IMPLEMENTASI METODE SUPPORT VECTOR MACHINE (SVM) DAN ZONING UNTUK PENGENALAN TULISAN TANGAN PADA KASUS PENGECEKAN JAWABAN UJIAN

Muhammad Ferderico Soleh[1], Ken Kinanti Purnamasari[2]

[1,2]Teknik Informatika - Universitas Komputer Indonesia
Jl. Dipatiukur 112-114 Bandung
E-mail : ferdericosoleh@email.unikom.ac.id[1], ken.kinanti@email.unikom.ac.id[2]

ABSTRAK

Tulisan tangan merupakan salah satu bagian bahasa dari bahasa secara umumnya. Tulisan tangan dapat dimiliki oleh semua orang dan tulisan setiap orang berbeda-beda, tergantung dengan tingkat keunikan dan karakteristik yang dimiliki. Telah banyak penelitian yang dikaji terkait pengenalan citra tulisan tangan. Hasil dari pengenalan citra tulisan tangan yang didapatkan bervariasi dan dipengaruhi oleh data latih dan metode ekstraksi fitur dan klasifikasi. Dalam penelitian ini, metode ekstraksi fitur yang digunakan adalah *Zoning(Image Centroid Zone-Zone Centroid Zone),* metode klasifikasi yang digunakan adalah *Support Vector Machine* untuk melakukan pengenalan citra tulisan tangan yang selanjutnya akan diimplementasikan pada kasu evaluasi jawaban ujian esai. Hasil evaluasi jawaban ujian pada penelitian ini dipengaruhi oleh akurasi pengenalan tulisan tangan yang didapatkan. Sebelum melakukan proses klasifikasi , citra tulisan tangan akan melalui tahap *Grayscale, Threshold,* Segmentasi, *Resize* dan ekstraksi fitur *Zoning.* Dan selanjutnya dilakukan proses pelatihan dan pengujian oleh metode *Support Vector Machine.* Setelah didapatkan hasil pengenalannya maka langkah terakhir yaitu evaluasi jawaban ujian dengan metode *rabin-karp* yang hanya bertugas untuk menentukan tingkat kesamaan antara hasil pengenalan dan jawaban ujian yang disediakan. Karakter tulisan tangan yang dapat dikenali ada 62 karakter terdiri dari A-Z, a-z dan 0-9. Data *training* yang digunakan pada penelitian ini menggunakan 1 sampel tulisan tangan untuk 62 data kelas yang ada, yang berarti (1 x 62 = 62) dan 100 sampel tulisan tangan untuk data latih yang artinya ada 6.200 karakter (100 x 62 = 6200). Kemudian data uji yang digunakan ada 1 data tulisan jawaban esai yang berbeda-beda. Berdasarkan hasil pengujian yang dilakukan terhadap data uji dengan menggunakan 10 kombinasi parameter pelatihan, maka didapatkan akurasi terbaik sebesar 77,6 %.

Kata kunci : Kecerdasan Buatan, *Support Vector Machine, Zoning,* pengenalan citra tulisan tangan, pengolahan citra, SVM.

1. PENDAHULUAN

Tulisan tangan merupakan salah satu bagian bahasa dari bahasa secara umumnya. Tulisan tangan dapat dimiliki oleh semua orang dan tulisan setiap orang berbeda-beda, tergantung dengan tingkat keunikan dan karakteristik yang dimiliki. Berdasarkan perbedaan tersebut banyak penelitian yang membahas tentang tulisan tangan. Penelitian ini dilakukan karena adanya keterbatasan kinerja yang dilakukan oleh sistem yang terbilang lambat dalam waktu pengecekkan dan penilaian jawaban ujian, dan belum ditemukannya hasil optimal dari penggunaan metode-metode sebelumnya yang membahas kajian tentang pengenalan tulisan tangan *Handwritting Recognition*. Pada penelitian tersebut mendapatkan rata-rata akurasi terbaik sebesar 55,7 % [1].

Support Vector Machine (SVM) adalah suatu sistem pembelajaran yang menggunakan ruang hipotesis dari suatu fungsi *linear* dalam suatu ruang dimensi berfitur tinggi yang dikembangkan oleh Boser, Guyon, Vapnik, dan pertama kali dipresentasikan pada tahun 1992 di *Annual Workshop on Computational Learning Theory.* Metode *Support Vector Machine* (SVM) dalam penelitian ini digunakan karena telah diimplementasikan untuk berbagai macam penelitian dengan mengangkat bahasan yang sama yaitu pengenalan tulisan tangan dan terbilang efisien dalam penggunaannya[2]. Penggunaan metode klasifikasi ini akan dikombinasikan dengan metode ekstraksi fitur *Zoning-Zone Centroid Zone dan Image Centroid Zone* yang diyakini mampu menghasilkan hasil yang akurat dalam pengenalan tulisan, metode ini dipilih karena berdasarkan penelitian sebelumnya apabila metode klasifikasi SVM dan Zoning dikombinasikan akan lebih akurat[2].

Metode *Support Vector Machine* sudah pernah digunakan dalam mengklasifikasikan beberapa kasus, seperti penerapan metode SVM pada pengenalan aksara sunda[2], penerapan metode klasifikasi *Support Vector Machine* pada data akreditasi sekolah SD[3], identifikasi tanda tangan [4], pengenalan pola aksara Jawi[5], pengenalan tulisan tangan Persia menggunakan SVM [6], dan pengenalan karakter tulisan tangan angka dan operator matematika berdasarkan Algoritma *Support*

Vector Machine dan Hybrid [7], metode *support vector machine* sudah pernah dikombinasikan dengan metode *Zoning* dalam kasus pengenalan tulisan tangan huruf hijaiyah[8], dan juga diterapkan dalam pengenalan aksara sunda dengan SVM[9]. Dari berbagai penelitian tersebut digunakan metode klasifikasi *Support Vector Machine* (SVM) dengan hasil akurasi yang berbeda-beda. Hasil tersebut dipengaruhi oleh data training dan metode ekstraksi ciri yang digunakan. Pada penelitian ini menggunakan metode *Support Vector Machine Multiclass (One Vs All),* metode ini digunakan karena pada penelitian perbandingan *multiclass (svm)*[23], telah mendapatkan rata-rata akurasi terbaik sebesar 90%.

Maka, dalam penelitian ini akan dibuat aplikasi pengenalan tulisan tangan jawaban esai dengan menggunakan metode klasifikasi *Support Vector Machine* dengan ekstraksi ciri *Zoning (Image Centroid Zone-Zone Centroid Zone)*. Kemudian setelahnya akan dilakukan perhitungan akurasi dilakukan dengan menghitung banyaknya karakter tulisan tangan yang dapat dikenali oleh aplikasi.

1.1 Pengolahan Citra Digital

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 [15]. Foto adalah contoh gambar berdimensi dua yang dapat diolah dengan mudah. Setiap foto dalam bentuk citra digital dapat diolah menggunakan perangkat lunak tertentu. Dalam penelitian ini pengolahan citra yang dilakukan diantaranya *grayscale*, *threshold*, segmentasi, *resize* dan *feature extraction.*

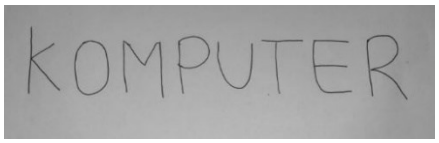
1.1.1 *Grayscale*

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 [19]. Berikut adalah rumus konversi citra berwana (RGB) menjadi *grayscale*.

$$I = (0.2989 * R)+(0.5870*G)+(0.1141*B) \quad (1)$$

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*



Gambar 1 Contoh Citra Hasil *Grayscale*

1.1.2 *Threshold*

Metode threshold yang digunakan dalam penelitian ini adalah *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* [18]. *Sauvola* termasuk dalam *local threshold* dimana nilai ambang ditentukan oleh nilai *pixel* tetangga. Berikut adalah rumus *sauvola threshold* T(x,y).

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

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

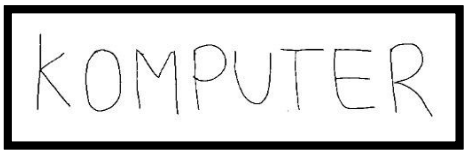
$$s(x,y) = \sqrt{\frac{\sum_{i=min}^{i=max} \sum_{j=min}^{j=max} (img(i,j)-m(x,y))^2}{(i*j)-1}} \quad (4)$$

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

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

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

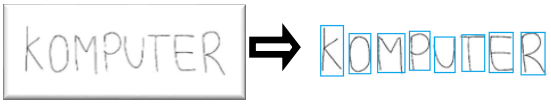


Gambar 2 Contoh Citra Hasil *Threshold*

1.1.3 Segementasi

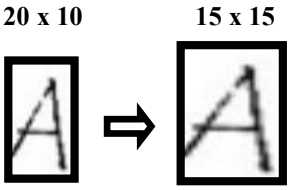
Segmentasi merupakan teknik untuk membagi suatu citra menjadi beberapa daerah (*region*) dimana setiap daerah memiliki kemiripan atribut [19]. Sampai sejauh mana pembagian tersebut dilakukan tergantung pada masalah yang dihadapi. Idealnya, langkah segmentasi tersebut dihentikan pada saat objek yang diinginkan sudah berhasil dipisahkan. Dalam penelitian ini segmentasi dilakukan dengan pemotongan baris dan kolom citra.

Pemotongan pertama dilakukan untuk seluruh baris citra tulisan tangan, kemudian dilanjutkan dengan pemotongan kolom untuk setiap citra hasil pemotongan baris.



1.1.4 *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.



Gambar 3 Contoh Hasil Segmentasi

1.1.6 Ekstraksi Ciri *Zoning*

Ekstraksi ciri Zoning merupakan, ekstraksi ciri dengan konsep utamanya adalah pembagian citra kedalam sejumlah zona yang sama agar dapat dikenali ciri dari setiap karakter huruf dan angka, yang nantinya akan menghasilkan sebuah nilai yang akan diproses pada tahapan klasifikasi. Langkah-langkah dalam algoritma *zoning-Image Centroid Zone*

.

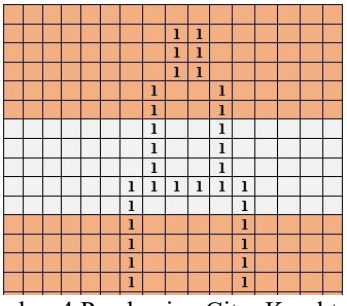
1. Hitung Nilai Centroid Citra masukan.

$$x_c = \frac{(x_1.p_1 + x_2p_2 + \ldots + x_np_n)}{(p_1 + p_2 + \ldots + p_n)}$$

$$y_c = \frac{(y_1.p_1 + y_2p_2 + \ldots + y_np_n)}{(p_1 + p_2 + \ldots + p_n)}$$



2. Bagi Ke dalam jumlah zona yang sama



Gambar 4 Pembagian Citra Karakter A Ke dalam Zona yang sama

3. Hitung Jarak Pixel hitam dari Centroid.

$$d(P, C)\sqrt{(x_p - x_c)^2 + (y_p - y_c)^2}$$

d	= jarak antara dua titik
P	= koordinat titik berat
C	= koordinat *pixel*
x_p	= koordinat x titik berat
y_p	= koordinat y titik berat
x_c	= koordinat x *pixel*
y_c	= koordinat y *pixel*

Perhitungan jarak setiap zona dapat dilihat dibawah ini.

Zona Atas (1)

(8,1) : Jarak $=\sqrt{(8-8)^2 + (1-8)^2} = 7$

(9,1):Jarak $=\sqrt{(9-8)^2 + (1-8)^2} = 7.07$

(8,2):Jarak $=\sqrt{(8-8)^2 + (2-8)^2} = 6$

4. Hitung Jarak Rata-Rata Antara Titik.

Zona 1 : Rataan Jarak

$$= \frac{(jarak(8,1)+Jarak(9,1)+Jarak(10,5)+\cdots+Jarak(2,9)}{Banyak\ Titik\ (10)}$$

$$= \frac{51.61592037200608827}{10} = 5.161592037200608827$$

Zona 2 : Rataan Jarak

$$= \frac{(jarak(7,1)+Jarak(10,1)+Jarak(7,2)+\cdots+Jarak(11,5)}{Banyak\ Titik\ (14)}$$

$$= \frac{71.2813549963260149}{14} = 5.0915253568804296357$$

Zona 3 : Rataan Jarak =

$$\frac{(jarak(6,1)+Jarak(6,2)+Jarak(6,3)+\cdots+Jarak(11,4)}{Banyak\ Titik\ (8)}$$

$$= \frac{48.6168949049599383}{8} = 6.07711186311999228$$

5. Didapatkan tiga nilai fitur rata-rata.

Zona 1(atas) = 5.1615920372006088

Zona 2(tengah) = 5.0915253568804296

Zona 3(bawah) = 6.0771118631199922

Langkah-langkah Ekstraksi Ciri *Zoning-Zone Centroid Zone.*

1. Langkah yang dilakukan sama seperti tahapan *Image centroid zone.* Bedanya adalah nilai fitur yang dihasilkan dan nilai centroid didapat dari setiap zona.

2. 3 Nilai Fitur yang dihasilkan

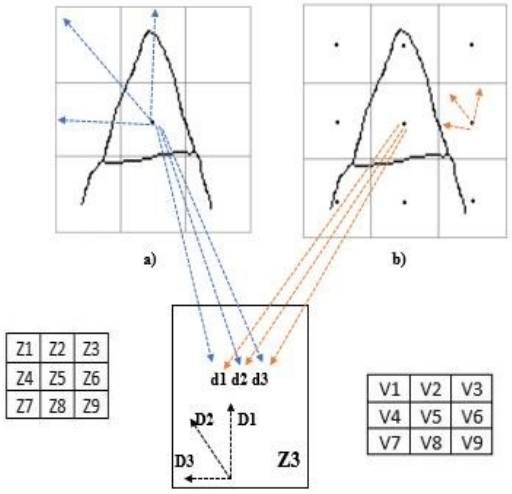
Zona 1(atas) = 1.436505818199174

Zona 2(tengah) = 2.11511398444594

Zona 3(bawah) = 2.7788337094433146

Karakter	Nilai Fitur ICZ (Image Centroid Zone)	Nilai Fitur ZCZ (Zone Centroid Zone)
	**ICZ** 1= 5.1615920372006 2= 5.0915253568804 3= 6.0771118631199	**ZCZ** 1= 1.43650581819917 2 = 2.11511398444594 3 = 2.77883370944331
	**ICZ** 1= 2.3030865650810 2 = 2.6943777608364 3 = 2.3030865650810	**ZCZ** 1 =2.9401919977662 2 =2.8277084050488 3 = 2.303086565081
	**ICZ** 1=3.0695304169120 2= 3.466455626024 3 3.3166247903553	**ZCZ** 1 = 2.631947231218 2 = 1.466455626024 3 = 2.3166247903553

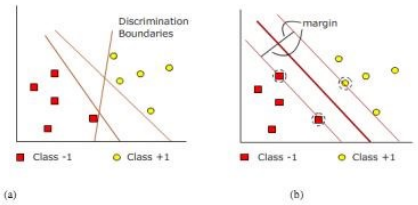
Gambar 5 Hasil Nilai Fitur Dari 3 Karakter



Gambar 6 Skema Penelsuran objek hitam

1.2 Jaringan Syaraf Tiruan (JST)

Support Vector Machine merupakan metode pembelajaran yang digunakan untuk klasifikasi biner, ide dasarnya adalah mencari hyperlane terbaik yang berfungsi sebagai pemisah dua *class* pada *input space* [21].



Gambar 7 Proses Mencari Hyperplane oleh SVM

Berbeda dengan pendekatan neural network yang berusaha mencari hyperlane pemisah antar class, SVM mencoba mencari dan memisah hyperlane yang terbaik yang berada pada dua kelas, gambar (a) menunjukan pola-pola yang merupakan anggota dari dua buah kelas +1 dan -1, pola yang berada pada anggota -1 kemudian disimbolkan dengan warna merah, sedangkan pola pada +1 warna

1.2.1 Proses Belajar Sistem

Belajar dalam konteks SVM adalah proses dimana parameter-parameter bebas SVM diadaptasikan melalui suatu proses pembelajaran biner dua kelas, yaitu :

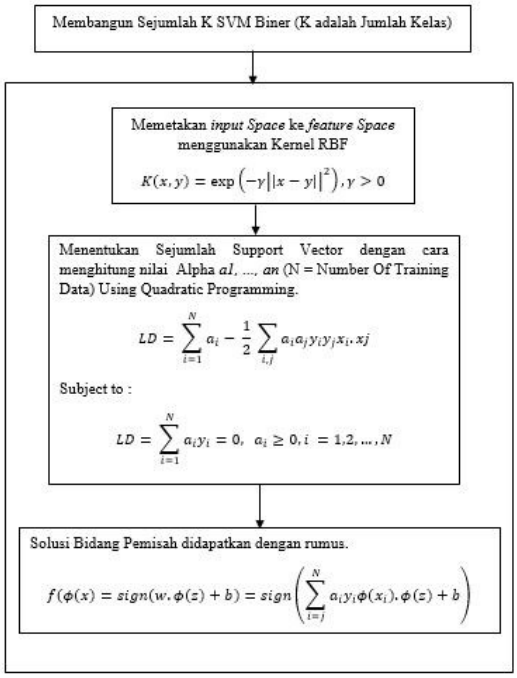
1. *Supervised Learning* (Belajar Dengan Pengawasan)
   Supervised atau *active learning* adalah proses belajar membutuhkan guru. Yang dimaksud guru disini adalah sesuatu yang memiliki pengetahuan tentang lingkungan. Guru bisa direpresentasikan sebagai sekumpulan sampel input-output. Pembangunan pengetahuan dilakukan oleh guru dengan memberikan respon yang diinginkan kepada SVM. Parameter-parameter jaringan berubah-ubah berdasarkan vektor latih dan sinyal kesalahan (sinyal kesalahan adalah perbedaan antara keluaran SVM dan respon yang diinginkan). Proses perubahan ini dilakukan secara berulang-ulang, selangkah demi selangkah, dengan tujuan agar SVM bisa memiliki kemampuan yang mirip dengan gurunya. Dengan kata lain, SVM dilatih untuk dapat memetakan sekumpulan sampel *input-output* dengan akurasi yang tinggi. Pada proses belajar ini, output yang diharapkan telah diketahui sebelumnya.
2. *Unsupervised Learning* (Belajar Tanpa Pengawasan)
   Unsupervised atau *self-organized learning* tidak membutuhkan guru untuk memantau proses belajar. Dengan kata lain, tidak ada sekumpulan sampel *input-output* atau fungsi tertentu untuk dipelajari oleh jaringan. Salah satu contoh *unsupervised learning* adalah *competitive learning*. Pada metode ini, tidak

ditentukan hasil yang seperti apakah yang diharapkan selama proses pembelajaran.

Dalam penelitian ini proses pembelajaran yang digunakan adalah *supervised learning.*

1.2.2 SVM *Training*

Support Vector Machine (SVM) termasuk dalam proses pembelajaran terawasi (*supervised learning*). SVM merupakan metode klasifikasi pola yang masing-masing unit output mewakili kategori atau kelas tertentu. Dalam metode ini, suatu lapisan kompetitif akan secara otomatis belajar untuk mengklasifikasikan vektor-vektor data input. Kelas-kelas yang didapatkan sebagai hasil lapisan kompetitif ini hanya tergantung pada jarak antara vektor input. Jika dua vektor mendekati sama, maka lapisan kompetitif akan meletakkan kedua vektor tersebut ke dalam kelas yang sama. Berikut merupakan arsitektur dari metode *Support Vector Machine*



Gambar 8 *Block Diagram* Algoritma Proses SVM *Training Multiclass*

Sedangkan algoritma metode SVM adalah sebagai berikut.

T : Target kelas data latih
c : Jarak selisih bobot terkecil
x : Data latih
W : Data bobot kelas
xi : Indeks suatu data
yi : Banyak data latih
N : Banyak data pada lapisan input / jumlah pixel citra
a : Lagrange Multiplier *(alpha)*

1.3 Rabin-Karp

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 [20]. Tahap tahap yang dilakukan dalam evaluasi jawaban esai yaitu proses *tokenizing* dan *case folding*, *filtering,* algoritma Rabin-Karp (K-gram, *hashing*, *similarity*).

1.3.1 *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*).

Contoh teks :

Bandung ada di Provinsi Jawa Barat

Hasil *tokenizing* dan *case folding* :

bandung, ada, di, provinsi, jawa, barat

1.3.2 *Filtering*

Filtering adalah tahap mengambil kata-kata penting dari hasil *tokenizing* dan *case folding*, dengan menggunakan *stoplist* atau *stopword* (membuang kata yang kurang penting). Contoh kata yang termasuk ke dalam *stoplist* seperti : ‘di’, ‘yaitu’, ‘adalah’, ‘merupakan’, ‘dan’, ‘yang’, ‘ini’. Berikut adalah contoh dari proses *filtering*.

Sebelum *filtering* :

bandung, ada, di, provinsi, jawa, barat

Setelah *filtering* :

bandung, provinsi, jawa, barat

1.3.3 K-gram

K-gram adalah rangkaian *terms* dengan panjang K. Kebanyakan yang digunakan sebagai *terms* adalah kata. K-gram merupakan sebuah metode yang diaplikasikan untuk pembangkitan kata atau karakter. Metode K-gram ini digunakan untuk mengambil potongan-potongan karakter huruf sejumlah k dari sebuah kata yang secara kontinuitas dibaca dari teks sumber hingga akhir dari kalimat atau dokumen [24]. Berikut adalah contoh K-gram dengan nilai k=4.

Contoh teks :

bandungprovinsijawabarat

Hasil K-gram :

band, andu, ndun, dung, ungp, ngpr, gpro, prov, rovi, ovin, vins, insi, nsij, sija, ijaw, jawa, awab, waba, abar, bara, arat

Pada kalimat di atas terdapat 21 K-grams.

1.3.4 *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. Fungsi untuk menghasilkan nilai ini disebut fungsi *hash*, sedangkan nilai yang dihasilkan disebut nilai *hash*. Nilai *hash* pada umumnya digambarkan sebagai *fingerprint* yaitu suatu *string* pendek yang terdiri atas huruf dan angka yang terlihat acak (data biner yang ditulis dalam heksadesimal) [21]. Berikut adalah persamaan untuk mendapatkan nilai *hasing* untuk setiap K-gram.

$$H = C_1*a^{(k-1)} + C_2*a^{(k-2)} + C_3*a^{(k-3)} \dots C_k*a^{0} \quad (13)$$

Keterangan :

H : Nilai *hashing*
C_k : Nilai ascii dari setiap huruf pada K-grams
a : nilai konstanta / basis
k : Banyak karakter

1.3.4 *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 Coefisient.*

$$S = \frac{2 * C}{A + B} \quad (14)$$

Keterangan :

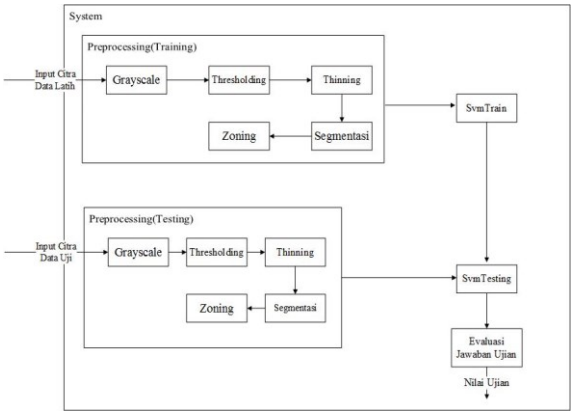
S = Nilai *similarity* antar kedua teks
A = Jumlah kumpulan K-grams dalam teks 1
B = Jumlah kumpulan K-grams dalam teks 2
C = Jumlah dari K-grams yang sama dari teks yang dibandingkan

2. ISI PENELITIAN

Dalam penelitian ini akan dibangun aplikasi pengenalan tulisan tangan dengan metode *Support Vector Machine* (SVM), kemudian input berupa citra jawaban ujian yang ditulis tangan. Data masukan diproses dengan pengolahan citra dan mengimplementasikan metode *Support Vector Machine* yang menggunakan data latih dari 100 sampel tulisan tangan, sehingga data keluaran atau akurasi yang diharapkan nilainya tinggi. Setelah citra tulisan tangan sudah dikenali, maka selanjutnya sistem dapat melakukan evaluasi jawaban ujian dari hasil pengenalan tulisan tersebut dengan menggunakan metode *Rabin-Karp.*

2.1 Arsitektur Umum Sistem

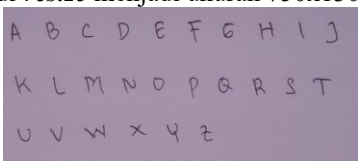
Proses yang dilakukan dalam sistem ini dibagi menjadi beberapa bagian dimana setiap proses memiliki peranan masing-masing dalam melakukan pengenalan tulisan tangan dan evaluasi jawaban ujian. Berikut proses yang dilakukan pada gambar 9.



Gambar 9 Arsitektur Sistem

2.1.1 Citra Tulisan Tangan

Data masukan yang digunakan dalam sistem ini, yaitu citra tulisan tangan dengan format gambar .jpg atau .png. Citra dapat diperoleh melalui scan atau foto dengan resolusi kamera minimal 8 *megapixel*. Citra tulisan tangan yang digunakan ditulis menggunakan pulpen dalam kertas dengan *background* putih. Selain itu warna citra juga tidak terlalu gelap agar aplikasi dapat lebih maksimal melakukan pengenalan karakter dari setiap tulisan tangan. Dalam penelitian ini data masukan secara otomatis di-*resize* menjadi ukuran 750x1300 *pixel*.



Gambar 10 Contoh Citra input Tulisan Tangan

2.1.2 Pengolahan Citra

Dalam pengolahan citra dilakukan beberapa proses agar citra input dapat digunakan pada tahap klasifikasi menggunakan metode *Support Vector Machine* (SVM). Berikut adalah penjabaran dari proses pengolahan citra yang dilakukan sesuai dengan arsitektur sistem yang dibuat.

1. *Grayscale*
   Setelah citra tulisan tangan sudah dimasukkan ke dalam aplikasi, kemudian dilakukan proses *grayscale* citra. *Grayscale* merupakan tahap awal yang dilakukan dalam tahap pengolahan citra. Hasil dari proses ini nantinya menghasilkan citra dengan range warna dari 0 sampai dengan 255 yang merupakan skala keabuan.
2. *Thresholding*

Langkah selanjutnya dalam pengolahan citra adalah thresholding. Proses ini dilakukan terhadap citra *grayscale*. Metode *threshold* yang digunakan dalam penelitian ini adalah *sauvola threshold*. Sauvola termasuk kedalam *local threshold*, dimana nilai ambang yang didapatkan tergantung dari jumlah tetangga yang digunakan. Hasil akhir dari proses ini adalah citra hitam putih.

3. Segmentasi

Dalam proses segmentasi, citra harus sudah melalui proses *thresholding*. Hal ini dilakukan agar sistem lebih mudah dalam melakukan pemotongan citra. Segmentasi dilakukan terhadap baris citra terlebih dahulu, sehingga didapatkan citra hasil pemotongan baris. Selanjutnya dilakukan pemotongan kolom terhadap setiap citra hasil pemotongan baris. Hasil dari proses ini yaitu mendapatkan citra per karakter dari suatu citra tulisan tangan.

4. *Resize*

Citra hasil segmentasi, tentu ukurannya masih beragam. Oleh karena itu perlu diseragamkan menjadi suatu ukuran tertentu agar sistem dapat konsisten membaca citra. Dalam penelitian ini citra di-*resize* menjadi ukuran 15 x 15 *pixel*.

5. Ekstraksi Ciri

Binerisasi merupakan proses terakhir dalam tahap pengolah citra yang dilakukan. Binerisasi digunakan untuk mendapatkan nilai 0 atau 1 dari suatu citra hitam putih. Hasil dari proses ini yaitu berupa deretan angka biner yang nantinya digunakan sebagai data input pada tahap klasifikasi menggunakan metode *Support Vector Machine* (SVM).

2.1.3 *Support Vector Machine* (SVM)

Metode klasifikasi yang digunakan dalam penelitian ini adalah *Support Vector Machine* (SVM). Klasifikasi dilakukan setelah citra tulisan tangan sudah melewati proses pengolahan citra. Adapun proses yang dilakukan pada tahap ini yaitu proses pelatihan dan pengujian. Berikut adalah penjelasan untuk setiap proses yang dilakukan.

1. Pelatihan SVM

Dalam proses pelatihan SVM digunakan data training yang terdiri dari data kelas dan data latih. Banyaknya data training yang digunakan dalam penelitian ini yaitu, data kelas sebanyak 62 citra (A-Z,a-z,0-9) dan data latih sebanyak 6.200 citra (A-Z,a-z,0-9). Kemudian juga digunakan parameter pelatihan (*learning rate*, pengurangan *learning rate*, Max. Epoch dan *error minimum*). Setelah proses pelatihan ini dilakukan, hasil akhirnya akan memperbaharui data bobot kelas dari semua data kelas. Data bobot inilah yang nantinya digunakan dalam tahap pengujian.

2. Pengujian SVM

Dalam proses pengujian SVM digunakan citra tulisan tangan jawaban esai, kemudian juga dilakukan pengolahan citra seperti proses sebelumnya.



Gambar 11 Contoh Citra Uji

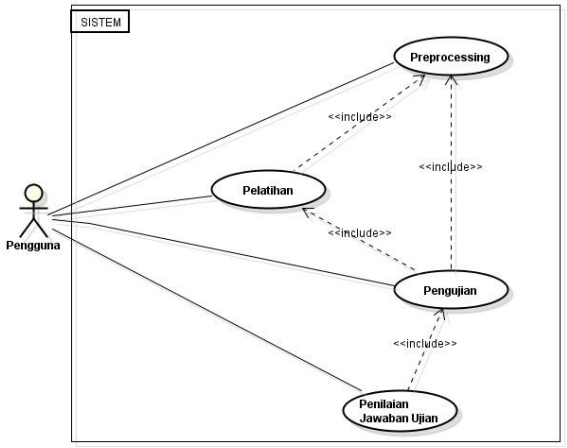
Setelah dilakukan proses pengolahan citra, kemudian dilanjutkan dengan proses pengujian dengan membandingkan nilai bobot dari setiap citra uji dengan bobot kelas yang sudah diperbaharui pada tahap pelatihan sebelumnya. Hasil dari pengujian ini adalah teks digital sesuai dengan yang ditulis pada citra uji. Namun, dalam proses pengujiannya ada kemungkinan hasil yang didapatkan kurang sempurna. Hasil dari pengujian inilah yang nantinya diolah kembali pada tahap evaluasi jawaban ujian.

2.1.4 Evaluasi Jawaban Ujian

Proses evaluasi jawaban ujian dalam penelitian ini hanya menentukan tingkat kemiripan antara kunci jawaban dengan jawaban ujian yang didapatkan dari proses pengujian SVM. Metode yang digunakan yaitu Rabin-Karp yang terdiri dari K-gram, *hashing* dan *similarity* dengan proses awal *tokenizing* dan *case folding* kemudian *filtering*. Hasil akhir yang didapatkan pada proses ini yaitu nilai ujian berdasarkan kesamaan antara teks jawaban dengan kunci jawaban ujian.

2.2 Use Case Diagram

Use case diagram merupakan pemodelan yang menunjukkan interaksi antara sistem dengan pengguna (aktor). Dalam *use case* yang dibuat menggunakan lima *use case*, dimana ada *use case* yang berelasi dengan *use case* lainnya. Berikut adalah *use case diagram* untuk aplikasi yang dibangun.



Gambar 12 *Use Case Diagram*

2.3 Pengujian Akurasi

Pada pengujian akurasi akan dihitung nilai ketepatan atau kecocokan dari hasil aplikasi pengenalan tulisan tangan jawaban esai. Dalam penelitian ini pengujian akurasi digunakan 10 kombinasi parameter pelatihan dengan 10 citra uji tulisan tangan jawaban esai yang berbeda. Berikut adalah tabel kombinasi parameter yang digunakan.

Tabel 1 Kombinasi Parameter Pelatihan

Kombinasi ke-	*Probability*	Gamma	*Nu*	C
1	1.5	0,5	3	1.2
2	1.8	0,7	2	1.1
3	1.9	0,9	1	1.5
4	2	0,5	0,5	1
5	2.6	0,2	0,4	1.7
6	2.1	0.5	0,5	2
7	3	1.5	0,6	1.4
8	1	0,5	0,7	1.6
9	0.5	2	0,8	1.8
10	1.2	1.1	0,5	1.3

Tabel 1 di atas merupakan kombinasi parameter pelatihan yang digunakan pada sistem untuk melakukan pengenalan untuk setiap citra uji tulisan tangan. Pengujian dilakukan satu persatu untuk setiap kombinasi parameter pelatihan terhadap semua citra uji yang digunakan, sehingga hasil yang didapatkan yaitu rata-rata akurasi pengenalan tulisan tangan. Berikut adalah tabel hasil pengujian yang didapatkan.

Tabel 2 Hasil Pengujian

Kombinasi ke-	Akurasi
1	70%
2	60,8%
3	76%
4	77,6%
5	71,4%
6	70,9%
7	66,6%
8	74%
9	71,7%
10	75,7%

Dari tabel hasil pengujian diatas, kombinasi parameter pelatihan terbaik adalah kombinasi parameter ke 4 dengan akurasi rata-rata yang didapatkan adalah sebesar 77,6%. Kombinasi parameter pelatihan yang digunakan yaitu *Probability* = 2, Gamma = 0,5, *Nu*= 0,5, dan C = 1.

3. PENUTUP

3.1 Kesimpulan

Berdasarkan hasil pengujian sistem pengenalan tulisan tangan pada kasus jawaban esai menggunakan metode *Support Vector Machine* maka diperoleh akurasi terbaik sebesar 77,6%. Akurasi ini dipengaruhi oleh parameter pelatihan, data training dan data uji yang digunakan.

3.2 Saran

Agar penelitian selanjutnya tentang pengenalan tulisan tangan menggunakan metode *Support Vector Machine* dalam kasus evaluasi jawaban esai memiliki akurasi yang lebih tinggi, maka berikut adalah saran yang dapat dijadikan pertimbangan, yaitu:

1. Dibutuhkan metode segmentasi citra tulisan tangan untuk kasus tulisan berdempet.
2. Dalam tahap evaluasi jawaban isian singkat, dapat digunakan metode evaluasi jawaban jawaban yang lain seperti metode *TF-Idf, Metaphone, Zhu-Takaoka,* dan *Cosine Similarity.*
3. Dibutuhkan metode ekstraksi fitur lain yang lebih baik untuk gambar seperti *Wavelet Transform, Hybrid Feature,* dan *Template matching*

DAFTAR PUSTAKA

[1] I. k. meganjaya, " pengenalan tulisan tangan menggunakan metode learning vector quantization(LVQ) dalam kasus evaluasi jawaban ujian,Unikom, Bandung, 2016.

[2] I. Mulia, "pengenalan aksara sunda menggunakan ekstraksi ciri zoning dan klasifikasi SVM," IPB, Bogor, 2012.

[3] Y. w. d. I. Pusphita Anna Octaviani, "penerapan metode klasifikasi support vector machine pada data akreditasi sekolah dasar," *Jurnal Gaussian,* Vol. %1 dari %23, No. 4 , no. Jurnal Gaussian, pp. 811-820, 2014.

[4] D. P. A. M. Endina Putri P, "Identifikasi tanda tangan dengan pendekatan SVM," *jurnal Sains,* vol. 12, no. jurnal Sains, pp. -, Juni 2015 .

[5] R. M. S., "Pengenalan Aksara Jawa Tulisan Tangan Menggunakan Ekstraksi Fitur Zoning dan Klasifikasi K-Nearest Neighbour," IPB, Bogor, 2013.

[6] M. S. A. Hadieh Sajedi, "Persian Handwritten Digit Recognition Using Support Vector Machines," *International Journal Of Computer Applications,* vol. 29, no. International Journal, p. 12, 2011.

[7] I. J. a. W. M. Muhammad Naeem Ayyaz, "Handwritten character recognition using multclass SVM classification with Hybrid Feature Extraction," *J.Engg & Appl,* vol. 10, no. J.Engg & Appl, pp. -, Januari 2012.

[8] N. Amelia, "pengenalan pola huruf hijaiyah menggunakan SVM," Universitas Sumatera Utara, Medan, 2016.

[9] M. I. a. Musthofa, "Sundanese Script Recognition Using Zoning Feature Extraction and Support Vector Machine Classification," *Department Of Computer Science,* vol. 21, no. Bogor Agricultural University, Bogor, p. 1, 2013.

[10] Sugiyono, Metode Penelitian Kombinasi (Mixed Methods), Bandung: Alvabeta, 2011.

[11] D. Susanto, Sistem Informasi Manajemen (Konsep dan Pengembangan), Bandung: Lingga Jaya, 2004.

[12] R. S. Pressman, Software Engineering Practitioner's Approach, United States: McGraw Hills, 2010.

[13] D. T. Sutoyo., Teori Pengolahan Citra Digital, Yogyakarta: AndiPublisher, 2009.

[14] C. M. Bishop, Pattern Recognition and Machine learning, Cambridge: Springer, 2006.

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

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

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

[18] R. P. Rajashekararadhya SV, "Novel Zone Based Feature Extraction Algorithm For Handwritten Numeral Recognition Of Four Indian Scripts," *Digital Technology Journal,* vol. 2, pp. 41-51, 2009.

[19] Suyanto, Artificial Intelligence (Searching,Reasoning, Planning and Learning), Bandung: Informatika Bandung, 2014.

[20] C. a. T. Breckon, Fundamentak of Digital Image Processing,, West Sussex : John Wiley, 2011.

[21] H. .. K. Kopriva, "Kernel Based Algorithms for mining huge data sets," dalam *17 edition*, USA, Springer, 2003, p. 460.

[22] Krisantus, Tutorial SVM Berbahasa indonesia, Yogyakarta: Krisantus, 2003.

[23] J. Marsha, "Klasifikasi Teks Dengan TF-IDF Pada Artikel Berbahasa Indonesia dengan Model LDA," Unikom, Bandung, 2016.

[24] H. B., Rekayasa Sistem Berorientasi Objek, Bandung: Informatika, Bandung, 2004.

[25] M. a. B. S. P. Hamza S, "Sistem koreksi Soal Essay Otomatis Dengan Menggunakan Metode Rabin-Karp," *Jurnal EECCIS,* vol. 7, pp. 153-158,

2013.

[26] R. Sianipar, Teori dan Implementasi Java, Bandung: Informatika, 2013.

[27] R. A. a. M.Shalahudin, Rekayasa Perangkat Lunak 'Terstruktur dan Berorientasi objek', Bandung: Informatika, Bandung, 2014.