

Pengenalan Tulisan Tangan Offline Dengan Algoritma Generalized Hough Transform dan Backpropagation

Farica Perdana Putri¹, Adhi Kusnadi²

Program Studi Informatika, Universitas Multimedia Nusantara, Tangerang, Indonesia
 farica@umn.ac.id
 adhi.kusnadi@umn.ac.id

Diterima 30 Maret 2018
 Disetujui 8 Juni 2018

Abstract—Offline handwriting recognition is a technique used to recognize handwriting in paper document which converting it to digital form. Each handwriting has a unique style and shape that can be used to identify the owner. This research aims to develop a method to recognize the digital data handwriting. The method combines two algorithms; the first is Generalized Hough Transform in feature extraction process to detect arbitrary objects on the image; the second algorithm is Backpropagation to train the neural network based on feature values from feature extraction process. Artificial Neural Network (ANN) is used to improve the accuracy of the recognition system. The experiments are performed by using 100 handwriting images of 10 different people. The number of hidden units is defined through experiment to obtain optimal neural network. The experiment result shows that the recognition accuracy is up to 80%.

Index Terms—Artificial Neural Network, Backpropagation, Generalized Hough Transform, Offline handwriting recognition

I. PENDAHULUAN

Tulisan tangan merupakan salah satu hal unik yang dapat dihasilkan oleh manusia selain tanda tangan. Seperti halnya tanda tangan, tulisan tangan juga dapat digunakan untuk mengidentifikasi pemiliknya. Beberapa penelitian pun telah memfokuskan pada teknik-teknik baru dan metode yang akan mengurangi waktu pemrosesan dan tingkat keakuratan yang lebih tinggi [1].

Pengenalan tulisan merupakan bagian dari pengenalan oleh komputer (*computer recognition*). Pengenalan oleh komputer merupakan proses otomatisasi yang menggunakan teknik pengenalan pola dan pembelajaran mesin untuk mengenal karakter atau kata yang diberikan [2]. Berdasarkan teknik yang digunakan untuk memperoleh input data, terdapat dua macam pengenalan tulisan tangan, yaitu *offline* dan *online*. Pada pengenalan tulisan *online*, data dikumpulkan ketika dibuat pada permukaan digital dengan menggunakan alat tulis seperti *stylus* [3]. Tetapi, pada pengenalan tulisan *offline*, data berupa

dokumen tulisan tangan yang dipindai terlebih dahulu [3]. Secara umum, terdapat lima tahapan dalam sistem pengenalan tulisan tangan *offline*, yaitu *image scanning*, *image preprocessing*, segmentasi, ekstraksi ciri, dan *neural network* [4].

Ada banyak algoritma yang dapat digunakan pada tahap ekstraksi ciri yang telah dikembangkan, salah satu diantaranya adalah *Generalized Hough Transform* (GHT). GHT merupakan pengembangan dari algoritma *Hough Transform* (HT). Berdasarkan rujukan [5] dan [6], GHT terbukti lebih efektif digunakan untuk mendeteksi bentuk yang berubah-ubah, seperti tulisan tangan, dibandingkan dengan HT. GHT adalah sebuah algoritma berbasis kontur yang hanya menggunakan informasi kontur untuk proses pengenalan pola, sehingga secara efektif dapat mengurangi pengaruh *noise* [7].

Setelah melalui ekstraksi ciri, ciri-ciri penting yang diperoleh digunakan pada tahap *neural network*. Jaringan syaraf tiruan (JST) atau *artificial neural network* digunakan untuk meningkatkan akurasi dalam mengenali tulisan tangan seseorang. Salah satu algoritma yang dapat digunakan untuk melatih jaringan syaraf adalah *Backpropagation*. Algoritma ini dapat meningkatkan kecepatan dari proses pembelajaran jaringan [4]. Penggunaan algoritma ini sesuai untuk aplikasi pengenalan pola dan merupakan salah satu algoritma pelatihan dengan supervisi yang dapat meningkatkan kinerja jaringan karena telah ditetapkan tujuan atau target dari pelatihan [8]. Penelitian ini bertujuan untuk mengimplementasikan algoritma ekstraksi ciri *Generalized Hough Transform* dan *Backpropagation* pada pengenalan tulisan tangan *offline*.

Pembahasan penelitian ini dibagi menjadi beberapa bagian. Analisis masalah dan penelitian dijelaskan pada bagian PENDAHULUAN. Landasan teori yang digunakan diuraikan pada bagian TELAAH LITERATUR. Perancangan dari arsitektur jaringan syaraf tiruan yang digunakan akan dijelaskan pada bagian PERANCANGAN ARSITEKTUR

JARINGAN SYARAF TIRUAN. Perancangan sistem pengenalan tulisan tangan dibahas pada bagian PERANCANGAN SISTEM. Uji coba dan hasil pengujian dibahas pada bagian IMPLEMENTASI APLIKASI dan UJI COBA. Adapun kesimpulan dan saran yang dapat digunakan untuk penelitian selanjutnya dijabarkan pada bagian SIMPULAN. Kemudian, saran yang dapat digunakan untuk penelitian selanjutnya diuraikan pada bagian SARAN.

II. TELAHAH LITERATUR

A. Pengenalan Tulisan Tangan Offline

Pengenalan tulisan tangan mengarah pada identifikasi karakter (huruf ataupun angka) yang tertulis [9]. Berdasarkan bagaimana input didapatkan, pengenalan tulisan tangan dibagi menjadi dua, yaitu pengenalan tulisan tangan *offline* dan pengenalan tulisan tangan *online* [4]. Pengenalan tulisan tangan *offline* merupakan pengenalan tulisan yang didasarkan pada citra dari sebuah teks tertulis [10].

Pengenalan tulisan tangan *offline* ini melibatkan konversi otomatis teks dalam citra ke dalam kode-kode huruf yang digunakan dalam komputer dan aplikasi pengolahan teks [10]. Data yang diperoleh dianggap sebagai representasi statis tulisan tangan. Tahapan dalam pengenalan tulisan tangan ini adalah [4]:

1. *Image Scanning*. Pada tahap ini, dokumen tertulis akan dipindai dengan menggunakan *scanner* dan kemudian citra tersebut akan menjadi input ke dalam aplikasi, khususnya tahap *image preprocessing*. Citra ini memiliki format tertentu seperti JPEG, BMP, dll.
2. *Image Preprocessing*. Keluaran dari tahap sebelumnya merupakan masukan bagi tahap *image preprocessing* [4]. Data mentah dari tulisan tangan akan diberikan sejumlah tahapan *preprocessing* untuk membuat data tersebut dapat berguna untuk tahapan selanjutnya [4]. Tahap *image preprocessing* bertujuan untuk mengekstraksi bagian-bagian tekstual, bagian yang terdapat tulisan atau karakter, yang relevan dan mempersiapkannya untuk tahap segmentasi dan ekstraksi ciri [4]. *Image preprocessing* pada sebuah citra diartikan sebagai penerapan beberapa prosedur seperti *thresholding*, penghalusan, penyaringan, pengubahan skala, dan normalisasi sehingga algoritma untuk klasifikasi akhir dapat lebih sederhana dan akurat.
3. Segmentasi. Sebuah citra ditampilkan dalam bentuk deretan karakter-karakter yang diuraikan ke dalam bagian-bagian citra dari individual karakter pada tahapan segmentasi [4].
4. Ekstraksi ciri. Ekstraksi ciri berkonsentrasi pada menemukan bentuk pada citra [11]. Setiap karakter memiliki ciri yang berbeda-beda, yang memainkan peranan penting dalam pengenalan

pola [4]. Tujuan utama dari ekstraksi ciri adalah untuk menyediakan informasi yang paling relevan dari data asli dan merepresentasikan informasi tersebut pada ruang dimensional yang lebih rendah [4].

5. *Neural Network*. Jaringan syaraf belajar dari citra pelatihan awal. Jaringan akan dilatih, kemudian dapat membuat identifikasi dari karakter. Setiap jaringan syaraf, secara unik mempelajari *properties* dan perbedaan dari citra-citra tersebut [10]. Selanjutnya, akan mencari *properties* yang sama pada citra untuk diidentifikasi [10]. *Backpropagation* merupakan algoritma yang digunakan untuk melatih jaringan syaraf. Jaringan dapat digunakan untuk mempelajari karakter pada pola dan melatih jaringan disimulasikan untuk pengenalan karakter yang ditampilkan dalam bentuk citra [4].

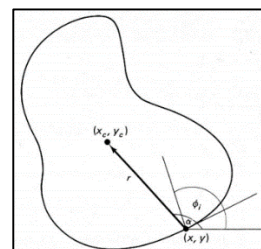
B. Algoritma Generalized Hough Transform

Hough Transform merupakan salah satu algoritma yang dapat digunakan untuk ekstraksi ciri. Ide dasar dari *Hough Transform* adalah untuk memetakan titik tepi citra ke parameter ruang, yang mewakili semua instance dari ciri yang terlihat pada citra [6]. *Hough Transform* dapat mengekstrak bentuk – bentuk yang sederhana seperti garis dan bentuk dua dimensi lainnya, tetapi tidak untuk bentuk – bentuk yang lebih kompleks dan berubah-ubah [7]. Oleh karena itu, digunakanlah algoritma *Generalized Hough Transform*.

Generalized Hough Transform merupakan penemuan dari DH. Ballard. *Generalized Hough Transform* adalah perkembangan dari *Hough Transform* untuk mendeteksi obyek yang berubah-ubah pada citra. Semua bentuk tidak dapat secara mudah dinyatakan dengan sekelompok kecil parameter [6]. Solusi yang diusulkan kemudian untuk membuat tabel, dinamakan *R-table*, untuk menyimpan semua piksel sisi dari target bentuk yang diberikan [6].

Implementasi *Generalized Hough Transform* menggunakan informasi sisi untuk mendefinisikan pemetaan dari orientasi titik tepi ke titik referensi dari bentuk tersebut. Dalam menerapkan algoritma *Generalized Hough Transform* terdapat beberapa tahapan [12], yaitu:

1. Menetapkan titik acuan (x_c, y_c) yang dapat dihitung menggunakan rumus berikut dengan contoh obyek citra seperti pada Gambar 1 di bawah ini.



Gambar 1. Contoh Obyek Citra Untuk Implementasi GHT [12]

Rumus yang digunakan untuk menghitung titik acuan adalah berikut.

$$x_c = x_i + r_i \cos(\alpha_i) \quad (1)$$

$$y_c = y_i + r_i \sin(\alpha_i) \quad (2)$$

2. Menggambar garis dari titik acuan ke titik pada tepi obyek (P_i) dan menghitung panjang dari r_i dan sudut α_i.
3. Menuliskan koordinat dari titik acuan (x_c, y_c) sebagai fungsi dari r_i dan α_i.
4. Menyimpan orientasi gradien (φ_i) pada P_i.
5. Membuat R-table dengan data-data di atas, dan sebagai indeks φ_i.

Angle measured from figure boundary to reference point	Set of radii {r ^k } where r = (r, α)
φ ₁	r ₁ ¹ , r ₂ ¹ , ..., r _{n₁} ¹
φ ₂	r ₁ ² , r ₂ ² , ..., r _{n₂} ²
⋮	⋮
⋮	⋮
φ _m	r ₁ ^m , r ₂ ^m , ..., r _{n_m} ^m

Gambar 2. Contoh R-table pada Generalized Hough Transform [12]

6. Menemukan versi kurva obyek setelah diputar, diperbesar/perkecil, dan ditranslasikan dengan membuat accumulator array dari titik acuan, diperbesar/perkecil dengan faktor S dan diputar dengan sudut θ. Untuk setiap tepi (x, y) pada citra, menghitung φ(x, y).

Kemudian untuk setiap (r, α) yang sesuai dengan φ(x, y) lakukan untuk setiap S dan θ:

$$x_c = x_i + r(f) S \cos[a(f) + q] \quad (3)$$

$$y_c = y_i + r(f) S \sin[a(f) + q] \quad (4)$$

$$A(x_c, y_c, S, q) = A(x_c, y_c, S, q) + 1 \quad (5)$$

Dan mencari maxima dari A.

C. Algoritma Backpropagation

Algoritma Backpropagation merupakan salah satu algoritma pembelajaran yang bekerja pada jaringan multilayer feed-forward, menggunakan gradien turunan pada bobot untuk meminimalkan kesalahan keluaran [13]. Terdapat dua fase pada algoritma Backpropagation [14]:

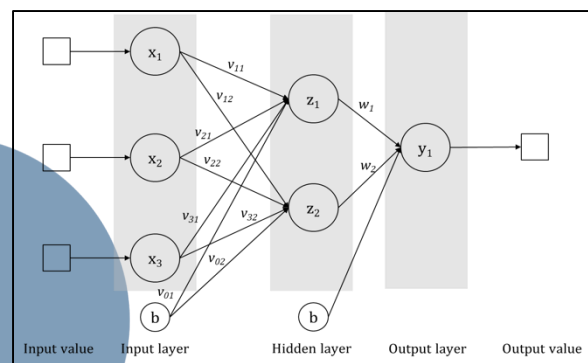
1. Fase Forward

Fase ini merupakan tempat dimana fungsi aktivasi merambat dari input layer ke output layer [14]. Fungsi aktivasi yang digunakan adalah sigmoid biner dengan rumus sebagai berikut.

$$f(x) = \frac{1}{1 + e^{-x}} \quad (6)$$

2. Fase Backward

Fase ini merupakan tempat dimana nilai aktual yang diamati dan nilai nominal yang diminta pada output layer dirambatkan mundur ke input layer sehingga bobot dapat dimodifikasi sampai nilai keluaran yang dihasilkan mendekati atau bahkan sama dengan diinginkan [14].



Gambar 3. Contoh Arsitektur Jaringan Syaraf Tiruan

Gambar 3. merupakan contoh dari arsitektur jaringan syaraf tiruan untuk Backpropagation. Tahapan dari Backpropagation adalah sebagai berikut.

1. Menginisialisasi nilai bobot dan bias dengan nilai secara acak yang bernilai antara [0,1] atau [-0,5, 0,5].
2. Menghitung keluaran dari hidden layer.

$$z_j = \frac{1}{1 + e^{v_{0j} + \sum_{i=1}^n v_{ij}x_i}} \quad (7)$$

z_j = keluaran dari unit ke-j pada hidden layer

v_{0j} = bobot bias pada input layer ke hidden unit ke-j

x_i = masukkan pada unit ke-i

v_{ij} = bobot dari input unit ke-i ke hidden layer ke-j

3. Menghitung keluaran dari output layer.

$$y_k = \frac{1}{1 + e^{w_{0k} + \sum_{j=1}^m w_{jk}z_j}} \quad (8)$$

y_k = keluaran dari output unit ke-k

w_{jk} = bobot dari hidden unit ke-j ke output unit ke-k

w_{0k} = bobot bias pada hidden layer ke output unit ke-k

4. Menghitung kesalahan untuk unit pada output layer.

$$\delta_k = (t_k - y_k) y_k (1 - y_k) \quad (9)$$

$$\Delta w_{jk} = \alpha \delta_k z_j \quad (10)$$

$$\Delta w_{0k} = \alpha \delta_k \quad (11)$$

δ_k = sinyal *error* untuk *output* unit ke- k pada *output layer*

α = laju pembelajaran (*learning rate*)

5. Menghitung kesalahan untuk unit pada *hidden layer*.

$$\delta_j = \sum_{k=1}^m w_{jk} \delta_k z_j (1 - z_j) \quad (12)$$

$$\Delta v_{ij} = \alpha \delta_j x_i \quad (13)$$

$$\Delta v_{0j} = \alpha \delta_j \quad (14)$$

δ_j = sinyal *error* untuk *output* unit ke- j pada *hidden layer*

6. Memperbaharui bobot pada *output layer*.

$$w_{jk}^{new} = w_{jk}^{old} + \Delta w_{jk} \quad (15)$$

7. Memperbaharui bobot pada *hidden layer*.

$$v_{ij}^{new} = v_{ij}^{old} + \Delta v_{ij} \quad (16)$$

8. Tes kondisi penghentian

III. PERANCANGAN ARSITEKTUR JARINGAN SYARAF TIRUAN

Perancangan arsitektur JST yang akan dilakukan terdiri dari dua tahap, yaitu pada tahap pelatihan dan tahap pengenalan. Pada tahap pelatihan JST, algoritma *Backpropagation* digunakan untuk mendapatkan bobot yang nantinya disimpan ke dalam *database*. Jaringan akan terus mengubah nilai bobot sampai kriteria tertentu terpenuhi, dalam aplikasi ini adalah nilai kesalahan yang lebih kecil dari toleransi kesalahan.

Tahap selanjutnya adalah tahap pengenalan, aliran pemrosesan dari jaringan hanya dapat bergerak maju, berbeda dengan tahap pelatihan. Bobot yang disimpan dalam *database* akan diambil dan kemudian diberikan kepada JST sebagai bobot untuk pengenalan sehingga didapatkan hasil keluaran.

Kedua tahap di atas, membutuhkan masukan yang didapatkan dari hasil ekstraksi ciri menggunakan algoritma GHT. Jika ingin menggunakan fungsi aktivasi sigmoid biner, data harus ditransformasikan dulu karena rentang keluaran fungsi aktivasi sigmoid adalah $[0,1]$ [8]. Normalisasi dilakukan dengan menggunakan rumus normalisasi sebagai berikut [8].

$$x' = \frac{0.8(x - \text{nilai terkecil})}{\text{nilai terbesar} - \text{nilai terkecil}} + 0.1 \quad (17)$$

Adapun arsitektur JST yang digunakan diuraikan sebagai berikut.

1. Jumlah masukan pada *input layer* ditetapkan adalah 40 buah masukan. GHT menghasilkan 10 titik ciri dari tiap karakter. Banyak masukan

pada jaringan syaraf tiruan didapatkan dari proses ekstraksi ciri dengan GHT (10 titik ciri) dikalikan dengan 4 huruf yang terdapat pada *input* citra.

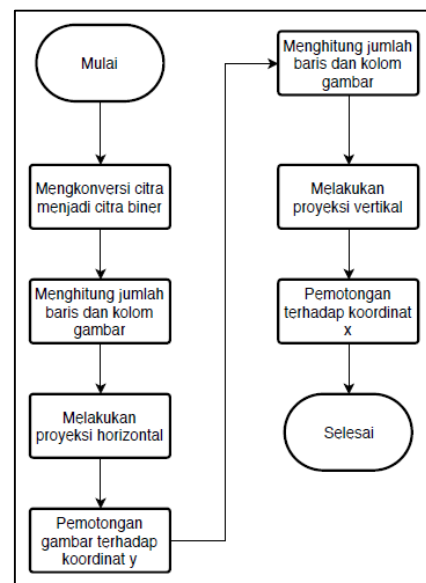
2. *Hidden layer* berjumlah 1 dan jumlah unit pada *hidden layer* akan ditentukan setelah proses pencarian jaringan optimum dengan menghitung nilai *root-mean-square error* (RMSE).
3. Antar *layer* terdapat koneksi satu arah yang memiliki bobot dengan nilai awal dipilih secara acak antara $-0,5 - 0,5$.
4. Jumlah keluaran pada *output layer* adalah 1 dengan nilai target yaitu 1.

IV. PERANCANGAN SISTEM

Sistem pengenalan tulisan tangan *offline* menggunakan algoritma GHT pada tahap ekstraksi ciri dan *Backpropagation* pada tahap *neural network classification* agar dapat mengenali tulisan tangan seseorang pada citra digital. Ciri-ciri yang didapatkan dari tahap ekstraksi ciri menggunakan GHT menjadi *input* untuk jaringan syaraf tiruan pada tahap *neural network classification*.

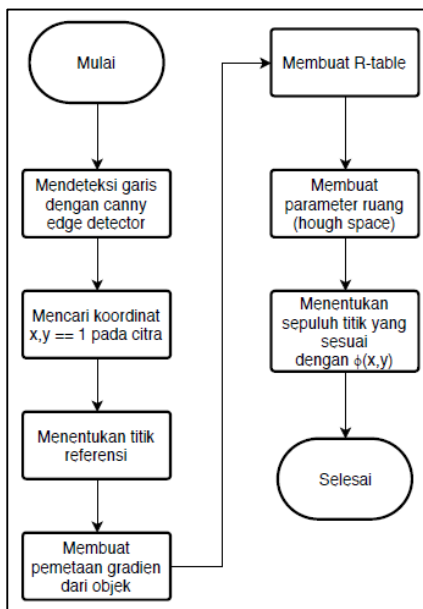
Pada perancangan sistem, digunakan *flowchart* untuk menggambarkan alur proses dan fungsi-fungsi pada sistem. *Flowchart* yang diberikan menggambarkan fungsi segmentasi tulisan tangan, ekstraksi ciri menggunakan GHT, pelatihan JST menggunakan *Backpropagation*, dan pengenalan tulisan tangan.

Gambar 4 merupakan alur dari proses segmentasi tulisan tangan yang dilakukan saat melakukan proses pelatihan ataupun pengenalan JST. Masukkan untuk proses segmentasi merupakan hasil keluaran dari proses *binarization* dari citra yang diberikan oleh *user*. Proses segmentasi adalah sebuah proses untuk memecah karakter atau huruf-huruf dari kata pada citra.

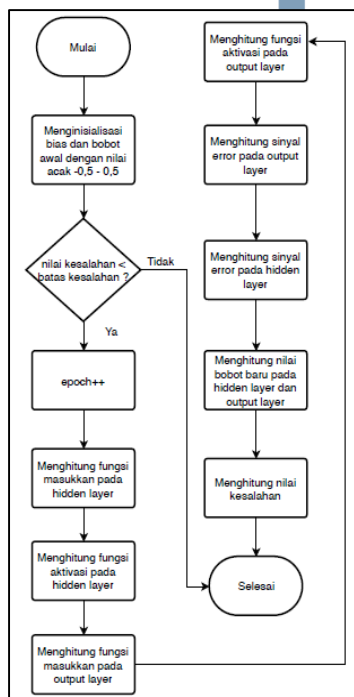


Gambar 4. Flowchart Proses Segmentasi Tulisan Tangan

Setelah proses segmentasi selesai, citra yang telah disegmentasi akan melalui tahap ekstraksi ciri menggunakan GHT. Proses ekstraksi ciri digambarkan pada Gambar 5. Algoritma GHT melakukan pendeteksian garis menggunakan algoritma *canny edge detector* sebelum pemrosesan GHT dilakukan.



Gambar 5. Flowchart Algoritma Generalized Hough Transform



Gambar 6. Flowchart

Hasil keluaran dari proses ekstraksi ciri digunakan sebagai masukan bagi JST pada tahap pelatihan ataupun pengenalan. Gambar 6 menggambarkan alur dari proses pelatihan JST menggunakan algoritma *Backpropagation*.

Selama proses pelatihan JST, bobot pada jaringan akan terus diperbaharui sampai nilai kesalahan lebih kecil dibandingkan dengan batas kesalahan, yaitu 0,00001. Nilai kesalahan dihitung menggunakan rumus sebagai berikut.

$$E = \frac{1}{2} \sum_i (t_i - o_i)^2 \quad (18)$$

E = kesalahan jaringan

t_i = target

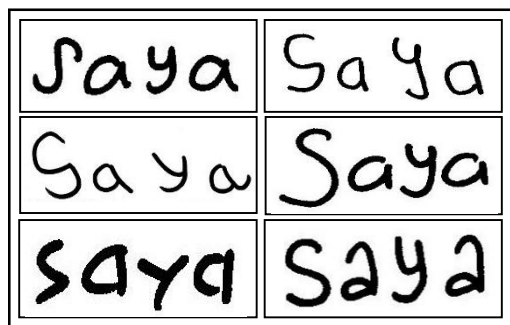
o_i = keluaran / output

Proses pengenalan tulisan tangan yang dilakukan hanya proses *feed-forward* atau maju, tidak ada perubahan bobot pada proses ini. Penghitungan yang dilakukan adalah fungsi masukan dan aktivasi dari masing-masing unit pada *hidden layer* dan *output layer*.

V. UJI COBA

A. Persiapan Pengujian

Sebelum tahap pengujian tulisan tangan, data yang digunakan untuk pelatihan dan pengujian dikumpulkan terlebih dahulu. Tulisan tangan yang digunakan merupakan tulisan cetak dan terdiri dari huruf latin. Data tersebut merupakan tulisan tangan dari 10 orang yang berbeda dan dipindai menjadi citra untuk diproses menggunakan teknik *image processing*. Setiap orang akan menulis sebanyak 10 kali kata “Saya” seperti contoh pada Gambar 7. Data citra dikelompokkan berdasarkan kepemilikan seperti diuraikan pada Tabel 1.



Gambar 7. Contoh Data Citra Tulisan Tangan

Tabel 1. Penamaan dan Pengelompokkan Data Citra Tulisan Tangan

Tulisan Tangan Orang ke -	Penamaan Data Citra
1	TT_1 sampai TT_10
2	TT_11 sampai TT_20
3	TT_21 sampai TT_30
4	TT_31 sampai TT_40
5	TT_41 sampai TT_50
6	TT_51 sampai TT_60
7	TT_61 sampai TT_70
8	TT_71 sampai TT_80
9	TT_81 sampai TT_90
10	TT_91 sampai TT_100

Kemudian, melakukan pemilihan arsitektur jaringan syaraf optimum yang akan digunakan pada tahap pelatihan dan pengenalan tulisan tangan. Pada tahap ini, arsitektur jaringan syaraf optimum, ditentukan menggunakan 40 masukan, 1 *hidden layer*, laju pembelajaran (*learning rate*) 0,5, batas kesalahan 0,00001, fungsi aktivasi yang digunakan adalah fungsi sigmoid biner dengan satu keluaran, dan target keluaran adalah 1. Sedangkan, jumlah unit pada *hidden layer* yang akan diuji coba adalah 2 sampai 10. Perubahan bobot akan terus berlangsung sampai nilai kesalahan lebih kecil dari batas kesalahan yang ditetapkan.

Tabel 2. Hasil Pengujian terhadap Jumlah Unit pada *Hidden Layer (Hidden Unit)*

Parameter	Pengujian				
	1	2	3	4	5
Laju Pembelajaran	0,5	0,5	0,5	0,5	0,5
Hidden Unit	2	3	4	5	6
Batas Kesalahan	10^{-5}	10^{-5}	10^{-5}	10^{-5}	10^{-5}
RMSE	7,24	7,23	7,19	7,18	7,27

Parameter	Pengujian			
	6	7	8	9
Laju Pembelajaran	0,5	0,5	0,5	0,5
Hidden Unit	7	8	9	10
Batas Kesalahan	10^{-5}	10^{-5}	10^{-5}	10^{-5}
RMSE	7,38	7,40	7,35	7,28

Tabel 2 merupakan hasil uji coba untuk menentukan jumlah unit pada *hidden layer* dengan menghitung nilai RMSE. Berdasarkan data pada Tabel 2, nilai RMSE

terkecil didapatkan saat jumlah unit pada *hidden layer* sebanyak 4 buah, yaitu 7,18. Ini dapat diartikan jumlah unit pada *hidden layer* untuk arsitektur JST yang optimum dengan laju pembelajaran 0,5, batas kesalahan 0,00001, 40 unit masukan, 1 *hidden layer*, dan 1 keluaran adalah 4 buah.

Selanjutnya, tahap pelatihan jaringan syaraf tiruan menggunakan 20 tulisan tangan dari 10 orang yang berbeda. Arsitektur jaringan yang digunakan sesuai dengan arsitektur jaringan yang telah diuji sebelumnya.

Tabel 3. Hasil Pelatihan Jaringan Syaraf Tiruan

Data Tulisan Tangan Orang ke-	Keluaran	E
1	0,99552831	9,99801E-06
2	0,99631147	6,80263E-06
3	0,99615998	7,37288E-06
4	0,99577645	8,91919E-06
5	0,99580301	8,80736E-06
6	0,99576086	8,98515E-06
7	0,99562138	9,58616E-06
8	0,99561736	9,60377E-06
9	0,99599469	8,02125E-06
10	0,99618233	7,2873E-06

Seperti yang ditunjukkan pada Tabel 3, nilai kesalahan (E) yang dihasilkan dari data pelatihan sudah lebih kecil dibandingkan dengan batas kesalahan, sehingga proses pelatihan dihentikan dan dapat dilakukan proses pengujian dengan bobot yang didapatkan dari hasil pelatihan tersebut.

B. Hasil Pengujian

Pada pengujian ini, dilihat tingkat akurasi aplikasi dalam melakukan pengenalan tulisan tangan. Pengujian dilakukan dengan:

1. Memasukkan citra tulisan tangan yang akan diuji dan memilih pemilik dari tulisan tangan yang akan diuji keasliannya sesuai yang ada pada data di *database*.
2. Sistem akan melakukan proses pengenalan tulisan tangan dimulai dari image processing, algoritma GHT, dan *feed-forward neural network*.
3. Mencatat hasil uji coba yang dilakukan ke dalam tabel, seperti pada Tabel 4.

$$\text{Akurasi} = \frac{\text{jumlah pengenalan benar}}{\text{total citra uji coba}} \times 100\% \quad (13)$$

Dari persamaan di atas dan data uji coba yang diberikan pada Tabel 3, maka persentase keberhasilannya adalah

$$\text{Akurasi} = \frac{16}{20} \times 100\% = 80\%$$

Tabel 4. Hasil Pengujian Pengenalan Tulisan Tangan

Uji Coba ke -	Tulisan Tangan 1	Tulisan Tangan 2	Keluaran	Hasil
1	TT_3	TT_1	0.99597653	Cocok
2	TT_12	TT_11	0.99602671	Cocok
3	TT_27	TT_21	0.49510268	Tidak cocok
4	TT_33	TT_31	0.99594956	Cocok
5	TT_42	TT_41	0.99589848	Cocok
6	TT_56	TT_51	0.99584048	Cocok
7	TT_64	TT_61	0.99576848	Cocok
8	TT_78	TT_71	0.49497389	Tidak cocok
9	TT_89	TT_81	0.99587663	Cocok
10	TT_100	TT_91	0.99591680	Cocok
11	TT_6	TT_70	0.99563015	Cocok
12	TT_12	TT_65	0.39472854	Tidak cocok
13	TT_13	TT_8	0.35531605	Tidak cocok
14	TT_14	TT_8	0.48539911	Tidak cocok
15	TT_15	TT_4	0.35516635	Tidak cocok
16	TT_16	TT_6	0.69602320	Cocok
17	TT_11	TT_5	0.46524995	Tidak cocok
18	TT_13	TT_3	0.48493201	Tidak cocok
19	TT_14	TT_7	0.79563189	Cocok
20	TT_15	TT_10	0.29539982	Tidak cocok

Pada Tabel 4 di atas, Tulisan Tangan 1 merupakan tulisan tangan yang akan diuji, Tulisan Tangan 2 merupakan tulisan tangan yang ada pada *database* dan sudah dilatih. Keluaran merupakan nilai keluaran dari jaringan syaraf pada tahap pengujian, dan hasil merupakan hasil apakah Tulisan Tangan 1 sesuai atau cocok dengan Tulisan Tangan 2 atau tidak.

C. Pembahasan

Berdasarkan pada pengujian yang telah dilakukan, dengan hasil yang telah diberikan sebelumnya, dapat ditentukan persentase akurasi dengan penghitungan seperti di bawah ini.

Persentase keberhasilan dari aplikasi pengenalan tulisan tangan *offline* menggunakan algoritma *Generalized Hough Transform* dan *Backpropagation* adalah 80%.

Persentase akurasi yang dihasilkan sudah cukup baik. Menurut hasil pelatihan jaringan syaraf tiruan pada Tabel 2, dapat disimpulkan bahwa proses pelatihan dengan arsitektur JST yang ada sudah baik dikarenakan nilai kesalahan sudah sangat kecil. Namun, proses pelatihan JST dapat ditingkatkan lagi dengan melakukan uji coba jumlah unit pada *hidden layer* sebesar 11 sampai dengan jumlah unit masukkan dikurangi 1.

VI. SIMPULAN DAN SARAN

Sistem pengenalan tulisan tangan dengan menggabungkan algoritma *Generalized Hough Transform* dan *Backpropagation* telah berhasil dilakukan. Algoritma *Generalized Hough Transform* digunakan untuk memperoleh ciri dari citra tulisan tangan pada tahap ekstraksi ciri. *Backpropagation* digunakan untuk melatih jaringan syaraf tiruan dalam mengenali citra tulisan tangan.

Sistem telah diuji dengan menggunakan model arsitektur jaringan berupa 40 buah masukkan, 1 *hidden layer* (dengan 4 buah unit), 1 keluaran, laju pembelajaran 0.5 dan batas kesalahan 0.00001. Fungsi aktivasi yang digunakan pada layar tersembunyi dan layar keluaran adalah fungsi sigmoid biner. Data yang digunakan berjumlah 100 dari 10 orang yang berbeda. Dari hasil uji coba yang dilakukan, didapatkan bahwa penggabungan algoritma *Generalized Hough Transform* dan *Backpropagation* dalam pengenalan tulisan tangan *offline* menghasilkan nilai persentase akurasi sebesar 80%.

Adapun saran untuk penelitian selanjutnya yaitu perlu dilakukan penelitian dengan menggunakan algoritma atau metode lain untuk proses pelatihan jaringan syaraf maupun ekstraksi ciri. Sistem yang dikembangkan pada kesempatan kali ini belum dapat menganalisis citra tulisan tangan yang berbentuk sambung. Oleh karena itu, penelitian selanjutnya diharapkan dapat menganalisis tulisan tangan yang berbentuk latin ataupun huruf lain seperti aksara Arab, Mandarin, Jepang, dan lain sebagainya.

DAFTAR PUSTAKA

- [1] Althaf, M. K. M. dan M. B. Begum, "Handwritten Characters Pattern Recognition Using Neural Networks," International Conference on Computing and Control Engineering, vol. 4, no. 7, 2012.

- [2] Liu, Zhi-Qiang, Jin-Hai Cai, dan Richard Buse, "Introduction" di dalam *Handwriting Recognition: Soft Computing and Probabilistic Approach*, edisi 1, Berlin: Springer, 2003, bab I.
- [3] Suryani, Dewi, Patrick Doesch, dan Hermann Ney, "On the Benefits of Convolutional Neural Network Combinations in Offline Handwriting Recognition," International Conference on Frontiers in Handwriting Recognition (ICHFR), Shenzhen, hal. 193-198, 2016.
- [4] Kumar, Ashok dan P. K. Bhatia, "Offline Handwritten Character Recognition Using Improved Back-Propagation Algorithm," International Journal of Advances in Engineering Sciences, vol. 3, no. 3, 2013.
- [5] Ballard, D. H., "Generalizing The Hough Transform to Detect Arbitrary Shapes," Pattern Recognition, vol. 13, no. 2, hal. 111-122, 1981.
- [6] Touj, S., Amara, N. B., & Amiri, H., "Generalized Hough Transform for Arabic Printed Optical Character Recognition," The International Arab Journal of Information Technology, vol. 2, no. 4, hal. 326-333, 2005.
- [7] Qian, Lisi dan Bin Fang, "Optimized Generalized Hough Transform for Road Making Recognition Application," SPAC, Shenzhen, hal. 361-367, 2017.
- [8] Siang, J. J., "Jaringan Syaraf Tiruan & Pemrogramannya Menggunakan Matlab," Yogyakarta: Penerbit Andi, 2009.
- [9] Kala, R., H. Vazirani, A. Shukla, dan R. Tiwari, "Offline Handwriting Recognition using Genetic Algorithm," IJCSI, vol. 7, no. 1, hal. 16-25, March 2010.
- [10] Chakravarthy, A. S. N., P. V. K. Raja, dan P. S. Avadhani, "Handwritten Text Image Authentication Using Back Propagation," IJNSA, vol. 3, no. 5, hal 121-130, Sep 2011.
- [11] Nixon, Mark dan A. Aguado, "Introduction" di dalam *Feature Extraction and Image Processing*, edisi 2, United Kingdom: Elsevier, 2008, bab I.
- [12] Ballard, D. H. dan C. M. Brown, "The Hough Method for Curve Detection" di dalam Computer Vision, edisi 1, New Jersey: Prentice-Hall, hal. 128-131, 1982, bab IV.
- [13] Russel, Stuart J. dan Peter Norvig, "Artificial Neural Network" di dalam *Artificial Intelligence: A Modern Approach*, edisi 3, New Jersey: Prentice-Hall, hal. 733, 2010, bab XVIII.
- [14] Lagudu, S. dan CH. V. Sarma, "Hand Writing Recognition Using Hybrid Particle Swarm Optimization & Back Propagation Algorithm," IJAIEM, vol. 2, no. 1, hal. 75-81, 2013.



UMN