Pengenalan Tulisan Tangan Dengan Menggunakan Metode Diagonal Feature Extraction dan K-Nearest Neighbour

Yustar Pramudana

Program Studi Teknik Informatika, Departemen Informatika
Telkom University
Jl. Telekomunikasi No. 1 Bojongsoang, Kabupaten Bandung
E-mail: yustar.shooter@gmail.com

Abstrak

Pada sebuah ban ketika pelanggan mengisi form dan diterima oleh *teller, teller* akan menginput kembali kedalam komputer. Proses akan lebih cepat jika ada sistem automasi. Dibangun sistem yang dapat mensimulasikan situasi diatas. Dibuat sistem untuk mengenali karakter tulisan tangan dengan menggunakan *Diagonal Based Feature Extraction* dan metode klasifikasi *K-Nearest Neighbour.* Dari hasil pengujian diperoleh akurasi terbaik 90% dengan metode *Diagonal Feature Extraction* dari kanan dan *K-NN* dengan k=2.

Kata kunci: *Diagonal Feature Extraction,* Pengenalan Tulisan Tangan, *Handwritten Recognition, k-Nearest Neighbour, k-NN*

1. PENDAHULUAN

1.1 Latar Belakang

Pada sebuah bank, nasabahnya menulis pada sebuah kertas untuk melakukan transaksi. Kemudian *teller* menginput kembali data ke dalam komputer. Seandainya ada alat yang dapat langsung membaca isi kertas dan langsung masuk kedalam komputer, proses transaksi yang terjadi akan lebih cepat. Untuk mensimulasikan situasi yang mirip dengan kondisi diatas, penulis membuat sistem yang dapat mendeteksi tulisan tangan perkarakter. Inputan dari sistem ini nantinya berupa karakter a-z dan A-Z. Metode ekstraksi ciri yang digunakan adalah *Diagonal Feature Extraction* dan metode klasifikasi dengan k-NN untuk mengetahui keakuratan dari sistem yang dibangun.

Mengacu dari paper yang berjudul *Diagonal Based Feature Extraction for Handwritten Alphabets Recognition System Using Neural Network*. Pada paper tersebut juga masih menghasilkan akurasi yang tinggi pada karakter yang telah dipisah. Dari permasalahan diatas penulis ingin penelitian tentang pengenalan tulisan tangan menggunakan ekstraksi ciri *Diagonal Based Feature Extraction*.

Penulis ingin penelitian tentang pengenalan tulisan tangan menggunakan metode klasifikasi dengan algoritma *k-Nearest Neighbour* dikarenakan tulisan tangan mempunyai kelas variansi yang tinggi. Tulian alphabet mempunyai karakter A-Z yang berarti 26 tetapi dari tulisan tangan bisa lebih banyak karena karakter penulisan orang berbeda-beda.

1.2 Tujuan

1. Membangun suatu sistem yang dapat mengenali tulisan tangan menggunakan ekstraksi ciri *Diagonal Feature* dan k-NN.

2. Melakukan *preprocessing* untuk tulisan tangan.
3. Implementasi ekstraksi ciri diagonal.
4. Implementasi metode k-NN untuk Klasifikasi.
5. Mengetahui akurasi *Diagonal Feature Extraction* dan k-NN dalam mengenali tulisan tangan.

2. LANDASAN TEORI

2.1 *Handwriting Recognition*

Handwriting recognition atau pengenalan tulisan tangan adalah suatu proses dimana komputer menerjemahkan tulisan tangan kedalam teks komputer. *Input* dapat berupa teks yang telah discan berbentuk image atau input langsung dari suatu *device* yang menggunakan pena virtual.

2.2 Pengloahan Citra

Pengolahan citra adalah setiap bentuk pengolahan sinyal dimana input adalah gambar, seperti foto atau *video* bingkai, sedangkan output dari pengolahan gambar dapat berupa gambar atau sejumlah karakteristik atau parameter yang berkaitan dengan gambar. Pengolahan citra merupakan proses pengolahan dan analisis citra yang banyak melibatkan persepsi visual. Proses ini mempunyai ciri data masukan dan informasi keluaran yang berbentuk citra. Istilah pengolahan citra digital secara umum didefinisikan sebagai pemrosesan citra dua dimensi dengan komputer. Dalam definisi yang lebih luas, pengolahan citra digital juga mencakup semua data dua dimensi. Citra digital adalah barisan bilangan nyata maupun kompleks yang diwakili oleh bit-bit tertentu [4].

Operasi-operaasi yang terdapat pada pengolahan citra, antara lain:

1. *Thresholding*

Thresholding adalah proses untuk mengklasifikasi sebuah citra ke dalam bagian yang berbeda yang didasarkan pada intensitas-intensitas atau derajat keabuan yang dominan. Tujuan dari thresholding adalah menyeleksi nilai threshold yang memisahkan citra dalam dua atau lebih derajat keabuan yang berbeda, selanjutnya memberi label tertentu piksel-piksel yang masuk dalam kelompok-kelompok derajat keabuan yang diinginkan.

2. *Cropping*

Cropping adalah proses pengambilan bagian dalam suatu citra pada bagian yang diinginkan saja. Pengambilan bagian pada suatu citra dilakukan agar citra yang akan diproses lebih fokus pada bagian yang akan diolah selanjutnya. Contohnya ketika akan melakukan pemrosesan pada citra wajah, citra yang menampilkan keseluruhan badan dapat di *crop* sehingga hanya menyisakan bagian wajah saja.

3. *Resize*

Mengubah besarnya ukuran citra digital dalam *pixel.*

4. *Binarization*

Citra biner disebut juga citra hitam dan putih. Hanya dibutuhkan 1 bit untuk mewakili setiap nilai citra biner.

2.3 Feature Extraction

Feature Extraction atau dalam bahasa Indonesia disebut juga Ekstrasi Ciri. *Feature Extraction* merupakan proses terpenting dalam pengenalan objek. Fitur adalah karakteristik atau hal-hal khusus yang mencirikan satu objek. Ekstraksi fitur atau *Feature Extraction* berarti mengambil dan menyimpan karakteristik-karakteristik khusus dari suatu objek pada citra. Fitur-fitur inilah yang kemudian akan digunakan sebagai pembanding untuk mengenali objek tertentu pada suatu citra. Karakteristik fitur yang baik sebisa mungkin memenuhi persyaratan sebagai berikut [3]

a. Dapat membedakan suatu objek dengan yang lainnya

b. Kompleksitas komputasi yang tidak terlalu rumit

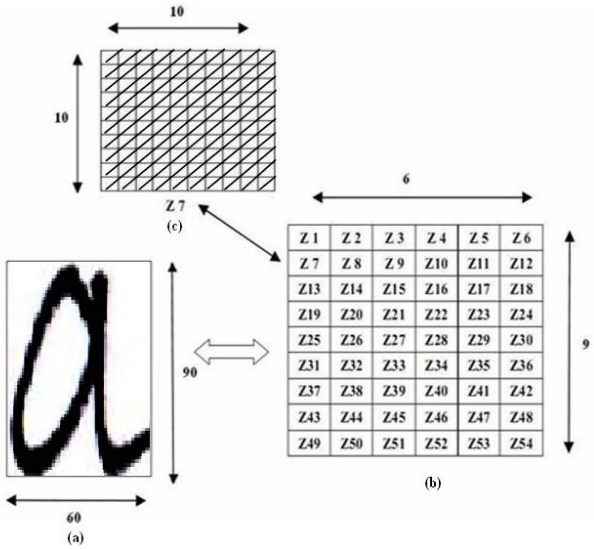
c. Tidak terikat (invarian) terhadap transformasi

d. Jumlahnya sedikit

2.4 Diagonal Based Feature Extraction

Diagonal Based Feature Extraction adalah algoritma ekstrasi ciri yang membagi ukuran piksel gambar menjadi piksel-piksel yang lebih kecil dan sama rata. Misal *character image* berukuran 100x100 *pixels* dibagi menjadi 100 zona yang sama rata. Setiap zona berukuran 10x10 *pixels* (Gambar 2.1). Ciri diextraksi dari tiap zona dengan bergerak diagonal dari masing-masing 10x10 *pixels*. Tiap zona memiliki 19 garis diagonal dan *foreground pixels* yang ada di setiap baris diagonal dijumlahkan untuk mendapatkan satu sub-ciri. 19 nilai sub-ciri ini akan dirata-ratakan untuk mendapatkan nilai ciri tunggal dan ditempatkan di zona yang sesuai (Gambar 2.1). Prosedur ini diulangi untuk semua zona [8].

Akan ada beberapa zona yang diagonalnya kosong dari *foreground pixels*. Nilai ciri untuk zona tersebut adalah nol. 100 ciri telah di ekstraksi utuk masing-masing karakter. Selain itu, 100 ciri didapatkan dari merata-ratakan nilai yang ditempatkan pada tiap zona baris dan kolom [8].



Gambar 2. 1 Diagonal Feature Extraction

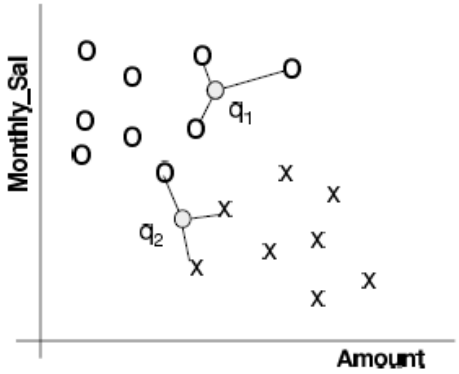
2.5 K-Nearest Neighbour

Algoritma *k-Nearest Neighbor (k-NN)* adalah metode yang digunakan untuk mengklasifikasikan suatu data. Pada data latih biasanya diambil lebih dari satu tetangga terdekat dengan data uji kemudian akan digunakan algoritma ini untuk ditentukan kelasnya. k–NN disebut juga teknik *lazy learning* yang proses klasifikasinya langsung berdasarkan pada data-data latih. Teknik *lazy learning* juga disebut dengan klasifikasi berdasarkan contoh atau klasifikasi berdasarkan kasus.

Ide utama dari algoritma *k-NN* ditunjukkan pada Gambar 2.2 yang menunjukkan *k-Nearest Neighbor* dengan nilai k=3, pada dua kelas masalah dalam ruang dua dimensi. Pada contoh ini keputusan untuk q1 sangat mudah karena semua dari ketiga tetangga terdekat adalah kelas O sehingga diklasifikasikan sebagai kelas O. Situasi dari q2 sedikit lebih rumit karena memiliki dua tetangga dari kelas X dan satu dari kelas O. Masalah pada q2 dapat diatasi dengan *voting* mayoritas sederhana.

Jadi klasifikasi *k-NN* mempunyai dua langkah, yaitu:

1. Menentukan tetangga-tetangga terdekat dari data tersebut.
2. Menentukan kelas dari masing-masing tetangga terdekat tersebut.



Gambar 2. 2 Klasifikasi k-NN untuk K=3

Pada q2 penetuan kelas sangat mudah yaitu dengan *voting* mayoritas sederhana yaitu dengan membandingkan jumlah kelas X terdekat ada dua dan jumlah kelas O ada satu buah, jadi dapat diambil kesimpulan bahwan q2 adalah kelas X. Tetapi akan menjadi masalah bila misal k yang diambil ada 2 dengan perincian kelas terdekat masing-masing kelas X dan kelas O satu buah maka menggunakan voting mayoritas sederhana tidak bisa dilakukan.

Untuk mengatasi masalah tersebut maka akan menggunakan rumus aturan jarak. Pada k-NN terdapat beberapa aturan jarak yang dapat digunakan, salah satunya adalah *Euclidean Distance.*

Euclidean Distance

$$L_2(X,Y) = \sqrt{\sum_{i=1}^{d}(X_i - Y_i)^2}$$

L disini menunjukan *Lenght* atau jarak. Jarak terdekat ketentanggaan dalam k-NN. Xi – Yi adalah jarak antara data uji dengan data model.

2.6 Perhitungan Akurasi

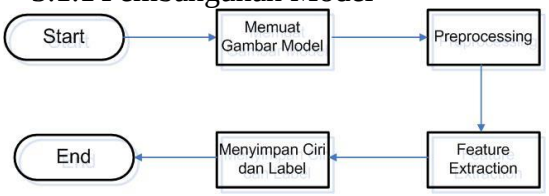
$$\boldsymbol{Akurasi = \frac{k - NN}{Jumlah\ Data\ Uji}}$$

Akurasi dari sistem dapat diketahui dengan membandingkan *output* dari k-NN dengan label kelas dari data uji. Nilai yang benar adalah ketika *output* k-NN sama dengan label kelas dari data uji. Kemudian jumlah benar dibagi dengan banyaknya data uji.

3. Desain Sistem

3.1 Gambaran Umum

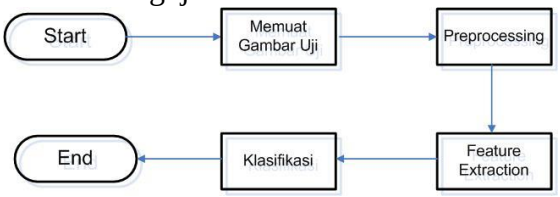
3.1.1 Pembangunan Model



Gambar 3. 1 Pembangunan Model

1. Memuat Gambar Model
   Data model yang dicrop perkarakter dari form data input di*load* semuanya kedalam sistem
2. *Preprocessing*
   Dilakukan pemprosesan *grayscale, thresholding, complement, boundingbox crop,* dan *resize.*
3. *Faeture Extraction*
   Ciri diekstraksi menggunakan *Diagonal Based Feature Extraction* untuk setiap data model.
4. Menyimpan Ciri dan Label
   Ciri yang terekstraksi disimpan dan diberikan label sesuai dengan data yang dimasukkan ke dalam sistem.

3.1.2 Pengujian Model



Gambar 3. 2 Pengujian Model

1. Memuat Gambar Uji
   Data uji yang dicrop perkarakter dari form data input di*load* semuanya kedalam sistem.
2. *Preprocessing*
   Dilakukan pemprosesan *grayscale, thresholding, complement, boundingbox crop,* dan *resize.*
3. Feature Extraction
   Ciri diekstraksi menggunakan *Diagonal Based Feature Extraction* untuk setiap data uji.
4. Klasifikasi
   Ciri dari data model dan data uji dibandingkan dengan k-NN untuk pengklasifikasian.

3.2 *Preprocessing*

Ada beberapa tahap dalam *preprocessing* yang akan mengolah data gambar :

1. *Input dataset/Akuisisi Data*
   Memasukkan data yang telah *crop* per karakter kedalam sistem
2. *Grayscale*
   Merubah citra rgb menjadi citra abu-abu yang memiliki range 0-1. 0 hitam dan 1 putih dengan mengeliminasi hue dan saturation.
3. *Thresholding*
   Konversi gambar dari citra grayscale menjadi citra biner berdasarkan threshold.
4. *Image Complement*
   Membalikkan gambar yang semula ber*background* putih dan karakter berwarna hitam menjadi ber*background* hitam dan karakter berwarna putih.
5. *Thinning*
   Menipiskan gambar agar lebih akurat dalam klasifikasi.

Gambar 3. 3 Gambar setelah dithinning

6. *Bounding Box Cropping*
   Cropping image menggunakan *Regionprops Boundingbox* karena ada beberapa data yang tulisannya lebih kecil dari kotak form atau terlalu besar sehingga terlalu dekat dengan garis form. Dengan *Boundingbox* ukuran akan menjadi sama dan mempermudah dalam deteksi.
7. *Resize*
   Resize citra menjadi berukuran 100x100 pixel

3.3 *Diagonal Feature Extraction*

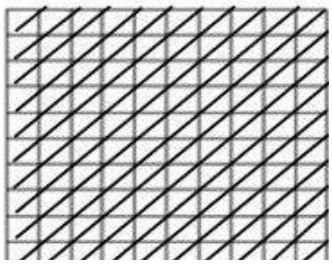
	1	2	3	4	5	6	7	8	9	10
1	10x10 cell	10x10 cell	10x10 cell	10x10 cell	10x10 cell	10x10 cell	10x10 cell	10x10 cell	10x10 cell	10x10 cell
2	10x10 cell	10x10 cell	10x10 cell	10x10 cell	10x10 cell	10x10 cell	10x10 cell	10x10 cell	10x10 cell	10x10 cell
3	10x10 cell	10x10 cell	10x10 cell	10x10 cell	10x10 cell	10x10 cell	10x10 cell	10x10 cell	10x10 cell	10x10 cell
4	10x10 cell	10x10 cell	10x10 cell	10x10 cell	10x10 cell	10x10 cell	10x10 cell	10x10 cell	10x10 cell	10x10 cell
5	10x10 cell	10x10 cell	10x10 cell	10x10 cell	10x10 cell	10x10 cell	10x10 cell	10x10 cell	10x10 cell	10x10 cell
6	10x10 cell	10x10 cell	10x10 cell	10x10 cell	10x10 cell	10x10 cell	10x10 cell	10x10 cell	10x10 cell	10x10 cell
7	10x10 cell	10x10 cell	10x10 cell	10x10 cell	10x10 cell	10x10 cell	10x10 cell	10x10 cell	10x10 cell	10x10 cell
8	10x10 cell	10x10 cell	10x10 cell	10x10 cell	10x10 cell	10x10 cell	10x10 cell	10x10 cell	10x10 cell	10x10 cell
9	10x10 cell	10x10 cell	10x10 cell	10x10 cell	10x10 cell	10x10 cell	10x10 cell	10x10 cell	10x10 cell	10x10 cell
10	10x10 cell	10x10 cell	10x10 cell	10x10 cell	10x10 cell	10x10 cell	10x10 cell	10x10 cell	10x10 cell	10x10 cell

Gambar 3. 4 Output dari pembagian zona

Setiap karakter dibagi menjadi 100 zona kecil dengan ukuran tiap zona 10x10 piksel

	1	2	3	4	5	6	7	8	9	10
1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	1	1
6	0	0	0	0	0	0	0	1	1	1
7	1	1	1	1	1	1	1	1	0	0
8	1	1	1	1	1	1	1	0	0	0
9	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0

Gambar 3. 5 Zona berukuran 10x10 piksel



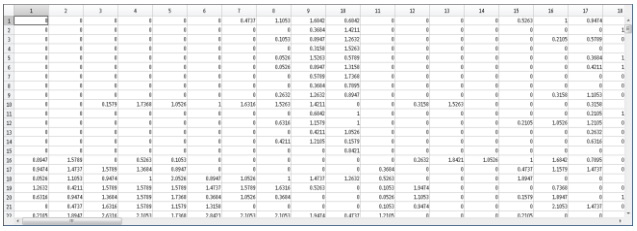
Gambar 3. 6 Ekstraksi ciri diagonal dari kiri

Dari zona berukuran 10x10 piksel Gambar 3.11 diekstraksi cirinya secara diagonal dengan bergerak dari kiri Gambar 3.12 . dilakukan berulang kali untuk 100 zona sehingga tiap karakter memiliki 100 ciri.

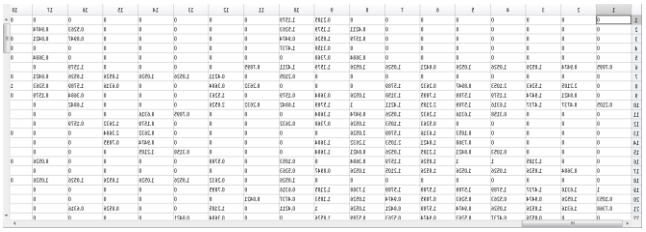
	1	2	3	4	5	6	7	8	9	10
1	0	0	0	0.0526	1.8947	1.2105	0.2105	0	0	0
2	0	0	0	1.1579	0	0	1.7368	0.1053	0	0
3	0	0	0.3684	0.8947	0	0	0	1.3158	0	0
4	0	0	1.4211	0	0	0	0	0.5263	0.6842	0
5	0	0.5263	0.6316	0	0	0	0	0.0526	1.8947	0
6	0	1	0.4737	0.3684	0.6842	1.3158	1.6842	1.1053	1.2105	0.3158
7	0.4737	0.9474	0	0	0	0	0	0	0	1
8	1.1053	0	0	0	0	0	0	0	0	1.1053
9	1.6842	0	0	0	0	0	0	0	0	1.5263
10	0.6842	0	0	0	0	0	0	0	0	1.5789

Gambar 3. 7 100 ciri yang terekstraksi

3.4 K-Nearest Neighbour



Gambar 3. 8 Hasil ekstraksi ciri data model



Gambar 3. 9 Hasil ekstraksi ciri data uji

Hasil dari data model dan data uji akan dibandingkan dengan menggunakan k-NN. Untuk pengukuran *distancenya* digunakan *Euclidean Distance.*

4. Pengujian dan Analisis

4.1 Spesifikasi

4.1.1 Spesifikasi Perangkat Keras

Perangkat keras yang digunakan untuk membangung *prototype* antara lain :

1. Processor : Intel Pentium Core i3 2350M @2.30 GHz
2. RAM : Corsair CMSO4GX3M1A1600C11 4096 MBytes
3. Harddisk : 464 GB
4. VGA :Intel(R) HD Graphics 3000

4.1.2 Spesifikasi Perangkat Lunak

Perangkat lunak yang digunakan untuk membangung *prototype* antara lain :

1. Sistem Operasi Windows 7 Professional
2. Matlab R2015a 8.5.0.197613
3. Adobe Photoshop CS3
4. Corel Draw Graphic Suite X4

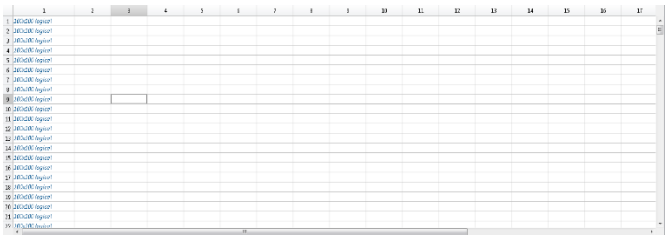
4.2 Pengujian

Setelah perancangan dan implementasi sistem, hal selanjutnya yang dilakukan adalah pengujian sistem. Pengujian sistem dilakukan dengan cara mengganti parameter di setiap sekenario untuk mengetahui dampak performansi sistem dari perubahan parameter.

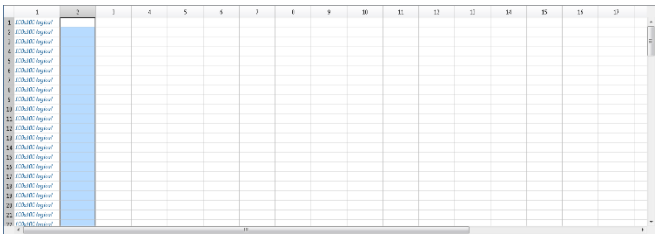
4.2.1 Tujuan Pengujian

1. Mengetahui pengaruh ekstraksi ciri diagonal dalam keakuratan sistem
2. Mengetahui pengaruh besar ukuran tiap zona pada diagonal feature extraction antara 10x10 pixel dan 25x25 pixel pada keakuratan sistem.
3. Mengetahui pengaruh langkah diagonal antara dari kiri ke kanan atau dari kanan ke kiri pada keakuratan sistem dan juga kombinasi dari diagonal kiri dan kanan.
4. Mengetahui pengaruh jumlah tetangga atau nilai *k* pada *k*-NN pada keakuratan sistem

Sebelum memasuki pengujian dilakukan *preprocessing* terhadap 390 data huruf kapital A-Z untuk dijadikan data model. Data model disimpan dan diberi label. 130 data uji juga di*preprocessing* terlebih dahulu.



Gambar 4. 1 Data model yang telah di preprocessing



Gambar 4. 2 Data uji yang telah di preprocessing

4.3 Analisis Hasil Pengujian

4.3.1 Pengujian Ekstraksi ciri diagonal

Dilakukan pengujian terhadap ekstraksi ciri kanan, ekstraksi ciri kiri, dan kombinasi ekstraksi ciri kanan dan kiri. Pada tiap ekstraksi ciri digunakan pembagian 100, 25, 16 zona kecil. Pada pengujian ini diuji dengan menggunakan k-NN dengan nilai k = 1 atau dengan pencocokan satu per satu sederhana.

4.3.1.1 Pengujian Diagonal Kiri

1. Pembagian 100 zona

   Ekstraksi ciri diagonal kiri dengan pembagian 100 zona. Tiap zona berukuran 10x10 piksel. Menghasil ciri sebanyak 100.

   Hasil dari ekstraksi ciri diklasifikasi dengan k-NN k=1.

	1
1	0.7615

Gambar 4. 3 Akurasi dengan ekstraksi ciri 100 zona

2. Pembagian 25 zona

   Ekstraksi ciri diagonal kiri dengan pembagian 25 zona. Tiap zona berukuran 20x20 piksel. Menghasil ciri sebanyak 25.

	1
1	0.7077

Gambar 4. 4 Akurasi diagonal kiri pembagian 25 zona

3. Pembagian 16 zona

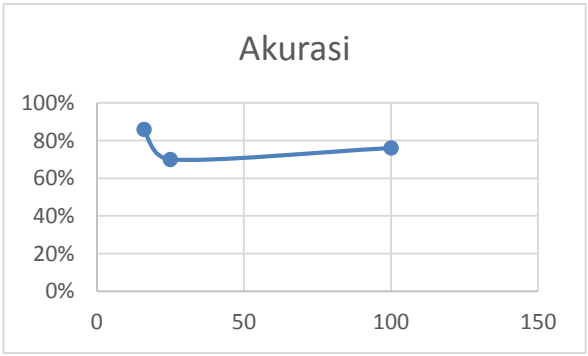
   Ekstraksi ciri diagonal kiri dengan pembagian 16 zona. Tiap zona berukuran 25x25 piksel. Menghasil ciri sebanyak 16.

	1
1	0.8615

Gambar 4. 5 Akurasi diagonal kiri pembagian 16 zona

Tabel 4. 1 Tabel akurasi dengan diagonal kiri

Pembagian Zona	Akurasi
100	76%
25	70%
16	86%



Gambar 4. 6 Grafik akurasi dengan diagonal kiri

Pada pembagian 100 zona kecil tiap citra akan mempunyai 100 ciri. Pada pembagian 25 zona tiap citra akan mempunyai 25 ciri dan pada pembagian 16 zona akan diperoleh 16 ciri tiap image.

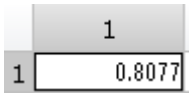
Pada pembagian 100 zona kecil tiap zona memiliki ukuran 10x10

pixel. Pada 25 zona tiap zona berukuran 20x20 pixel dan pada 20 zona tiap zonanya berukuran 25x25 pixel. Semakin banyak pembagian zona ciri yang diperoleh semakin banyak. Akan tetapi pada pembagian 100 zona banyak ciri yang bernilai nilai 0 karena ada beberapa zona yang tidak terlewati objek huruf atau hanya mendeteksi *background* yang berwarna hitam. Hal ini menimbulkan keambiguan pada saat klasifikasi karena pada image lain akan banyak jg zona yang menghasilkan. Jika ukuran zona semakin besar maka ciri yang bernilai 0 lebih sedikit dan keambiguan klasifikai antar karakter akan lebih sedikit. Hal ini meningkatkan akurasi sistem cukup signifikan.

4.3.1.2 Pengujian Diagonal Kanan

1. Pembagian 100 zona

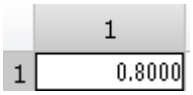
   Ekstraksi ciri diagonal kanan dengan pembagian 100 zona. Tiap zona berukuran 10x10 piksel. Menghasil ciri sebanyak 100.



Gambar 4. 7 Akurasi diagonal kanan pembagian 100 zona

2. Pembagian 25 zona

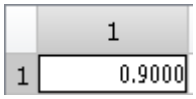
   Ekstraksi ciri diagonal kanan dengan pembagian 25 zona. Tiap zona berukuran 20x20 piksel. Menghasil ciri sebanyak 25.



Gambar 4. 8 Akurasi diagonal kanan pembagian 25 zona

3. Pembagian 16 zona

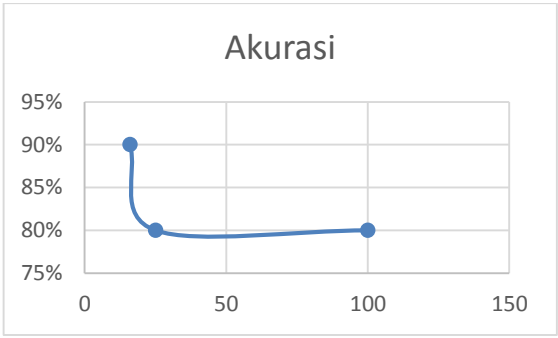
   Ekstraksi ciri diagonal kanan dengan pembagian 16 zona. Tiap zona berukuran 25x25 piksel. Menghasil ciri sebanyak 16.



Gambar 4. 9 Akurasi diagonal kanan pembagian 16 zona

Tabel 4. 2 Tabel akurasi dengan diagonal kanan

Pembagian Zona	Akurasi
100	80%
25	80%
16	90%



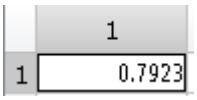
Gambar 4. 10 Grafik akurasi sistem menggunakan diagonal kanan

Untuk ekstraksi ciri dari kanan menghasilkan peningkatan akurasi sebanyak 4-10% dibandingkan dengan ekstraksi ciri diagonal kiri. Yang memberikan perbedaan cukup signifikan adalah besar tiap zona yang digunakan untuk membagi citra.

4.3.1.3 Pengujian Kombinasi Diagonal Kiri dan Kanan

1. Pembagian 100 zona

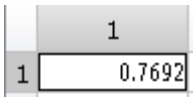
   Kombinasi ekstraksi ciri diagonal kanan dan kiri dengan pembagian 100 zona. Tiap zona berukuran 10x10 piksel. Menghasil ciri sebanyak 100.



Gambar 4. 11 Akurasi kombinasi diagonal kanan dan kiri pembagian 100 zona

2. Pembagian 25 zona

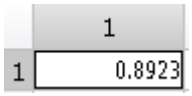
Kombinasi ekstraksi ciri diagonal kanan dan kiri dengan pembagian 25 zona. Tiap zona berukuran 20x20 piksel. Menghasil ciri sebanyak 25.



Gambar 4. 12 Akurasi kombinasi diagonal kiri dan kanan pembagian 25 zona

3. Pembagian 16 zona

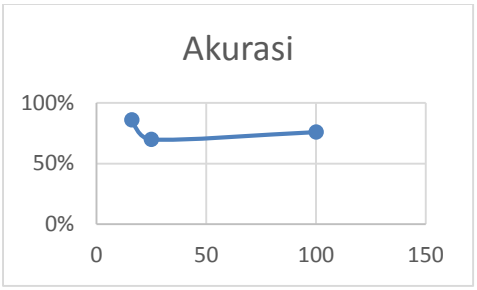
Kombinasi ekstraksi ciri diagonal kanan dan kiri dengan pembagian 16 zona. Tiap zona berukuran 25x25 piksel. Menghasil ciri sebanyak 16.



Gambar 4. 13 Akurasi kombinasi diagonal kiri dan kanan pembagian 16 zona

Tabel 4. 3 Tabel akurasi kombinasi diagonal kanan dan kiri

Pembagian Zona	Akurasi
100	79%
25	76%
16	89%

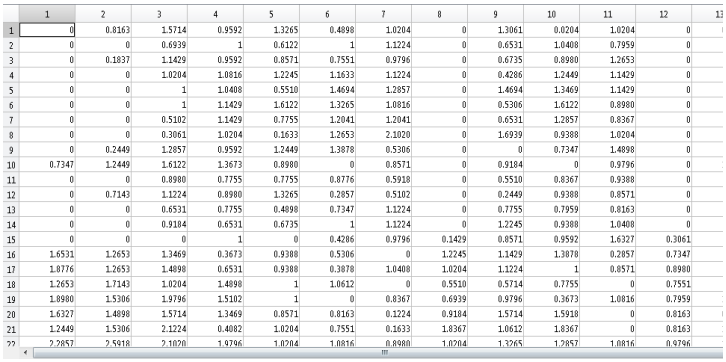


Gambar 4. 14 Grafik akurasi sistem dengan menggunakan kombinasi diagonal kanan dan kiri

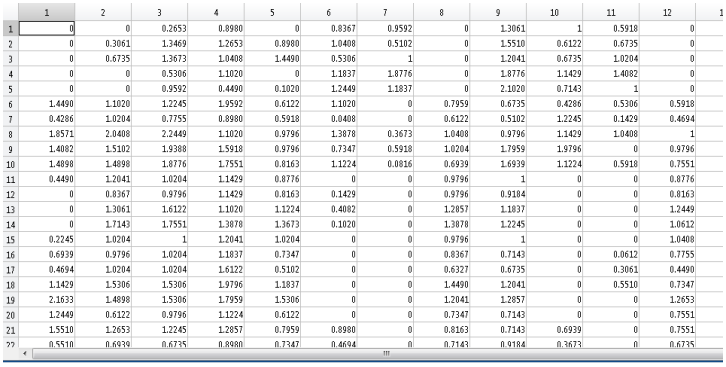
Pengujian menggunakan kombinasi diagonal kanan dan kiri ini tidak terlalu memberikan peningkatan akurasi malah justru mengurangi akurasi sistem. Penurunan akurasi sebanyak 3% dibandingkan dengan diagonal kiri. 1-4% terhadap diagonal kanan. Untuk penggunaan metode ini kurang baik karena waktu pemprosesan lebih lama. Melakukan diagonal kiri dan kemudian kanan yang kemudian baru dijumlahkan.

4.3.2 Pengujian k pada k-NN

Dari pengujian ekstraksi ciri diagonal diperoleh hasil akurasi terbaik yaitu 90% pada ekstraksi ciri diagonal dari kanan dengan pembagian 16 zona kecil. Ekstraksi ciri diagonal dari kanan dengan pembagian 16 zona dipakai untuk pengujian banyaknya tetangga dalam klasifikasi k-NN. Jumlah tetangga yang digunakan dalam pengujian adalah k=3, k=4, k=5.

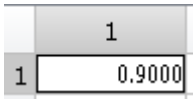


Gambar 4. 15 Ciri data model dengan 16 zona



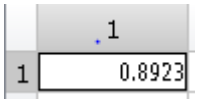
Gambar 4. 16 Ciri data uji dengan 16 zona

Dari hasil ekstraksi ciri diagonal kanan dengan pembagian 16 zona diklasifikasi menggunakan k-NN dengan k=2, k=3, k=4, dan k=5.



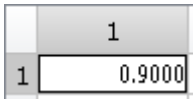
	1
1	0.9000

Gambar 4. 17 Akurasi dengan k=2



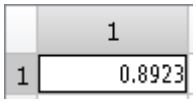
	1
1	0.8923

Gambar 4. 18 Akurasi dengan k=3



	1
1	0.9000

Gambar 4. 19 Akurasi dengan k=4



	1
1	0.8923

Gambar 4. 20 Akurasi dengan k=5

Pengujian terhadap k diperoleh akurasi terbaik yaitu 90%. Pada pengujian k, antara k=2, k=4 dan k=3, k=5 menunjukan akurasi yang sama, yaitu 89% dan 90%. Hal ini dikarenakan apabila jumlah tetangga ganjil, maka akan dilakukan voting mayoritas. Untuk jumlah tetangga genap voting dilakukan dengan menggunakan *euclidean distance.* Dalam hal ini k=2 adalah yang terbaik karena pemprosesan lebih cepat dengan membandingkan 2 tetangga.

4.4 Analisis Kesalahan

Sistem yang penulis bangun ini memiliki beberapa kelemahan dalam mendeteksi beberapa huruf sebagaimana ditampilkan pada Tabel 4.4 .

Tabel 4. 4 Tabel kesalahan

	P1	P2	P3	P4	P5	Total
A	1	1	1	1	1	5
B	5	4	2	2	2	3
C	3	3	3	3	3	5
D	4	4	4	4	4	5
E	5	3	5	5	5	4
F	6	6	6	6	6	5
G	7	7	7	15	3	3
H	8	8	8	8	8	5
I	9	25	9	9	9	4
J	10	10	10	10	10	5
K	11	9	11	11	8	3
L	12	12	12	12	12	5
M	13	13	13	13	13	5
N	8	14	14	14	14	4

	P1	P2	P3	P4	P5	Total
O	15	15	15	15	4	4
P	16	6	16	16	16	4
Q	17	17	17	17	17	5
R	18	12	18	18	18	4
S	26	19	19	19	19	4
T	20	20	20	20	20	5
U	21	21	21	21	21	5
V	22	22	22	22	22	5
W	23	23	23	23	23	5
X	24	24	24	24	24	5
Y	25	25	25	25	25	5
Z	26	26	26	26	26	5
Total						117
Akurasi						0.9

Data uji diambil dari 5 orang yang menuliskan huruf A-Z. Dari 5 orang ini dilambangkan dengan P1, P2, P3, P4, dan P5. Jika A=1, B=2,..., Z=26 maka data tersebut benar. A=1 berarti A terbaca A, jika A=2 maka A terbaca B. Hal ini dikarenakan karena beberapa hal. Penggunaan *regionprops boundingbox* untuk *cropping* dan penggunaan ekstraksi ciri diagonal itu sendiri.

Gambar 4. 21 E terbaca C

Pada huruf B, B dibaca E dan D karena proses ekstraksi ciri diagonal mengekstraksi B ciri yang nilainya lebih dekat ke E dan D. Pada Huruf E terbaca huruf C karena garis tengah E yang terpisah dibaca sebagai *noise* bukan objek oleh *regionprops*.

5. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil penelitian didapatkan bahwa kesimpulan dari penelitian ini antara lain :

1. Semakin besar ukuran tiap zona akan mengurangi keambiguan tiap karakter karena lebih sedikit ciri yang bernilai 0.
2. Ekstraksi ciri diagonal kanan memberikan peningkatan akurasi sebesar 4-10% karena langkah yang dilakukan sama hanya berbeda koordinat saat deteksi.
3. Kombinasi 2 ekstraksi ciri dari kiri dan kanan memperlihatkan penurunan akurasi 3-6% terhadap ekstraksi diagonal kiri. Penurunan sebesar 1-4% terhadap ekstraksi diagonal kanan. Kombinasi 2 ekstraksi ini akan memakan waktu pemprosesan lebih lama.
4. Klasifikasi k-NN dengan k=2 dan k=4 menghasilkan akurasi 90%. Untuk k=1 dan k=5 menghasilkan akurasi 89%.

5.2 Saran

Saran yang diperlukan untuk pembangunan sistem lebih lanjut adalah:

1. Pembangunan sistem dengan lebih memperhatikan *noise*.
2. Menggunakan input yang lebih bervariasi seperti tulisan tangan pada kertas yang rusak.

6. DAFTAR PUSTAKA

[1] Anonim. *Jaringan Syaraf Tiruan.* http://dc372.4shared.com/doc/M0lSKXEx/preview.html

[2] Akbar, Ryan Harry. 2012. *Klasifikasi Kualitas Kelayakan Daging Sapi Pada Citra*

Digital Dengan Metode Run Lenght dan K-Nearest Neighbour. Bandung. Institut Teknologi Telkom.

[3] Arfian, Riki. 2012. *Pengenalan Huruf Pada Form Menggunakan Metode Ekstraksi Ciri Diagonal Dan Algoritma Probabilistic Neural Network.* Bandung. Institute Teknologi Telkom

[4] Dwi Putra, M. Iqbal. 2013. *Pengenalan Tanda Tangan Dengan Menggunakan Probabilistik Neural Network.* Bandung. Institute Teknologi Telkom.

[5] Fransiskus. 2010. *OCR – Optical Character Recognition.* Blogspot. http://fransarea.blogspot.com/2010/05/ocr-optical-character-recognition.html.

[6] Jatmiko, Sasongko. 2013. *Analisis dan Implementasi Penggunaan Scale Invariant Feature Transform (SIFT) Pada Sistem Verifikasi Tanda Tangan.* Bandung. Telkom University.

[7] Kadir, Abdul; Susanto Adhi. 2013. *Teori dan Aplikasi Pengolahan Citra.* Yogyakarta. Penerbit Andi

[8] Pradeep, J.; Srinivasan, E.; and Himavathi, S. . 2011. *Diagonal Based Feature Extraction for Handwritten Alphabets Recognition System Using Neural Network.* India. Department of ECE Pondicherry College Engineering. International Journal of Computer Science & Information Technology (IJCSIT).

[9] Pradhana, Andhika. 2013. *LISTIE : Indonesia License Plat Identification.* Bandung. Institute Teknologi Telkom.

[10] Putra, Alfian Pramudita. 2013. *Fiture Extraction.* UNAIR. http://alfian-p-p-fst10.web.unair.ac.id/artikel_detail-76028-INTELLIGENT%20SYSTEM-fiture%20extraction.html