

Analisa Perbandingan Pengenalan Wajah Dengan Principal Component Analysis Dan Bidirectional Associative Memory

Etini Harefa^{1*}, Anda Yanny²

^{1*}Teknik Informatika, Univeristas Budi Darma, Indonesia

²Univeristas Budi Darma, Indonesia

e-mail: ^{1*}etiniharefa424@gmail.com

Keywords:

*Comparison,
Face recognition,
Principal Component Analysis,
Bidirectional Associative
Memory,
Image processing.*

ABSTRACT (10 PT)

This research aims to compare two methods in face recognition systems, namely Principal Component Analysis (PCA) and Bidirectional Associative Memory (BAM), to determine the most accurate method in recognizing face images. In this experiment, both methods are tested using the same face image samples, so that the results can be compared objectively. PCA is used to reduce the data dimension and extract facial features through an eigenface approach, while BAM applies a heteroassociative neural network to recognize input patterns based on bidirectional synaptic weights. The results show that the PCA method produces a higher level of accuracy than BAM. This is evidenced by the smallest Euclidean Distance value of 455, which indicates a match between the test image and the training image. In contrast, the BAM method did not successfully identify the test image because the final result did not match the target, which shows its limitation in recognizing facial image patterns. Thus, PCA is more recommended to be applied in image-based face recognition systems.

Kata Kunci

*Perbandingan,
Pengenalan wajah,
Principal Component Analysis,
Bidirectional Associative
Memory,
Pengolahan citra.*

ABSTRAK

Penelitian ini bertujuan untuk membandingkan dua metode dalam sistem pengenalan wajah, yaitu Principal Component Analysis (PCA) dan Bidirectional Associative Memory (BAM), guna menentukan metode yang paling akurat dalam mengenali citra wajah. Dalam eksperimen ini, kedua metode diuji menggunakan sampel citra wajah yang sama, sehingga hasilnya dapat dibandingkan secara objektif. PCA digunakan untuk mereduksi dimensi data dan mengekstrak fitur wajah melalui pendekatan eigenface, sedangkan BAM menerapkan jaringan saraf heteroasosiatif untuk mengenali pola input berdasarkan bobot sinaptik dua arah. Hasil penelitian menunjukkan bahwa metode PCA menghasilkan tingkat akurasi yang lebih tinggi dibandingkan BAM. Hal ini dibuktikan dengan nilai Euclidean Distance terkecil sebesar 455, yang menunjukkan kesesuaian antara citra uji dan citra latih. Sebaliknya, metode BAM tidak berhasil mengidentifikasi citra uji karena hasil akhir tidak sesuai dengan target, yang menunjukkan keterbatasannya dalam mengenali pola citra wajah. Dengan demikian, PCA lebih direkomendasikan untuk diterapkan dalam sistem pengenalan wajah berbasis citra.

Korespondensi Penulis *):

Etini Harefa
Universitas Budi Darma
Jalan Sisingamangaraja No. 338 Kota Medan

Diajukan: 02-08-2024/ Diterima: 02-08-2024 / Diterbitkan: 30-08-2024

1. PENDAHULUAN

Identifikasi merupakan hal yang sangat penting dalam dunia digital. Salah satu sistem pengenalan yang banyak dikembangkan pada saat ini adalah sistem informasi biologis yaitu retina, sidik jari, wajah, retina dan lain-lain. Salah satu informasi pada biologis yang sekarang banyak dikembangkan dan memiliki tingkat keakuratan tinggi yaitu citra wajah. Citra wajah seseorang memiliki ciri yang unik masing-masing yang dapat diidentifikasi. Identifikasi tersebut dapat digunakan untuk sistem pencarian seseorang di dalam sebuah gambar yang berisi wajah. Eigenface merupakan salah satu metode pengenalan wajah yang berdasarkan pada *Principal Component Analysis (PCA)* dan *Bidirectional Associative Memory (BAM)* yang mudah di implementasi.

Sistem face recognition merupakan sebuah solusi dalam melakukan identifikasi wajah seseorang. Banyak metode yang telah diterapkan untuk pengenalan wajah manusia dengan kelebihan dan kekurangan tertentu. Hal ini disebabkan karena wajah manusia mempresentasikan sesuatu yang kompleks, sehingga untuk mengembangkan model komputasi yang ideal untuk pengenalan wajah manusia adalah suatu hal yang sulit, tidak dibantahkan masih banyak pembuat sistem belum mengetahui metode yang lebih akurat untuk diterapkan pada suatu sistem pengenalan wajah.

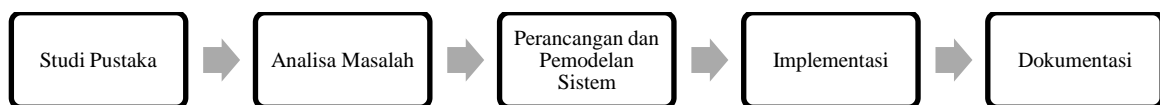
Permasalahan pada penelitian ini adalah mencari pengenalan wajah yang paling akurat untuk diterapkan dengan membandingkan dua hasil dari metode yang berbeda, dimana sampel input di gunakan citra wajah yang sama, sehingga metode Principal Component Analysis Dan Bidirectional Associative Memory akan didapatkan hasil perbandingan pengenalan citra wajah

Perbandingan antara metode *Principal Component Analysis* dengan *Bidirectional Associative Memory* menjadi solusi dalam menggunakan metode yang paling akurat diterapkan pada sistem pengenalan (identifikasi) seseorang, baik dalam pencarian seseorang maupun pada keamanan data.

Penelitian Terdahulu yang dilakukan oleh ‘Rina Firlina dengan judul Implementasi *Principal Component Analysis (PCA)* Untuk Pengenalan Wajah Manusia dengan menyimpulkan Tingkat akurasi dipengaruhi nilai threshold ukuran pixel, jumlah data set dan noise” [1]. Penelitian yang dilakukan oleh Rizqi Elmuna Hidayah, S.Si, M.Kom dengan judul Implementasi Metode Principal Component Analysis Pada Pengenalan Wajah Berbasis Eigenface menyimpulkan dengan menggunakan citra training sebanyak 25 citra yang sama terdiri dari 5 orang yang berbeda, dan tiap orang mewakili 5 citra yang berbeda dengan melakukan pengujian presentasi kesuksesan mencapai 81.82% dengan 1 orang mengalami kegagalan indentifikasi [2]. Penelitian yang dilakukan oleh Zainal Arifin dengan judul Jaringan Syaraf Tiruan Bidirectional Associative Memory (BAM) Sebagai Identifikasi Pola Sidik jari Manusia menyimpulkan Bidirectional Associative Memory (BAM) tidak dapat mengenali secara tepat beberapa pola sidik jari manusia yang dikenai noise [3]. sedangkan penelitian yang dilakukan oleh Aperius Giawa dengan judul Implementasi Metode *bidirectional associative memory* pada absensi berbasis identifikasi wajah menyimpulkan sistem yang dengan metode BAM dapat melakukan pengenalan terhadap citra wajah input [4].

2. METODE PENELITIAN

2.1 Kerangka Kerja Penelitian



Gambar 1. Pelaksanan Penelitian

1. Studi Pustaka, merupakan salah satu metode pengumpulan data dalam kualitatif yang tidak memiliki pengetahuan mendalam akan literatur dari kemampuan yang digunakan tertentu dari pihak peneliti.
2. Analisa Masalah, data yang diperoleh dari hasil studi pustaka dianalisa untuk merencanakan pembuatan sistem yang sesuai dengan kebutuhan.
3. Perancangan dan Pemodelan, setelah analisa masalah, dilakukan pemodelan dan interface, dari pemodelan sistem yang dibuat sesuai dengan tujuan dan batasan masalah yang ada.
4. Implementasi, sistem yang dirancang diimplementasikan dengan bahasa pemrograman visual basic 2008 dengan menerapkan metode yang diteliti.
5. Dokumentasi, dari hasil implementasi dilakukan penyusunan laporan dalam format penulisan penelitian.

2.2 Principal Component Analysis (PCA)

Principal Component Analysis memiliki tahapan yaitu mereduksi atau mengurangi informasi data yang besar dari sebuah citra wajah tanpa menghilangkan informasi yang ada pada sebuah citra wajah. Algoritma PCA melakukan penguraian citra wajah kedalam kumpulan fitur karakteristik yang disebut sebagai “Eigenface”. Hal tersebut yang kemudian dikatakan sebagai Principal Component dalam sebuah data trainingset, fitur utama dari sebuah algoritma PCA adalah merekonstruksi dari beberapa citra asli dari trainingset dengan mengkombinasikan *eigenface*.

Penyusunan algoritma PCA akan dijelaskan sebagai berikut:

1. Pencarian nilai *eigen* terlebih dahulu perlu melakukan perhitungan nilai rata-rata dari sebuah *citra* dengan persamaan sebagai berikut:

$$\text{Mean} = \frac{i=1}{n} \sum^n X_1 \dots \dots \dots (1)$$

Keterangan :

X_i =data ke i dari variabel X

n =jumlah data

Perhitungan nilai rata-rata guna mengurangi dimensi yang akan di hitung pada proses selanjutnya.

2. Menghitung matriks normalisasi dengan persamaan berikut :

$$A = T - \text{Means} \dots \dots \dots (2)$$
3. Proses mencari nilai dari matriks kovarian *citra* dengan persamaan sebagai berikut :

$$L = A \times A \dots \dots \dots (3)$$

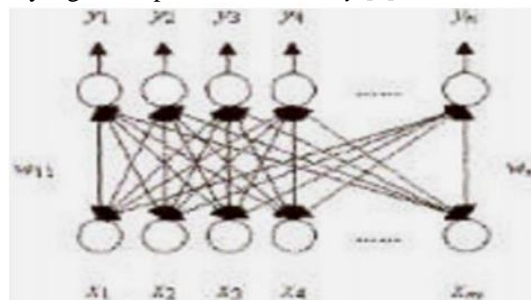
Tujuan mencari matriks kovarian ini adalah untuk mempermudah mencari nilai *eigen* dan *vectoreigen*.
4. Mengurutkan nilai *eigen value* (D) dan *eigen vector* (V) dari besar ke kecil berdasarkan urutan nilai *eigen*.
Kemudian hitung nilai matriks *eigenface* dengan persamaan berikut :

$$\text{Eigenface} = A \times \text{Eigenvector} \dots \dots \dots (4)$$
5. Hitung *project image* dari *citra* dengan persamaan sebagai berikut :

$$\text{Project Image} = \text{eigenface}^t \times A \dots \dots \dots (5)$$

2.3 Bidirectional Associativ Memory

Bidirectional associativ memory merupakan model *heteroassociative* yang terdiri dari dua lapisan memori (A dan B) yang dapat diakses, kedua pola vector dan menggunakan aliran perambatan informasi arah umpan maju dan arah umpan balik untuk menuju pola yang tersimpan dalam *memory* [4]



Gambar 2. Arsitektur *Bidirectional Associative Memory* (BAM)

1. Jika jaringan mengirim sinyal dari *X-layer* ke *Y-layer* maka bobot jaringan dituliskan sebagai vektor berbentuk W.
2. Jika jaringan mengirim sinyal dari *Y-layer* ke *X-layer* maka bobot jaringan dituliskan sebagai vektor bentuk WT.

Algoritma jaringan syaraf tiruan *Bidirectional Associative Memory* (BAM) adalah sebagai berikut:

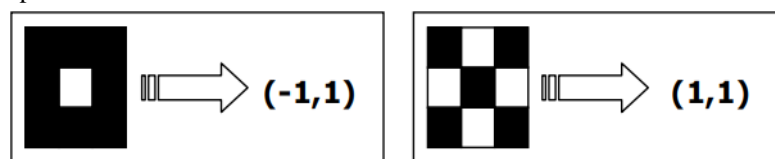
- Langkah 0 : inisialisasi bobot untuk disimpan pada P *vector*
- Langkah 1 : untuk masing-masing masukan, lakukan langkah 2-6.
- Langkah 2a : berikan pola masukan X pada *X-layer* (dengan mengatur aktivitas pada pola masukan X).
- Langkah 2b : berikan pola masukan Y pada *Y-layer* (dengan pemberian nilai dari pola masukan X)
- Langkah 3 : saat nilai aktivitas tidak dapat ditemukan, lakukan langkah 4-6
- Langkah 4 : perbaharui aktivasi pada unit *neuron Y-layer*, dengan cara :
hitung jaringan *input* : y_{inj}
hitung aktivasi $y_i = f(y_{inj})$
kemudian kirim sinyal ke *X-layer*

$$\sum i = w_{ij}, x_i \dots \dots \dots 2.7$$
- Langkah 5 : perbaharui aktivasi pada unit *neuron X-layer*, dengan cara :
hitung jaringan *input* : x_{inj}
hitung aktivasi $x_i = f(y_{inj})$
kemudian kirim sinyal ke *X-layer*

$$\sum i = w_{ij}, y_i \dots \dots \dots 2.8$$

Contoh Kasus :

Misalkan kita memiliki 2 matriks 3x3 yang mewakili bilangan huruf O dan X. Tiap matriks berhubungan dengan kode bipolar seperti



Gambar 3. Matriks 3x3 untuk BAM

- Atur pola *input* dan pola *output*
Kita bisa membawa tiap-tiap matriks menjadi satu bentuk vektor dengan elemen-elemennya berupa bilangan biner -1 atau 1, sebagai berikut:
Pola 1 : Pola O
 $x_1: 1 \ 1 \ 1 \ 1 \ -1 \ 1 \ 1 \ 1 \quad y_1 \rightarrow -1 \ 1$
Pola 2 : Pola X
 $x_2: 1 \ -1 \ 1 \ -1 \ 1 \ -1 \ 1 \ -1 \quad y_2 \rightarrow -1 \ 1$
- Hitung matriks bobot dari pola X ke pola Y
Matriks bobot untuk pola 1 :

$$W_1 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ -1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} * [-1 \quad 1] = \begin{bmatrix} -1 & 1 \\ -1 & 1 \\ -1 & 1 \\ -1 & 1 \\ 1 & -1 \\ 1 & -1 \\ 1 & -1 \\ 1 & -1 \\ 1 & -1 \end{bmatrix}$$

Matriks bobot untuk pola 2 :

$$W_1 = \begin{bmatrix} 1 \\ -1 \\ 1 \\ -1 \\ 1 \\ -1 \\ 1 \\ -1 \\ 1 \end{bmatrix} * [1 \quad 1] = \begin{bmatrix} 1 & 1 \\ -1 & -1 \\ 1 & 1 \\ -1 & -1 \\ 1 & 1 \\ -1 & -1 \\ 1 & 1 \\ -1 & -1 \\ 1 & 1 \end{bmatrix}$$

Matriks bobot untuk W_{Total} :

$$w_{total} = \begin{bmatrix} -1 & 1 \\ -1 & 1 \\ -1 & 1 \\ -1 & 1 \\ 1 & -1 \\ -1 & 1 \\ -1 & 1 \\ -1 & 1 \\ -1 & 1 \end{bmatrix} + \begin{bmatrix} 1 & 1 \\ -1 & -1 \\ 1 & 1 \\ -1 & -1 \\ 1 & 1 \\ -1 & -1 \\ 1 & 1 \\ -1 & -1 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 2 \\ -2 & 0 \\ 0 & 2 \\ -2 & 0 \\ 2 & 0 \\ -2 & 0 \\ 0 & 2 \\ -2 & 0 \\ 0 & 2 \end{bmatrix}$$

Matriks bobot tersebut menghubungkan antara neuron-neuron di lapisan input ke neuron-neuron yang ada di lapisan output. Sedangkan matriks bobot yang menghubungkan antara neuron-neuron di lapisan output ke neuron-neuron yang ada di lapisan input adalah W_t

- Recall pola input
Tentukan nilai y_i dengan membandingkan nilai y_{in_i} dengan *threshold*
Jika $y_{in_i} \geq \theta$ maka $y_i = 1$
Jika $y_{in_i} < \theta$ maka $y_i = -1$

$$(w_{total})^t = \begin{bmatrix} 0 & -2 & 0 & -2 & 2 & -2 & 0 & -2 & 0 \\ 2 & 0 & 2 & 0 & 0 & 0 & 2 & 0 & 2 \end{bmatrix}$$

Kita bisa menguji bobot tersebut. Misalkan kita cobakan vektor input pertama yang mewakili Huruf O, maka output yang diperoleh adalah:

$$(w_{total})^t = [1 \quad 1 \quad 1 \quad 1 \quad -1 \quad 1 \quad 1 \quad 1 \quad 1] * \begin{bmatrix} 0 & 2 \\ -2 & 0 \\ 0 & 2 \\ -2 & 0 \\ 2 & 0 \\ -2 & 0 \\ 0 & 2 \\ -2 & 0 \\ 0 & 2 \end{bmatrix} = [-10 \quad 8]$$

Karena $(y_{in_1}(1) = -10 < 0$, maka $y_1(1) = -1$) dan $(y_{in_1}(2) = 8 > 0$, maka $y_1(2) = 1$), maka nilai $y_1 = [-1 \ 1]$, sama dengan target yang diharapkan

Untuk vektor input kedua yang mewakili Huruf X, maka output yang diperoleh adalah:

$$(w_{total})^t = [1 \quad -1 \quad 1 \quad -1 \quad 1 \quad -1 \quad 1 \quad -1 \quad 1] * \begin{bmatrix} 0 & 2 \\ -2 & 0 \\ 0 & 2 \\ -2 & 0 \\ 2 & 0 \\ -2 & 0 \\ 0 & 2 \\ -2 & 0 \\ 0 & 2 \end{bmatrix} = [-10 \quad 8]$$

Karena $(y_{in2}(1) = 10 > 0$, maka $y_2(1) = 1$) dan $(y_{in2}(2) = 8 > 0$, maka $y_2(2) = 1$), maka nilai $y_2 = [1 \ 1]$, sama dengan target yang diharapkan

4. Balik arah, cari nilai x_{in}

Sekarang apabila dibalik, y digunakan sebagai input untuk mendapatkan x . Misalkan kita cobakan vektor input pertama $y_1 = [-1 \ 1]$, maka output yang diperoleh adalah:

$$(w_{total})^t = [-1 \quad 1] * \begin{bmatrix} 0 & -2 & 0 & -2 & 2 & -2 & 0 & -2 & 0 \\ 2 & 0 & 2 & 0 & 0 & 0 & 2 & 0 & 2 \end{bmatrix} \\ = 2[2 \ 2 \ 2 \ -2 \ 2 \ 2 \ 2] \Rightarrow [1 \ 1 \ 1 \ -1 \ 1 \ 1 \ 1]$$

Outputnya sama dengan target, yaitu Huruf O.

Untuk vektor input kedua yaitu $y_2 = [1 \ 1]$, maka output yang diperoleh adalah:

$$(w_{total})^t = [1 \quad 1] * \begin{bmatrix} 0 & -2 & 0 & -2 & 2 & -2 & 0 & -2 & 0 \\ 2 & 0 & 2 & 0 & 0 & 0 & 2 & 0 & 2 \end{bmatrix} \\ = 2[-2 \ 2 \ -2 \ 2 \ -2 \ 2 \ -2 \ 2] \Rightarrow 1[-1 \ 1 \ -1 \ 1 \ -1 \ 1 \ -1]$$

Outputnya sama dengan target, yaitu Huruf X

3. HASIL DAN ANALISIS

3.1 Penerapan Principal Component Analysis

Langkah 1 :

Menyusun seluruh *training image* menjadi 1 matrik seperti berikut.

$$C_1 = \begin{bmatrix} 295 & 259 & 408 \\ 484 & 454 & 444 \\ 388 & 409 & 248 \end{bmatrix} \\ = 295 \quad 259 \quad 408 \quad 484 \quad 454 \quad 444 \quad 288 \quad 409 \quad 248 \\ C_2 = \begin{bmatrix} 527 & 341 & 350 \\ 461 & 281 & 611 \\ 566 & 350 & 719 \end{bmatrix} \\ = 527 \quad 341 \quad 350 \quad 461 \quad 281 \quad 611 \quad 566 \quad 350 \quad 719$$

Langkah 2 :

Mencari nilai rata-rata (*mean flatvector*) dengan menjumlahkan seluruh baris dengan jumlah *image*, seperti berikut :

$$Mean(f_v) = \begin{matrix} 295 & 259 & 408 & 484 & 454 & 444 & 288 & 409 & 248 \\ 527 & 341 & 350 & 461 & 281 & 611 & 566 & 350 & 719 \end{matrix} \\ = \left[\frac{822 \quad 600 \quad 758 \quad 945 \quad 735 \quad 1055 \quad 854 \quad 759 \quad 967}{2} \right] \\ = 411 \quad 300 \quad 379 \quad 473 \quad 368 \quad 528 \quad 427 \quad 380 \quad 484$$

Dengan hasil hitung nilai pada citra *mean flatvector*, maka nilai *eigenface* untuk matriks *flatvector* yang sudah disusun tersebut dapat dihitung nilai *eigenfacenya*.

Langkah 3 :

Menghitung nilai *eigenface* dengan cara mengurangi baris-baris pada matriks *flatvector* dengan nilai rata-rata *flatvector*. Jika didapatkan nilai dibawah 0 (nilai minus), maka nilai diganti dengan 0 (nol).

$$C_1 = \begin{matrix} 295 & 259 & 408 & 484 & 454 & 444 & 288 & 409 & 248 \\ 411 & 300 & 279 & 473 & 368 & 528 & 427 & 380 & 484 \end{matrix} - \\ \begin{matrix} 0 & 0 & 129 & 211 & 86 & 0 & 0 & 29 & 0 \\ 527 & 341 & 350 & 461 & 281 & 611 & 566 & 350 & 719 \end{matrix} \\ C_2 = \begin{matrix} 411 & 300 & 279 & 473 & 368 & 528 & 427 & 380 & 484 \end{matrix} - \\ \begin{matrix} 116 & 41 & 71 & 0 & 0 & 83 & 139 & 0 & 235 \end{matrix}$$

Untuk mengenali citra tes (*testface*) pada saat uji, langkah identifikasinya adalah menghitung nilai *eigenface* untuk matriks *testface* dengan cara seperti sebelumnya yaitu dimulai dari awal penentuan nilai *flatvector* kemudian dilakukan pengurangan dengan nilai rata-rata *flatvector* (dari citra *training*), dan mendapatkan *eigenface* untuk *testface* seperti berikut ini :

$$C_t = \begin{bmatrix} 315 & 258 & 186 \\ 354 & 420 & 447 \\ 300 & 327 & 354 \end{bmatrix}$$

$$C_T = \begin{array}{cccccccccc} 315 & 258 & 186 & 354 & 420 & 447 & 300 & 327 & 354 & \\ 411 & 300 & 279 & 473 & 368 & 528 & 427 & 380 & 484 & \\ \hline & 0 & 0 & 0 & 0 & 52 & 0 & 0 & 0 & 0 \\ C_t = 0 & 0 & 0 & 0 & 52 & 0 & 0 & 0 & 0 & \end{array}$$

Langkah 4 :

Melakukan identifikasi dengan *euclidean distance* yaitu menentukan jarak terdekat antara nilai *eigenface training* dengan nilai *tesface*.

$$C_1 = \begin{array}{cccccccc} 0 & 0 & 129 & 211 & 86 & 0 & 0 & 29 & 0 \\ 0 & 0 & 0 & 0 & 52 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 129 & 211 & 86 & 0 & 0 & 29 & 0 \end{array}$$

$$C_1 = 0+0+129+211+86+0+0+29+0$$

$$C_1 = 455$$

$$C_2 = \begin{array}{cccccccc} 116 & 41 & 71 & 0 & 0 & 83 & 139 & 0 & 235 \\ 0 & 0 & 0 & 0 & 52 & 0 & 0 & 0 & 0 \\ \hline 116 & 41 & 71 & 0 & 52 & 83 & 139 & 0 & 235 \end{array}$$

$$C_2 = 116+41+71+0+52+83+139+0+235$$

$$C_2 = 737$$

Dari hasil perhitungan yang dilakukan diatas, diperoleh jarak *citra face* satu memiliki nilai terkecil yaitu 455, maka hasil identifikasi menyimpulkan bahwa *tesface* lebih mendekati pada *face* satu dari pada *face* dua.

3.2 Penerapan Principal Component Analysis

Perhitungan menggunakan metode *Principal Component Analysis* menunjukkan *citra 1 tesface* terkecil, maka pada proses metode *bidirectional associative memory* akan digunakan nilai *citra 1*, untuk melihat perbandingan antara dua mete yang berbeda.

Langkah pertama :

1. Menentukan *citra input* untuk diproses

Table 1. Nilai citra RGB 3x3 pixel

R	G	B
295	259	408
484	454	444
388	409	248

Dari nilai *citra* RGB pada tabel 4.3 di atas, maka dilakukan pencarian nilai *citra grayscale* seperti berikut ini :

$$295 = \frac{117 + 83 + 55}{3} = 98$$

Langkah yang sama dilakukan pada nilai RGB, Sehingga didapatkan nilai *citra grayscale* dari *citra* RGB sebagai berikut :

Tabel 2. Nilai *citra grayscale*

R	G	B
98	86	136
161	151	148
129	136	83

2. Dari hasil nilai *citra grayscale* pada *table 4.4* diatas, maka dilakukan proses pengubahan nilai ke nilai *biner* dengan menggunakan *threshold* sebesar 128, dimana nilai piksel yang kurang dari 128 akan di ubah menjadi 0 sedangkan jika nilai pikselnya sama dengan 128 atau lebih akan dirubah menjadi 1, seperti berikut ini.

Table 3. nilai *biner*

0	0	1
1	1	1
1	1	0

3. Dari nilai *citra biner* yang diperoleh dari tabel 4.5 diatas, maka dilakukan proses perubahan ke nilai *bipolar*, dimana proses ini jika *citra* bernilai 0 maka dirubah menjadi -1 sedangkan nilai 1 nilainya tetap, maka diperoleh nilai bipolar seperti pada tabel berikut :

Tabel 4. nilai *bipolar*

-1	-1	1
1	1	1
1	1	-1

4. Proses menentukan pola *input* (x_{11}) dan *output* berdasarkan *Bidirectional Associative memory*.
target $[1,9] = \{-1, -1, -1, -1, 1, 1, -1, -1, -1\}$

- Menghitung matriks bobot (w_{total}) atau $W_{11} = X_{11} * target1$, hasil perkalian diperoleh seperti berikut ini :
 $W_{total}(W_{11}) = \{-1, -1, 1, 1, 1, 1, 1, -1\}$
- Melakukan recal (x_{in11}) pola pada *inputan*, nilai *transpose* dari W_{total} yang dihasilkan pada tahap ke 5 akan dikalikan dengan seluruh target yaitu variabel *target1*. Proses ini dilakukan untuk membuktikan hasil perkalian antara W_{total} dengan *targer1*
 $W_{in11} = \{1, -1, 1, 1, 1, 1, 1, -1\}$

Kesimpulan :

Jika nilai y_{in11} sama dengan nilai target maka identifikasi berhasil dilakukan atau sesuai dengan target, sebaliknya jika nilai y_{in11} berbeda maka identifikasi gagal dilakukan atau tidak mengenali target. Dari hasil perhitungan antara dua metode yang di bandingkan, maka penulis menyimpulkan metode BAM kurang efektif dalam pengenalan wajah seseorang dibuktikan dengan hasil yang didapat pada langkah 6 nilai y_{in11} tidak sama dengan nilai target.

3.3 Hasil Pengujian PCA

Pada pengujian dengan metode *Principal Component Analysis* ini akan diberikan *citra* wajah sebagai *inputan* seperti berikut :

- Citra inputan* wajah *training*



Gambar 4. *citra inputan* sampel metode PCA

Dari *citra input* wajah sampel pada gambar 4.19 diatas, di dapatkan jumlah nilai *Euclidean Distance* = 1,791.647 dapat dilihat pada gambar berikut :

97	76	86	91	96	83	106	125	140	153	155	151	147	147	149	151	152	158	156	149	156	156	142	144
58	76	77	85	100	84	76	73	92	106	126	143	151	152	151	150	153	157	155	149	152	152	142	139
123	131	97	95	116	131	135	115	127	134	145	154	155	149	143	141	140	152	153	152	147	146	145	137
163	148	111	99	92	122	141	150	147	140	152	156	155	150	147	147	147	150	154	154	145	142	150	140
146	141	135	118	87	83	119	138	148	149	150	150	150	150	151	151	148	148	149	147	134	132	135	119
146	149	142	130	130	100	95	122	157	100	150	151	146	147	147	145	146	141	139	134	123	115	101	76
122	120	105	110	149	132	101	103	123	137	147	146	146	152	151	144	144	134	121	100	100	92	70	47
95	95	92	100	148	153	136	119	115	134	147	145	146	153	151	141	139	122	98	72	65	64	49	38
106	102	99	133	108	101	151	150	135	151	155	149	149	152	148	141	132	105	76	56	49	49	40	31
95	101	124	158	159	151	149	140	155	150	151	148	149	148	144	133	119	87	57	42	38	37	31	28
115	124	142	157	150	153	154	142	150	154	150	153	150	142	134	112	101	69	44	36	34	32	29	31
145	156	150	152	145	145	138	123	113	137	153	161	152	139	123	84	71	51	38	33	31	30	29	31
154	156	149	134	120	121	101	81	85	124	149	157	148	140	115	58	44	39	39	34	29	30	30	30
125	106	100	120	123	106	92	91	102	132	140	149	147	147	116	52	32	35	40	35	30	32	33	33
66	77	85	79	95	107	109	111	136	140	151	147	146	145	113	59	28	27	32	32	29	30	32	34
100	95	71	55	85	97	102	130	144	147	146	141	135	126	96	57	28	21	24	30	31	30	31	37
141	148	121	59	59	66	67	115	143	141	138	131	119	103	63	47	28	26	26	29	31	32	33	34
145	151	147	88	67	82	96	150	139	139	136	127	109	84	48	36	29	27	26	29	32	34	34	35
140	137	136	129	128	145	149	148	140	142	136	123	99	68	40	33	30	28	27	30	34	37	38	37
140	143	152	161	149	153	147	127	141	141	126	106	83	57	39	33	29	29	30	32	36	39	41	41
145	135	140	130	124	143	140	140	139	131	103	75	58	44	34	27	26	30	33	34	35	38	41	43
137	144	158	140	141	140	143	152	144	124	87	51	42	38	31	24	39	39	44	44	45	46	45	45
138	144	130	141	149	144	143	136	150	120	88	42	39	40	33	28	28	32	34	34	38	46	51	52
140	146	140	145	153	147	145	145	149	114	75	38	37	40	32	29	31	33	33	35	42	54	60	60
140	145	146	147	151	150	146	146	139	94	62	38	39	27	32	31	33	36	34	40	44	56	75	75
139	143	143	144	147	145	141	141	120	79	50	32	36	27	31	32	32	37	37	44	53	67	87	88
139	142	143	143	146	145	141	140	111	63	41	31	36	29	33	31	35	40	42	52	65	82	100	100
139	142	145	146	148	147	145	143	109	60	42	35	38	32	35	34	41	46	49	61	78	95	100	100
137	140	145	147	148	147	145	142	112	63	45	38	37	33	37	41	47	51	56	71	92	106	111	100
138	139	143	147	147	146	143	136	99	57	44	39	38	34	35	42	50	53	62	81	103	114	114	113
141	139	141	145	146	146	141	132	81	47	41	39	41	38	35	44	52	56	67	88	110	118	116	117
143	138	139	143	145	146	141	129	75	44	38	37	41	43	40	53	55	58	71	92	113	118	115	119
144	143	144	146	149	150	143	132	81	49	42	36	44	47	50	56	58	62	78	101	115	117	116	117
143	143	145	147	151	153	144	131	80	40	44	39	47	50	53	58	60	66	83	105	117	117	116	117
138	139	141	144	149	152	142	126	70	41	41	38	48	52	57	61	62	71	89	109	118	117	116	118
144	144	143	144	148	152	140	122	66	39	43	42	48	52	57	59	68	76	92	108	115	116	116	117
143	142	142	143	148	152	144	128	68	41	47	45	49	54	61	62	68	78	94	108	114	115	116	117
140	139	140	143	147	151	144	132	64	37	44	41	48	55	65	67	72	83	98	110	115	116	116	117
144	142	143	145	148	149	143	132	60	34	41	41	46	55	67	69	75	86	102	113	116	116	116	117
143	143	143	144	149	152	144	133	54	37	42	39	44	59	69	72	80	89	102	111	114	116	117	116
143	143	142	143	148	151	143	132	62	40	44	43	49	62	70	74	82	93	104	112	113	115	117	117
142	141	140	142	147	150	142	131	67	42	44	46	53	65	72	77	86	95	106	112	113	114	116	118
140	140	139	141	146	149	140	129	69	40	43	46	56	65	73	82	91	99	108	112	114	114	116	118

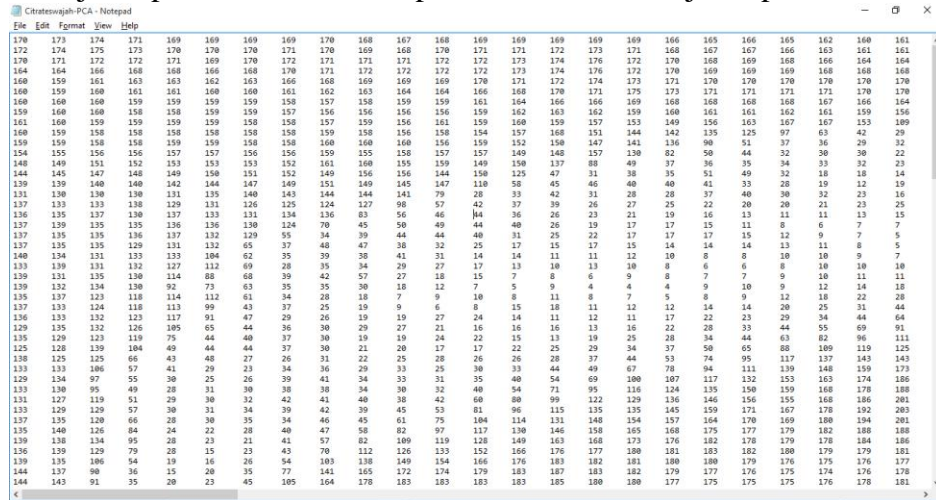
Gambar 5. Nilai *citra inputan* sampel metode PCA

Nilai *Euclidean Distance* 1,791.647 dari *citra* sampel digunakan sebagai target pada *citra* wajah yang akan dikenali nantinya.

- Melakukan identifikasi menggunakan metode PCA, dengan memberi *input citra* wajah sebagai berikut :



Gambar 6. *Citra* wajah *input* identifikasi





3. Dari *citra* wajah input identifikasi di dapatkan nilai *citra* wajah seperti berikut


The image shows a Notepad window titled 'Citrastewajah-PCA - Notepad'. It contains a large grid of numerical data, likely representing the output of a PCA analysis on face images. The data is organized in rows and columns, with values ranging from approximately 125 to 188. The grid is dense and covers most of the window area.

Gambar 7. Nilai *citra* wajah input identifikasi

Dari hasil proses pada citra uji didapatkan *Euclidean Distance*=1,584.278, karena jarak nilai *Euclidean Distance* medekati nilai target maka wajah dikenali, hasil pengujian yang telah dilakukan dapat dilihat pada tabel berikut :

Tabel 5. Hasil Pengenalan Dengan Metode PCA

No	Citra Sampel	Citra Input Target	Nilai Euclidean Distance	Keterangan
1			1,584.278	Etini Harefa
2			1,684.278	Etini Harefa

3.4 Hasil pengujian Metode BAM

Dalam pengujian ini, sistem diberikan 2 (dua) *inputan* dengan gambar wajah yang sebelumnya digunakan pada metode PCA sehingga perbandingan pengenalan wajah dengan menggunakan metode berbeda dapat ditentukan, berikut ini adalah hasil pengujian dengan metode BAB.

1. *Input citra* latihGambar 8. *Citra input* latih rgb

2. Dari gambar 4.25 di atas mengalami proses digitalisasi *citra*, sehingga menghasilkan *citra grayscale* seperti berikut ini :



Gambar 9. *citra grayscale* latih

Dari gambar 4.26 diatas *citra grayscale* memiliki nilai yang tersimpan pada database *notepad* sebagai berikut :

[illegible]

Gambar 9dNilai *citra grayscale*latih

3. Dari *citra grayscale* pada gambar 10 diatas mengalami prose konversi ke *bipolar*, sehingga mendapatkan *citra bipolar* sebagai berikut :



Gambar 10. *citra bipolar* latih

Citra bipolar pada gambar 4.28 diatas, memiliki nilai yang tersimpan pada database berbentuk *notepad* sebagai berikut :

[illegible]

Gambar 11. nilai *citra bipolar* latih

Dari nilai hasil *citra* wajah yang di dapatkan pada gambar 4.29 di atas, di jadikan sebagai target pada wajah uji, sebagai berikut :

4. *Input citra wajah identifikasi*



Gambar 11. *Citra input rgb identifikasi*

Dari gambar 4.30 dilakukan proses digitalisasi *citra* RGB-Grayscale sehingga mendapatkan *citra grayscale* dan dengan nilai yang tersimpan pada database *notepad* sebagai berikut :



Gambar 13. Citra *input grayscale* identifikasi
Citra *grayscale* pada gambar 4.31 di atas, memiliki nilai piksel sebagai berikut :

Gambar 12. Nilai citra *Grayscale*

5. Dari citra *grayscale* pada 5 di atas, dengan nilai pada gambar 4.32 maka, dilakukan proses konversi citra *grayscale* ke hitam putih (*bipolar*) seperti terlihat berikut ini :



Gambar 13. Citra *bipolar* Uji
Citra *bipolar* di atas memiliki nilai sebagai berikut :





Gambar 14. Nilai Citra *Bipolar*

6. Dari hasil nilai citra *bipolar* pada gambar 8 di atas, dilakukan perkalian pada nilai citra bipolar target sehingga mendapatkan hasil sebagai berikut :

Gambar 15. Nilai hasil perkalian

Dari hasil yang yang di dapatkan maka pada pengujian metode BAM dapat dilihat pada tabel berikut ini :

Tabel 6. hasil uji metode BAM

No	Citra Input (Training)	Citra input (Target)	Output
1			Tidak Dikenal
2			Tidak dikenal

4. KESIMPULAN

Setelah melakukan implementasi dari Analisa Perbandingan Pengenalan Wajah Dengan Principal Component Analysis Dan Bidirectional Associative Memory, maka diperoleh kesimpulan sebagai berikut: 1. Pada hasil implementasi pada penelitian dengan wajah berhasil dilakukan pengenalan untuk menentukan perbandingan pengenalan wajah. 2. Hasil pencarian antara wajah sampel dengan wajah uji, di dapatkan hasil perbandingan pengenalan dimana metode PCA lebih di efektif di gunakan di bandingkan dengan metode BAM. 3. Hasil perbandingan pengenalan wajah dengan metode Principal Component Analysis dengan Bidirectional Associative Memory pada pengenalan wajah dapat ditentukan

REFERENSI

- [1] R. F. e. all, "Implementasi Principal Component Analysis (PCA) Untuk Pengenalan Wajah Manusia," Nusantara of Engineering, vol. 2, no. 2355-6684, p. 69, 215.
- [2] S. M. Rizqi Elmuna Hidayah, "Implementasi Metode Principal Component Analysis pada pengenalan wajah berbasis Eigenface," Technologia, vol. 7, no. 2656-8047, pp. -, 2016.
- [3] Z. Arufun, "Jaringan Syaraf Tiruan Bidirectional Associative Memory (BAM) Sebagai Identifikasi Pola Sidik jari Manusia," Jurnal Informatika Mulawarman, vol. 4, p. 26, 2009.
- [4] A. Giawa, "Implementasi Metode Bidirectional associative memory pada absensi berbasis identifikasi wajah," Jurnal Pelita Informatika, vol. 8, no. 2301-9425, p. 111, 2019.
- [5] A. Sudarsono, "Jaringan Syaraf Tiruan Untuk Memprediksi Laju Pertumbuhan Penduduk Menggunakan Metode Bacpropagation (Studi Kasus Di Kota Bengkulu)," Jurnal Media Infotama, vol. 12, no. 1858-2680, pp. 62-63, 2016.
- [6] A. B. S. Suma'inna and H. Maulana, "Pengenalan Citra Wajah Sebagai Identifier Menggunakan Metode Principal Component Analysis (PCA)," JURNAL TEKNIK INFORMATIKA, vol. 9, no. 1979-9160, pp. 166-175, 2016.
- [7] Salamun and F. Wazir, "Rancangan Bangun Sistem Pengenalan Wajah Dengan Metode Principal Component Analysis," RABIT: Jurnal Teknologi dan Sistem Informasi Univrab, vol. 1, no. 2502-891X, pp. 62-65, 2016.
- [8] A. R. Syakhala, D. Puspitaningrum and E. P. P. , "Perbandingan Metode Principal Component Analysis (Pca) Dengan Metode Hidden Markov Model (Hmm) Dalam Pengenalan Identitas Seseorang Melalui Wajah," Jurnal Rekursif, vol. 3, no. 2303-0755, pp. 70-71, 2015.
- [9] M. Fikry, F. Fadlisyah and D. Putri, "Sistem Pendeteksi Pola Citra Tajwid Alquran Mad Lazim Mutsaqal Kilmi Menggunakan Metode Algoritma BAM & FAM," TECHSI-Jurnal Teknik Informatika, vol. 10, no. DOI: <https://doi.org/10.29103/techsi.v10i2.885>, pp. 101-102, 2018.
- [10] Santoso and R. Nurmalina, "Perencanaan dan Pengembangan Aplikasi Absensi Mahasiswa Menggunakan Smart Card Guna Pengembangan Kampus Cerdas (Studi Kasus Politeknik Negeri Tanah Laut)," Jurnal Integrasi, vol. 9, no. 2548-9828, pp. 86-87, 2017.
- [11] F. Ayu and N. Permatasari, "Perancangan Sistem Informasi Perancangan Data Praktek Kerja Lapangan (PKL) Pada Devisi Humas PT Pegadaian," Jurnal Intra-Tech, vol. 2, no. 2549-0222, p. 13, 2018.
- [12] R. Irviani and R. Oktaviana, "Aplikasi Perpustakaan Pada SMA N1 Kelumbayan Barat Menggunakan Visual Basic," Jurnal TAM (Technology Acceptance Model), vol. 8, no. 2579-4221, pp. 63-69, 2017.
- [13] Muhammad Saed Novendri, Ade Saputra and Chandra Eri Firman, "Aplikasi Inventaris Barang Pada MTS Nurul Islam Dumai Menggunakan PHP dan MYSQL," LENTERA DUMAI, vol. 10, no. 2528-1062, pp. 48-49, 2019.
- [14] Y. Heriyanto, "Perancangan Sistem Informasi Rental Mobil Berbasis Web Pada PT.APM Rent Car," Jurnal Intra-Tech, vol. 2, no. 2549-0222, pp. 66-69, 2018.