

PEMAMPATAN DAN REKONSTRUKSI CITRA BERWARNA 24-BIT MENGUNAKAN ANALISIS KOMPONEN UTAMA (PCA)

Rofi Yuliansyah¹, Budi Setiyono², R. Rizal Isnanto²

Abstrak - Selama ini penelitian mengenai pemampatan dan rekonstruksi citra digital masih menggunakan teknik pemampatan konvensional, seperti JPEG, PNG, GIF, dan JPEG 2000 yang telah menjadi standar dari teknik pemampatan data citra digital. Teknik yang digunakan pada teknik pemampatan konvensional bukan menggunakan analisis mengenai komponen-komponen pembentuk data citra. Oleh sebab itu perlu dilakukan penelitian mengenai pemampatan dan rekonstruksi citra menggunakan metode Analisis Komponen Utama (PCA) untuk menganalisis komponen pembentuk data citra.

PCA melakukan pemampatan dengan cara menyajikan sekumpulan ciri-ciri citra ke dalam sebuah ruang baru yaitu ruang eigen dengan setiap ciri tidak saling berkorelasi, kemudian menghilangkan beberapa ciri citra atau vektor eigen yang kurang penting. Langkah pertama yang dilakukan dalam melakukan pemampatan data citra menggunakan PCA adalah menyiapkan citra yang akan dimampatkan. Langkah yang kedua adalah melakukan pengolahan awal terhadap citra, yaitu dengan memisahkan komponen RGB menjadi komponen R, G, dan B, kemudian mengolah masing-masing komponen sampai mendapatkan vektor-vektor eigen. Langkah yang ketiga melakukan proses pemampatan dengan cara membentuk data citra termampat dan matriks alihragam yang berisi vektor-vektor eigen terpilih. Langkah yang terakhir adalah merekonstruksi citra dengan cara mengalihragamkan balik data citra termampat.

Hasil pengujian sistem menunjukkan bahwa penggunaan PCA untuk pemampatan data citra dapat memberikan tingkat prosentase pemampatan yang cukup tinggi dengan hasil citra rekonstruksi sama dengan citra asli. Untuk citra yang diujikan dapat diperoleh prosentase pemampatan terbaik sebesar 60% yaitu citra dengan resolusi tertinggi sedangkan prosentase pemampatan terburuk adalah 20% untuk citra yang memiliki tekstur lebih detail dibandingkan dengan citra uji lainnya. Dengan demikian, aplikasi yang dibuat sudah cukup layak digunakan untuk metode pemampatan citra digital.

Kata-kunci : Analisis Komponen Utama, citra digital, resolusi, vektor eigen

I. PENDAHULUAN

Citra digital adalah gambar dua dimensi yang dihasilkan dari gambar analog dua dimensi yang kontinu menjadi gambar diskrit melalui proses sampling. Citra digital merupakan suatu data yang membutuhkan ruang memori cukup besar untuk menyimpannya.

Ukuran berkas citra bergantung pada resolusi dan kedalaman warna yang digunakan. Oleh sebab itu perlu dilakukan pemampatan terhadap data citra. Teknik pemampatan secara umum terbagi dua, yaitu pemampatan berugi (lossy) dan tak berugi (lossless). Pada tugas akhir ini dikembangkan teknik pemampatan citra menggunakan Analisis Komponen Utama (PCA). Teknik PCA termasuk ke dalam teknik pemampatan berugi.

II. TEORI ANALISIS KOMPONEN UTAMA

Analisis komponen utama adalah teknik statistik untuk mengidentifikasi pola dalam suatu data dan mengekspresikan suatu data sedemikian rupa sehingga diperoleh persamaan dan perbedaannya. PCA melakukan pemampatan dengan cara menyajikan sekumpulan ciri-ciri citra kedalam sebuah ruang baru yaitu ruang eigen dengan setiap ciri tidak saling berkorelasi, kemudian menghilangkan beberapa ciri citra atau vektor eigen yang kurang penting. Teknik ini disebut juga alihragam Karhunen-Loeve atau alihragam Hotelling.

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

Tinggi (X)	Berat (Y)
154 cm	50 kg
171 cm	69 kg
180 cm	74 kg
165 cm	60 kg

2. Penyesuaian data, yaitu mencari nilai rerata masing-masing dimensi dan mengurangi setiap nilai sampel dengan nilai rerata sesuai dengan dimensinya.

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 cm	50 kg	-13,5	-13,25
171 cm	69 kg	3,5	5,75
180 cm	74 kg	12,5	10,75
165 cm	60 kg	-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.

1) Mahasiswa Jurusan Teknik Elektro UNDIP
2) Staff Pengajar Jurusan Teknik Elektro UNDIP

$$\text{cov}(X, Y) = \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{n-1} \quad (2.1)$$

$$C^{n \times n} = (c_{i,j}, c_{i,j} = \text{cov}(\text{Dim}_i, \text{Dim}_j)) \quad (2.2)$$

Dimana

C = Matriks varians-kovarians

$\text{Dim}_i, \text{Dim}_j$ = Dimensi ke-I, dimensi ke-j

n = Jumlah sampel setiap dimensi

$\text{cov}(X, Y)$ = Nilai kovarian antara dimensi X dan Y

Contoh 1:

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

4. 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.

$$|C - \lambda I| = 0 \quad (2.3a)$$

$$|C - \lambda I| \alpha = 0 \quad (2.3b)$$

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:

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

5. 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:

$$\text{Matriks Ciri} = \begin{bmatrix} -0,7293 & 0,6841 \\ -0,6841 & -0,7293 \end{bmatrix}$$

6. Membentuk data akhir dengan mengalikan operasi tranpos terhadap matriks ciri tersebut dan mengalikannya dengan himpunan data hasil penyesuaian yang telah ditranspos.

$$DA = MCB \times DPB \quad (2.4)$$

Dimana

DA = Data akhir

MCB = Matriks citra yang ditranspos

DPB = Data penyesuaian yang ditranspos

Contoh 4:

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

7. Merekonstruksi kembali himpunan data yang asli menggunakan persamaan 2.5.

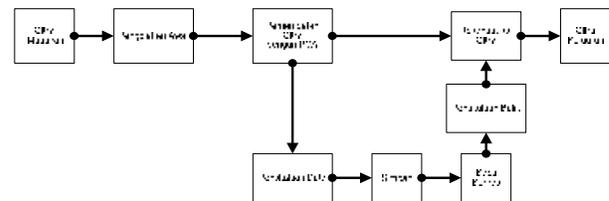
$$DPB = (MCB)^T \times DA \quad (2.5)$$

Dimana

DPB = Data penyesuaian yang ditranspos hasil rekonstruksi

III. PERANCANGAN SISTEM

3.1 Penerapan PCA pada Pemampatan Citra



Gambar 3.1 Diagram blok program utama

Untuk melakukan pemampatan terhadap suatu citra berwarna, maka citra tersebut harus dipisahkan dahulu komponen-komponen warnanya menjadi komponen warna satuan, yaitu komponen warna merah(R), hijau(G) dan biru(B). Tiap komponen warna harus dibagi terlebih dahulu menjadi beberapa blok dan ukuran resolusi tiap blok citra tersebut harus sama.

IB1	IB3
IB2	IB4

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.

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

Selanjutnya membentuk matriks penyesuaian ($MCitP$) dengan mengurangkan setiap sample

dimensi dengan rerata dari dimensi yang bersangkutan.

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

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

$$C = \begin{bmatrix} \text{cov}(B_1, IB_1) & \text{cov}(B_1, IB_2) & \dots & \text{cov}(B_1, IB_n) \\ \text{cov}(B_2, IB_1) & \text{cov}(B_2, IB_2) & \dots & \text{cov}(B_2, IB_n) \\ \vdots & \vdots & \ddots & \vdots \\ \text{cov}(B_n, IB_1) & \text{cov}(B_n, IB_2) & \dots & \text{cov}(B_n, IB_n) \end{bmatrix} \quad (3.3)$$

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

$$MC = [\alpha_1 \quad \alpha_2 \quad \dots \quad \alpha_k] \quad (3.4)$$

Dimana $k = n$ 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 kembali sebuah matriks alihragam (MA).

$$MA = (MC)^T \quad (3.5)$$

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

$$MDA = MA \times (MCitP)^T \quad (3.6)$$

Untuk merekonstruksi citra digunakan persamaan 3.7. hasil rekonstruksi berupa matriks penyesuaian terekonstruksi yang ditranspos.

$$(MCitP)^T = (MA)^T \times MDA \quad (3.7a)$$

$$MCitP = \left\{ (MCitP)^T \right\}^T \quad (3.7b)$$

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

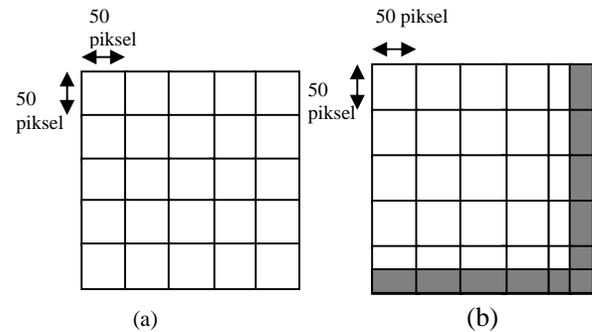
Untuk mendapatkan kembali matriks citra maka matriks penyesuaian dijumlahkan dengan rerata \bar{X} .

$$(MCitRek) = MCitP + \bar{X} \quad (3.8)$$

Matriks citra kemudian disusun kembali menjadi blok-blok citra terekonstruksi. Blok citra kemudian digabungkan kembali menjadi satu citra utuh. Citra utuh ini masih citra berbentuk citra satuan (komponen R, G, dan B), untuk itu citra komponen R, G, dan B disatukan kembali menjadi citra RGB.

3.2 Pengolahan Awal

1. Membaca citra yang akan dimampatkan
2. Memisahkan komponen warna dari citra masukan menjadi komponen warna satuan R, G dan B.
3. Membagi nilai elemen piksel komponen warna R dengan 255.
4. Memecah citra komponen R menjadi blok-blok kecil yang berukuran 50×50 piksel.
5. Menambahkan data *dummy* jika resolusi citra tidak habis dibagi 50.

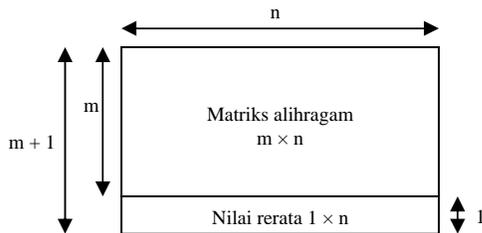


Gambar 3.2 (a) Pembentukan blok citra tanpa penambahan *dummy*
(b) Pembentukan blok citra dengan penambahan *dummy*

6. Menyusun matriks citra.
7. Menghitung rerata dari setiap kolom pada matriks citra
8. Membentuk matriks penyesuaian dari matriks citra
9. Membentuk matriks varians-kovarians.
10. Mencari vektor eigen dari matriks varians-kovarians.
11. Setelah proses pengolahan komponen warna R selesai, dilakukan proses pengolahan komponen warna G dan B seperti proses pengolahan komponen warna R.

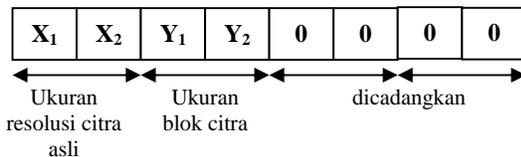
3.3 Pemampatan citra

1. Menyusun matriks alihragam komponen warna R 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 dengan 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 $1 \times 8 \times 32$ bit.



Gambar 3.4 Format header citra saat proses pemampatan

7. Melakukan proses pemampatan untuk komponen warna citra lainnya.

3.4 Rekonstruksi Citra

1. Mempersiapkan data citra termampat, *header* citra, *footer* citra komponen warna citra R.
2. Mengekstrak informasi nilai maksimum dan 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 .
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 komponen warna R.

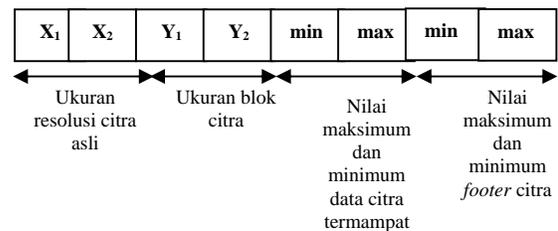
10. Menghilangkan data *dummy* dari citra tersebut dengan cara dipotong berdasarkan ukuran resolusi citra asli.
11. Melakukan proses rekonstruksi untuk citra komponen warna G dan B.
12. Menyatukan citra komponen R,G, dan B menjadi citra RGB terekonstruksi.

3.5 Penskalaan Data Citra Termampat.

1. Menyiapkan data citra termampat, *header* citra, dan *footer* citra komponen warna R,G, dan B.
2. Mencari nilai maksimum dan minimum dari data citra termampat dan *footer* citra tiap komponen warna.
3. Menskalaan data citra termampat dan *footer* citra tiap komponen warna menggunakan persamaan 3.9.

$$\text{nilai terskala} = \text{round} \left(\frac{\text{nilai asli} - \text{nilai min}}{(\text{nilai maks} - \text{nilai min})/255} \right) \quad (3.9)$$

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.

3.6 Penyimpanan Data ke Berkas

1. Menyiapkan data citra termampat terskala, *header* citra, dan *footer* citra terskala tiap komponen warna.
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.
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

Header Citra R (1×8)	
Header Citra G (1×8)	
Header Citra B (1×8)	
Data citra termampat terskala R ($m \times k$)	
Data citra termampat terskala G ($m \times k$)	
Data citra termampat terskala B ($m \times k$)	
Footer Citra R ($((m+1) \times n)$)	
Footer Citra G ($((m+1) \times n)$)	
Footer Citra B ($((m+1) \times n)$)	

Gambar 3.6 Format data yang disimpan ke berkas

3.7 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 berdasarkan informasi ukuran data citra termampat terskala.
7. Membaca footer citra terskala berdasarkan informasi ukuran footer citra terskala.
8. Menutup kembali berkas yang sudah dibacanya.
9. Membuang informasi ukuran data citra termampat terskala dan ukuran footer citra terskala dari header citra.

3.8 Penskalaan Balik

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

$$NA = NT \times \left(\frac{Nmaks - Nmin}{255} \right) + Nmin \quad (3.10)$$

Dimana:

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

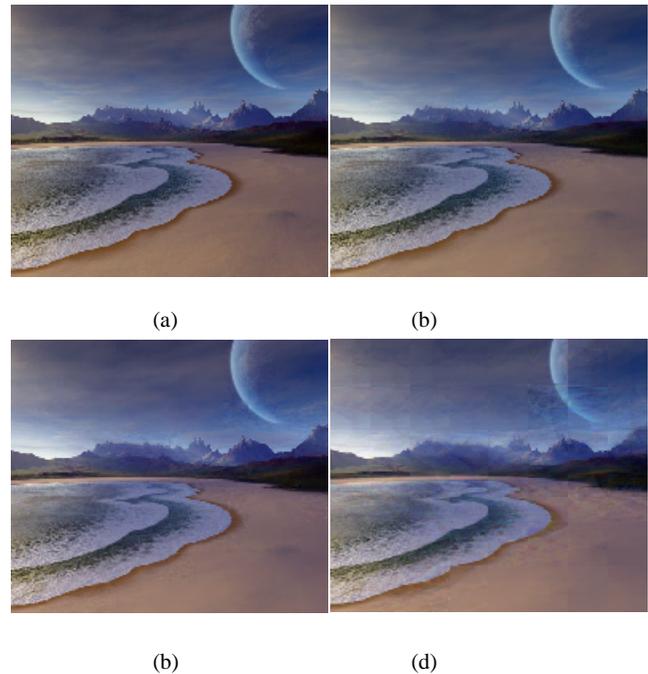
IV. HASIL PENELITIAN DAN PEMBAHASAN

4.1 Pengujian Menggunakan Tiga Citra dengan Resolusi Berbeda

Beberapa pengujian dilakukan terhadap perangkat lunak pemampatan citra berwarna

menggunakan analisis komponen utama dengan tujuan untuk mengetahui faktor yang mempengaruhi proses pemampatan citra menggunakan perangkat lunak ini. Pada pengujian pertama dilakukan pengujian menggunakan tiga citra dengan resolusi yang berbeda-beda. Setiap citra uji akan dimampatkan dengan nilai prosentase pemampatan yang berbeda-beda. Citra yang akan diuji adalah citra **beachside.bmp** yang memiliki resolusi 800×600 piksel, 600×450 piksel, dan 400×300 piksel.

Untuk mempersingkat pembahasan, hanya ditampilkan hasil pengujian untuk nilai prosentase pemampatan 60% untuk ketiga citra yang ditunjukkan pada Gambar 4.1.



Gambar 4.1 (a) Citra **beachside.bmp** asli
 (b) Citra rekonstruksi **beachside.bmp** 800×600
 (c) Citra rekonstruksi **beachside.bmp** 600×450
 (d) Citra rekonstruksi **beachside.bmp** 400×300

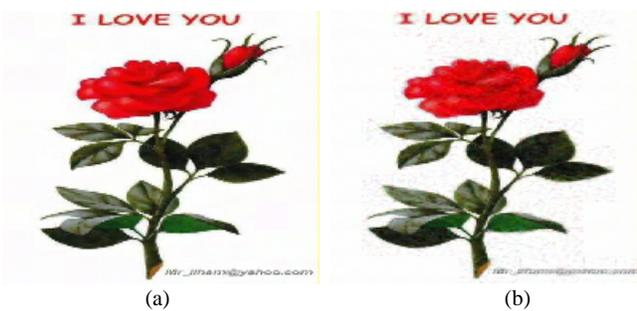
Secara singkat hasil pengujiannya adalah sebagai berikut. Hasil rekonstruksi citra resolusi 800×600 menunjukkan prosentase pemampatan tertinggi dengan citra rekonstruksi masih terlihat seperti citra asli adalah 60%. Hasil rekonstruksi citra resolusi 600×450 , prosentase pemampatan tertingginya adalah 50% dengan kualitas citra rekonstruksi seperti citra asli. Hasil rekonstruksi citra resolusi 400×300 , prosentase pemampatan tertinggi yang didapat adalah 40% dengan kualitas citra rekonstruksi masih terlihat seperti citra asli.

Hasil pengujian menunjukkan citra yang resolusinya besar jika dimampatkan dengan prosentase pemampatan yang tinggi, citra rekonstruksinya lebih baik dibanding dengan citra yang resolusinya lebih rendah.

4.2 Pengujian Menggunakan Tiga Citra Berbeda dengan Resolusi yang Sama

Pengujian dilakukan menggunakan tiga citra berbeda dengan resolusi yang sama. Citra yang diujikan adalah citra **rose.bmp**, **empire.bmp**, dan **beachside.bmp**. ketiganya mempunyai resolusi 500×500 piksel. pengujian ini dilakukan dengan tujuan untuk mengetahui kualitas citra hasil rekonstruksi dibandingkan citra aslinya maupun dengan citra yang lain dengan resolusi yang sama jika dimampatkan dengan variasi nilai prosentase pemampatan, selain itu akan diketahui citra seperti apa yang baik dan cocok untuk dilakukan pemampatan dengan analisis komponen utama.

Untuk mempersingkat pembahasan, hanya ditampilkan hasil pengujian untuk nilai prosentase pemampatan 60% untuk ketiga citra yang ditunjukkan pada Gambar 4.2, 4.3, 4.4.



Gambar 4.2 (a) Citra **rose.bmp** asli
(b) Citra rekonstruksi **rose.bmp**



Gambar 4.3 (a) Citra **empire.bmp** asli
(b) Citra rekonstruksi **empire.bmp**



Gambar 4.4 (a) Citra **beachside.bmp** asli
(b) Citra rekonstruksi **beachside.bmp**

Dari ketiga citra yang diuji, hasil rekonstruksi menunjukkan berdasarkan penglihatan, prosentase pemampatan tertinggi dengan kualitas citra rekonstruksi masih terlihat seperti citra masukan untuk 3 citra dengan resolusi yang sama yaitu 500×500 piksel adalah 40% untuk citra **rose.bmp**, 20% untuk citra **empire.bmp**, dan 50% untuk citra **beachside.bmp**.

Dapat disimpulkan bahwa citra rekonstruksi **beachside.bmp** lebih baik dibandingkan dua rekonstruksi citra uji lainnya. Hal ini disebabkan detail/tekstur dan persebaran warna dari citra **beachside.bmp** lebih seragam dibanding citra uji lainnya, begitu juga tekstur citra **rose.bmp** lebih seragam dibandingkan citra **empire.bmp**.

4.3 Pengaruh Prosentase Pemampatan dan Resolusi Terhadap Ukuran Berkas Termampat.

TABEL 4.1 HASIL PENGUJIAN CITRA 540 × 486 DAN 600 × 500

Citra 1			Citra 2		
Prosentase	Rasio Pemampatan	Selisih	Prosentase	Rasio Pemampatan	Selisih
3%	106,47%	9,47%	3%	101,36%	4,36%
6%	102,49%	8,49%	6%	98,74%	4,74%
9%	99,50%	8,50%	9%	95,24%	4,24%
12%	96,52%	8,52%	12%	92,62%	4,62%
15%	92,54%	7,54%	15%	89,13%	4,13%
18%	89,56%	7,56%	18%	85,64%	3,64%
21%	86,58%	7,58%	21%	83,02%	4,02%
24%	83,59%	7,59%	24%	79,52%	3,52%
27%	79,62%	6,62%	27%	76,90%	3,90%
30%	76,63%	6,63%	30%	73,41%	3,41%
33%	73,65%	6,65%	33%	69,92%	2,92%
36%	69,67%	5,67%	36%	67,30%	3,30%
39%	66,69%	5,69%	39%	63,81%	2,81%
42%	63,70%	5,70%	42%	61,19%	3,19%
45%	59,73%	4,73%	45%	57,69%	2,69%
48%	56,74%	4,74%	48%	54,20%	2,20%
51%	53,76%	4,76%	51%	51,58%	2,58%
54%	50,78%	4,78%	54%	48,09%	2,09%
57%	46,80%	3,80%	57%	45,47%	2,47%
60%	43,82%	3,82%	60%	41,97%	1,97%
63%	40,83%	3,83%	63%	38,48%	1,48%
66%	36,85%	2,85%	66%	35,86%	1,86%
69%	33,87%	2,87%	69%	32,37%	1,37%
72%	30,89%	2,89%	72%	29,75%	1,75%
75%	26,91%	1,91%	75%	26,25%	1,25%
78%	23,93%	1,93%	78%	22,76%	0,76%
81%	20,94%	1,94%	81%	20,14%	1,14%
84%	17,96%	1,96%	84%	16,65%	0,65%
87%	13,98%	0,98%	87%	14,03%	1,03
90%	11,00%	1,00%	90%	10,54%	0,54

Untuk mengetahui pengaruh prosentase pemampatan citra terhadap ukuran berkas citra termampat akan digunakan parameter rasio pemampatan. Indikatornya adalah jika rasio

pemampatan semakin mendekati nilai yang sebenarnya maka ukuran berkas termampat akan semakin optimal. Untuk pengujian akan digunakan 2 buah citra. Citra 1 memiliki resolusi 540×486 piksel, pada citra 1 ini ditambahkan data *dummy* pada proses pemampatannya dan citra kedua memiliki resolusi 600×500 piksel.

Citra 1 memiliki selisih maksimal 9,87% pada prosentase pemampatan 3% dan citra 2 memiliki selisih maksimal 4,74% pada prosentase pemampatan 6%. Hal ini menunjukkan bahwa semakin besar resolusi citra masukan, mengakibatkan rasio pemampatan kurang optimal pada prosentase pemampatan yang kecil, selain itu untuk citra yang ditambahkan data *dummy*, memiliki rasio pemampatan yang kurang optimal dan berkas citra hasil pemampatan yang disimpan juga lebih besar

V. PENUTUP

5.1 Kesimpulan

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

1. Berdasarkan hasil pengamatan menggunakan sistem penglihatan manusia, rekonstruksi citra hasil pemampatan citra **beachside.bmp** dengan resolusi 800×600 piksel menunjukkan prosentase pemampatan tertinggi dengan citra rekonstruksi masih terlihat seperti citra masukan adalah 60% .
2. Berdasarkan perhitungan, hasil pemampatan citra **beachside.bmp** dengan resolusi 800×600 piksel memiliki prosentase pemampatan tertinggi dengan kualitas hasil rekonstruksi citra hampir sama dengan citra asli adalah 48%.
3. Hasil pemampatan citra **beachside.bmp** dengan resolusi 600×450 piksel, menunjukkan prosentase pemampatan tertinggi dengan citra rekonstruksi masih terlihat seperti citra masukan adalah 50%. Sedangkan berdasarkan perhitungan prosentase pemampatan tertinggi dengan kualitas hasil rekonstruksi citra hampir sama dengan citra asli adalah 42%.
4. Hasil pemampatan citra **beachside.bmp** dengan resolusi 400×300 piksel, menunjukkan prosentase pemampatan tertinggi dengan citra rekonstruksi masih terlihat seperti citra masukan adalah 40%. Sedangkan berdasarkan perhitungan prosentase pemampatan tertinggi dengan kualitas hasil rekonstruksi citra hampir sama dengan citra asli adalah 20,8%.
5. Berdasarkan penglihatan mata, prosentase pemampatan tertinggi dengan kualitas citra rekonstruksi masih terlihat seperti citra masukan untuk 3 citra dengan resolusi yang sama yaitu 500×500 piksel adalah 40% untuk citra **rose.bmp**, 20% untuk citra **empire.bmp**, dan 50% untuk citra **beachside.bmp**

6. Berdasarkan perhitungan, prosentase pemampatan tertinggi dengan kualitas hasil rekonstruksi citra hampir sama dengan citra asli untuk citra **rose.bmp**, **empire.bmp**, dan citra **beachside.bmp** berturut-turut adalah 33%, 1%, dan 41%.
7. Citra yang memiliki tekstur kurang detail jika dimampatkan dengan prosentase pemampatan yang tinggi akan memiliki citra rekonstruksi yang lebih baik dibanding citra yang bertekstur detail pada prosentase pemampatan yang sama.
8. Penambahan data *dummy* akan mengakibatkan ukuran berkas termampat kurang optimal. Hal ini disebabkan oleh adanya penambahan nilai 0 pada kedua sisi citra (sisi vertikal dan sisi horisontal), sehingga ukuran resolusi citra bertambah.
9. *Header* citra berisi ukuran resolusi citra asli, ukuran blok citra, nilai minimum dan maksimum data citra termampat, nilai minimum dan maksimum *footer* citra, ukuran data citra termampat terskala, ukuran *footer* citra terskala. Isi *header* citra ini berguna saat proses rekonstruksi citra.
10. *Footer* citra berisi matriks alihragam dan nilai rerata pada baris terakhir. *Footer* citra berfungsi saat rekonstruksi citra untuk proses alihragam balik.

DAFTAR PUSTAKA

- [1] Anton, H. dan P. Silaban, *Aljabar Linier Elementer*, PT. Gelora Aksara Pratama, Jakarta, 1988.
- [2] Chitwong, S., A. Somboonkaew, F. Cheevasuvit, K. Dejhan, and S. Mithatha, *PCA Colour Image Compression Using Vector Quantization*, Faculty of Engineering King Mongkut's Institute of Technology Ladkraband, Thailand, 1999.
- [3] Godbole, A.S., *Data Communication and Networks*, McGraw-Hill, 2003.
- [4] Gunadi, K. dan S.R. Pongsitanan, *Pembuatan Perangkat Lunak Pengenalan Wajah Menggunakan Principal Components Analysis*, Universitas Kristen Petra, kgunadi@peter.petra.ac.id.
- [5] Jain, A.K., *Fundamentals of Digital Image Processing*, Prentice Hall, 1989.
- [6] Pitas, I., *Digital Image Processing Algorithms*, Prentice Hall, 1993.
- [7] Sid-Ahmed, M.A., *Image Processing Theory, Algorithms, and Architectures*, McGraw-Hill, Inc., 1995.
- [8] Smith, L.I, *A tutorial on Principal Components Analysis*, <http://vision.auc.dk/sig/>, 2002.
- [9] Wu, M., *Review 1 : Image Processing Basic & Compression*, <http://www.ece.umd.edu/class/enee631/>, 2001



Rofi Yuliansyah (L2F 000 633)

Lahir di Tangerang, 18 Juli 1982.
Saat ini sedang menempuh pendidikan strata 1 di Jurusan Teknik Elektro UNDIP dengan konsentrasi Elektronika dan Telekomunikasi.

Menyetujui dan mengesahkan

Pembimbing I

Budi Setiyono, S.T., M.T.
NIP. 132 283 184
Tanggal :

Pembimbing II

R. Rizal Isnanto, S.T., M.M., M.T.
NIP. 132 288 515
Tanggal :