

## Pengenalan Karakter Tulisan Tangan Dengan K-Support Vector Nearest Neighbor

Aditya Surya Wijaya<sup>1</sup>, Nurul Chamidah\*<sup>2</sup>, Mayanda Mega Santoni<sup>3</sup>

<sup>1,2,3</sup>Fakultas Ilmu Komputer, Universitas Nasional Pembangunan Veteran, Jakarta

e-mail: <sup>1</sup>[adityasuryawijaya519@gmail.com](mailto:adityasuryawijaya519@gmail.com), \*<sup>2</sup>[nurul.chamidah@upnvj.ac.id](mailto:nurul.chamidah@upnvj.ac.id),

<sup>3</sup>[megasantoni@upnvj.ac.id](mailto:megasantoni@upnvj.ac.id)

### Abstrak

Karakter pada tulisan tangan cukup sulit untuk dikenali dengan mesin karena setiap orang memiliki gaya penulisan yang berbeda-beda. Penelitian ini mengenali pola karakter angka dan huruf pada tulisan tangan dengan memanfaatkan algoritma klasifikasi K-Nearest Neighbor (KNN). Proses pengenalan tulisan tangan dilakukan dengan memproses citra tulisan tangan, melakukan segmentasi untuk mendapatkan karakter tunggal dari sekumpulan karakter pada citra, ekstraksi ciri untuk mendapatkan fitur, dan klasifikasi yang terdiri dari proses pelatihan dan pengujian. Ekstraksi ciri dilakukan dengan metode Zona yang akan dimanfaatkan untuk klasifikasi dengan membagi data kedalam data latih dan data uji. Data latih dari hasil ekstraksi ciri direduksi menggunakan metode K-Support Vector Nearest Neighbor (K-SVNN) dengan hanya mengambil support vector sebagai data latih. Proses terakhir, untuk mengenali pola tulisan tangan dari data uji digunakan K-Nearest Neighbor (KNN). Hasil pengujian menunjukkan bahwa proses reduksi data latih K-SVNN berhasil meningkatkan akurasi pengenalan karakter tulisan tangan.

**Kata kunci**—Tulisan Tangan, KSVNN, KNN, Zona, Klasifikasi

### Abstract

Handwritten characters are difficult to be recognized by machine because people had various own writing styles. This research recognizes handwritten character pattern of numbers and alphabet using K-Nearest Neighbour (KNN) algorithm. Handwritten recognition process is done by preprocessing handwritten image, segmentation to obtain separate single characters from a set of character in an image, feature extraction to get feature vector, and classification consisting of training and testing process. Features extraction is done by utilizing Zone method that will be used for classification by splitting features data into training data and testing data. Training data from extracted features is reduced by K-Support Vector Nearest Neighbor (K-SVNN) by collecting only support vector data as training data. Last process, recognizing handwritten pattern from testing data, we used K-Nearest Neighbor (KNN). Test result shows that reducing training data using K-SVNN able to improve handwritten character recognition accuracy.

**Keywords**—Handwritten, KSVNN, KNN, Zone, Classification

## 1. PENDAHULUAN

Optical Character Recognition (OCR) merupakan suatu software yang biasa digunakan untuk mengkonversi teks tercetak menjadi teks digital yang dapat diolah dengan mesin [1]. Untuk mengkonversi, teks tercetak perlu dipindai dan diubah menjadi suatu citra digital yang tidak dapat langsung dikenali sebagai teks oleh mesin. Oleh karena itu, penelitian-penelitian terdahulu telah banyak dilakukan untuk menghasilkan pengenalan dari OCR yang mampu mentransformasi dokumen yang berupa citra menjadi format yang dapat dibaca oleh mesin.

Pengenalan karakter pada tulisan hasil ketikan atau cetakan mesin lebih mudah dilakukan karena tulisan mesin lebih konsisten dengan jenis huruf yang sama, sangat berbeda dengan karakter pada tulisan tangan yang umumnya variatif dari satu orang dengan orang yang lain. Beberapa penelitian telah dilakukan sebelumnya untuk mengenali pola karakter tulisan tangan baik berupa huruf maupun angka, seperti rekapitulasi hasil pemilihan umum pada form C1[2], pengurutan surat berdasarkan alamat pos [3], dan pencarian kata kunci dari beberapa dokumen berdasarkan kata *spotting* [4]. OCR ini sangat membantu untuk mengamankan berbagai macam literatur yang tidak didigitalkan [5].

Optical Character Recognition telah menjadi metode yang sangat menarik dikalangan peneliti untuk menyediakan input dokumen secara otomatis yang dilakukan secara cepat, baik untuk penyimpanan maupun pemrosesan lainnya[6]. Banyaknya kelas pada klasifikasi tulisan tangan karakter angka dan huruf dalam OCR juga menjadi masalah, maka diperlukan suatu metode klasifikasi yang digunakan dalam banyak kelas. Penelitian Zhang dan Zhou [7] menyinggung bahwa KNN sangat kompetitif dalam klasifikasi banyak kelas. Namun, [8] menyinggung bahwa KNN lebih banyak menggunakan memori pada proses pelatihan dan kecepatan klasifikasi rendah.

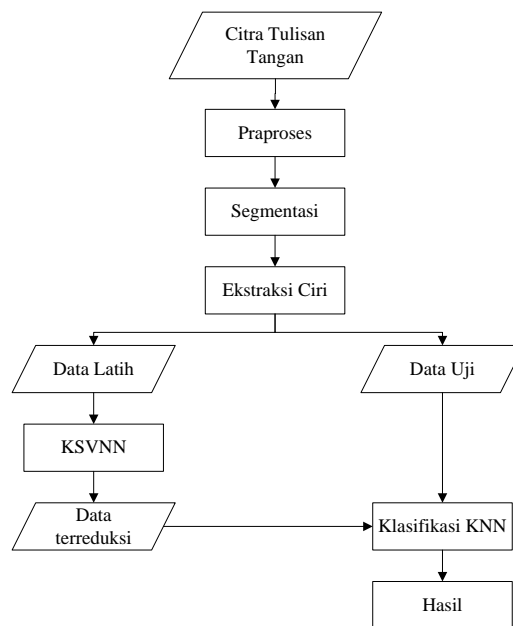
Purwandari [9] melakukan penelitian mengenai identifikasi tanda tangan dengan pendekatan *Support Vector Machine* (SVM), untuk mendapatkan ciri dari setiap tanda tangan digunakan metode *Image Zone and Centroid* (ICZ) dan *Zone Centroid and Zone* (ZCZ). Pada penelitian tersebut, metode yang diusulkan belum dapat mengenali citra tanda tangan secara maksimal pada sudut-sudut kemiringan tertentu. Selanjutnya, Kadhmi [10] melakukan penelitian untuk mengenali kata dengan tulisan tangan menggunakan algoritma SVM, data sampel yang digunakan yaitu citra tulisan tangan *arabic* sebanyak 2913 *record*. Akurasi yang diperoleh mencapai 96.317%, namun hasil segmentasi terdapat bagian yang saling tumpang tindih.

Penelitian yang dilakukan oleh Ilmi [2] memanfaatkan Local Binary Pattern Variance dan *K-Nearest Neighbour* (KNN) menghasilkan akurasi pengenalan terbaik sebesar 89.65% pada skenario pertama dengan menggunakan data sampel MNIST dimana skenario yang digunakan sebanyak 3 skenario serta data sampel MNIST dan Form C1 yang digunakan. Misnadin [11] melakukan penelitian pengenalan tulisan tangan dengan metode *K-Nearest Neighbor*, nilai  $k$  yang digunakan yakni  $k = \{1, 3, 5\}$ . Penggunaan 3 data sampel yang berbeda jumlahnya menghasilkan akurasi sebesar 56.1% untuk data sampel 1, 32.53% untuk data sampel 2, dan 32.52% untuk data sampel 3. Selanjutnya, Prasetyo [12] melakukan penelitian menghasilkan akurasi yang cukup baik yakni memperoleh selisih akurasi terkecil 0.66% dengan metode KNN atau SVM untuk data sampel iris dan selisih terbesar 20.29% dengan metode KNN untuk data sampel wine.

Oleh karena itu, diperlukan modifikasi pada metode KNN untuk mengatasi masalah memori dan waktu. Pada penelitian ini digunakan suatu metode untuk mereduksi data latih, sehingga diharapkan penggunaan memori dapat berkurang dan proses klasifikasi pengujian cepat tanpa mengabaikan hasil akurasi. Salah satu metode yang efektif untuk mereduksi data latih yakni menggunakan *K-Support Vector Nearest Neighbor* (KSVMN) [12]. Selain itu, penelitian ini juga akan memanfaatkan metode Zona (ICZ dan ZCZ) untuk ekstraksi ciri dan *K-Nearest Neighbor* untuk klasifikasi tulisan tangan.

## 2. METODE PENELITIAN

Penelitian ini dilakukan dengan tahapan-tahapan yang dapat dilihat pada Gambar 1. Citra hasil *scan* tulisan tangan yang berisi kumpulan huruf yang selanjutnya di-preproses untuk meningkatkan kualitas citra. Hasil preproses akan dilakukan segmentasi untuk memperoleh citra huruf yang terpisah yakni satu huruf pada satu citra serta pelabelan pada citra tersebut. Setelah dilakukan segmentasi, dilakukan proses ekstraksi ciri untuk mendapatkan ciri-ciri dari setiap citra. Data hasil ekstraksi ciri dibagi kedalam data uji dan data latih, yang untuk selanjutnya, data latih akan direduksi dengan metode *K-Support Vector Nearest Neighbour* (KSVNN). Data yang telah direduksi dengan KSVNN digunakan untuk pengklasifikasian data uji dengan algoritma KNN.



Gambar 1 Metode Penelitian

### 2.1 Data

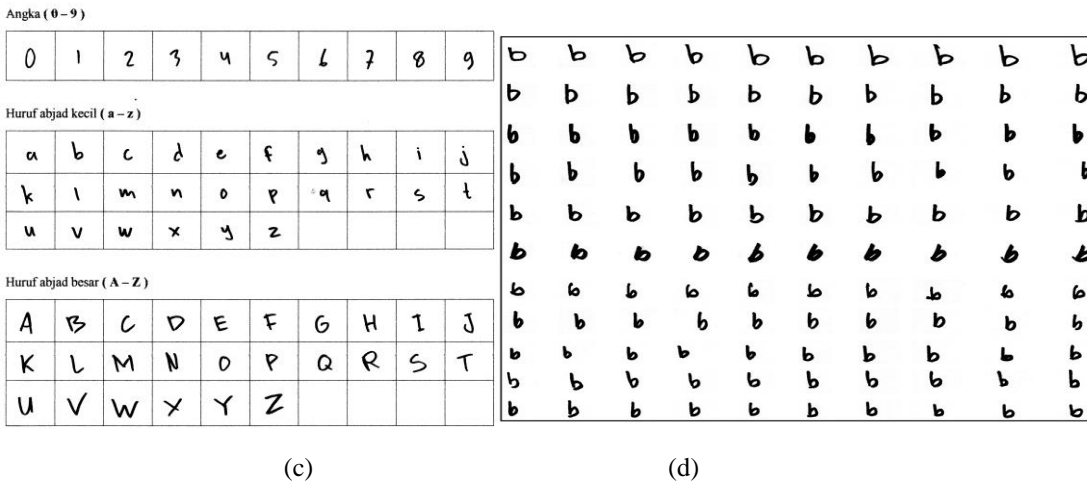
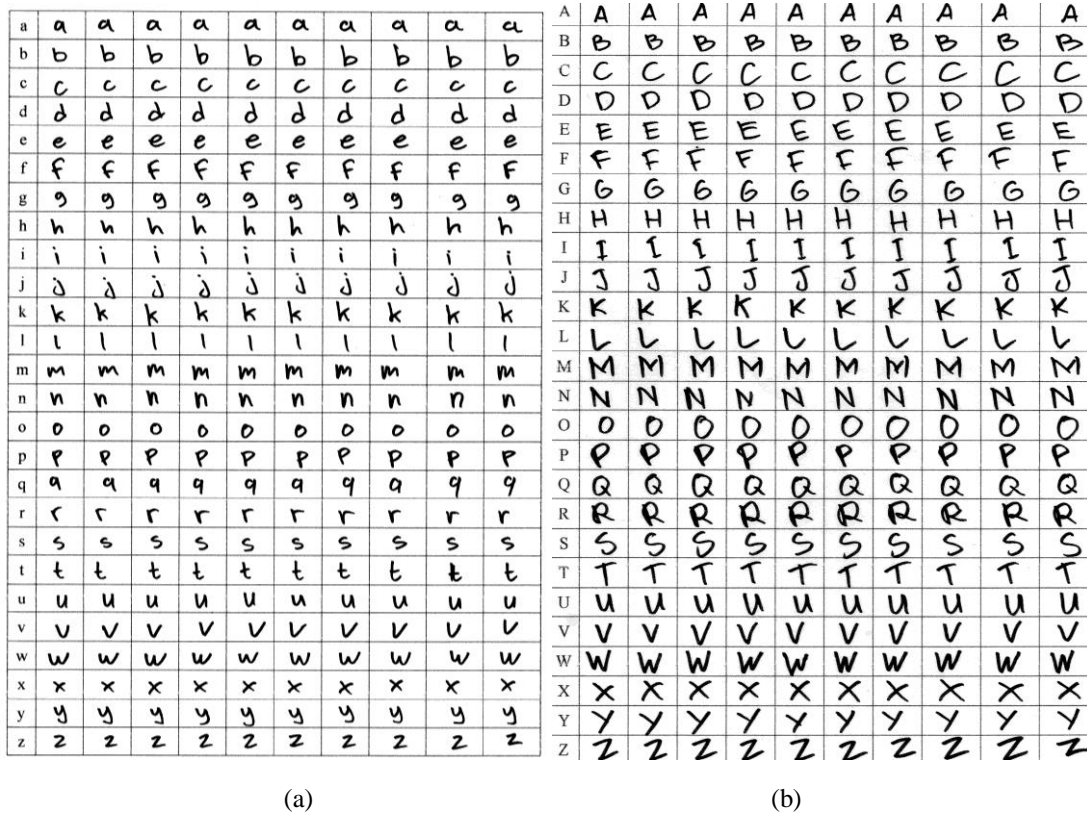
Tahap ini dilakukan dengan memberikan dua form secara acak kepada 31 responden, dimana 20 responden menulis satu kali karakter angka dan huruf pada form 1 dan 11 responden berikutnya menulis sepuluh kali karakter angka dan huruf, sehingga untuk setiap karakter diperoleh sebanyak 130 karakter. Total seluruh data sampel sebanyak 8060 yang terdiri dari 1300 data sampel angka, 3380 data sampel huruf kecil, dan 3380 data sampel huruf besar. Label atau kelas pada penelitian ini dapat direpresentasikan pada Tabel 1.

Tabel 1 Representasi Kelas

Karakter	Kelas
0-9	1-10
a-z	11-36
A-Z	37-62

Tahap *scanning* dilakukan pada data sampel yang telah diperoleh dengan resolusi

600dpi. Setelah data sampel menjadi citra dipotong secara manual untuk mengelompokkan citra karakter berdasarkan jenis atau kelas. Hasil pengelompokkan tersebut menghasilkan 130 karakter tiap kelas. Gambar 2 merupakan hasil citra tulisan tangan.



Gambar 2 Data sampel penelitian (a) Citra form 1 huruf kecil, (b) Citra form 1 huruf besar, (c) Citra form 2, (d) Hasil pemotongan citra

2.2 Praproses

Tahap praproses dilakukan untuk mengubah citra menjadi citra grayscale dan biner. Selain itu, untuk memperbaiki citra yang masih rusak. Berikut tahap praproses yang dilakukan,

### 2.2.1 Konversi RGB ke grayscale

Data citra yang diperoleh pada tahap pengumpulan data merupakan citra RGB, untuk memudahkan pada tahap berikutnya citra perlu dikonversi menjadi *grayscale* [13] dengan persamaan (1).

$$Gray = 0.2989 \times R + 0.5870 \times G + 0.1140 \times B \quad (1)$$

### 2.2.2 Konversi grayscale ke biner

Hasil *grayscale* dapat dikonversi menjadi citra biner. Citra tulisan tangan lebih optimal jika citra dikonversi ke dalam bentuk biner. Pada tahap ini, digunakan metode *Local Adaptive Thresholding* [14] dengan persamaan (2),

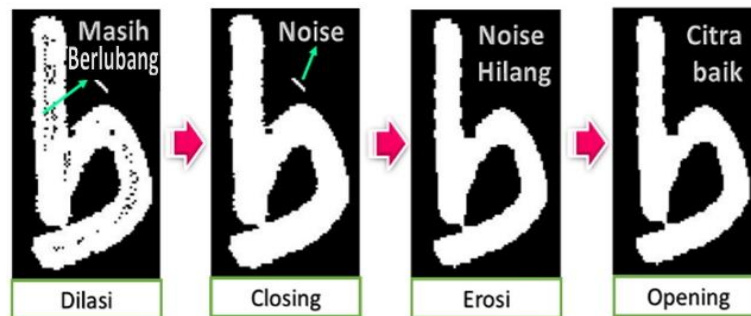
$$T = \frac{\sum_{(x,y)} \sum_{\epsilon \in W} f(x,y)}{N_w} - C \quad (2)$$

dengan syarat fungsi yang ditunjukkan pada persamaan (3).

$$f(x,y) = \begin{cases} 0 & , \quad f(x,y) < T \\ 1 & , \quad f(x,y) \geq T \end{cases} \quad (3)$$

### 2.2.3 Operasi Morfologi

Untuk memperbaiki hasil dari akuisisi citra (*scanning*) yang memiliki noise, maka perlu dilakukan proses perbaikan citra dengan menggunakan operasi morfologi. Penggunaan operasi morfologi meliputi proses dilasi, erosi, *opening* dan *closing* yang diilustrasikan pada Gambar 3.



Gambar 3 Operasi Morfologi

Dapat dilihat pada Gambar 2, citra hasil akuisisi, terdapat beberapa dari citra yang berlubang. Untuk memperbaiki citra tersebut maka perlu ditutup dengan menggunakan operasi dilasi [15]. Dilasi dari himpunan A dan himpunan B dapat ditunjukkan dengan  $A \oplus B$  pada persamaan (4).

$$A \oplus B = \{z \mid (\hat{B})_z \cap A \neq \emptyset\} \quad (4)$$

Hasil dilasi, masih meninggalkan lubang pada objek. Lubang tersebut ditutup kembali dengan proses dilasi dan dilanjutkan dengan penipisan *noise* yang ada pada citra dengan menggunakan proses erosi. Proses dilasi yang dilanjutkan dengan proses erosi disebut dengan proses *closing*. Erosi dari himpunan A dan himpunan B dapat ditunjukkan dengan  $A \ominus B$  pada persamaan (5).

$$A \ominus B = \{z \mid (B)_z \subseteq A\} \quad (5)$$

Efek yang diberikan dari proses *closing* adalah mengisi lubang-lubang kecil (*noise*)

pada objek. Proses *closing* ditunjukkan dengan  $A \cdot B$  pada persamaan (6).

$$A \cdot B = (A \oplus B) \ominus B \quad (6)$$

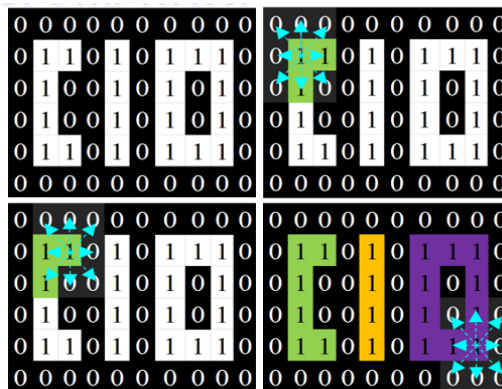
Selanjutnya, untuk menghaluskan batasan pada objek tanpa mengubah area objek secara signifikan pada tahap perbaikan citra sebelumnya maka dilakukan proses *opening* [16]. Proses *opening* merupakan proses erosi yang dilanjutkan dengan proses dilasi yang ditunjukkan dengan  $A \circ B$  pada persamaan (7).

$$A \circ B = (A \ominus B) \oplus B \quad (7)$$

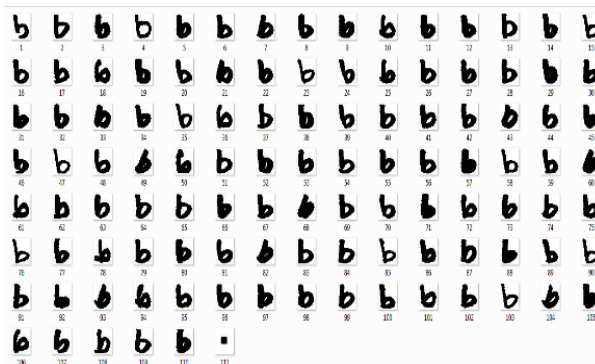
### 2.3 Segmentasi

Segmentasi merupakan proses mengelompokkaa citra kedalam bagian yang berbeda untuk mempermudah melakukan proses identifikasi citra [17]. Pengertian lain dari segmentasi merupakan teknik pembagian citra menjadi beberapa daerah (*region*) dimana setiap daerah memiliki fitur yang sama [18]. Metode yang digunakan untuk segmentasi karakter yaitu *8-connected component*.

Pemberian label pada tahap segmentasi juga dilakukan untuk mempermudah pemotong citra menjadi bagian yang berbeda secara otomatis. Dengan melakukan identifikasi 8 komponen yang saling terhubung. Gambar 4 merupakan ilustrasi dari metode *8-connected component*, sedangkan hasil segmentasi dapat dilihat pada Gambar 5.



Gambar 4 Ilustrasi *8-connected component*



Gambar 5 Hasil segmentasi

#### 2.3.1 Cropping

Setelah pemberian label dengan metode *8-connected component* maka citra tulisan

tangan dipotong berdasarkan label yang diberikan.

### 2.3.2 Penskalaan

Hasil pemotong citra menghasilkan ukuran yang berbeda-beda untuk memudahkan pada tahap ekstraksi ciri *zoning*, dimana diperlukan citra dengan ukuran yang sama maka hasil citra potong perlu diubah ukurannya. Penskalaan merupakan sebuah operasi untuk memperbesar atau memperkecil objek citra [18]. Penskalaan tersebut dapat dilakukan dengan persamaan (8) dan (9).

$$P_o = S_p \cdot P_i \quad (8)$$

$$L_o = S_l \cdot L_i \quad (9)$$

Keterangan :

$P$  dan  $L$  = Ukuran citra

$S$  = Skala yang diinginkan

### 2.4 Ekstraksi Ciri

Ekstraksi ciri yang digunakan pada penelitian ini yaitu metode *zoning*. Metode ini membagi citra menjadi beberapa zona, dimana setiap zona memiliki ukuran yang sama. Pada penelitian ini, zona dibagi menjadi 5x4 zona dan menghasilkan 20 zona untuk setiap karakter sehingga hasil ciri yang diperoleh dengan menggunakan metode ICZ dan ZCZ menghasilkan 40 ciri setiap karakter.

Ada beberapa algoritma dari ekstraksi ciri *zoning* untuk klasifikasi dan pengenalan pola, yaitu ekstraksi ciri *Image Centroid and Zone (ICZ)*, *Zone Centroid and Zone (ZCZ)*, dan gabungan dari algoritma keduanya ICZ+ZCZ [19]. Dalam penelitian ini, ekstraksi ciri menggunakan algoritma ICZ+ZCZ, langkah-langkah yang digunakan sebagai berikut:

- 1) Menghitung centroid dari citra masukan.
- 2) Membagi citra masukan menjadi  $n$  zona yang sama.
- 3) Menghitung jarak antara centroid citra dengan setiap pixel yang ada dalam zona.
- 4) Mengulangi ke langkah 3 untuk semua pixel yang ada dalam zona.
- 5) Menghitung jarak rata-rata antara titik-titik tersebut.
- 6) Menghitung centroid tiap zona.
- 7) Menghitung jarak antara centroid zona dengan setiap pixel yang ada dalam zona.
- 8) Mengulangi langkah 7 untuk semua pixel yang ada dalam zona.
- 9) Menghitung jarak rata-rata antara titik-titik tersebut.
- 10) Mengulangi langkah 3-9 untuk semua zona secara berurutan.
- 11) Akhirnya, akan didapatkan  $2n$  ciri untuk klasifikasi dan pengenalan.

### 2.5 Normalisasi Data

Data yang dihasilkan dari ekstraksi ciri merupakan data yang tidak teratur dalam rentang tertentu. Oleh karena itu, data perlu dinormalisasi dengan menskalakan data agar berada pada rentang yang telah ditentukan. Ada banyak teknik atau metode untuk melakukan normalisasi data, yaitu normalisasi Min-Max, normalisasi Z-Score, dan normalisasi penskalaan desimal [20]. Dalam penelitian menggunakan normalisasi Min-Max. Normalisasi Min-Max merupakan salah satu metode normalisasi yang baik untuk digunakan dalam normalisasi data set pada metode apapun [21]. Range yang digunakan yaitu 0 sampai dengan 1 dengan persamaan (10).

$$X_{ij} = \frac{(X_{ij} - \min_i)}{(\max_i - \min_i)} \quad (10)$$

## 2.6 N-Fold Cross Validation

Setelah data dinormalisasi selanjutnya data sampel dibagi menjadi data latih dan data uji dengan metode *N-fold cross validation*, dimana nilai  $N=10$ . Penelitian [22] menyinggung bahwa menggunakan nilai  $N=10$  (*10-fold cross validation*) lebih baik dari pada *leave one out cross validation* untuk memilih suatu model, dan nilai tersebut lebih baik dibandingkan pilihan nilai  $N$  lainnya.

## 2.7 Reduksi Data dengan K-Support Vector Nearest Neighbor (K-SVNN)

Data sampel terbagi menjadi data latih dan data uji. Selanjutnya, data latih direduksi dengan K-SVNN dimana nilai  $K=3$ . Nilai  $k$  dapat menentukan kinerja dari metode K-SVNN, semakin kecil nilai  $k$  maka *support vector* diperoleh semakin sedikit dan akurasi yang diperoleh semakin kecil, tetapi besarnya nilai  $k$  tidak dapat menjamin akurasi semakin besar [12]. Berikut merupakan algoritma dari K-SVNN:

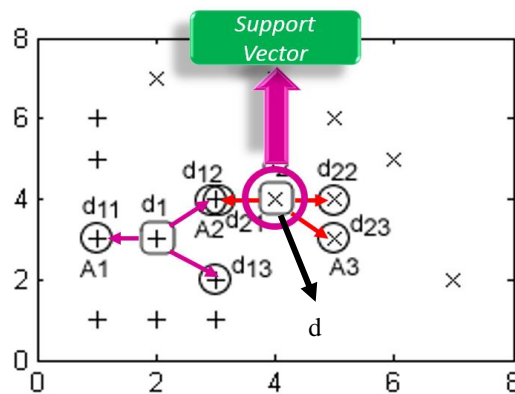
1. Melakukan inisialisasi  $D$  merupakan set data latih,  $K$  sebagai jumlah tetangga terdekat,  $T$  merupakan threshold *Significant Degree* ( $SD$ ), *Left Value* ( $LV$ ) dan *Right Value* ( $RV$ ) untuk semua data latih = 0.
2. Untuk setiap data latih  $d_i \in D$ , lakukan langkah 3 hingga 5.
3. Menghitung jarak dari  $d_i$  ke data latih lain. untuk mendapatkan jarak digunakan persamaan (11) sebagai berikut,

$$D_{p,q} = \sqrt{\sum_{i=1}^n (p_i - q_i)^2} \quad (11)$$

4. Pilih  $d_t$  sebagai  $K$  tetangga terdekat.
5. Untuk setiap data dalam  $d_t$ , jika memiliki kelas yang sama, berikan nilai 1 pada  $LV_i$ , jika tidak berikan nilai 1 pada  $RV_i$ .
6. Untuk setiap data  $d_i$ , hitung  $SD_i$  menggunakan persamaan (12) dibawah,

$$SD_i = \begin{cases} 0, & LV_i = RV_i = 0 \\ \frac{LV_i}{RV_i}, & LV_i < RV_i \\ \frac{RV_i}{LV_i}, & LV_i > RV_i \\ 1, & LV_i = RV_i \end{cases} \quad (12)$$

7. Pilih data dengan  $SD_i \geq T$ , simpan dalam variabel.



Gambar 6 Contoh *Support Vector* Data Latih



Gambar 6 menunjukkan proses mendapatkan *support vector* pada data latih. Support vector inilah yang dijadikan sebagai sampel pada data latih sehingga data latih menjadi tereduksi. Jarak minimum yang diperoleh pada data  $d_1$ , yaitu pada data  $d_{11}$ ,  $d_{12}$ , dan  $d_{13}$ . Tiga tetangga terdekat tersebut memiliki kelas yang sama dengan  $d_1$  maka nilai  $LV=3$  dan  $RV=0$ . Sedangkan, data  $d_2$  memiliki jarak minimum terhadap 3 tetangga terdekat, yaitu  $d_{21}$ , yang memiliki kelas yang berbeda dengan  $d_2$  sehingga  $RV=1$ , serta  $d_{22}$ , dan  $d_{23}$  mempunyai kelas yang sama dengan  $d_2$  sehingga nilai  $LV=2$ . Sesuai pada persamaan 12, nilai derajat signifikansi pada data  $d_1$  yaitu  $SD=0$  sedangkan data  $d_2$   $SD=0,5$ . Maka dari itu, data  $d_2$  dapat ditetapkan sebagai *support vector*.

### 2.8 K-Nearest Neighbor (K-NN)

Tahap klasifikasi dilakukan dengan K-NN, nilai  $k=1, 3$ , dan  $5$ . Perhitungan klasifikasi dengan KNN didasarkan pada perhitungan jarak antara data sampel dengan data lainnya [2]. Jarak diperoleh menggunakan rumus *euclidean distance* pada persamaan (11).

### 2.9 Evaluasi

Tahap dilakukan untuk mengetahui akurasi dari metode yang diterapkan pada penelitian. Dari akurasi tersebut dapat dianalisa bahwa metode yang dalam penelitian baik untuk digunakan. Akurasi dapat diperoleh dengan persamaan (13) sebagai berikut,

$$\text{Akurasi} = \frac{\text{Jumlah Benar}}{\text{Banyak Data}} \times 100 \quad (13)$$

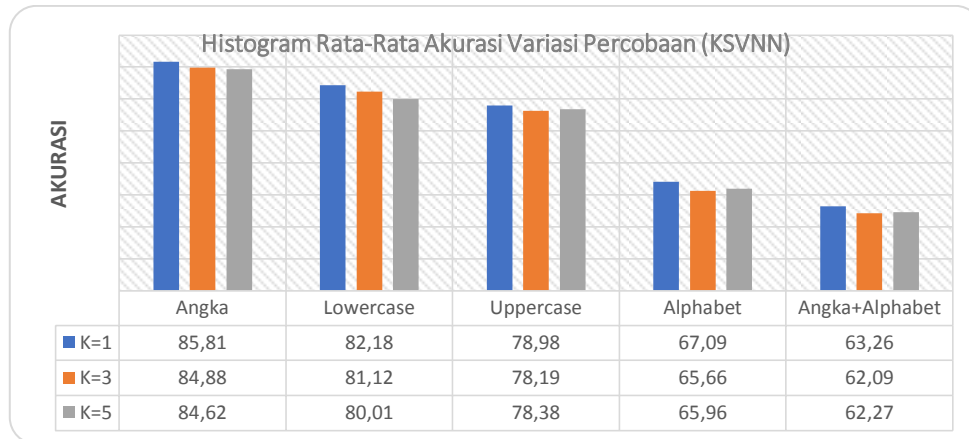
## 3. HASIL DAN PEMBAHASAN

Pada tahap klasifikasi dibagi menjadi 5 variasi pengujian yaitu pengenalan angka, huruf kecil, huruf besar, (huruf kecil + huruf besar), dan (angka + huruf kecil + huruf besar). Data sampel yang digunakan sebesar 1302 record untuk pengenalan angka, 3442 record untuk huruf kecil, 3419 record untuk huruf besar, 6861 record untuk (huruf kecil + huruf besar), dan 8163 record untuk (angka + huruf kecil + huruf besar). Data sampel setiap variasi pengujian dibagi menjadi data latih dan data uji dengan *10-fold cross validation*.

Pada setiap variasi pengujian dilakukan 5 kali percobaan untuk menghindari hasil akurasi yang kebetulan dengan menggunakan metode klasifikasi K-NN, dimana  $k=1,3,5$ . Hasil dari 5 kali percobaan yang dilakukan, selanjutnya diambil rata-rata akurasi pada setiap  $k$  yang digunakan dengan menjumlahkan akurasi dari setiap percobaan dan dibagi dengan banyaknya percobaan.

#### a. Reduksi Data Latih dengan K-SVNN

Sebelum dilakukan proses pelatihan pada klasifikasi, dilakukan reduksi data latih dengan K-SVNN sebagaimana dibahas pada sub sebelumnya, dimana nilai  $k=3$  untuk mendapatkan *support vector* yang akan digunakan sebagai data latih. Hasil dari pelatihan ialah model yang digunakan untuk pengujian. Gambar 7 merupakan hasil pengujian dengan reduksi data latih.

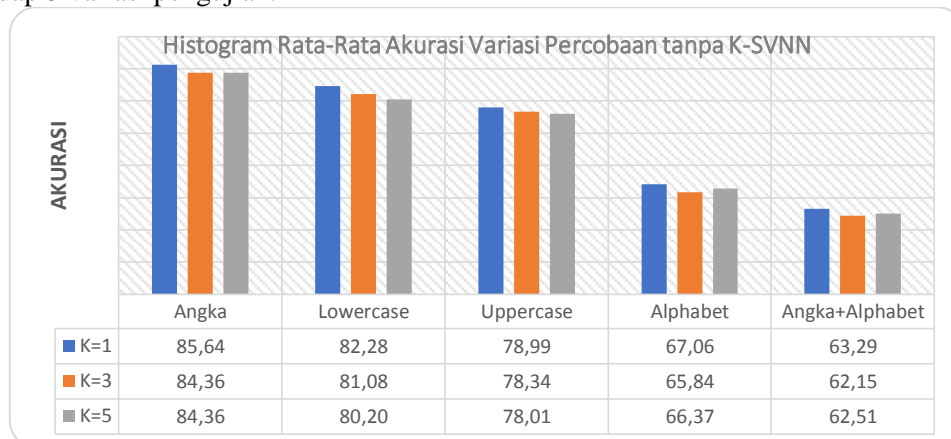


Gambar 7 Grafik Rata-Rata Akurasi Setiap Variasi Pengujian

Pengujian yang dilakukan pada 5 variasi pengujian menghasilkan akurasi yang optimal yaitu pada pengenalan angka dengan nilai akurasi sebesar 85.81% untuk  $k=1$ , 84.88% untuk  $k=3$ , dan 84.62% untuk  $k=5$ . Selain itu, untuk setiap variasi pengujian yang dilakukan menghasilkan akurasi yang optimal pada saat nilai  $k$  yang digunakan sama dengan 1. Untuk pengujian pengenalan (angka + huruf kecil + huruf besar) menghasilkan akurasi paling kecil karena terdapat beberapa kemiripan dari citra angka dan huruf.

b. Tanpa Reduksi Data Latih

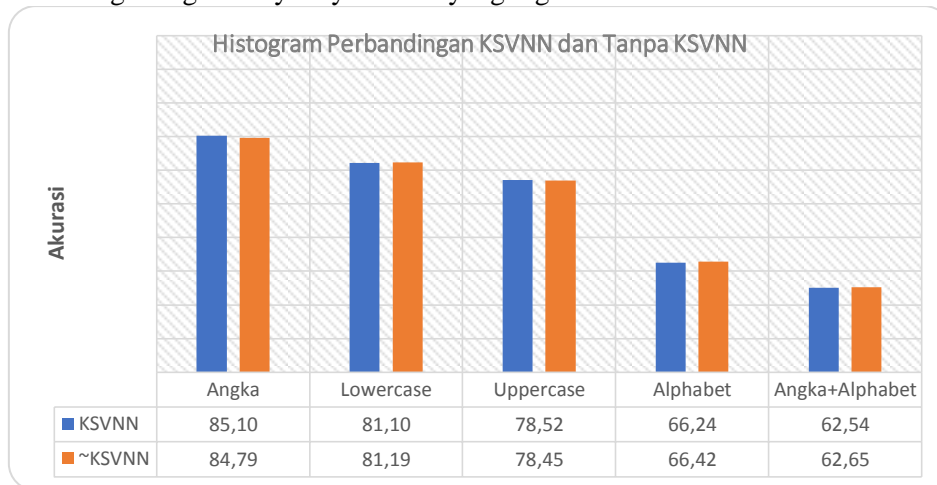
Pengujian juga dilakukan tanpa melakukan reduksi data latih. Hal ini dilakukan untuk mengetahui akurasi yang dihasilkan tanpa reduksi. Gambar 8 merupakan hasil pengujian terhadap 5 variasi pengujian.



Gambar 8 Grafik Rata-Rata Akurasi Setiap Variasi Pengujian Tanpa K-SVNN

Dari hasil pengujian yang dilakukan dalam 5 variasi pengujian tanpa melakukan reduksi terhadap data latih, rata-rata akurasi yang diperoleh dari 5 percobaan pada setiap pengujian yaitu akurasi pengujian angka mencapai 85.64%, 84.36%, dan 84.36% pada nilai  $k = 1, 3, 5$ . Akurasi pada pengujian *lowercase* mencapai 82.28%, 81.08%, dan 80.20%. Pada nilai  $k$  yang sama untuk pengujian *uppercase* menghasilkan akurasi sebesar 78.99%, 78.34%, dan 78.01%. Selanjutnya, pengujian *alphabet* menghasilkan akurasi sebesar 67.06%, 65.84%, dan 66.37%. Terakhir, pengujian angka dan *alphabet* dilakukan pada nilai  $k$  yang sama menghasilkan akurasi sebesar 63.29%, 62.15%, dan 62.51%.

Kemudian dari penjelasan hasil pengujian menggunakan K-SVNN dan tanpa K-SVNN dihitung kembali rata-rata akurasi dengan cara menjumlahkan akurasi variasi pengujian pada setiap k dan dibagi dengan banyaknya nilai k yang digunakan.



Gambar 9 Perbandingan KSVNN dan Tanpa KSVNN

Pada Gambar 9 dapat diketahui bahwa hasil pengenalan pola tulisan tangan dengan menggunakan metode *K-Support Vector Nearest Neighbor* sebagai metode reduksi data latih dapat meningkatkan akurasi pada pengenalan tulisan tangan angka dan huruf besar (*uppercase*). Sementara itu, pada pengujian huruf kecil (*lowercase*) memiliki selisih sebesar 0.09, pengujian alfabet (*lowercase, uppercase*) memiliki selisih sebesar 0.18, dan pada pengujian karakter angka dan huruf alfabet memiliki selisih sebanyak 0.11.

#### 4. KESIMPULAN

Metode K-SVNN dapat meningkatkan akurasi dibandingkan dengan tanpa menggunakan K-SVNN pada pengenalan karakter angka dimana akurasi mencapai 85.81% untuk  $k=1$ , 84.88% untuk  $k=3$ , dan 84.62% untuk  $k=5$ . Kemudian pada pengenalan huruf besar mencapai 78.98% untuk  $k=1$ , 78.19% untuk  $k=3$ , dan 78.33% untuk  $k=5$ . Pengenalan pola tulisan tangan yang terbagi menjadi 5 variasi pengujian, yaitu pengenalan angka, huruf kecil, huruf besar, alfabet, dan angka+alfabet dapat menghasilkan akurasi tertinggi pada pengujian pengenalan angka yaitu 85,10%. Akurasi tertinggi juga diperoleh pada pengenalan pola tulisan tangan dengan menggunakan nilai k sama dengan 1 pada metode klasifikasi yang digunakan yaitu K-NN dimana k yang digunakan dalam penelitian  $k=1,3,5$ . Akurasi yang diperoleh selalu mengalami penurunan jika nilai k lebih besar dari 1.

#### 5. SARAN

Adapun saran yang diberikan, yaitu menambah metode peningkatan kualitas citra pada tahap praproses, sehingga dapat meningkatkan kualitas citra. Selain itu juga, disarankan juga untuk menggunakan metode klasifikasi yang dapat meminimalkan waktu proses klasifikasi.

#### DAFTAR PUSTAKA

- [1] N. Islam, Z. Islam, and N. Noor, "A Survey on Optical Character Recognition System," *J. Inf. Commun. Technol.*, vol. 10, no. 2, pp. 1–4, 2016.

- [2] N. Ilmi, W. T. A. Budi, and R. K. Nur, "Handwriting digit recognition using local binary pattern variance and K-Nearest Neighbor classification," in *2016 4th International Conference on Information and Communication Technology (ICoICT)*, 2016, pp. 1–5.
- [3] F. Kimura and M. Shridhar, "Handwritten numerical recognition based on multiple algorithms," *Pattern Recognit.*, vol. 24, no. 10, pp. 969–983, Jan. 1991.
- [4] L. M. Lorigo and V. Govindaraju, "Offline Arabic handwriting recognition: a survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 5, pp. 712–724, May 2006.
- [5] N. Mehta and J. Doshi, "A Review of Handwritten Character Recognition," *Int. J. Comput. Appl.*, vol. 165, no. 4, pp. 37–40, May 2017.
- [6] S. IMPEDOVO, L. OTTAVIANO, and S. OCCHINEGRO, "OPTICAL CHARACTER RECOGNITION — A SURVEY," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 05, no. 01n02, pp. 1–24, Jun. 1991.
- [7] Min-Ling Zhang and Zhi-Hua Zhou, "A k-nearest neighbor based algorithm for multi-label classification," in *2005 IEEE International Conference on Granular Computing*, 2005, p. 718–721 Vol. 2.
- [8] M. Kibanov, M. Becker, J. Mueller, M. Atzmueller, A. Hotho, and G. Stumme, "Adaptive kNN using expected accuracy for classification of geo-spatial data," in *Proceedings of the 33rd Annual ACM Symposium on Applied Computing - SAC '18*, 2018, pp. 857–865.
- [9] E. Purwandari, D. Puspitaningrum, and A. Mirfen, "Identifikasi Tanda Tangan Dengan Pendekatan Support Vector Machine," *Jurnal Sains, Teknologi dan Industri*, vol. 12, no 2, 2015.
- [10] M. S. Kadhm and A. P. D. A. K. Abdul, "Handwriting Word Recognition Based on SVM Classifier," *Int. J. Adv. Comput. Sci. Appl.*, vol. 6, no. 11, 2015.
- [11] R. A. Misnadin, S. A. S. Mola, and A. Fanggaldae, "Pengenalan Pola Tulisan Tangan Dengan Metode K – Nearest Neighbor," *J. Komput. dan Inform.*, vol. 2, no. 1, pp. 65–72, Nov. 2016.
- [12] E. Prasetyo, *K-Support Vector Nearest Neighbor Untuk Klasifikasi Berbasis K-NN*. 2012.
- [13] S. G. Wu, F. S. Bao, E. Y. Xu, Y.-X. Wang, Y.-F. Chang, and Q.-L. Xiang, "A Leaf Recognition Algorithm for Plant Classification Using Probabilistic Neural Network," Jul. 2007.
- [14] N. P. Sutramiani, Ik. G. Darmaputra, and M. Sudarma, "Local Adaptive Thresholding Pada Preprocessing Citra Lontar Aksara Bali," *Maj. Ilm. Teknol. Elektro*, vol. 14, no. 1, Jun. 2015.
- [15] R. C. Gonzalez and R. E. (Richard E. Woods, *Digital image processing*. Prentice Hall, 2008.
- [16] B. Jähne, *Digital Image Processing*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002.
- [17] Y. Deng, B. S. Manjunath, and H. Shin, "Color Image Segmentation N," *Comput. Eng.*, vol. 18, no. 10, pp. 2330–9, 2009.
- [18] D. Putra, *Pengolahan Citra Digital*. Penerbit Andi.
- [19] S. V Rajashekararadhya, "Efficient Zone Based Feature Extration Algorithm for Handwritten Numeral Recognition of Four Popular South Indian Scripts," *J. Theor. Appl. Inf. Technol.*, vol. 4, no. 12, pp. 1171–1181, 2008.
- [20] Y. Kumar Jain and S. Kumar Bhandare, "Min Max Normalization Based Data Perturbation Method for Privacy Protection," *Int. J. Comput. Commun. Technol.*, 2011.
- [21] L. Al Shalabi, Z. Shaaban, and B. Kasasbeh, "Data Mining: A Preprocessing Engine," *J. Comput. Sci.*, vol. 2, no. 9, pp. 735–739, Sep. 2006.
- [22] R. Kohavi, "A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection," *Appear. Int. Jt. Conf. Artificial Intell.*, vol. 5, pp. 1–7, 1995.