

APLIKASI DEKOMPOSISI NILAI SINGULAR MATRIKS QUATERNION TERHADAP PERBAIKAN CITRA *LOW* DAN *HIGH PASS FILTERING*

APPLICATION OF QUATERNION MATRIX SINGULAR VALUE DECOMPOSITION ON LOW AND HIGH PASS FILTERING

Regita Agustin Wahyu Fibriyanti*, Karyati

Jurusan Pendidikan Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Negeri Yogyakarta, Yogyakarta 55281, Indonesia

*email korespondensi: regittaagustinn@gmail.com

Abstrak

Dekomposisi nilai singular matriks quaternion (*Quaternion Singular Value Decomposition/SVDQ*) sangat berguna dalam pengolahan citra, seperti dalam kompresi ukuran gambar, *eigen-images*, dan perbaikan kualitas citra, khususnya perbaikan suatu citra *low pass filtering* dan *high pass filtering*. Tujuan penelitian ini adalah mengetahui bagaimana aplikasi dekomposisi nilai singular matriks quaternion pada perbaikan suatu citra *low pass filtering* dan *high pass filtering*, serta memberikan suatu aplikasi berbasis GUI MATLAB. Dasar dari dekomposisi nilai singular matriks quaternion didapatkan dengan menggunakan representasi terhadap matriks kompleks, dengan kata lain menggunakan isomorfisma matriks quaternion terhadap matriks kompleks. Hal itu bertujuan untuk mempermudah pendekomposisian matriks quaternion. Matriks quaternion tidak lepas dari nilai eigen matriks quaternion. Selanjutnya, nilai eigen matriks quaternion dapat dihitung menggunakan nilai eigen dari matriks kompleksnya. Melalui dekomposisi nilai singular matriks quaternion, diperoleh perbaikan citra *low pass filtering* dan *high pass filtering* yang hampir menyerupai citra aslinya. Untuk perbaikan citra *low pass filtering* berhenti pada $\alpha = 1,06$ dan untuk *high pass filtering* berhenti pada $\alpha = 0,97$.

Kata kunci: dekomposisi nilai singular, quaternion, matriks kompleks, low pass filtering, high pass filtering

Abstract

Quaternion singular value decomposition is beneficial in image processing, such as in image size compression, eigen-images, and image enhancement, especially low pass filtering and high pass filtering image enhancement. This research aims to know how quaternion singular value decomposition is applied in low pass filtering and high pass filtering image enhancement, as well as to give a MATLAB GUI based application. The base of quaternion singular value decomposition is obtained by using the representation to the complex matrices, or in another word, using isomorphism of quaternion matrix to complex matrices. This aims to ease the quaternion matrix decomposition. Quaternion matrix is not detached from eigen matrix quaternion value. Eigen value of quaternion matrix is counted using the eigen value of the complex matrices. Using quaternion singular value decomposition, low pass filtering or high pass filtering image enhancement is obtained which resemble the real image. Low pass filtering image enhancement stopped at $\alpha=1.06$ and high pass filtering stopped at $\alpha=0.97$.

Keywords: singular value decomposition, quaternion, complex matrix, low pass filtering, high pass filtering

Pendahuluan

Kemajuan teknologi berdampak pada peningkatan konten multimedia dari informasi digital pada setiap harinya. Terutama yang terdiri dari citra foto atau video. Citra-citra tersebut diubah dari satu bentuk ke bentuk lainnya dengan proses seperti pencitraan, pemindaian, atau transmisi, sehingga gambar yang dihasilkan mungkin lebih rendah daripada citra input asli. Untuk mengatasi masalah ini, perlu dilakukan suatu perbaikan citra. Salah satu caranya yaitu dengan meningkatkan jangkauan dinamis dari fitur yang dipilih dalam gambar, yang pada dasarnya adalah suatu proses *image enhancement* (perbaikan citra) [1]. Tujuan utama perbaikan citra

adalah memberikan detail yang tersembunyi dalam suatu gambar atau untuk meningkatkan kontras pada gambar dengan kontras rendah [2].

Citra dapat direpresentasikan sebagai sebuah matriks [3]. Perbaikan citra dapat dilakukan dengan pembobotan nilai singular suatu citra, dan langkah tepat yang berkaitan dengan nilai singular yaitu suatu dekomposisi nilai singular (*Singular Value Decomposition/SVD*) [4]. SVD merupakan suatu metode aljabar dalam pendekomposisian matriks menjadi tiga buah matriks U, Σ , dan V^T . Matriks diagonal Σ berisi nilai-nilai singular sebagai entri diagonalnya. Nilai-nilai singular tersebut merepresentasikan intensitas sebuah informasi dari citra input dan setiap perubahan pada nilai singular akan

mengubah intensitas informasi tersebut [5]. Metode SVD dianggap dapat mereduksi data citra cukup baik dengan tingkat kompleksitas perhitungan yang tidak rumit dan besarnya *rank* yang diambil tergantung kebutuhan aplikasi serta merupakan pemilihan antara besarnya rasio kompresi dan kualitas citra hasil reduksi [6].

Secara umum, jenis dari perbaikan citra menggunakan SVD terdapat dua macam, yaitu pembobotan nilai singular secara linear dan non linear. Pembobotan nilai singular secara linear dapat dipandang sebagai *low pass filtering*, sedangkan pembobotan nilai singular secara non linear dapat dipandang sebagai *high pass filtering*. Kedua proses tersebut bekerja pada manipulasi langsung piksel sebuah citra [7]. Quaternion, disebut juga bilangan *hypercomplex*, merupakan sebuah bilangan perluasan dari bilangan kompleks. Dalam penelitian ini digunakan *Singular Value Decomposition Quaternion* (SVDQ) untuk perbaikan suatu citra melalui pembobotan nilai singularnya.

Telah banyak penelitian sebelumnya mengenai SVDQ, pada tahun 1947, H.C. Lee memberikan penjelasan mengenai perhitungan nilai eigen dan vektor eigen matriks quaternion yang berperan penting pada perhitungan SVD quaternion nantinya [8]. Pada 1997, Zhang memberikan detail mengenai quaternion dan matriks quaternion [9]. Sangwine dan Ell menggunakan quaternion untuk *auto and cross-correlation* suatu warna citra pada tahun 1999 [10]. Pada tahun 2003, Pei memperkenalkan beberapa kegunaan SVDQ dalam berbagai proses warna citra [4]. Perbaikan citra menggunakan DWT, DCT, dan SVD telah diperkenalkan oleh Mohan dan Linda pada tahun 2014 [5]. Selanjutnya tahun 2015, Marda menggunakan isomorfisma matriks kompleks terhadap matriks quaternion untuk mencari dekomposisi nilai singular matriks quaternion [11]. Namun, dari beberapa penelitian di atas, belum ada yang memperkenalkan aplikasi GUI secara gamblang untuk perbaikan suatu citra *low pass filtering* dan *high pass filtering* menggunakan SVDQ.

Berdasarkan uraian di atas, penelitian ini akan menunjukkan algoritma perhitungan dekomposisi nilai singular matriks quaternion menggunakan representasinya pada matriks kompleks secara manual yang merupakan salah satu alat dalam perbaikan citra. Selanjutnya dibuat suatu aplikasi berbasis GUI MATLAB tentang perbaikan citra *low pass filtering* dan *high pass filtering* menggunakan SVDQ.

Metode Penelitian

Penelitian ini menggunakan metode studi literatur berupa jurnal-jurnal ilmiah yang terkait dengan topik penelitian dan buku-buku referensi. Pada tahap awal dipelajari tentang konsep vektor, matriks, dekomposisi nilai singular, bilangan quaternion, dan perbaikan citra. Selanjutnya, dipelajari representasi kompleks bilangan quaternion, vektor quaternion, dan matriks quaternion yang merupakan langkah menentukan dekomposisi nilai singular matriks quaternion. Langkah terakhir adalah menerapkan dekomposisi nilai singular yang diperoleh untuk membuat rancangan algoritma, sehingga diperoleh teknik perbaikan citra berdasarkan operasi dan sifat-sifat yang dimilikinya menggunakan GUI MATLAB.

Hasil dan Pembahasan

Pada bagian ini terlebih dahulu diberikan pembahasan mengenai representasi kompleks bilangan quaternion, matriks quaternion, dan vektor quaternion.

Representasi Kompleks

Berikut diberikan pembahasan mengenai representasi kompleks suatu quaternion menggunakan isomorfismanya.

Representasi Kompleks Bilangan Quaternion

Definisi 1 [9], misalkan \mathbb{C} dan \mathbb{R} menunjukkan lapangan (field) kompleks dan bilangan riil secara berurutan, lalu \mathbb{Q} merupakan suatu ruang vektor dimensi empat atas \mathbb{R} dengan sebuah basis berurutan, ditunjukkan dengan $\mathbf{e}, \mathbf{i}, \mathbf{j}$, dan \mathbf{k} .

$$x = x_0\mathbf{e} + x_1\mathbf{i} + x_2\mathbf{j} + x_3\mathbf{k} \in \mathbb{Q} \quad (1)$$

Suatu quaternion riil, secara sederhana disebut quaternion, adalah suatu vektor dengan koefisien riil x_0, x_1, x_2, x_3 dan $\mathbf{e}, \mathbf{i}, \mathbf{j}, \mathbf{k}$ memenuhi aturan perkalian tak komutatif:

$$\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{ijk} = -\mathbf{1}, \quad (2)$$

$$\begin{aligned} \mathbf{ij} &= -\mathbf{ji} = \mathbf{k}, \\ \mathbf{ki} &= -\mathbf{ik} = \mathbf{j}, \\ \mathbf{jk} &= -\mathbf{kj} = \mathbf{i}. \end{aligned} \quad (3)$$

Selanjutnya, sebuah quaternion $q = q_0 + q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k}$ secara unik dapat ditulis dengan bentuk

$$q = \alpha + \beta j, \tag{4}$$

dengan

$$\alpha = q_0 + q_1 i \text{ dan } \beta = q_2 + q_3 i, \alpha, \beta \in \mathbb{C} \tag{5}$$

Notasi seperti itu disebut dengan notasi Cayley-Dickson. α dapat dipandang sebagai proyeksi q pada bidang kompleks. Persamaan 4 membentuk suatu *field* komutatif \mathbb{C} secara tertutup. Menurut Lee [8], \mathbb{C} secara jelas isomorfis terhadap *field* bilangan kompleks, sehingga quaternion q pada Persamaan 4 yang terletak pada \mathbb{C} secara sederhana dapat disebut sebagai *bilangan kompleks*. Sebuah bilangan quaternion isomorfis terhadap matriks kompleks 2×2 [12].

Diberikan teorema mengenai representasi bilangan quaternion terhadap bilangan kompleks.

Teorema 1, untuk $q = q_0 + q_1 i + q_2 j + q_3 k \in \mathbb{Q}$, dengan $q_0, q_1, q_2, q_3 \in \mathbb{R}$ dapat didefinisikan sebuah pemetaan fungsi $f: \mathbb{Q}^{m \times n} \rightarrow \mathbb{C}^{2 \times 2}$ sedemikian sehingga,

$$f(q) = \begin{bmatrix} q_0 + q_1 i & q_2 + q_3 i \\ -(q_2 - q_3 i) & q_0 - q_1 i \end{bmatrix} \in \mathbb{C}^{2 \times 2} \tag{6}$$

merupakan suatu isomorfisma [13].

Menggunakan persamaan 5, persamaan 6 dapat ditulis dengan bentuk,

$$f(q) = \begin{bmatrix} \alpha & \beta \\ -(\bar{\beta}) & \bar{\alpha} \end{bmatrix} \in \mathbb{C}^{2 \times 2}. \tag{7}$$

Persamaan di atas merupakan representasi kompleks bilangan quaternion terhadap matriks kompleks 2×2 . Isomorfisma ini dapat digunakan untuk mendekomposisikan matriks quaternion nantinya. Bukti dari isomorfisma di atas telah dipaparkan oleh Abdal et al. [13]. Berikut diberikan bukti dari teorema tersebut. Bukti,

Jika diberikan $q, p \in \mathbb{Q}$ dengan $q = q_0 + q_1 i + q_2 j + q_3 k$ dan $p = p_0 + p_1 i + p_2 j + p_3 k$, maka dari definisi fungsi f diperoleh,

$$f(q) = \begin{bmatrix} q_0 + q_1 i & q_2 + q_3 i \\ -(q_2 - q_3 i) & q_0 - q_1 i \end{bmatrix}$$

dan

$$f(p) = \begin{bmatrix} p_0 + p_1 i & p_2 + p_3 i \\ -(p_2 - p_3 i) & p_0 - p_1 i \end{bmatrix}.$$

Untuk membuktikan bahwa pemetaan fungsi f merupakan suatu isomorfisma quaternion real terhadap matriks kompleks, maka

1) Ditunjukkan $f(p + q) = f(p) + f(q)$

$$\begin{aligned} f(p + q) &= \\ & \begin{bmatrix} (p_0 + q_0) + (p_1 i + q_1 i) & (p_2 + q_2) + (p_3 i + q_3 i) \\ -(p_2 + q_2) + (p_3 i + q_3 i) & (p_0 + q_0) - (p_1 i + q_1 i) \end{bmatrix} \\ &= \begin{bmatrix} (p_0 + p_1 i) & (p_2 + p_3 i) \\ -(p_2 - p_3 i) & (p_0 - p_1 i) \end{bmatrix} + \\ & \begin{bmatrix} (q_0 + q_1 i) & (q_2 + q_3 i) \\ -(q_2 - q_3 i) & (q_0 - q_1 i) \end{bmatrix} \\ &= f(p) + f(q). \end{aligned}$$

2) Ditunjukkan $f(pq) = f(p)f(q)$

Menggunakan definisi dot product dua quaternion didapatkan,

$$\begin{aligned} f(pq) &= f((p_0 + p_1 i + p_2 j + p_3 k)(q_0 + q_1 i + q_2 j + q_3 k)) \\ &= f((p_0 q_0 - p_1 q_1 - p_2 q_2 - p_3 q_3) + \\ & (p_0 q_1 + p_1 q_0 + p_2 q_3 - p_3 q_2) i + \\ & (p_0 q_2 + p_2 q_0 + p_3 q_1 - p_1 q_3) j + (p_0 q_3 + p_3 q_0 + \\ & p_1 q_2 - p_2 q_1) k) \\ &= \begin{bmatrix} p_0 + p_1 i & p_2 + p_3 i \\ -(p_2 - p_3 i) & p_0 - p_1 i \end{bmatrix} \begin{bmatrix} q_0 + q_1 i & q_2 + q_3 i \\ -(q_2 - q_3 i) & q_0 - q_1 i \end{bmatrix} \\ &= f(p)f(q). \end{aligned}$$

3) Ditunjukkan bahwa f merupakan fungsi satu-satu (injektif), yaitu jika $f(p) = f(q)$, maka

$$\begin{aligned} p &= q. \\ f(p) &= f(q) \\ \begin{bmatrix} p_0 + p_1 i & p_2 + p_3 i \\ -(p_2 - p_3 i) & p_0 - p_1 i \end{bmatrix} &= \\ \begin{bmatrix} q_0 + q_1 i & q_2 + q_3 i \\ -(q_2 - q_3 i) & q_0 - q_1 i \end{bmatrix} \end{aligned}$$

Dari kesamaan matriks $f(p) = f(q)$ di atas, diperoleh empat persamaan entri matriks yaitu :

- i. $p_0 + p_1 i = q_0 + q_1 i$
- ii. $p_2 + p_3 i = q_2 + q_3 i$
- iii. $-(p_2 - p_3 i) = -(q_2 - q_3 i)$
- iv. $p_0 - p_1 i = q_0 - q_1 i$

Selanjutnya, mencari nilai p_0, p_1, p_2, p_3 menggunakan metode eliminasi pada keempat persamaan di atas, sehingga didapatkan

$$\begin{aligned} p_0 &= q_0, p_1 = q_1, p_2 = q_2, p_3 = q_3, \text{ maka} \\ p &= p_0 + p_1 i + p_2 j + p_3 k = q_0 + q_1 i + q_2 j + q_3 k = q \end{aligned}$$

4) Selanjutnya ditunjukkan bahwa f merupakan fungsi surjektif (onto), yaitu untuk setiap $b \in \mathbb{C}^{2 \times 2}$, terdapat $p \in \mathbb{Q}$ sedemikian sehingga $f(p) = b$.

Ambil $b \in \mathbb{C}^{2 \times 2}$, misalkan $b =$

$$\begin{bmatrix} w_0 + w_1 i & -(x_0 + x_1 i) \\ y_0 - y_1 i & z_0 - z_1 i \end{bmatrix}.$$

Karena $f(p) = b$ maka,

$$\begin{bmatrix} p_0 + p_1 i & p_2 + p_3 i \\ -(p_2 - p_3 i) & p_0 - p_1 i \end{bmatrix} = \begin{bmatrix} w_0 + w_1 i & (x_0 + x_1 i) \\ -(y_0 - y_1 i) & z_0 - z_1 i \end{bmatrix}.$$

Dari persamaan dua matriks di atas, diperoleh empat persamaan entri-entri matriks yaitu;

- i. $p_0 + p_1 i = w_0 + w_1 i$
- ii. $p_2 + p_3 i = x_0 + x_1 i$
- iii. $-(p_2 - p_3 i) = -(y_0 - y_1 i)$
- iv. $p_0 - p_1 i = z_0 - z_1 i$

Selanjutnya, mencari nilai p_0, p_1, p_2, p_3 menggunakan metode eliminasi pada keempat persamaan di atas, sehingga didapatkan

$$\begin{aligned} p_0 &= \frac{(w_0 + w_1 i) + (z_0 - z_1 i)}{2} \\ p_1 &= \frac{(w_0 + w_1 i) + (z_0 - z_1 i)}{2i} \\ p_2 &= \frac{(x_0 + x_1 i) + (y_0 - y_1 i)}{2} \\ p_3 &= \frac{(w_0 + w_1 i) + (y_0 - y_1 i)}{2i} \end{aligned}$$

sehingga didapatkan

$$\begin{aligned} p &= p_0 + p_1 i + p_2 j + p_3 k, \\ &= \frac{(w_0 + w_1 i) + (z_0 - z_1 i)}{2} + \frac{(w_0 + w_1 i) + (z_0 - z_1 i)}{2i} i \\ &\quad + \frac{(x_0 + x_1 i) + (y_0 - y_1 i)}{2} j + \frac{(w_0 + w_1 i) + (y_0 - y_1 i)}{2i} k \end{aligned}$$

Berdasarkan pembuktian, dapat disimpulkan f merupakan isomorfisma atau $\mathbb{Q} \cong \mathbb{C}^{2 \times 2}$, dan pembuktian tersebut dapat digunakan untuk menunjukkan adanya isomorfisma antara matriks quaternion terhadap matriks kompleks.

Representasi Kompleks Matriks Quaternion

Matriks quaternion akan direpresentasi ke dalam matriks kompleks untuk mempermudah dalam mendekomposisi matriks quaternion.

Definisi 2, misalkan $Q_{(q)}$ adalah suatu matriks quaternion $n \times n$ dan $Q_{(q)} = A_{(c)} + B_{(c)}$ dengan $A_{(c)}$ dan $B_{(c)}$ merupakan dua matriks kompleks, sehingga matriks kompleks yang ekuivalen dengan $Q_{(q)}$ yaitu,

$$Q_{e(c)} = \begin{bmatrix} A_{(c)} & B_{(c)} \\ -\bar{B}_{(c)} & \bar{A}_{(c)} \end{bmatrix}_{2n \times 2n} \quad (8)$$

Isomorfisma bilangan quaternion terhadap matriks kompleks dapat digunakan untuk mencari isomorfisma matriks quaternion terhadap matriks kompleks. Selanjutnya menurut Rodman [14] jika terdapat $Q_{(q)} \in \mathbb{Q}^{2 \times 2}$ yaitu

$$\begin{aligned} Q_{(q)} &= \\ &\begin{bmatrix} a_0 + a_1 i + a_2 j + a_3 k & b_0 + b_1 i + b_2 j + b_3 k \\ c_0 + c_1 i + c_2 j + c_3 k & d_0 + d_1 i + d_2 j + d_3 k \end{bmatrix} \\ &\in \mathbb{Q}^{2 \times 2} \end{aligned}$$

Isomorfisma matriks kompleks dapat ditulis sebagai,

$$Q_{e(c)} = \begin{bmatrix} a_0 + a_1 i & b_0 + b_1 i & a_2 + a_3 i & b_2 + b_3 i \\ c_0 + c_1 i & d_0 + d_1 i & c_2 + c_3 i & d_2 + d_3 i \\ -a_2 + a_3 i & -b_2 + b_3 i & a_0 - a_1 i & b_0 - b_1 i \\ -c_2 + c_3 i & -d_2 + d_3 i & c_0 - c_1 i & d_0 - d_1 i \end{bmatrix}$$

$$\text{dengan } A_{(c)} = \begin{bmatrix} a_0 + a_1 i & b_0 + b_1 i \\ c_0 + c_1 i & d_0 + d_1 i \end{bmatrix} \text{ dan } B_{(c)} = \begin{bmatrix} a_2 + a_3 i & b_2 + b_3 i \\ c_2 + c_3 i & d_2 + d_3 i \end{bmatrix}.$$

Selanjutnya, direpresentasikan vektor quaternion ke bentuk kompleks.

Representasi Kompleks Vektor Quaternion

Demikian halnya dengan matriks quaternion, vektor dari matriks quaternion juga dapat direpresentasikan ke dalam bentuk vektor kompleks.

Definisi 3, diberikan sebuah vektor quaternion $x \in \mathbb{Q}^n$, menggunakan notasi Cayley-Dickson, $x = x_1 + x_2 j$ dimana $x_1 \in \mathbb{C}^n$ dan $x_2 \in \mathbb{C}^n$, sebuah vektor quaternion dapat didefinisikan oleh $f: \mathbb{Q}^n \rightarrow \mathbb{C}^{2n}$, sehingga didapatkan

$$f(x) = \begin{bmatrix} x_1 \\ -\bar{x}_2 \end{bmatrix} \quad (9)$$

dengan \bar{x}_2 adalah konjugat dari x_2 [15].

Dekomposisi Nilai Singular Matriks Quaternion

Definisi 4, jika diketahui $Q_{(q)}$ adalah suatu matriks quaternion $n \times n$ dengan rank r , maka terdapat dua matriks uniter quaternion $U_{(q)}$ dan $V_{(q)}$ sedemikian sehingga

$$U_{(q)}^* \cdot Q_{(q)} \cdot V_{(q)} = \begin{bmatrix} \Lambda_r & 0 \\ 0 & 0 \end{bmatrix} \equiv \Lambda \quad (10)$$

dengan $\Lambda_r = \text{diag} \{ \sigma_1, \dots, \sigma_r \}$ dan σ merupakan nilai singular positif dari $Q_{(q)}$, $U_{(q)}$ dan $V_{(q)}$ secara berurutan memuat vektor singular quaternion kiri dan kanan dari Λ dan $U_{(q)}^*$ menunjukkan transpos konjugat dari $U_{(q)}$ [4].

Selanjutnya, akan diberikan teorema mengenai hubungan antara dekomposisi nilai singular matriks quaternion dan dekomposisi nilai singular dari matriks kompleks ekuivalensinya.

Teorema 2, jika diketahui dekomposisi nilai singular matriks quaternion dan dekomposisi nilai singular matriks kompleks ekuivalensi secara berturut-turut $Q_{(q)} = U_{(q)} \cdot \Lambda \cdot V_{(q)}^*$ dan $Q_{e(c)} = U_{(c)} \cdot \Lambda' \cdot V_{(c)}^*$, maka

$$\begin{aligned} (i) \quad & Q_{(q)} Q_{(q)}^* = U_{(q)} \cdot \Lambda^2 \cdot U_{(q)}^* \\ (ii) \quad & Q_{(q)}^* Q_{(q)} = V_{(q)} \cdot \Lambda^2 \cdot V_{(q)}^* \\ (iii) \quad & Q_{e(c)} Q_{e(c)}^* = U_{(c)} \cdot (\Lambda')^2 \cdot U_{(c)}^* \\ (iv) \quad & Q_{e(c)}^* Q_{e(c)} = V_{(c)} \cdot (\Lambda')^2 \cdot V_{(c)}^* \end{aligned} \quad (11)$$

Bukti :

$$\begin{aligned} (i) \quad & Q_{(q)} = U_{(q)} \Lambda V_{(q)}^*, \\ & \Leftrightarrow Q_{(q)} Q_{(q)} = U_{(q)} \Lambda V_{(q)}^* U_{(q)} \Lambda V_{(q)}^*, \\ & \Leftrightarrow Q_{(q)} Q_{(q)}^* = U_{(q)} \Lambda V_{(q)}^* (U_{(q)} \Lambda V_{(q)}^*)^*, \\ & \quad = U_{(q)} \Lambda V_{(q)}^* (V_{(q)}^*)^* \Lambda^* U_{(q)}^*, \\ & \quad = U_{(q)} \Lambda V_{(q)}^* V_{(q)} \Lambda^* U_{(q)}^*, \\ & \quad = U_{(q)} \Lambda I \Lambda^* U_{(q)}^*, \\ & \quad = U_{(q)} \Lambda^2 U_{(q)}^*. \end{aligned}$$

$$\begin{aligned} (ii) \quad & Q_{(q)} = U_{(q)} \Lambda V_{(q)}^*, \\ & \Leftrightarrow Q_{(q)} Q_{(q)} = U_{(q)} \Lambda V_{(q)}^* U_{(q)} \Lambda V_{(q)}^*, \\ & \Leftrightarrow Q_{(q)}^* Q_{(q)} = (U_{(q)} \Lambda V_{(q)}^*)^* U_{(q)} \Lambda V_{(q)}^*, \\ & \quad = (V_{(q)}^*)^* \Lambda^* U_{(q)}^* U_{(q)} \Lambda V_{(q)}^*, \\ & \quad = V_{(q)} \Lambda^* U_{(q)}^* U_{(q)} \Lambda V_{(q)}^*, \\ & \quad = V_{(q)} \Lambda I \Lambda^* V_{(q)}^*, \\ & \quad = V_{(q)} \Lambda^2 V_{(q)}^*. \end{aligned}$$

$$\begin{aligned} (iii) \quad & Q_{e(c)} = U_{(c)} \Lambda' V_{(c)}^*, \\ & \Leftrightarrow Q_{e(c)} Q_{e(c)} = U_{(c)} \Lambda' V_{(c)}^* U_{(c)} \Lambda' V_{(c)}^*, \\ & \Leftrightarrow Q_{e(c)} Q_{e(c)}^* = \\ & \quad U_{(c)} \Lambda' V_{(c)}^* (U_{(c)} \Lambda' V_{(c)}^*)^*, \\ & \quad = \\ & \quad U_{(c)} \Lambda' V_{(c)}^* (V_{(c)}^*)^* (\Lambda')^* U_{(c)}^*, \end{aligned}$$

$$\begin{aligned} & = U_{(c)} \Lambda' V_{(c)}^* V_{(c)} (\Lambda')^* U_{(c)}^*, \\ & = U_{(c)} \Lambda' I (\Lambda')^* U_{(c)}^*, \\ & = U_{(c)} (\Lambda')^2 U_{(c)}^* \end{aligned}$$

$$\begin{aligned} (iv) \quad & Q_{e(c)} = U_{(c)} \Lambda' V_{(c)}^*, \\ & \Leftrightarrow Q_{e(c)} Q_{e(c)} = \\ & \quad U_{(c)} \Lambda' V_{(c)}^* U_{(c)} \Lambda' V_{(c)}^*, \\ & \Leftrightarrow Q_{e(c)}^* Q_{e(c)} = \\ & \quad (U_{(c)} \Lambda' V_{(c)}^*)^* U_{(c)} \Lambda' V_{(c)}^*, \\ & \quad = \\ & \quad (V_{(c)}^*)^* (\Lambda')^* U_{(c)}^* U_{(c)} \Lambda' V_{(c)}^*, \\ & \quad = \\ & \quad V_{(c)} (\Lambda')^* U_{(c)}^* U_{(c)} \Lambda' V_{(c)}^*, \\ & \quad = V_{(c)} (\Lambda')^* I \Lambda' V_{(c)}^*, \\ & \quad = V_{(c)} (\Lambda')^2 V_{(c)}^* \end{aligned}$$

Dari pembuktian di atas, menurut Zhang [9] dapat diketahui bahwa dekomposisi nilai singular matriks quaternion merupakan perkalian matriks yang terdiri dari :

- Matriks Λ^2 merupakan matriks diagonal dengan entri nilai eigen riil dari $Q_{(q)} Q_{(q)}^*$ dan $Q_{(q)}^* Q_{(q)}$.
- Vektor kolom $U_{(q)}$ dan $V_{(q)}$ secara berurutan merupakan vektor eigen dari $Q_{(q)} Q_{(q)}^*$ dan $Q_{(q)}^* Q_{(q)}$.

Selanjutnya, karena $Q_{e(c)}$ merupakan matriks representasi kompleks dari $Q_{(q)}$ sehingga berdasarkan Teorema 2, $Q_{e(c)} Q_{e(c)}^*$ dan $Q_{e(c)}^* Q_{e(c)}$ secara berturut-turut merupakan matriks representasi kompleks dari $Q_{(q)} Q_{(q)}^*$ dan $Q_{(q)}^* Q_{(q)}$. Sehingga dapat diuraikan beberapa hubungan antara Λ' dan Λ , $U_{(c)}$ dan $U_{(q)}$, serta $V_{(c)}$ dan $V_{(q)}$ di antaranya adalah :

- Matriks Λ^2 dan $(\Lambda')^2$ secara berurutan merupakan matriks diagonal nilai eigen riil dari $Q_{(q)} Q_{(q)}^*$ dan $Q_{e(c)} Q_{e(c)}^*$, sehingga setiap elemen pada Λ^2 akan menjadi dua kali pada $(\Lambda')^2$. Oleh karena itu, dengan menghapus setengah duplikat nilai eigen pada Λ' akan menjadi Λ .
- Vektor kolom $U_{(q)}$ dan $U_{(c)}$ secara berurutan merupakan vektor eigen dari $Q_{(q)} Q_{(q)}^*$ dan $Q_{e(c)} Q_{e(c)}^*$. Dengan Teorema 3 dapat dihitung vektor kolom $U_{(q)}$ menggunakan vektor kolom $U_{(c)}$.
- Vektor kolom $V_{(q)}$ dan $V_{(c)}$ secara berurutan merupakan vektor eigen dari $Q_{(q)}^* Q_{(q)}$ dan $Q_{e(c)}^* Q_{e(c)}$. Dengan Teorema 3 dapat dihitung

vektor kolom $V_{(q)}$ menggunakan vektor kolom $V_{(c)}$.

Selanjutnya, setelah didapatkan representasi kompleks bilangan quaternion. Matriks quaternion dan vektor quaternion akan dilakukan perhitungan dekomposisi nilai singular matriks quaternion, yang merupakan tujuan dari penelitian ini.

Algoritma Perhitungan Dekomposisi Nilai Singular Matriks Quaternion

Pada perhitungan dekomposisi nilai singular matriks quaternion, digunakan representasi kompleks dari matriks quaternion untuk mempermudah perhitungan. Adapun bentuk persamaan dekomposisi nilai singular matriks representasi kompleksnya yaitu

$$Q_{e(c)} = U_{(c)} \cdot \Lambda' \cdot V_{(c)}^*, \quad (12)$$

dengan vektor kolom $U_{(c)}$ dan $V_{(c)}$ masing-masing merupakan elemen dari \mathbb{C}^{2n} dan \mathbb{C}^{2m} , serta $V_{(c)}^*$ merupakan transpos konjugat dari $V_{(c)}$. Berikut akan diberikan penjelasan lebih lanjut mengenai matriks $U_{(c)}$, Λ' , dan $V_{(c)}^*$.

Matriks $V_{(c)}^*$

Matriks $V_{(c)}$ merupakan matriks kolom pada \mathbb{C}^{2m} yang memuat vektor singular kanan dari $Q_{e(c)}$, sedemikian sehingga merupakan vektor eigen dari $Q_{e(c)}^* Q_{e(c)}$. Misalkan

$$V_{(c)} = (\vec{V}_{1(c)}, \vec{V}_{2(c)}, \vec{V}_{3(c)}, \dots, \vec{V}_{r(c)}, \vec{V}_{r+1(c)} \dots, \vec{V}_{m(c)})$$

agar membentuk himpunan ortonormal maka didefinisikan $\vec{V}_{i(c)}$ sebagai

$$\vec{V}_{i(c)} = \frac{1}{\|V_{i(c)}\|} V_{i(c)}, \quad (13)$$

$V_{i(c)}$ merupakan vektor eigen yang bersesuaian dengan nilai eigen λ_i , dengan $1 \leq i \leq m$.

Matriks $U_{(c)}$

Matriks $U_{(c)}$ merupakan matriks kolom pada \mathbb{C}^{2n} yang memuat vektor singular kiri dari $Q_{e(c)}$, sedemikian sehingga merupakan vektor eigen dari $Q_{e(c)} Q_{e(c)}^*$. Misalkan

$$U_{(c)} = (\vec{U}_{1(c)}, \vec{U}_{2(c)}, \vec{U}_{3(c)}, \dots, \vec{U}_{r(c)}, \vec{U}_{r+1(c)} \dots, \vec{U}_{n(c)})$$

Untuk membentuk basis ortonormal pada matriks $U_{(c)}$, maka didefinisikan $\vec{U}_{i(c)}$ sebagai

$$\vec{U}_{i(c)} = \frac{1}{\sigma_i} A \vec{V}_{i(c)} \quad (14)$$

dengan $1 \leq i \leq n$.

Matriks Λ'

Matriks Λ' merupakan matriks diagonal dengan entri diagonalnya merupakan nilai singular dari $Q_{e(c)}$. Matriks Λ' berukuran $n \times n$ dengan *rank* r memiliki bentuk

$$\Lambda' = \text{diag}(\sigma_1, \sigma_1, \sigma_2, \sigma_2, \dots, \sigma_r, \sigma_r) \\ = \begin{bmatrix} \sigma_1 & 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & \sigma_1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & \sigma_2 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & \sigma_2 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & \sigma_r & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 & \sigma_r \end{bmatrix}. \quad (15)$$

Selanjutnya, diberikan algoritma untuk menghitung dekomposisi nilai singular matriks quaternion $Q_{(q)}$ melalui dekomposisi nilai singular representasi matriks kompleksnya $Q_{e(c)}$. Algoritma ini dapat diuraikan sebagai berikut,

Langkah 1 : Menghitung representasi matriks kompleks $Q_{e(c)}$ dengan Definisi 2.

Langkah 2 : Menghitung dekomposisi nilai singular matriks $Q_{e(c)}$ dengan perhitungan dekomposisi nilai singular matriks kompleks. Asumsikan $Q_{e(c)} = U_{(c)} \cdot \Lambda' \cdot V_{(c)}^*$.

Langkah 3 : Menghitung matriks singular Q_q, Λ , melalui Λ' pada Teorema 2. Menghitung dua matriks uniter $Q_{(q)}, U_{(q)}$ dan $V_{(q)}$, melalui $U_{(c)}$ dan $V_{(c)}$ pada Teorema 2.

Aplikasi SVD Matriks Quaternion

Sebuah matriks quaternion $f_{(q)}(m, n)$ dapat merepresentasikan suatu warna citra, $f_{(q)}(m, n)$ dapat dinyatakan sebagai suatu persamaan

$$f_{(q)}(m, n) = f_R(m, n)\mathbf{i} + f_G(m, n)\mathbf{j} + f_B(m, n)\mathbf{k}, \quad (16)$$

dengan $f_R(m, n)$, $f_G(m, n)$, dan $f_B(m, n)$ merepresentasikan bagian R, G, B dari warna citra.

Dari representasi di atas, dekomposisi nilai singular matriks quaternion untuk suatu warna citra dapat ditulis sebagai

$$f_{(q)} = U_{(q)} \cdot \Lambda \cdot V_{(q)}^*, \quad (17)$$

dengan $U_{(q)}$, Λ , dan $V_{(q)}^*$ dapat dihitung pada pembahasan sebelumnya.

Dari dekomposisi nilai singular warna citra di atas, banyak penggunaan metode proses citra yang dapat diuraikan menjadi suatu warna citra tanpa memisahkan warna citra menjadi tiga *channel* citra. Dalam tulisan ini, pembahasan mengenai aplikasi dekomposisi nilai singular matriks quaternion dibatasi pada perbaikan citra *low pass filter* dan *high pass filter*. Perbaikan citra dapat dilakukan dengan pembobotan nilai singular. Perbaikan citra menggunakan dekomposisi nilai singular terbagi menjadi dua jenis proses, yaitu pembobotan linear (*linear weighting*) dan pembobotan nonlinear (*nonlinear weighting*).

Pembobotan linear berdasarkan dekomposisi nilai singular matriks quaternion dapat ditulis sebagai persamaan

$$[f]_{(q)} = \sum_{i=1}^R (1 + m \cdot i) \lambda_i \cdot u_{i(q)} \cdot v_{i(q)}^* \quad (18)$$

$(m \geq 0)$.

Pembobotan linear akan memperbaiki komponen frekuensi tinggi, sehingga dapat dipandang sebagai suatu *high pass filter*.

Sedangkan untuk pembobotan nonlinear berdasarkan dekomposisi nilai singular matriks quaternion dapat ditulis sebagai persamaan

$$[f]_{(q)} = \sum_{i=1}^R \lambda_i^\alpha \cdot u_{i(q)} \cdot v_{i(q)}^* \quad (19)$$

Jika $\alpha > 1$, maka pembobotan nonlinear dapat dipandang sebagai suatu *low pass filter*, sebaliknya jika $\alpha < 1$, maka pembobotan nonlinear dapat dipandang sebagai suatu *high pass filter*.



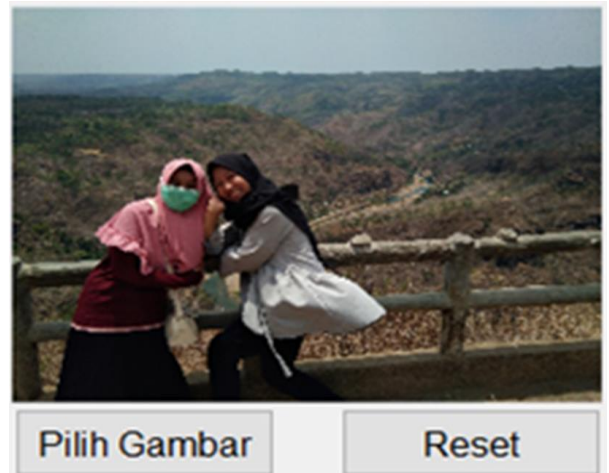
Gambar 1. Citra Gita dan Maya

Pada penelitian ini digunakan persamaan 19 untuk secara mudah dilakukan perbaikan citra menggunakan GUI Matlab. Sebagai contoh aplikasi perbaikan citra di bawah ini digunakan Citra Gita dan Maya (Gambar 1) untuk kasus *low pass filtering*. Langkah-langkah berikut juga digunakan untuk kasus *high pass filtering* menggunakan Citra Sekolah. Sementara itu, berikut ini diberikan langkah-langkah contoh pengaplikasian dekomposisi nilai singular matriks quaternion menggunakan *software* MATLAB terhadap Citra Gita dan Maya yaitu

Langkah 1. Membaca file citra.

Untuk membaca Citra Gita dan Maya pada MATLAB (Gambar 2), digunakan perintah `gambar=imread(fullfile(pathname, filename));`

Menurut Firdausi et al. [16], perintah di atas berfungsi mengonversi Citra Gita dan Maya menjadi data angka yang tersimpan dalam tiga matriks yang menyusun warna merah, hijau, dan biru. Setiap entri matriks menandakan kepekatan warna dan merupakan bilangan riil.



Gambar 2. Pilih gambar citra Gita dan Maya

Langkah 2. Menentukan ukuran matriks dari Citra Gita dan Maya. Untuk menentukan ukuran matriks dari Citra Gita dan Maya digunakan perintah

```
gambar = imresize(gambar,
    [min(lebar, tinggi)
    min(lebar, tinggi)]);
```

Langkah 3. Mengonversi 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:

```
gambarDouble=double(gambar)/255.0;
```

Langkah 4. Mengonversi entri matriks hasil langkah 3 menjadi entri bilangan quaternion.

Pada MATLAB, hal ini dapat dilakukan dengan menggunakan perintah:

```
imshow(imresize(gambar2, [lebar
tinggi]));
[m,n,o] = size(gambar2);
X1 = complex(zeros(m,n),
gambar2(:,:,1));
X2 = complex(gambar2(:,:,2),
gambar2(:,:,3));
```

Langkah 5. Menghitung dekomposisi nilai singular matriks penyusun warna.

Pada MATLAB, hal ini dapat dilakukan dengan menggunakan perintah:

```
Quaternion = [X1,X2;-
conj(X2),conj(X1)];
[U,S,V] = svd(Quaternion);
S = diag(S);
```

Langkah 6. Pembobotan matriks singular

Selanjutnya dilakukan pembobotan pada matriks singular S. Misalkan untuk *low pass*, berdasarkan Persamaan 19 mengambil nilai α melalui slider 1. Pada MATLAB, hal ini dapat dilakukan dengan menggunakan perintah:

```
lowPass =
get(handles.slider1,'Value')
S_ = S.^lowPass;
Z = U*diag(S_)*V';
```

Langkah 7. 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 19 untuk *low pass filter*. Pada MATLAB, hal ini dapat dilakukan dengan menggunakan perintah:

```
gambarLowPass = zeros(n,m,3);
gambarLowPass(:,:,1) =
imag(Z(1:n,1:m));
gambarLowPass(:,:,2) =
real(Z(1:n,(m+1):(2*m)));
gambarLowPass(:,:,3) =
imag(Z(1:n,(m+1):(2*m)));
```

sedangkan untuk *high pass filter*

```
gambarHighPass = zeros(n,m,3);
gambarHighPass(:,:,1) =
imag(Z(1:n,1:m));
gambarHighPass(:,:,2) =
real(Z(1:n,(m+1):(2*m)));
gambarHighPass(:,:,3) =
imag(Z(1:n,(m+1):(2*m)));
```

Langkah 8. Tampilkan hasil

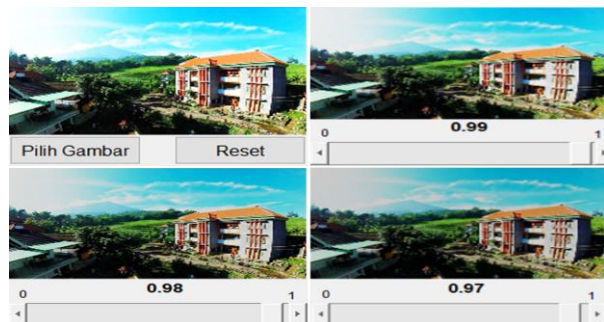
```
axes(handles.axes2);
imshow(imresize(gambarLowPass,
[lebar tinggi]));
axes(handles.axes3);
imshow(imresize(gambarHighPass,
[lebar tinggi]));
```

Hasil *Low Pass Filtering* ditunjukkan pada Gambar 3.



Gambar 3. Citra Gita dan Maya

Hasil *high pass filtering* ditunjukkan pada Gambar 4.



Gambar 4. Hasil perbaikan *high pass filtering*

Berdasarkan hasil pengolahan citra menggunakan Matlab di atas, didapatkan pada *low pass filter* perbaikan citra berhenti pada $\alpha = 1,06$ dan *high pass filter* pada $\alpha = 0,97$.

Kesimpulan

Berdasarkan hasil penelitian dan pembahasan yang telah dilakukan, dapat disimpulkan bahwa algoritma perhitungan dekomposisi nilai singular matriks quaternion dapat dilakukan dengan langkah-langkah yang meliputi pertama yaitu menghitung representasi matriks kompleks $Q_{e(c)}$ dengan Definisi 2, kedua yaitu menghitung dekomposisi nilai singular matriks $Q_{e(c)}$ dengan perhitungan dekomposisi nilai singular matriks kompleks. Asumsikan $Q_{e(c)} = U_{(c)} \cdot \Lambda' \cdot V_{(c)}^*$, dan kemudian menghitung matriks singular Q_q, Λ , melalui Λ' pada Teorema 2. Selanjutnya, menghitung dua matriks uniter $Q_{(q)}, U_{(q)}$ dan $V_{(q)}$, melalui $U_{(c)}$ dan $V_{(c)}$ pada Teorema 2. Pada *low pass filter* perbaikan citra berhenti pada $\alpha = 1,06$ dan *high pass filter* pada $\alpha = 0,97$.

Ucapan Terima Kasih

Kami berterima kasih kepada semua civitas akademika Program Studi Pendidikan Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Negeri Yogyakarta yang telah mendukung penelitian ini.

Daftar Pustaka

- [1] Acharya, T., & Ray, A. (2005). *Image processing, principles, and applications*. New Jersey: John Wiley & Sons, Inc.
- [2] Bansal, D., Bagoria, R., Sharma, V., & Student, P. (2015). Digital image enhancement by improving contrast, removal of noise and motion blurring. *International Journal of Innovative Research in Science, Engineering and Technology*, 4(4), 2601-2606.
- [3] Andrews, H., & Patterson, C. (1976). Singular value decompositions and digital image processing. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 24(1), 26-53.
- [4] Pei, S. C., Chang, J. H., & Ding, J. J. (2003). Quaternion matrix singular value decomposition and its applications for color image processing. *International Conference on Image Processing* (pp. 805-808). Spain: IEEE.
- [5] Mohan, A. & Linda P. A. M. (2014). Image enhancement using DWT CDT and SVD. *International Journal of Engineering Research and Applications*, 1(2), 36-40.
- [6] Sulaiman, S., & Agoes, S. (2017). Analisis reduksi data citra menggunakan metode dekomposisi nilai singular. *CITEE*, 5(1), 21-25.
- [7] Gonzalez, R. C., & Woods, R. (1992). *Digital image processing 2nd edition*. New Jersey: Prentice Hall.
- [8] Lee, H. C. (1950). Eigenvalues and canonical forms of matrices with quaternion coefficients. *JSTOR*, 2(1), 253-260.
- [9] Zhang, F. (1997). Quaternions and matrices of quaternions. *Linear Algebra and Its Applications*, 251(1), 21-57.
- [10] Ell, T. A., Bihan, N., & Sangwine, S. (2014). *Quaternion fourier transforms for signal and image processing*. USA: John Wiley & Sons, Inc.
- [11] Marda, N., Amir, A., & Erawaty, N. (2015). *Dekomposisi nilai singular matriks quaternion*. Makassar: UNHAS.
- [12] Bunse-Gerstner, A., Byers, R., & Mehrmann, V. (1989). A quaternion QR algorithm. *Numerische Mathematik*, 55(1), 83-95.
- [13] Abdal, A. M., Kamal, A., & Erawaty, N. (2015). *Isomorfisma dari matriks quaternion kompleks ke matriks kompleks dan sifat-sifatnya*. Makassar: UNHAS.
- [14] Rodman, L. (2014). *Topics in quaternion linear algebra*. New Jersey: Princeton University Press.
- [15] Le Bihan, N., & Mars, J. (2004). Singular value decomposition of quaternion matrices: A new tool for vector-sensor signal processing. *Signal Processing*, 84(7), 1177-1199.
- [16] Firdausi, A., Syafwan, M., & Bakar, N. N. (2015). Aplikasi dekomposisi nilai singular pada kompresi ukuran file gambar. *Jurnal Matematika UNAND*, 4(1), 31-39.