

**PEMAMPATAN DAN REKONSTRUKSI CITRA
MENGUNAKAN ANBALISIS KOMPONEN UTAMA (PRINCIPAL COMPONENTS ANALYSIS)**

Praditya*, Budi Setiono**, Rizal Isnanto**

Abstrak - Saat ini sudah banyak aplikasi berbasis citra digital maupun pengalihan(transfer) data dalam bentuk citra digital. Citra digital merupakan suatu data yang membutuhkan ruang memori cukup besar untuk penyimpanannya dan membutuhkan waktu pengalihan data yang lama sesuai dengan ukurannya. Hal ini akan sangat terasa pada aplikasi-aplikasi yang menggunakan citra digital dalam jumlah besar atau pada saat pengalihan data dalam jumlah besar sedangkan kapasitas memori dan lebar bidang media sangat terbatas. Untuk mengurangi masalah ini, ukuran citra harus direduksi melalui proses pemampatan citra.

Sudah banyak metode pemampatan citra yang dikembangkan dan diaplikasikan. Pada tugas akhir ini dikembangkan teknik pemampatan citra digital menggunakan Analisis Komponen Utama atau metode Karhunen-Loeve. Metode ini termasuk dalam pemampatan berugi, dimana beberapa elemen citra yang kurang penting dihilangkan. Namun metode ini dapat mengurangi ukuran citra secara signifikan. Metode ini mereduksi dimensi citra menjadi lebih ringkas menggunakan proses alihragam. Untuk merekonstruksi citra dilakukan proses alihragam balik.

Pada akhirnya dapat disimpulkan bahwa Analisis Komponen Utama cukup layak digunakan untuk metode pemampatan citra. Hasil pemampatan yang optimal diperoleh pada citra yang bertekstur sederhana serta mempunyai resolusi yang tinggi.

Kata kunci : Analisis Komponen Utama, pemampatan berugi, citra digital, dimensi citra, alihragam, alihragam balik.

I. PENDAHULUAN

Tujuan pemampatan citra digital adalah untuk menghemat memori penyimpanannya serta mempercepat pengalihan data. Teknik pemampatan secara umum terbagi dua, yaitu pemampatan berugi (*lossy*) dan tak berugi (*lossless*). Pemampatan berugi menghilangkan sebagian informasi yang kurang signifikan dari data yang dimampatkan, sedangkan pemampatan tak berugi tidak menghilangkan informasi apapun dari data yang dimampatkan. Pemampatan menggunakan Analisis Komponen Utama (PCA) termasuk pemampatan berugi.

II. TEORI ANALISIS KOMPONEN UTAMA

Analisis Komponen Utama (PCA) adalah teknik statistik untuk mengidentifikasi pola dalam suatu data, dan mengekspresikan suatu data sedemikian rupa sehingga diperoleh persamaan-persamaan dan perbedaan-perbedaanya^[5]. Salah satu keunggulan PCA adalah sekali pola suatu data dapat ditemukan, maka dapat dilakukan pemampatan data tanpa mengurangi

banyak informasi pada data itu, misalnya dengan mengurangi jumlah dimensi. Teknik ini digunakan pada pemampatan citra digital dan disebut juga alihragam Karhunen-Loeve atau alihragam Hotelling^[6].

Langkah-langkah dalam PCA adalah sebagai berikut.

1. Mengumpulkan himpunan data *n* dimensi dengan *m* sampel setiap dimensinya.

Tabel 2.1 Himpunan data dengan 2 dimensi 4 sampel

X	Y
154	50
171	69
180	74
165	60

2. Penyesuaian data, yaitu mencari nilai rerata masing-masing dimensi dan mengurangi setiap nilai sampel dengan nilai rerata sesuai dengan dimensinya, $x_i - \bar{x}$, dimana $i = 1, 2, \dots, m$.

Tabel 2.2 Contoh data hasil penyesuaian

X	Y	$x_i - \bar{x}$	$y_i - \bar{y}$
$\bar{x} = 167,5$	$\bar{y} = 63,25$	Penyesuaian X	Penyesuaian Y
154	50	-13,5	-13,25
171	69	3,5	5,75
180	74	12,5	10,75
165	60	-2,5	-3,25

3. Membentuk matriks varians-kovarians dari data yang telah diperoleh menggunakan persamaan 2.1 dan 2.2. Matriks varians-kovarians ditunjukkan pada Contoh 1.

$$cov(X, Y) = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{n - 1} \tag{2.1}$$

$$C^{n \times n} = (c_{i,j}, c_{i,j} = cov(Dim_i, Dim_j)) \tag{2.2}$$

Dimana

C = matriks varians-kovarians

Dim_i, Dim_j = dimensi ke-i, dimensi ke-j

n = jumlah sampel setiap dimensi

cov(X, Y) = nilai kovarian antara dimensi X dan Y

Tabel 2.3 Contoh hasil perhitungan varians dan kovarians

$x_i - \bar{x}$	$y_i - \bar{y}$	$(x_i - \bar{x})^2$	$(y_i - \bar{y})^2$	$(x_i - \bar{x})(y_i - \bar{y})$
-13,5	-13,25	182,25	175,5625	178,875
3,5	5,75	12,25	33,0625	20,125
12,5	10,75	156,25	115,5625	134,375
-2,5	-3,25	6,25	10,5625	8,125
		$\Sigma = 375$	$\Sigma = 334,75$	$\Sigma = 314,5$

*Mahasiswa Jurusan Teknik Elektro UNDIP

**Staf Pengajar Jurusan Teknik Elektro UNDIP

Contoh 1:

$$C = \begin{bmatrix} 125 & 104,83 \\ 104,83 & 111,58 \end{bmatrix}$$

- Mencari vektor eigen dan nilai eigen dari matriks varians-kovarians yang memenuhi persamaan 2.3a dan 2.3b. Hasil perhitungan vektor eigen dan nilai eigen ditunjukkan pada Contoh 2a dan 2b.

$$|C - \lambda I| = 0 \tag{2.3a}$$

$$|C - \lambda I| \alpha = 0 \tag{2.3b}$$

Dimana

λ = nilai eigen

α = vektor eigen

Contoh 2a:

$$\alpha_1 = \begin{pmatrix} 0,6841 \\ -0,7293 \end{pmatrix} \lambda_1 = 13,2455$$

Contoh 2b:

$$\alpha_2 = \begin{pmatrix} -0,7293 \\ -0,6841 \end{pmatrix} \lambda_2 = 223,3345$$

- Mengelompokkan vektor-vektor eigen ke dalam suatu Matriks Ciri (*MC*), yang urutannya mulai dari vektor eigen dengan nilai terbesar ke vektor eigen dengan nilai lebih kecil.

Contoh 3:

$$MC = \begin{bmatrix} -0,7293 & 0,6841 \\ -0,6841 & -0,7293 \end{bmatrix}$$

- Membentuk data akhir dengan mengalikan operasi tranpos terhadap Matriks Ciri tersebut dan mengalikannya dengan himpunan data hasil penyesuaian yang telah ditranspos.

$$DA = MCB \times DPB \tag{2.4}$$

Dimana

DA = Data akhir

MCB = Matriks citra yang ditranspos

DPB = Data penyesuaian yang ditranspos

Contoh 4:

$$DA = \begin{bmatrix} -0,7293 & -0,6841 \\ 0,6841 & -0,7293 \end{bmatrix} \times \begin{bmatrix} -13,5 & 3,5 & 12,5 & -2,5 \\ -13,25 & 5,75 & 10,75 & -3,25 \end{bmatrix}$$

- Merekonstruksi kembali himpunan data yang asli menggunakan persamaan 2.5.

$$DPB = (MCB)^T \times DA \tag{2.5}$$

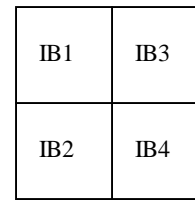
Dimana

DPB = Data penyesuaian yang ditranspos hasil rekonstruksi

III. PERANCANGAN SISTEM

Penerapan PCA pada Pemampatan Citra

Untuk melakukan pemampatan terhadap suatu citra, maka citra tersebut dibagi terlebih dahulu menjadi beberapa blok dan ukuran resolusi tiap blok citra tersebut harus sama.



Gambar 3.1 Pembagian citra menjadi beberapa blok

Blok-blok tersebut disusun sebagai kolom-kolom yang berbeda yang disebut sebagai matriks citra (*MCit*). Matriks citra ini merupakan representasi dari himpunan data dengan dimensi berupa blok-blok citra.

$$MCit = \begin{bmatrix} \text{vektor-1} \\ \text{vektor-2} \\ \vdots \\ \text{vektor-m} \end{bmatrix} = \begin{bmatrix} D_{1,1} & D_{1,2} & \dots & D_{1,n} \\ D_{2,1} & D_{2,2} & \dots & D_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ D_{m,1} & D_{m,2} & \dots & D_{m,n} \end{bmatrix} \tag{3.1}$$

Selanjutnya membentuk matriks penyesuaian (*MCitP*) dengan mengurangi setiap sampel dimensi dengan rerata dari dimensi yang bersangkutan.

$$MCitP = \begin{bmatrix} D_{1,1} - \bar{IB}_1 & D_{1,2} - \bar{IB}_2 & \dots & D_{1,n} - \bar{IB}_n \\ D_{2,1} - \bar{IB}_1 & D_{2,2} - \bar{IB}_2 & \dots & D_{2,n} - \bar{IB}_n \\ \vdots & \vdots & \ddots & \vdots \\ D_{m,1} - \bar{IB}_1 & D_{m,2} - \bar{IB}_2 & \dots & D_{m,n} - \bar{IB}_n \end{bmatrix} \tag{3.2}$$

Langkah selanjutnya adalah membentuk matriks varians-kovarians (*C*) dari matriks penyesuaian (*MCitP*). Matriks varians-kovarians selalu berbentuk bujursangkar dan berukuran $n \times n$, dimana n adalah jumlah dimensi (jumlah blok), karena matriks varians-kovarians merepresentasikan hubungan timbal balik antara dimensi/blok pada matriks penyesuaian.

$$C = \begin{bmatrix} \text{cov}(IB_1, IB_1) & \text{cov}(IB_1, IB_2) & \dots & \text{cov}(IB_1, IB_n) \\ \text{cov}(IB_2, IB_1) & \text{cov}(IB_2, IB_2) & \dots & \text{cov}(IB_2, IB_n) \\ \vdots & \vdots & \ddots & \vdots \\ \text{cov}(IB_n, IB_1) & \text{cov}(IB_n, IB_2) & \dots & \text{cov}(IB_n, IB_n) \end{bmatrix} \tag{3.3}$$

Dari matriks varians-kovarians dapat dicari vektor-vektor eigen beserta nilai eigen-nya. Vektor eigen ini diurutkan nilai eigen-nya dari terbesar ke terkecil, kemudian disusun ke dalam sebuah matriks ciri (*MC*). Matriks ciri selalu mempunyai jumlah kolom sebanyak n (jumlah dimensi/blok).

$$MC = [\alpha_1 \ \alpha_2 \ \dots \ \alpha_k] \tag{3.4}$$

dimana $k \leq n$ adalah jumlah vektor eigen yang diambil.

Untuk kepentingan pemampatan, sebagian vektor eigen yang nilai eigen-nya kecil dibuang dan disisakan sebanyak k vektor eigen. Dari vektor eigen yang tersisa pada matriks ciri dapat dibentuk sebuah matriks alihragam (*MA*).

$$MA = (MC)^T \tag{3.5}$$

Proses pemampatan dilakukan dengan mengalikan matriks alihragam dengan matriks penyesuaian yang ditranspos sehingga diperoleh matriks data akhir (*MDA*)

$$MDA = MA \times (MCitP)^T \tag{3.6}$$

Persamaan 3.7 digunakan untuk merekonstruksi citra. Hasil rekonstruksi berupa matriks penyesuaian terekonstruksi yang ditranspos.

$$(MCitP)^T = (MA)^T \times MDA \tag{3.7a}$$

$$MCitP = \{(MCitP)^T\}^T \tag{3.7b}$$

Matriks citra penyesuaian terekonstruksi akan tepat sama nilainya dengan matriks citra penyesuaian sebelum proses pemampatan citra) jika dan hanya jika seluruh vektor eigen diikutsertakan ke dalam matriks ciri.

Untuk mendapatkan kembali matriks citra maka matriks penyesuaian dijumlahkan dengan rerata \bar{x} .

$$(MCitRek) = MCitP + \bar{x} \tag{3.8}$$

Matriks citra ini kemudian disusun kembali menjadi blok-blok citra terekonstruksi.

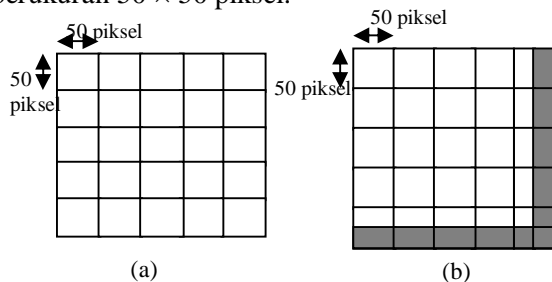
Algoritma Program

Algoritma Program Utama

1. Membaca citra yang akan dimampatkan.
2. Melakukan proses pengolahan awal.
3. Melakukan proses pemampatan citra.
4. Melakukan proses rekonstruksi citra.
5. Menampilkan hasil rekonstruksi citra.
6. Menskalakan data citra termampat.
7. Menyimpan data citra termampat terskala ke dalam berkas.
8. Membaca data citra termampat terskala dari berkas.
9. Menjalankan kembali algoritma program mulai langkah keempat.

Algoritma Pengolahan Awal

1. Membagi nilai elemen piksel dengan 255.
2. Menambahkan data dummy jika resolusi citra tidak habis dibagi 50.
3. Memecah citra menjadi blok-blok kecil yang berukuran 50 x 50 piksel.



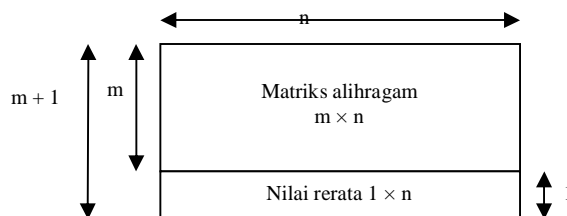
Gambar 3.2 (a) Pembentukan blok citra tanpa penambahan dummy; (b) Pembentukan blok citra dengan penambahan dummy

4. Menyusun matriks citra.
5. Menghitung rerata dari setiap kolom pada matriks citra.
6. Membentuk matriks penyesuaian dari matriks citra.
7. Membentuk matriks varians-kovarians.

8. Mencari vektor eigen dari matriks varians-kovarians.

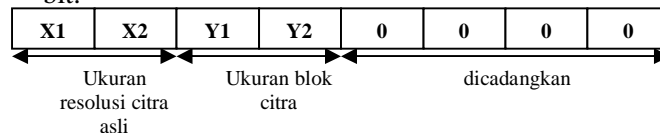
Algoritma Pemampatan Citra

1. Menyusun matriks alihragam dengan mentranspos vektor-vektor eigen terpilih.
2. Menghitung informasi ukuran resolusi citra asli yang diperlukan saat rekonstruksi citra untuk menghilangkan data dummy.
3. Menghitung informasi ukuran blok citra yang diperlukan saat rekonstruksi citra untuk membentuk kembali potongan-potongan blok citra.
4. Memampatkan citra dengan mengalikan matriks alihragam dan matriks penyesuaian yang telah ditranspos.
5. Membuat footer citra yang berisi matriks alihragam dan nilai rerata.



Gambar 3.3 Format footer citra

6. Membuat header citra yang berukuran 1 x 8 x 32 bit.



Gambar 3.4 Format header citra saat proses pemampatan

Algoritma Rekonstruksi Citra

1. Mempersiapkan data citra termampat, header citra, dan footer citra.
2. Mengekstrak informasi nilai maksimum dan nilai minimum data citra termampat dari header citra.
3. Mengekstrak matriks alihragam dari footer citra.
4. Mengekstrak nilai rerata dari footer citra.
5. Mengekstrak informasi ukuran blok citra dari header citra.
6. Mengekstrak informasi ukuran resolusi citra asli dari header citra.
7. Menghitung matriks penyesuaian terekonstruksi melalui proses alihragam balik dengan mentranspos hasil perkalian antara matriks alihragam yang telah ditranspos dengan data citra termampat.
8. Menambahkan matriks penyesuaian terekonstruksi dengan nilai rerata sehingga diperoleh matriks citra terekonstruksi.
9. Mengubah matriks citra terekonstruksi menjadi potongan-potongan blok citra menggunakan

informasi ukuran blok citra, kemudian menyatukan kembali menjadi satu citra utuh.

10. Menghilangkan data *dummy* dari citra tersebut dengan cara dipotong berdasarkan ukuran resolusi citra asli dengan posisi titik kiri atas citra sebagai patokannya.

Algoritma Penskalaan Data Citra Termampat

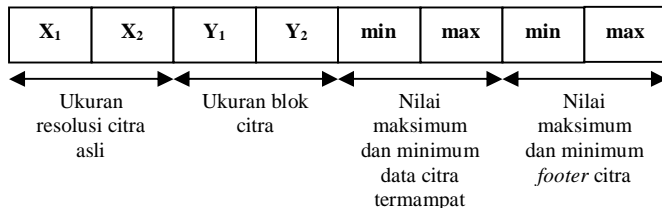
1. Menyiapkan data citra termampat, *header* citra, dan *footer* citra.
2. Mencari nilai maksimum dan nilai minimum dari data citra termampat dan *footer* citra.
3. Menskalakan data citra termampat dan *footer* citra menggunakan persamaan 3.9.

$$NT = \text{round}\left(\frac{NA - N_{\min}}{(N_{\max} - N_{\min})/255}\right) \quad (3.9)$$

Dimana :

- NA = Nilai asli
- NT = Nilai terskala
- Nmax = Nilai maksimum
- Nmin = Nilai minimum

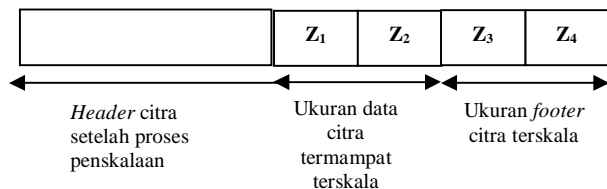
4. Meletakkan informasi nilai maksimum dan nilai minimum data citra termampat serta *footer* citra secara berurutan pada kolom ke-5 dan ke-6 serta kolom ke-7 dan ke-8 *header* citra.



Gambar 3.5 Format *header* citra setelah proses penskalaan

Algoritma Penyimpanan ke Berkas

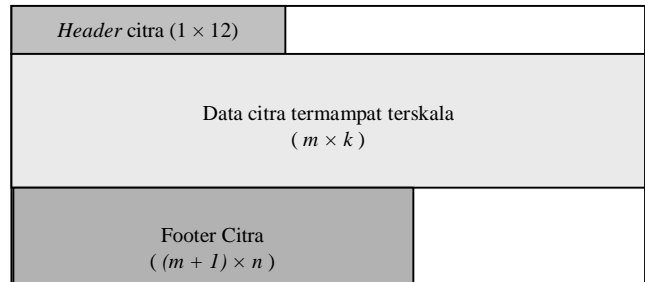
1. Menyiapkan data citra termampat terskala, *header* citra, dan *footer* citra terskala.
2. Menghitung resolusi citra termampat terskala.
3. Menghitung resolusi *footer* citra terskala.
4. Menambahkan informasi resolusi citra termampat terskala dan resolusi *footer* citra terskala ke dalam *header* citra.



Gambar 3.6 Format *header* citra saat proses penyimpanan

5. Menyiapkan berkas dengan kondisi *append* supaya dapat ditulisi dengan data citra termampat terskala, *header* citra, dan *footer* citra terskala.

6. Menyimpan *header* citra ke berkas dengan tipe data *single* (32-bit).
7. Menyimpan data citra termampat terskala yang bertipe data *integer* tak bertanda 8-bit (*uint8*) ke dalam berkas.
8. Menyimpan *footer* citra terskala yang bertipe data *integer* tak bertanda 8-bit (*uint8*) ke dalam berkas.
9. Menutup kembali berkas yang telah diisi.



Gambar 3.7 Format data yang disimpan ke berkas

Algoritma Pembacaan Data dari Berkas

1. Menentukan nama dan lokasi berkas.
2. Membuka dan menyiapkan berkas dengan kondisi *read*.
3. Membaca *header* citra dari dalam berkas.
4. Mengambil informasi ukuran data citra termampat terskala dari *header* citra.
5. Mengambil informasi ukuran *footer* citra terskala dari *header* citra.
6. Membaca data citra termampat terskala yang bertipe *uint8* dari berkas berdasarkan informasi ukuran data citra termampat terskala.
7. Membaca *footer* citra terskala yang bertipe *uint8* dari berkas berdasarkan informasi ukuran *footer* citra terskala.
8. Menutup kembali berkas yang sudah dibaca isinya.
9. Membuang informasi ukuran data citra termampat terskala dan ukuran *footer* citra terskala dari *header* citra. Informasi ini dibuang karena sudah tidak digunakan lagi.

Algoritma Penskalaan Balik

1. Mempersiapkan data citra termampat terskala dan *footer* citra termampat terskala serta nilai maksimum dan nilai minimumnya masing-masing yang diambil dari *header* citra.
2. Melakukan proses penskalaan balik menggunakan persamaan 3.10.

$$NA = NT \times ((N_{\max} - N_{\min}) / 255) + N_{\min} \quad (3.10)$$

Dimana :

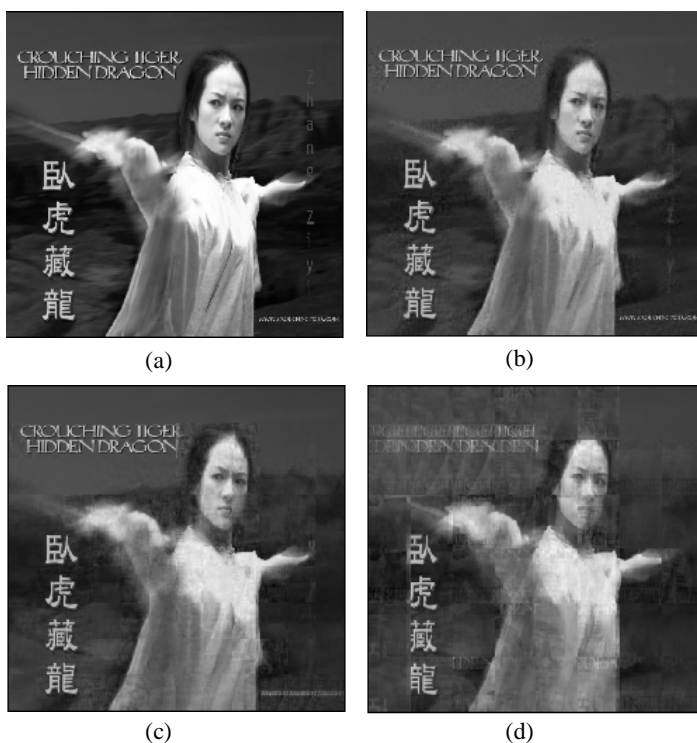
- NA = Nilai asli
- NT = Nilai terskala
- Nmaks = Nilai maksimum
- Nmin = Nilai minimum

IV. HASIL PENELITIAN DAN PEMBAHASAN

Pengujian Menggunakan Tiga Citra dengan Resolusi Berbeda

Pengujian dilakukan menggunakan tiga citra yang resolusinya berbeda-beda kemudian dimampatkan dengan variasi nilai prosentase pemampatan, kemudian hasilnya dibandingkan satu sama lain. Citra yang akan diujikan adalah **ziyi.bmp** yang mempunyai resolusi 800×600 , 600×450 , 400×300 . Tujuan pengujian ini untuk mengetahui pengaruh resolusi citra masukan terhadap kualitas citra hasil rekonstruksi.

Untuk mempersingkat pembahasan, hanya ditampilkan hasil pengujian untuk nilai prosentase pemampatan 70% untuk ketiga citra yang ditunjukkan pada Gambar 3.8.



Gambar 3.8 (a) Citra **ziyi.bmp** asli; (b) Citra rekonstruksi **ziyi.bmp** 800×600 dengan prosentasi pemampatan 70%; (c) Citra rekonstruksi **ziyi.bmp** 600×450 dengan prosentasi pemampatan 70%; (d) Citra rekonstruksi **ziyi.bmp** 400×300 dengan prosentasi pemampatan 70%;

Secara singkat hasil pengujiannya adalah sebagai berikut. Hasil rekonstruksi citra resolusi 800×600 mulai buruk saat rasio pemampatan di atas 75%. Hasil rekonstruksi citra resolusi 600×450 mulai buruk saat rasio pemampatan di atas 60%. Hasil rekonstruksi citra resolusi 800×600 mulai buruk saat rasio pemampatan di atas 40%.

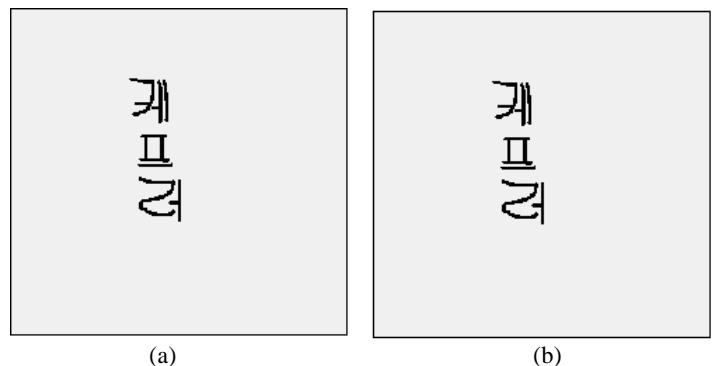
Hasil pengujian menunjukkan citra yang resolusinya besar lebih tahan terhadap prosentase pemampatan yang tinggi. Indikatornya adalah hasil

rekonstruksinya lebih baik dibanding dengan citra yang resolusinya lebih rendah. Hal ini disebabkan pada citra yang resolusinya tinggi, bidang citra yang dicakup oleh setiap potongan blok lebih sempit, sehingga jumlah dimensi (ruang pengukuran) akan semakin banyak. Akibatnya identifikasi pola antar dimensi akan semakin detail dan akurat yang ditunjukkan dengan semakin banyaknya jumlah vektor eigen.

Pengujian Menggunakan Tiga Citra Berbeda dengan Resolusi yang Sama

Pengujian dilakukan menggunakan tiga citra berbeda namun resolusinya sama. Citra yang diujikan adalah citra **hangeul.bmp**, **ziyi.bmp**, dan **hutan.bmp**. Ketiganya mempunyai resolusi 400×400 piksel. Tujuan pengujian ini untuk mengetahui pengaruh detail/tekstur citra terhadap kualitas citra hasil rekonstruksi.

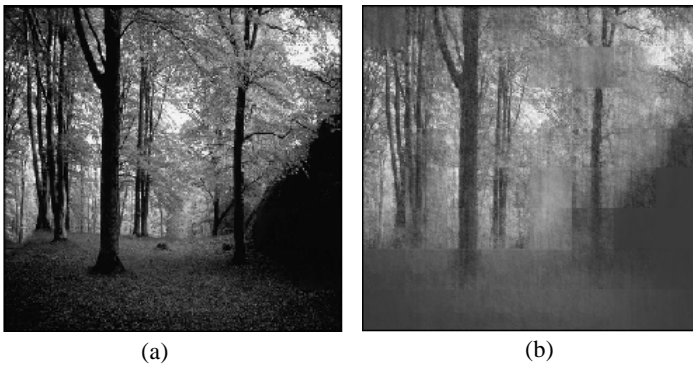
Untuk mempersingkat pembahasan, hanya ditampilkan hasil pengujian untuk nilai prosentase pemampatan 70% untuk ketiga citra yang ditunjukkan pada Gambar 3.9, 3.10, dan 3.11.



Gambar 3.9 (a) Citra **hangeul.bmp** asli; (b) Citra rekonstruksi **hangeul.bmp** dengan prosentase pemampatan 70%



Gambar 3.10 (a) Citra **ziyi.bmp** asli; (b) Citra rekonstruksi **ziyi.bmp** dengan prosentase pemampatan 70%



Gambar 3.11 (a) Citra **hutan.bmp** asli; (b) Citra rekonstruksi **hutan.bmp** dengan prosentase pemampatan 70%

Dari ketiga citra yang diuji, dapat diamati hasil rekonstruksi citra **hangeul.bmp** untuk prosentase pemampatan 70% masih lebih baik dibanding kedua citra lainnya. Hasil rekonstruksi citra **ziyi.bmp** sedikit lebih baik dibanding dengan hasil rekonstruksi citra **hutan.bmp**. Hal ini disebabkan detail/tekstur dari citra **hangeul.bmp** lebih seragam dibandingkan dengan dua citra yang lain, begitu juga tekstur citra **ziyi.bmp** teksturnya lebih seragam dibandingkan citra **hutan.bmp**. Citra yang bertekstur seragam, persebaran nilai eigen-nya cenderung mengelompok pada vektor-vektor eigen tertentu, sehingga banyak vektor-vektor eigen yang nilainya 0 atau mendekati 0. Citra seperti ini lebih tahan terhadap prosentase pemampatan yang tinggi.

Secara singkat hasil pengujiannya adalah sebagai berikut. Citra **hangeul.bmp** menunjukkan bahwa untuk rasio pemampatan hingga 85% hasil rekonstruksinya masih baik. Berdasarkan pengamatan mata, tampak citra **ziyi.bmp** dan **hutan.bmp** hingga 30% hasil rekonstruksinya masih baik, tetapi berdasarkan persebaran nilai eigen pada vektor eigen-nya citra **ziyi.bmp** sedikit lebih baik dibandingkan citra **hutan.bmp**.

Pengaruh Penambahan Data Dummy terhadap Ukuran Berkas Termampat

Pengujian dilakukan menggunakan dua citra yang mempunyai resolusi berbeda dan dimampatkan dengan prosentase pemampatan yang sama. Citra1 mempunyai resolusi 177×181 yang tidak habis dibagi 50, citra2 mempunyai resolusi 200×200 yang habis dibagi 50.

Hasil pengujiannya ditunjukkan pada tabel 4.1. Citra1 yang resolusinya lebih kecil (177×181) ukuran berkas hasil pemampatannya sama dengan ukuran berkas pemampatan citra2 yang resolusinya lebih besar (200×200). Secara wajar seharusnya ukuran berkas hasil pemampatan citra1 lebih kecil dibandingkan dengan citra2 karena prosentase pemampatannya sama

yaitu 40%. Hal ini menunjukkan bahwa penambahan data *dummy* mengakibatkan ukuran berkas hasil pemampatan kurang optimal. Ukuran berkas merupakan hasil penjumlahan dari ukuran *header* citra, ukuran data citra, dan ukuran *footer* citra.

Tabel 4.1 Hasil pengujian penambahan data *dummy* terhadap ukuran berkas citra termampat dengan prosentase pemampatan 40%

Keterangan Pengujian	Citra 1 (177×181 piksel)	Citra 2 (200×200 piksel)
Ukuran Citra Asli	33,658 kB	41,078 kB
Total Piksel Asli	32037 piksel	40000 piksel
Ukuran Citra Setelah Ditambah <i>Dummy</i> (perhitungan)	33,658 kB + 7,963 kB = 41,621 kB	41,078 kB + 0 kB = 41
Total Piksel Setelah Ditambah <i>Dummy</i>	32037 piksel + 7963 = 40000 piksel	40000 piksel + 0 piksel = 40000 piksel
Ukuran <i>Header</i> citra	1 x 8 piksel = 32 byte	1 x 8 piksel = 32 byte
Ukuran data citra	10 x 2500 piksel = 25 kB	10 x 2500 piksel = 25 kB
Ukuran <i>footer</i> citra	11 x 16 piksel = 0,176 kB	11 x 16 piksel = 0,176 kB
Ukuran berkas termampat	25,224 kB	25,224 kB
Rasio pemampatan	74,942%	61,405%

Ukuran *header* citra selalu tetap untuk resolusi berapa pun, namun data citra dan *footer* citra ukurannya akan berubah-ubah terhadap resolusi citra asli maupun terhadap prosentase pemampatan karena isi dari data citra adalah data hasil pemampatan yang dipengaruhi oleh ukuran resolusi citra asli dan prosentase pemampatan, sedangkan *footer* citra berisi matriks alihragam dan nilai rerata yang ukurannya juga dipengaruhi oleh resolusi citra asli dan prosentase pemampatan.

Dengan penambahan data *dummy*, ukuran data citra1 dan *footer* citra1 (resolusi 171×181 piksel) menjadi sama dengan citra2 (resolusi 200×200 piksel) karena resolusi pembulatan terdekat yang habis dibagi 50 adalah 200×200 piksel, yang kebetulan resolusinya sama dengan citra2. Hal inilah yang menyebabkan ukuran berkas pemampatan citra1 menjadi lebih besar dari yang seharusnya.

V. PENUTUP

Kesimpulan

Berdasarkan hasil pengujian, analisa, dan pembahasan maka dapat diambil kesimpulan sebagai berikut.

1. Citra yang resolusinya besar lebih tahan terhadap prosentase pemampatan yang tinggi dibandingkan citra yang resolusinya lebih kecil, karena setiap blok (dimensi) citra menyajikan bagian citra yang lebih sempit sehingga jumlah vektor eigen-nya menjadi lebih banyak. Dengan membandingkan blok-blok citra yang lebih sempit maka hasilnya akan lebih akurat.
2. Semakin tinggi prosentase pemampatan maka kualitas citra rekonstruksi akan semakin buruk karena komponen utama pada vektor-vektor eigen-nya lebih banyak terbuang.
3. Citra yang teksturnya detail, persebaran komponen utamanya pada vektor eigen lebih merata dibandingkan citra yang kurang detail. Citra yang kurang detail, persebaran komponen utama cenderung mengelompok pada vektor-vektor eigen tertentu.
4. Persebaran komponen utama mengelompok pada vektor-vektor eigen tertentu mengakibatkan sebagian besar vektor eigen-nya bernilai 0. Citra dengan karakteristik seperti ini lebih tahan terhadap prosentase pemampatan yang tinggi, karena sebagian besar vektor eigen yang bernilai 0 dapat dibuang. Menghilangkan vektor eigen dalam jumlah besar sama artinya dengan menerapkan prosentase pemampatan yang tinggi.
5. Citra yang sangat detail kurang tahan terhadap prosentase pemampatan yang tinggi karena persebaran komponen utama pada vektor-vektor eigen-nya lebih merata. Hampir sebagian besar vektor eigen-nya mempunyai nilai eigen, sehingga jika diterapkan prosentase pemampatan yang tinggi maka vektor-vektor eigen yang nilai eigen-nya tidak 0 akan ikut terbuang.
6. Penambahan data *dummy* mengakibatkan ukuran berkas termampat menyimpang dari nilai rasio pemampatan yang seharusnya. Besarnya penyimpangan ini tergantung nilai pembulatan resolusi terdekat yang habis dibagi 50.
7. Prosentase pemampatan yang semakin besar mengakibatkan ukuran berkas termampat akan semakin kecil terutama untuk citra yang resolusinya tinggi dan citra yang ditambah dengan data *dummy*. Tetapi kualitas hasil rekonstruksinya akan semakin buruk.
8. Faktor yang mempengaruhi hubungan antara prosentase pemampatan dengan kualitas hasil

rekonstruksi adalah : ukuran resolusi citra, nilai prosentase pemampatan itu sendiri, serta detail citra. Semakin besar resolusi citra, kualitas rekonstruksi citra akan semakin tahan untuk prosentase pemampatan yang tinggi. Semakin kecil detail citra, kualitas rekonstruksi citra akan semakin tahan untuk prosentase pemampatan yang tinggi.

9. Faktor-faktor yang mempengaruhi ukuran berkas termampat adalah resolusi citra dan penambahan data *dummy*.

DAFTAR PUSTAKA

- [1]. Pitas, I., *Digital Image Processing Algorithms*, Prentice Hall, 1993.
- [2]. Godbole, A.S., *Data Communication And Networks*, McGraw Hill, 2003.
- [3]. Jain, A. K., *Fundamentals of Digital Image Processing*, Prentice Hall, 1989.
- [4]. Calvo, R. and M. Partridge, *Fast Dimensionality Reduction and Simple PCA*, Departement of Electrical Engineering University of Sydney NSW, Australia, 1997.
- [5]. Smith, L.I., *A tutorial on Principal Components Analysis*, <http://vision.auc.dk/sig/>, 2002.
- [6]. Gunadi, K. dan S.R. Pongsitanan, *Pembuatan Perangkat Lunak Pengenalan Wajah Menggunakan Principal Components Analysis*, Universitas Kristen Petra, kgunadi@peter.petra.ac.id.
- [7]. Sid-Ahmed, M.A., *Image Processing Theory, Algorithms, and Architectures*, McGraw-Hill, Inc., 1995.
- [8]. Hanselman, D. and B. Littlefield, *Mastering MATLAB 5 A Comprehensive Tutorial and Reference*, Prentice Hall, Inc., 1998.
- [9]. Wu, M., *Review 1 : Image Processing Basics & Compression*, Electrical & Computer Engineering University of Maryland, College Park, 2001.



Praditya

L2F 000 627

Lahir tanggal 29 September 1982 di Semarang. Saat ini sedang menyelesaikan pendidikan Strata 1 di Jurusan Teknik Elektro Universitas Diponegoro dengan konsentrasi Elektronika Telekomunikasi

Mengetahui/Mengesahkan
Pembimbing I

Budi Setiyono, S.T., M.T.

NIP. 132 283 184

Pembimbing II

R. Rizal Isnanto, S.T., M.M., M.T.

NIP. 132 288 515

