PENERAPAN ALGORITMA *K-NEAREST NEIGHBOR* DAN *PRINCIPAL COMPONENT ANALYSIS* PADA PENGENALAN TULISAN TANGAN DALAM KASUS PENILAIAN JAWABAN ESAI

Yuyus Bahtiar Gustani[1], Galih Hermawan[2]

[1,2] Teknik Informatika – Universitas Komputer Indonesia
Jl. Dipatiukur 112-114 Bandung
Email : yuyus_rsbi2@yahoo.co.id[1], galih.hermawan@email.unikom.ac.id[2]

ABSTRAK

Optical Character Recognition (OCR) adalah suatu alat yang digunakan untuk menterjemahkan karakter pada citra digital menjadi format teks. Penelitian ini menggunakan metode *Principal Component Analysis* (PCA) dan *K-Nearest Neighbor* (K-NN). Tujuan dari PCA adalah untuk mereduksi dimensi yang besar dari ruang data (*observed variables*) menjadi dimensi intrinsik yang lebih kecil dari ruang fitur (*independent variables*), yang dibutuhkan untuk mendeskripsikan data lebih sederhana. K-NN adalah metode klasifikasi yang didasarkan pada kelas dari tetangga terdekat. Citra yang digunakan pada penelitian ini citra pelatihan dan citra pengujian. Citra pelatihan adalah citra tulisan tangan karakter terurut dari A-Z, a-z, dan 0-9. Citra pengujian adalah citra tulisan tangan jawaban esai. Tahap yang dilakukan pada penelitian adalah *preprocessing* yang meliputi proses *grayscal*e, *thresholding, dilasi, erosi*, segementasi dan *resize* yang selanjutnya akan dilakukan proses ekstraksi ciri menggunakan metode PCA. Tahapan pengenalan dilakukan dengan cara mengklasifikasi hasil ciri yang didapat menggunakan metode K-NN. Setelah didapatkan hasil pengenalan, selanjutnya akan di lakukan proses penilaian esai dengan mengukur tingkat kesamaan dari hasil pengenalan dengan kunci jawaban menggunakan metode *Rabin Karp*. Berdasarkan hasil pengujian yang dilakukan menggunakan 5 kombinasi parameter nilai K pada pengujian didapatkan akurasi pengujian terbaik sebesar 77,79% dengan nilai K=5.

Kata kunci : Pengenalan Tulisan Tangan, *Principal Component Analysis*, Ekstraksi Ciri, *K-Nearest Neighbour*, Pengolahan Citra Digital.

1. PENDAHULUAN

Optical Character Recognition (OCR) adalah suatu alat yang digunakan untuk menterjemahkan karakter pada citra digital menjadi format teks [1]. OCR sudah banyak digunakan dalam penerapannya diantaranya pengenalan plat nomor, tulisan tangan huruf Jawa, huruf Jepang, dan aksara Bali, bahkan masih banyak lagi penerapan lainnya dari penggunaan OCR. Pada pengolahan citra, untuk mendapatkan hasil yang baik dalam pengenalan, maka digunakan dua metode yakni metode untuk ekstraksi ciri dan metode untuk pengenalan.

Metode ekstraksi ciri pada penelitian ini akan menggunakan *Principal Components Analysis* (PCA). Tujuan dari PCA adalah untuk mereduksi dimensi yang besar dari ruang data (*observed variables*) menjadi dimensi intrinsik yang lebih kecil dari ruang fitur (*independent variables*), yang dibutuhkan untuk mendeskripsikan data lebih sederhana [2]. PCA dapat digunakan sebagai ekstraksi ciri pada pengenalan angka tulisan tangan [3]. Penggunaan PCA dalam mengenali pola bergantung pada dimensi dan jumlah zona [3]. Sedangkan metode pembelajaran yang digunakan dalam penelitian ini adalah metode *K-Nearest Neighbor* (K-NN). K-NN adalah metode klasifikasi yang didasarkan pada kelas dari tetangga terdekat. K-NN dapat digunakan untuk tulisan tangan angka arabic (indian) [4]. Nilai akurasi terbaik dari penggunaan metode K-NN adalah 86% dengan k=1 [4].

Pengenalan tulisan tangan merupakan salah satu bentuk kemampuan komputer dalam mengenali pola. Penelitian dalam bidang pengenalan tulisan tangan telah berkembang dalam kurun waktu yang cukup lama. Penelitian tersebut dilakukan karena semakin banyak dipergunakan model tulisan tangan dalam kehidupan sehari-hari, seperti identifikasi dokumen-dokumen penting, bukti pengesahan dalam dunia perbankan, dan hal lainnya yang melibatkan tulisan tangan di dalamnya.

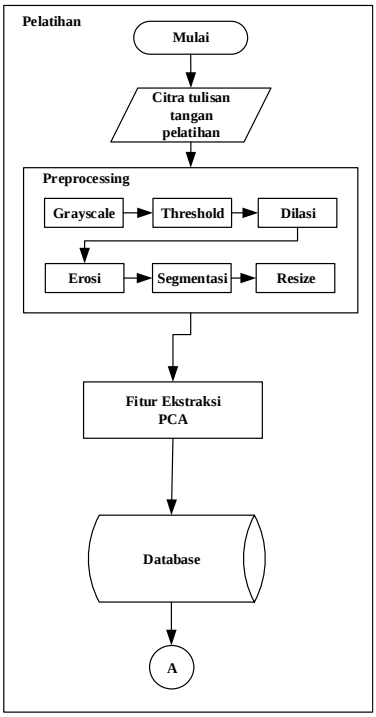
Berdasarkan penelitian mengenai pengenalan tulisan tangan yang telah dilakukan ditemukan kendala pada proses pengenalannya, hal ini dikarenakan beberapa huruf alfabet memiliki kemiripan bentuk dengan huruf lainnya [5], serta pada penerapannya pengenalan tulisan tangan baru diimplementasikan pada kasus yang secara umum. Pada penelitian ini pengenalan tulisan tangan akan diimplementasikan pada kasus penilaian esai. Penelitian mengenai penilaian esai pun telah banyak dilakukan akan tetapi terbatas hanya pada teks digital saja [6]. Sehingga dalam penelitian ini peneliti akan membuat prototipe penerapan algoritma *K-Nearest Neighbor* dan *Principal Components Analysis* pada pengenalan tulisan tangan dalam penilaian jawaban

esai. Kemudian untuk akurasi, akan dilakukan perhitungan dengan menghitung banyaknya karakter tulisan tangan yang dapat dikenali.

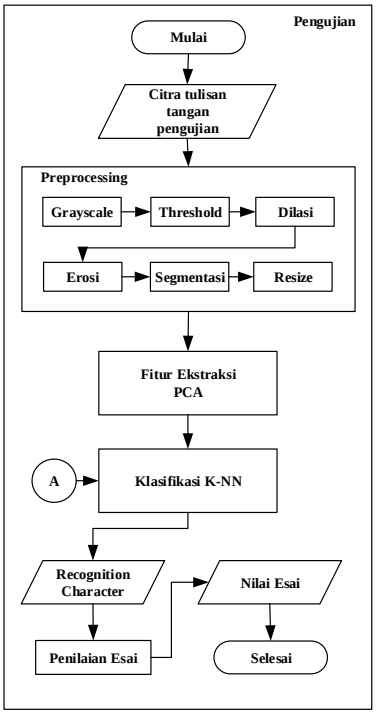
2. ISI PENELITIAN

Masalah pada penelitian ini adalah bagaimana cara mengenali tulisan tangan menggunakan metode K-NN dan PCA. Pada penelitian sebelumnya pengenalan tulisan tangan telah digunakan untuk mengenali huruf alfabet, namun terdapat kendala dalam proses pengenalannya, hal ini disebabkan karena beberapa huruf alfabet memiliki kemiripan bentuk dengan huruf lainnya serta pada penerapannya penganalan tulisan tangan baru diimplementasikan pada kasus secara umum. Pada penelitian ini metode PCA akan digunakan untuk menghasilkan ciri dari citra tulisan tangan yang selanjutnya akan diklasifikasikan menggunakan K-NN. Hasil dari pengenalan tulisan akan digunakan untuk proses penilaian esai. Akurasi pengenalan tulisan tangan yang didapatkan akan mempengaruhi hasil penilaian esai, sehingga perlu diketahui akurasi pengenalannya.

Sistem yang akan dibangun meliputi proses pengenalan tulisan tangan terbagi menjadi dua tahap, dimana tahap pelatihan dan pengujian. Pada tahap pelatihan terdapat proses *preprocessing*, dan fitur ekstraksi dengan PCA. Selanjutnya pada tahap pengujian terdapat *preprocessing*, fitur ekstraksi dengan PCA, klasifikasi K-NN, dan Penilaian Esai. Sedangkan data masukan yang digunakan dibagi menjadi dua data yakni data pelatihan yang berupa citra dengan ekstensi *.jpg* yang terdapat tulisan tangan kumpulan karakter huruf A-Z, a-z, dan 0-9. Data masukan pengujian adalah citra tulisan tangan dengan ekstensi *.jpg* yang berisikan jawaban esai. Berikut adalah arsitektur umum sistem yang akan dibangun.



Gambar 1 Tahap Pelatihan

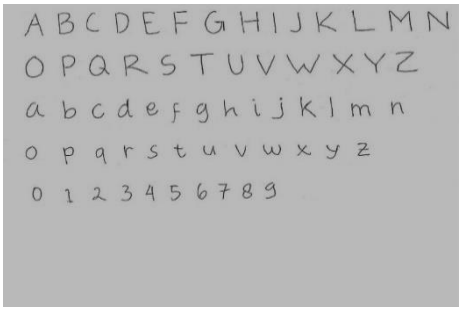


Gambar 2 Tahap Pengujian

2.1. Data Masukan

Data masukan yang digunakan pada penelitian ini, yaitu citra tulisan tangan dengan format gambar *.jpg*. Citra dapat diperoleh melalui *scan*. Citra tulisan tangan yang digunakan ditulis menggunakan pulpen dalam kertas HVS dengan *background* putih. Selain itu warna citra juga tidak terlalu gelap agar aplikasi

dapat lebih maksimal melakukan pengenalan karakter dari setiap tulisan tangan.



Gambar 3 Data Masukan

2.2. *Prepocessing*

Pengolahan citra digital artinya melakukan pemrosesan gambar berdimensi dua melalui komputer digital. Menurut Efford (2000), pengolahan citra adalah istilah umum untuk berbagai teknik yang keberadaanya untuk memanipulasi dan memodifikasi citra dengan berbagai cara [7]. Pada proses *preprocessing* data masukan citra akan diolah dalam beberapa tahapan yakni *grayscale*, *thresholding*, *dilasi*, segmentasi dan *resize*. Proses ini dilakukan agar data masukan citra dapat memenuhi kriteria tertentu sebelum dilakukan proses ekstraksi ciri dan klasifikasi.

A. *Grayscale*

Citra *grayscale* menangani gradasi warna hitam dan putih, yang tentu saja menghasilkan efek warna abu-abu. Warna gambar dinyatakan dengan intensitas. Dalam hal ini, intensitas berkisar antara 0 sampai dengan 255. Nilai 0 menyatakan hitam dan nilai 255 menyatakan putih [8]. Mengubah format warna menjadi *grayscale* berfungsi untuk mengecilkan range warna dari 0 sampai dengan 255. Proses ini akan memudahkan ketika ingin melakukan *thresholding* citra menjadi citra hitam putih. Berikut adalah rumus yang digunakan pada proses *grayscale*.

$$I = (0.2989*R)+(0.587*G)+(0.1141*B) \quad (1)$$

Keterangan:
- R = Komponen nilai merah (*Red*) dari suatu titik *pixel*
- G = Komponen nilai hijau (*Green*) dari suatu titik *pixel*
- B = Komponen nilai biru (*Blue*) dari suatu titik *pixel*

Maka dari itu, warna-warna ini diberi bobot yang lebih tinggi untuk memastikan bahwa keseimbangan intensitas *relatif* dalam citra *grayscale* yang dihasilkan mirip dengan citra warna RGB [9].

B. *Thresholding*

Metode *thresholding* yang digunakan dalam penelitian ini yaitu menggunakan metode *Sauvola Threshold*. *Sauvola threshold* merupakan metode *threshold* yang mampu memberikan peningkatan dengan cara menghitung ambang menggunakan rentang nilai dinamis dari nilai standar deviasi citra *grayscale* [10]. Sauvola merupakan kategori dari metode *local threshold*, di mana nilai ambang pada setiap *pixel* tergantung dari jumlah tetangga yang digunakan. Berikut adalah rumus yang digunakan pada proses *thresholding*.

$$T(x,y) = m(x,y) * (1 + k * (\frac{s(x,y)}{R} - 1)) \quad (2)$$

$$m(x,y) = \frac{\sum_{i=min}^{i=max}\sum_{j=min}^{j=max} img(i,j)}{i*j} \quad (3)$$

$$s(x,y) = \sqrt{\frac{\sum_{i=min}^{i=max}\sum_{j=min}^{j=max} (img(i,j)-m(x,y))^2}{(i*j)-1}} \quad (4)$$

Keterangan:
- R = Nilai maksimum dari standar deviasi (128 untuk citra *grayscale*)
- k = Kernel dengan nilai antara 0.2 – 0.5
- m = Fungsi yang menghasilkan nilai rata-rata dari sejumlah *pixel* citra
- s = Fungsi yang menghasilkan nilai standar deviasi dari sejumlah *pixel* citra
- T = Fungsi yang menghasilkan nilai *threshold* (ambang)
- x = Nilai Koordinat lebar citra
- y = Nilai koordinat tinggi citra
- i = Nilai ukuran *pixel* tetangga
- j = Nilai ukuran *pixel* tetangga

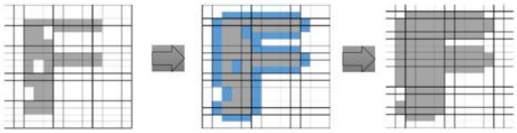
Setelah nilai ambang T(x,y) sudah didapatkan, selanjutnya masukkan ke persamaan di bawah ini.

$$f(x,y) = \begin{cases} 0, & \text{img}(x,y) < T(x,y) \\ 255, & \text{img}(x,y) \geq T(x,y) \end{cases} \quad (5)$$

Keterangan:
- f = Fungsi yang menghasilkan nilai 0 atau 255
- img = Nilai citra *grayscale*

C. *Dilasi*

Proses *dilasi* berfungsi untuk memperbaiki pixel pada citra yang pecah dengan cara membandingkan setiap piksel citra input dengan nilai pusat SE.



Gambar 4 Contoh Dilasi

D. *Erosi*

Proses *erosi* merupakan kebalikan dari proses dilasi. Jika dalam proses dilasi menghasilkan objek yang lebih luas maka dalam proses erosi akan menghasilkan objek yang menyempit (mengecil). Operasi *erosi* dapat dinyatakan sebagai berikut [8].

$$D(A,B)=A\ominus B=\{x:B_x\cap A\neq\emptyset\} \quad (6)$$

Keterangan:
A = Citra input
B = *Structuring Elements (SE)*
E = Fungsi yang menghasilkan citra *erosi*

E. Segmentasi

Segmentasi merupakan teknik untuk membagi suatu citra menjadi beberapa daerah (*region*) dimana setiap daerah memiliki kemiripan atribut [7].



Gambar 5 Contoh Segmentasi

F. *Resize*

Resize citra artinya mengubah besarnya ukuran citra digital dalam *pixel*. Adakala ukurannya berubah menjadi lebih kecil dari ukuran aslinya dan ada kalanya sebaliknya. Proses *Resize* citra dapat mempengaruhi *size* citra lebih besar atau lebih kecil dan juga kualitas citra menjadi lebih baik atau lebih buruk. *Resize/*Normalisasi dengan ukuran 16x16 [11].



Gambar 6 Contoh Resize

G. Algoritma *Principal Component Analysis* (PCA)

PCA adalah suatu teknik handal untuk mengekstraksi struktur dari suatu set data dengan dimensi yang cukup banyak. Masalah dalam PCA adalah menemukan *eigenvalue* dan *eigenvector*. PCA adalah transformasi *orthogonal* (tegak lurus) dari sistem koordinat dimana kita mendeskripsikan data. Koordinat baru kita merepresentasikan data dinamakan *principal component* atau PC. Koordinat ini dipilih dimana variansi dari data mencapai maksimum. Seringkali hanya dengan beberapa *principal component* sudah cukup untuk menjelaskan struktur dari data aslinya. Jika data dalam dimensi asli sulit untuk direpresentasikan melalui grafik, maka dengan dua PC atau satu PC bisa digambarkan melalui grafik [12]. Berikut adalah metodologi dari PCA [13]:

Setiap gambar karakter 2-D digambarkan sebagai vektor 1-D dengan menggabungkan setiap kolom (atau baris) ke dalam vektor tipis yang panjang. Jadi, vektor yang dihasilkan harus hadir sebagai:

$$\bar{x}^i = [x_1^i \dots x_N^i]^T \quad (7)$$

Keterangan:
$\bar{x}$ = Kumpulan citra karakter 1D
T = *Transpose*
X = Citra karakter 1D
N = Banyak karakter

Gambar rata-rata adalah vektor kolom dimana setiap entri adalah rata-rata dari semua piksel gambar pelatihan yang sesuai. Teori sebelumnya dapat dinyatakan dalam ungkapan berikut:

$$\bar{x}^i = x^i - m \quad (8)$$

Keterangan:
$\bar{\bar{x}}^i$ = Citra karakter 1D
m = Matriks kolom rata-rata karakter

dimana

$$m = \frac{1}{p}\sum_{i=1}^{p} x^i \quad (9)$$

Keterangan:
m = Matriks kolom rata-rata citra karakter
P = Banyak data
X = Citra karakter 1D ke-i
I = Banyak karakter

Data pusat:

$$\bar{X} = [\bar{x}^1|\bar{x}^2| \dots |\bar{x}^p|] \quad (10)$$

Keterangan:
$\bar{x}$ = Kumpulan citra karakter 1D
p = Banyak data
x = Citra karakter 1D

Matriks kovarian

$$\Omega = \bar{X}\bar{X}^T \quad (11)$$

Keterangan:
Ω = Matriks kovarian
$\bar{X}$ = Kumpulan citra karakter 1D
T = *Transpose*

$$det(C - \lambda) = 0 \quad (12)$$

Keterangan:
C = Matriks kovarian
λ = Kumpulan citra karakter 1D

Mengurutkan urutan *vector eigen*:

$$V = [V_1|V_2| \ldots |V_p|] \quad (13)$$

Keterangan:
V = Matriks *Eigenvector*
p = Banyak data

Proyeksi gambar :

$$\tilde{x}^i = V^T \bar{x}^i \quad (14)$$

Keterangan:
$\tilde{x}$ = Matriks proyeksi latih
V = Matriks *eigenvector*
T = *Transpose*
$\bar{x}$ = Gambar citra 1D

Mengidentifikasi gambar baru:

$$\bar{y}^i = y^i - m \quad (15)$$

Keterangan:
$\tilde{y}$ = Matriks proyeksi baru
$\bar{y}$ = Pusat data citra karakter 1D baru
V = Matriks *eigenvector*
T = *Transpose*

H. Algoritma *K-Nearest Neighbor* (K-NN)

Teknik ini termasuk dalam kelompok klasifikasi *nonparametric*. Teknik ini sangat sederhana dan mudah diimplementasikan. Mirip dengan teknik *clustering*, pengelompokan suatu data baru berdasarkan jarak data baru itu ke beberapa data/tetangga (*neighbor*) terdekat. Dalam hal ini jumlah data/tetangga terdekat ditentukan oleh user yang dinyatakan dengan k. Misalkan tentukan nilai k = 5, maka setiap data *testing* dihitung jaraknya terhadap data *training* dan pilih 5 data training yang jaraknya paling dekat ke data *testing*. Setelah diketahui 5 data ini, periksa *output* atau labelnya masing-masing. Lalu tentukan *output* mana yang frekuensinya paling banyak [12].

Untuk menghitung jarak antara dua titik x dan y bisa digunakan jarak Euclidean sebagai berikut:

$$d(x,y) = \| x - y \|^2 = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \quad (16)$$

Keterangan:
$d(x,y)$ = Jarak antara dua titik
x_i = Data latih ke-i
y_i = Data uji ke-i
i = Bilangan cacah i = (1,2,3, … , n)

I. Algoritma Rabin Karp

Algoritma *Rabin-karp* adalah algoritma pencocokan string yang menggunakan fungsi *hash* sebagai pembanding antara string yang dicari (m) dengan substring pada teks (n). Algoritma *Rabin-karp* didasarkan pada fakta jika dua buah *string* sama maka harga *hash value*-nya pasti sama. Akan tetapi ada dua masalah yang timbul dari hal ini, masalah pertama yaitu ada begitu banyak *string* yang berbeda, permasalahan ini dapat dipecahkan dengan meng-*assign* beberapa *string* dengan *hash value* yang sama. Masalah yang kedua belum tentu *string* yang mempunyai *hash value* yang sama cocok untuk mengatasinya maka untuk setiap *string* yang di-*assign* dilakukan pencocokan *string* secara *Brute-Force*. Kunci agar algoritma *Rabin-Karp* efisien, terdapat pada pemilihan *hash value*-nya. Salah satu cara yang efektif adalah memperlakukan setiap *substring* sebagai suatu bilangan dengan basis tertentu [14]. Tahap-tahap yang dilakukan dalam evaluasi jawaban esai yaitu proses *tokenizing* dan *case folding*, *filtering*, algoritma *Rabin-Karp* (K-gram, *hashing*, *similarity*).

- ***Tokenizing* dan *Case Folding***

Pada tahap *tokenizing* dan *case folding* dilakukan proses pemisahan teks menjadi per kata, kemudian mengubah semua kata yang sudah ditoken menjadi huruf kecil (*lowercase*). Berikut adalah contoh *Tokenizing* dan *Case Folding*.

Tabel 1 *Tokenizing* dan *Case Folding*

Teks	Memori adalah sistem yang sangat berstruktur, yang menyebabkan organisme sanggup merekam fakta tentang dunia dan menggunakan pengetahuannya untuk membimbing perilakunya
Hasil *tokenizing* dan *case folding*	memori | adalah | sistem | yang | sangat

	| berstruktur | yang | menyebabkan | organisme | sanggup | merekam | fakta | tentang | dunia | dan | menggunakan | pengetahuannya | untuk | membimbing | perilakunya |

- ***Filtering***

Setelah melalui proses *tokenizing* dan *case folding*, selanjutnya dilakukan proses *filtering* untuk menghilangkan kata yang termasuk ke dalam *stoplist* / *stopword*. Kata 'adalah', 'yang', 'sangat', 'tentang', 'dan', 'menggunakan', dan 'untuk', dihilangkan sehingga hasilnya sebagai berikut.

Tabel 2 *Filtering*

Teks	memori | adalah | sistem | yang | sangat | berstruktur | yang | menyebabkan | organisme | sanggup | merekam | fakta | tentang | dunia | dan | menggunakan | pengetahuannya | untuk | membimbing | perilakunya |
Hasil *filtering*	memori | sistem | berstruktur | menyebabkan organisme | sanggup | merekam | fakta | dunia | pengetahuannya | membimbing | perilakunya

- ***K-gram***

Pada proses ini, teks yang sudah melalui proses *tokenizing* dan *case folding* dan *filtering* dilakukan pemecahan sebanyak K huruf secara kontinyu. Dalam penelitian ini menggunakan K=4 maka hasil K-gramnya sebagai berikut:

Tabel 3 K-gram

Teks	memori | sistem | berstruktur | menyebabkan organisme | sanggup | merekam | fakta | dunia | pengetahuannya | membimbing | perilakunya
Hasil K-gram	memo | emor | mori | oris | risi | isis | sist | iste | stem | temb | embe | mber | bers | erst | rstu | stuk | tukt | uktu | ktur | turm | urme | rmen | meny | enye | nyeb | yeba | ebab | babk | abka | bkan | kano | anor | norg | orga | rgan | gani | anis | nism | isme | smes | mesa | esan | sang | angg | nggu | ggup | gupm | upme | pmer | mere | erek | reka | ekam | kamf | amfa | mfak | fakt | akta | ktad | tadu | adun | duni | unia | niap | iape | apen | peng | engt | ngta | gtah | tahu | ahua | huan | uann | anny | nnya | nyam | yame | amem | memb | embi | mbim | bimb | imbi | mbin | bing | ingp | ngpe | gper | peri | eril | rila | ilak | laku | akun | kuny | unya |

- ***Hashing***

Hashing adalah suatu cara untuk mentransformasi sebuah *string* menjadi suatu nilai yang unik dengan panjang tertentu (*fixed-length*) yang berfungsi sebagai penanda *string* tersebut. Berikut adalah persamaan untuk mendapatkan nilai *hashing* untuk setiap K-gram.

$$H = C_1 * a^{(k-1)} + C_2 * a^{(k-2)} + \ldots + C_3 * a^{(0)} \quad (17)$$

Keterangan:

H = Nilai *hashing*
C_k = Nilai ascii dari setiap huruf pada K-grams
A = Nilai konstanta / basis
K = Banyak karakter

- ***Similarity***

Similarity digunakan untuk menghitung tingkat kemiripan suatu teks, dimana yang dibandingkan adalah nilai hasil proses *hashing*. Untuk mengukur persentase kemiripan menggunakan metode *Dice's*

Similarity Coefisient dimana *hash value* yang sama dari dua dokumen teks abstrak yang berbeda dihitung dan jumlahnya disimpan dalam bentuk array. Berikut adalah persamaan *similarity* metode *Dice's Similarity Coefficient*.

$$S = \frac{2 * C}{A + B} \quad (18)$$

Keterangan:

S = Nilai *similarity* antar kedua teks
A = Jumlah kumpulan K-grams dalam teks 1
B = Jumlah kumpulan K-grams dalam teks 2
C = Jumlah dari K-grams yang sama dari teks yang dibandingkan

Penelitian yang dilakukan berfokus untuk pengenalan tulisan tangan yang diperoleh dari hasil ekstraksi ciri PCA. Dimana ekstraksi ciri yang dihasilkan adalah 13 ekstraksi ciri dan akan diklasifikasikan menggunakan metode K-NN. Pada proses pelatihan citra latih dilakukan pelatihan dengan cara mengekstraksi citra karakter terurut dari A-Z, a-z, 0-9 untuk diambil *eigenvalue* dan *principal component* (PC). Setelah didapatkan *eigenvalue* dan *principal component* (PC) dari setiap citra karakter, maka *eigenvalue* dan *principal component* (PC) akan disimpan dalam *database* yang akan digunakan kembali pada proses pengujian. Data yang digunakan 13 data tulisan tangan citra pelatihan dan 5 data tulisan tangan citra pengujian.

2.3. Pengujian Akurasi

Pada pengujian akurasi akan dihitung nilai ketepatan atau kecocokan dari hasil aplikasi pengenalan tulisan tangan jawaban esai. Pengujian dilakukan setelah data *training* sudah dilakukan proses pelatihan.

Parameter yang digunakan pada penelitian ini yaitu nilai *K* dari metode klasifikasi K-NN yang digunakan pada tahap pengujian. Pengujian ini dilakukan terhadap 806 data set. Berikut merupakan percobaan yang dilakukan dengan parameter yang berbeda pada pengujian citra.

Tabel 4 Hasil Pengujian

No	K	Akurasi (%)
1	1	74.83
2	2	74.83
3	3	76.62
4	4	77.55
5	5	77.79

Berdasarkan hasil pengujian akurasi terhadap 5 data uji yang telah dilakukan, evaluasi hasil akurasi yang didapatkan terhadap pengenalan tulisan tangan esai yaitu sebagai berikut:

a. Hasil akurasi yang memiliki persentase paling tinggi yaitu pengujian ke-1 menggunakan K = 5 dengan akurasi sebesar 77,79% terhadap data uji tulisan tangan.
b. Hasil akurasi yang memiliki persentase paling rendah yaitu pengujian ke-5 menggunakan K = 1 dan K = 2 dengan akurasi sebesar 74,83% terhadap data uji tulisan tangan.

Semakin nilai K besar maka tingkat akurasi semakin tinggi. Nilai K sangat berpengaruh pada klasifikasi K-NN. Dan sistem dapat memberikan nilai penilaian jawaban esai menggunakan algoritma *Rabin Karp* dengan cara membenadingkan kesamaan teks antara pengenalan jawaban dengan kuci jawaban.

3. PENUTUP

Pada bagian ini akan menjelaskan kesimpulan dan saran dari penelitian.

3.1. Kesimpulan

Pada penelitian ini telah menghasilkan sebuah sistem pengenalan tulisan tangan untuk penilian esai menggunakan metode KNN dan PCA. Pengenalan dilakukan terhadap 62 jenis output yang terdiri dari huruf A-Z, a-z dan angka 0-9.

Berdasarkan hasil pengujian yang telah dilakukan, pengujian terhadap sistem pengenalan tulisan tangan sebagai berikut:

1. Sistem pengenalan tulisan tangan menggunakan metode K-NN dan PCA diperoleh akurasi terbaik sebesar 77,79% dengan nilai K=5. Akurasi ini dipengaruhi oleh PCA Builder yang dihasilkan pada proses pelatihan, jumlah data latih dan data uji yang digunakan.
2. Sistem pengenalan tulisan tangan dapat memberikan nilai penilaian esai.

3.2. Saran

Sistem yang saat ini dibangun masih dapat dikembangkan lagi sesuai denngan kebutuhan pengguna dalam mendapatkan hasil akurasi yang lebih baik. Saran untuk pengembangan selanjutnya pada sistem yang dapat dijadikan pertimbangan yaitu:

1. Menggunakan metode segmentasi yang dapat digunakan untuk segmentasi citra tulisan tangan berdempet dan tulisan tangan bersambung.
2. Menggunakan metode ekstraksi ciri dan klasifikasi yang lebih baik untuk mendapatkan akurasi terbaik.
3. Menggunakan metode essay scoring yang lebih baik pada proses penilaian esai.
4. Memperhatikan makna dalam kalimat agar dapat memberikan penilaian jawaban yang tepat.

DAFTAR PUSTAKA

[1] R. S. Bahri dan I. Maliki, “Perbandingan Algoritma Template Matching dan Feature Extraction Pada Optical Character Recognition,” *Jurnal Komputer dan Informatika (KOMPUTA)*, vol. I, no. 1, hal. 29-35, 2012.

[2] M. F. Miftah, “Pengenalan Karakter Huruf Tulisan Tangan Menggunakan Metode Principal Components Analysis,” dalam *Tugas Akhir Jurusan Teknik Informatika Fakultas Ilmu Komputer*, Bandung, Universitas Dian Nuswantoro, 2013.

[3] D. Puspitaningrum, D. K. Sari dan B. Susilo, “Dampak Reduksi Sampel Menggunakan Principal Components Analysis (PCA) Pada Pelatihan Jaringan Syaraf Tiruan Terawasi (Studi Kasus : Pengenalan Angka Tulisan Tangan),” *Jurnal Pseudecode*, vol. II, no. 1, hal. 83-89, 2014.

[4] R. Akbar dan E. A. Sarwoko, “Studi Analisis Pengenalan Pola Tulisan Tangan Angka Arabic (Indian) Menggunakan Metode K-Nearest Neighborsdan Connected Components Labeling,” *Dinamika Rekayasa*, vol. XII, no. 2, hal. 45-51, 2016.

[5] I. A. Ersyaputra, "Analisis Dan Implementasi Pengenalan Citra Tulisan Tangan Alfabet Menggunakan Metode Modified Direction Feature dan Klasifikasi Jaringan Saraf Tiruan-Learning Vector Quantization," Ilmu Komputasi, Fakultas Teknik Informatika, Universitas Telkom, Bandung, 2013.

[6] S. Hamza, M. Sarosa dan P. Budi Santoso, “Sistem Koreksi Soal Essay Otomatis Dengan Menggunakan Metode Rabin Karp,” *Jurnal EECCIS*, vol. 7, no. 2, hal. 153-158, 2013.

[7] D. Putra, Pengolahan Citra Digital, Yogyakarta: ANDI, 2010.

[8] A. Kadir dan A. Susanto, Teori dan Aplikasi Pengolahan Citra, Yogyakarta: ANDI, 2013.

[9] C. Solomon and T. Breckon, Fundamental of Digital Image Processing, West Sussex: John Wiley & Sons, Ltd, 2011.

[10] K. Khurshid, I. Siddiqi, C. Faure and N. Vincent, "Comparison of Niblack Inspired Binarization Methods for Ancient Document," in *Document Recognition and Retrieval XVI*, San Jose, 2009.

[11] R. Lesmana, "Pengenalan Karakter Tulisan Tangan Menggunakan Ekstraksi Fitur PCA & LDA," *Jurnal Teknik Elektro Universitas Kristen Petra*, 2003.

[12] B. Santosa, Data Mining: Teknik Pemanfaatan Data untuk Keperluan Bisnis, Yogyakarta: Graha Ilmu, 2007.

[13] A. Khan and H. Farooq, "Principal Component Analysis-Linear Discriminant Analysis Feature Extractor for Pattern Recognition," *IJCSI International Journal of Computer Science Issues*, vol. VIII, no. 2, hal. 267-270, 2011.

[14] H. Noprisson, B. Susilo dan E. , “Implementasi Algoritma Rabin-Karp Untuk Menentukan Keterkaitan Antar Publikasi Penelitian Dosen Tahun 2013,” *Jurnal Teknologi Informasi*, vol. 10, no. 1, hal. 110-127, 2014.