

Aplikasi Grafologi dari Huruf “t” Menggunakan Jaringan Syaraf Tiruan

Iwan Awaludin¹, Aulia Khairunisa²

Abstract— Graphology is a branch of science which classifies human personality from handwriting. Graphologists observe the patterns of handwriting and compare it with personality class database. Computers can be trained to do the same procedure of human personality classification based on handwriting. The procedure is to perform digital image processing that extracts features from handwriting images. The features will become input for Artificial Neural Network. Neural networks that are already configured with a number of hidden layers, the number of neurons, activation function, and the particular learning algorithm will be able to recognize certain classes of human handwriting, thus his personality. Tested configurations include: changing the number of neurons in the hidden layer of eight to twelve, binary image resizing, changing the activation function, and also changing the learning algorithm. Results of simulation and analysis are also provided.

Intisari— Grafologi adalah cabang ilmu yang mengklasifikasikan kepribadian manusia berdasarkan tulisan tangan. Cara grafologi mengenal kepribadian manusia adalah dengan mengamati pola-pola tulisan tangan dan membandingkannya dengan basis data kelas kepribadian. Komputer bisa dilatih untuk mengenali pola yang sama dan mengklasifikasikan penulis berdasarkan kelas yang sudah ada. Caranya adalah dengan melakukan pemrosesan citra digital untuk mendapatkan fitur citra yang siap menjadi *input* bagi Jaringan Syaraf Tiruan. Jaringan Syaraf Tiruan yang sudah dikonfigurasi dengan jumlah *hidden layer*, jumlah neuron, fungsi aktivasi, dan algoritme belajar tertentu ini dapat mengenali dengan pasti kelas-kelas tulisan tangan manusia yang berelasi dengan kepribadiannya. Konfigurasi yang diujicobakan diantaranya adalah dengan mengubah jumlah neuron di *hidden layer* dari delapan sampai dua belas buah, mengubah besar citra biner, mengubah fungsi aktivasi, dan juga mengubah algoritme belajar. Hasil simulasi dan analisisnya juga diberikan.

Kata Kunci— grafologi, jaringan syaraf tiruan, levenberg marquadt.

I. PENDAHULUAN

Grafologi adalah cabang ilmu psikologi yang mencoba mencari hubungan antara tulisan tangan dengan kepribadian manusia. Hubungan ini dicari dari bentuk dan pola tulisan tangan. Tulisan tangan dipercaya sebagai representasi perintah otak yang melalui penelitian statistik telah dipetakan menjadi beberapa kategori kepribadian manusia[1].

¹Dosen, Jurusan Teknik Komputer dan Informatika POLBAN. Jl. Geger kalong Hilir Bandung 40164 (telp: 022-2013789; fax: 022-2013889; e-mail: awaludin@jtk.polban.ac.id)

²Alumni, Jurusan Teknik Informatika Universitas Komputer Indonesia, Jln. Dipati Ukur 112 Bandung (e-mail: aulikhairunisa1@gmail.com)

Biasanya pekerjaan menganalisis tulisan tangan ini dilakukan oleh grafologis, yaitu seorang psikolog yang mendapatkan pelatihan untuk membaca tulisan tangan. Pekerjaan ini dilakukan dengan mengambil beberapa contoh tulisan tangan seperti dari sebuah dokumen, dari huruf, atau dari tanda tangan. Dari contoh ini akan dicari pola untuk dibandingkan dengan data yang ada.

Proses pengambilan contoh dan pencarian pola sebenarnya bisa digantikan oleh komputer. Pengambilan contoh tulisan tangan dalam bentuk gambar digital bisa dilakukan dengan menggunakan pemindai atau juga kamera digital. Hasil gambar digital ini kemudian diproses dengan pengolahan citra digital untuk dicari fitur-fitur pembedanya. Berdasarkan fitur pembeda inilah proses pencarian pola dilakukan untuk mendapatkan kemiripan dengan pola yang sudah ada dalam basis data.

Salah satu metode untuk mencari pola yaitu memakai Jaringan Syaraf Tiruan (JST). JST sering dipakai karena pola di tulisan tangan manusia cukup sulit untuk dicari menggunakan metode analitis. Pola tulisan tangan manusia tidak linier sehingga sulit dicari misalnya dengan menurunkan persamaan matematikanya.

Membentuk JST yang mampu mengenali pola tulisan tangan manusia bukan pula hal yang mudah. Perlu ada proses pengolahan citra dan pertimbangan fitur apa saja yang akan menjadi pembeda dari tulisan tangan manusia sehingga nanti bisa diklasifikasi kepada sifat-sifat manusia yang ada dalam basis data.

Dalam makalah ini akan disajikan investigasi proses pengolahan citra tulisan tangan dan pencarian pola dengan memakai JST. Sebenarnya ada beberapa huruf yang biasa digunakan untuk mengenali sifat manusia seperti “t”, “d”, dan “i”. Tetapi dalam makalah ini tulisan tangan yang dipakai hanya mengenali pola dari huruf “t”. Hal ini dilakukan untuk menguji kemampuan awal JST sebagai pengklasifikasi sifat manusia berdasarkan tulisan tangan. Setelah potensi JST dapat dikenali dengan pasti maka penelitian lanjut dengan huruf yang lain dapat dilakukan.

II. GRAFOLOGI

Ada banyak tulisan tangan yang bisa dipakai untuk mengenali kepribadian penulisnya. Di antara tulisan tangan itu ada dalam bentuk kalimat, satu huruf, atau tanda tangan. Pembeda dari pendekatan yang diambil dari setiap peneliti adalah pemilihan jenis tulisan dan fitur yang akan dipakai untuk mengenali pola.

Dalam [2], Widiastuti dkk menggunakan tanda tangan sebagai contoh tulisan yang akan diklasifikasikan. Jenis JST yang digunakan adalah *Multilayer Perceptron* dengan *feedforward* dan belajar dengan *backpropagation*. Tidak

dijelaskan berapa buah *hidden layer*, jumlah neuron, dan jenis fungsi aktivasi yang digunakan.

Peneliti lainnya dalam [3] menggunakan tanda tangan dalam melakukan klasifikasi. Dari kata ini fitur yang dilihat adalah arah dan kemiringan tulisan. Untuk melakukan klasifikasi dijelaskan proses yang dilalui oleh tulisan tangan. Peneliti ini melakukan pemindaian terhadap tanda tangan untuk mendapatkan citra digital. Citra ini kemudian diatur ukurannya sesuai dengan *template*. Setelah itu citra dirampingkan untuk mendapatkan postur yang lebih kecil. Postur tulisan yang sudah kecil lalu dipotong dan diatur posisinya. Hal ini dibutuhkan agar tulisan bisa diketahui kemiringannya. Jenis JST yang digunakan adalah *backpropagation* dengan fungsi aktivasi yang tidak disebutkan.

Selanjutnya dalam [4] digunakan tulisan beberapa kata sebagai pembeda kepribadian manusia. Pengolahan citra yang dikerjakan lebih banyak, misalnya menggunakan analisis histogram dan segmentasi. Tidak disebutkan konfigurasi JST yang digunakan. Jumlah kelas yang dicari memang disebutkan tetapi tidak dijelaskan berapa jumlah data yang menjadi bahan latihan dan pengujian.

III. KONFIGURASI PERCOBAAN

Dalam makalah ini akan dicari JST yang bisa digunakan untuk membuat klasifikasi kepribadian manusia berdasarkan tulisan tangan. JST ini akan digunakan oleh dua aktor yaitu pengembang dan pengguna. Pengembang akan membuat JST dan melatihnya agar saat pengguna memakainya, JST yang sudah dilatih dapat dengan akurat mengklasifikasi kepribadian manusia.

Pengembang memerlukan data tulisan tangan dan kepribadian yang berkaitan dengannya. Data tulisan tangan ini bisa didapatkan dengan memindai tulisan-tulisan yang ada dalam buku psikologi, menambahkannya dengan *noise*, dan kemudian dipakai untuk melatih JST.

Citra hasil pemindaian biasanya berupa gambar tulisan dalam format warna RGB atau *grayscale*. Huruf yang menjadi acuan untuk klasifikasi harus dipisahkan dengan tulisan lainnya. Cara pemisahan manual dipilih untuk mempermudah proses.

Huruf yang dipilih sebagai acuan klasifikasi adalah huruf “t”. Berdasarkan [1] terdapat 11 jenis kepribadian manusia dari cara penulisan huruf “t”. Untuk setiap jenis kepribadian diperlukan 20 buah contoh tulisan huruf “t”. Karena itu contoh tulisan yang digunakan adalah 220 buah.

Contoh tulisan ini nanti akan dipisahkan untuk fase pelatihan dan fase pengujian JST. Biasanya jumlah contoh tulisan yang dijadikan bahan pelatihan sebanyak 80% sedangkan untuk pengujian sebanyak 20%. Karena itu jumlah contoh tulisan yang dijadikan bahan pelatihan adalah 176 buah dan bahan pengujian adalah 44 buah. JST akan dilatih dan diuji untuk dicari konfigurasi yang memberikan hasil terbaik untuk pelatihan dan pengujian.

Sebelum citra dijadikan bahan untuk pelatihan dan pengujian maka diperlukan pra-pengolahan terhadap citra tersebut. Pra-pengolahan yang dilakukan dalam makalah ini diilustrasikan dalam Gbr. 1.

Proses mengubah citra menjadi grayscale dapat diterangkan dalam algoritme berikut. Citra berwarna biasanya memiliki dua dimensi. Dimensi ini dinyatakan dalam bentuk jumlah *pixel* dalam baris dikalikan dengan jumlah *pixel* dalam kolom. Artinya setiap citra memiliki jumlah *pixel* sebanyak hasil perkalian jumlah *pixel* di baris dan di kolom.

Setiap *pixel* mengandung informasi komponen warna merah (*red*), hijau (*green*), dan biru (*blue*). Informasi *pixel* dan komponen warna disimpan dalam bentuk matriks. Untuk mendapatkan *grayscale* dari gambar RGB, diperlukan suatu proses yang mengambil informasi komponen warna. Misalnya yang dilakukan di [5], yaitu dari informasi RGB dikonversi ke ruang warna YPQ. Kemudian perbedaan warna antar *pixel* diambil menggunakan pasangan Gaussian. Sumbu dominan dari kontras kromatik dihitung untuk selanjutnya dikombinasikan dengan nilai luminasi dari data kromatik dan akhirnya kalibrasi terhadap citra *grayscale*.



Gbr. 1 Pra-pengolahan citra sebelum menjadi *input* ke JST.

Konversi dari nilai RGB ke nilai YPQ diberikan seperti dalam (1).

$$\begin{bmatrix} Y_i \\ P_i \\ Q_i \end{bmatrix} = \begin{bmatrix} 0,2989 & 0,5870 & 0,1140 \\ 0,5000 & 0,5000 & 1,000 \\ 1,000 & -1,000 & 0 \end{bmatrix} \begin{bmatrix} R_i \\ G_i \\ B_i \end{bmatrix} \quad (1)$$

Dari (1) bisa dihitung *hue* dan *saturation* dari gambar berdasarkan (2).

$$H_i = \frac{1}{\pi} \tan^{-1} \left(\frac{Q_i}{P_i} \right) \quad (2)$$

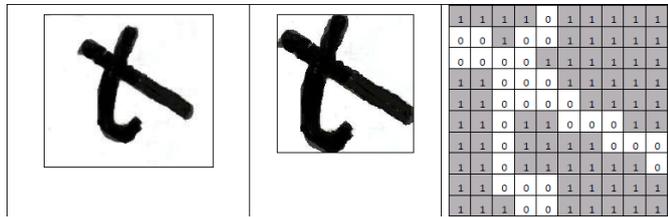
$$S_i = \sqrt{P_i^2 + Q_i^2} \quad (3)$$

Detail dari penurunan persamaan dan implementasinya bisa dirujuk pada [5].

Setelah mendapatkan citra dalam format warna *grayscale*, selanjutnya dengan menggunakan filter batas citra akan diubah menjadi biner. Artinya hanya garis tegas saja yang tetap membentuk gambar sedangkan citra lain yang tidak tegas akan menjadi warna putih. Filter batas cukup mudah diterapkan karena hanya dengan menetapkan batas angka yang akan diubah menjadi 1 dan 0 bisa didapatkan gambar biner.

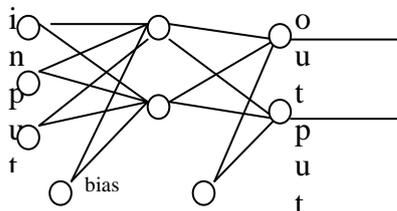
Untuk keperluan klasifikasi yang lain diperlukan informasi dari mana penulis memulai tulisan, berapa besar tekanan, dan arah penulisan. Informasi ini bisa didapat dari berapa besar

kumpulan *pixel* yang berwarna hitam. Tetapi untuk makalah ini, informasi yang diperlukan adalah bentuk huruf “t” nya sendiri dengan tidak memperhatikan darimana huruf itu mulai ditulis. Karena itu, tidak diperlukan data ketebalan huruf dimaksud sehingga selanjutnya bisa dilakukan proses perampangan terhadap citra. Gbr. 2 menunjukkan hasil pemrosesan citra huruf t menjadi matriks biner.



Gbr. 2 Hasil konversi huruf t ke matriks biner

Hasil akhir dari citra yang sudah diolah adalah berupa matriks yang besarnya bisa diatur dengan isi biner. Matriks ini yang akan menjadi *input* untuk JST. Arsitektur umum JST diperlihatkan pada Gbr. 3.



Gbr. 3 Konfigurasi umum JST.

Untuk konfigurasi JST ada beberapa hal penting yang harus diperhatikan. Dari sisi *input* dan *output* adalah jumlah neuronnya. Jumlah neuron di sisi *output* ditentukan oleh jumlah kelas kepribadian manusia yang akan dicari. Menurut buku panduan grafologi, dari huruf “t” dapat dipisahkan 11 kelas kepribadian manusia. Artinya jumlah neuron di sisi *output* JST yang dibangun ada 11 buah.

Selanjutnya di sisi *input* JST. Ada dua kemungkinan jumlah neuron di sisi *input*. Kemungkinan pertama ditentukan oleh jumlah fitur yang akan dipakai untuk membedakan tulisan tangan. Kemungkinan kedua ditentukan oleh jumlah *pixel* citra biner hasil pra-pengolahan.

Untuk kemungkinan pertama ada proses tambahan yang harus dilalui oleh citra biner. Proses ini adalah mengambil fitur pembeda. Misalnya perbandingan panjang dan lebar komponen pembentuk huruf, titik potong panjang dan lebar huruf, dan kemiringan dari lebar huruf. Terlihat bahwa jumlah fitur ini cukup kecil sehingga JST yang dibangunpun tidak akan besar. Isi dari setiap fitur dapat direpresentasikan dalam nilai yang menjadi nilai maksimum dan minimum dari input JST. Misalnya saja kemiringan dari lebar huruf atau garis horizontal dari huruf “t” ada dari -30° sampai 30° bisa dinyatakan sebagai nilai -1 sampai 1 di input JST. Sayangnya memang kemungkinan pertama ini memerlukan pengolahan lanjut dari citra biner.

Sedangkan untuk kemungkinan kedua hasil citra biner bisa langsung digunakan untuk *input* JST. Perlu diperhatikan bahwa *input* JST berupa matriks kolom sedangkan citra biner

berupa matriks dua dimensi. Perlu ada konversi terlebih dahulu dari matriks dua dimensi ke matriks satu dimensi. Prosedurnya cukup mudah misalnya dengan memindahkan kolom kedua sampai ke-n menjadi baris baru di kolom pertama.

Dimensi matriks *input* JST tergantung dari dimensi citra biner. Sebelumnya citra biner yang dihasilkan dibuat berdimensi sama besar baris dan kolom. Dengan demikian didapat jumlah baris *input* JST sebesar baris x kolom dari citra biner.

Memang bila diperhatikan, jumlah baris ini akan semakin besar bila dimensi citra biner semakin besar. Bahkan pembesaran ini bersifat kuadratik. Ada keraguan nanti JST tidak bisa beroperasi karena besarnya dimensi *input*. Meskipun demikian kemungkinan kedua ini yang dipilih untuk diinvestigasi karena kemudahan prosesnya.

Selanjutnya setelah *input* dan *output* JST ditentukan, maka perlu ditetapkan berapa banyak *hidden layer* dan neuron yang ada di dalamnya. Selain itu perlu ditentukan juga fungsi aktivasi. Biasanya jumlah *hidden layer* yang kecil sudah memberikan hasil yang memuaskan untuk fungsi nonlinier sederhana. Karena itu ditetapkan jumlah *hidden layer* untuk JST yang dibangun adalah satu.

Jumlah neuron juga *trivial*. Tidak ada panduan khusus tentang berapa jumlah neuron yang tepat untuk satu kasus, apalagi panduan jumlah neuron yang berlaku umum. Penentuan jumlah neuron biasa dilakukan secara *trial* dan *error*. Ditetapkan jumlah neuron yang akan dicoba bervariasi dari delapan sampai dua belas neuron.

Fungsi aktivasi biasanya dikaitkan dengan tujuan JST. Secara umum ada dua tujuan yaitu membuat klasifikasi atau membuat regresi. Klasifikasi yaitu memisahkan *output* menjadi beberapa kelas, sedangkan regresi membuat JST mengeluarkan nilai yang kontinu dalam selang tertentu.

Jenis aktivasi yang digunakan berbeda. Untuk keperluan klasifikasi biasa digunakan fungsi aktivasi yang serupa dengan sigmoid. Fungsi aktivasi seperti ini biasanya akan menarik *output* mendekati nilai maksimum atau nilai minimum. Untuk keperluan regresi biasa digunakan fungsi aktivasi linier. Fungsi ini tidak menarik *output* menuju nilai maksimum atau minimum melainkan menempatkannya di titik yang ada dalam garis linier. Meskipun tujuan penggunaannya berbeda, ada kalanya fungsi aktivasi sigmoid bisa digunakan untuk keperluan regresi, atau juga sebaliknya, fungsi aktivasi linier digunakan untuk keperluan klasifikasi. Untuk investigasi yang dilakukan, dipilih fungsi aktivasi linier di *hidden layer* dan sigmoid di *output layer*.

Setelah ini perlu ditentukan *bias*. *Bias* adalah bagian dari *input* ke *hidden layer* dan *output layer* yang tidak berhubungan dengan *input* dari layer sebelumnya. Misalnya di *hidden layer bias* tidak berhubungan dengan *input* dari citra biner. *Bias* berfungsi seperti konstanta di persamaan linier yang tidak berhubungan dengan peubah. Selain menggeser nilai, *bias* juga berfungsi untuk mengakomodasi nilai lain yang berpengaruh terhadap *output* tetapi tidak dinyatakan sebagai *input*. Pada makalah ini, *bias* di *hidden layer* dan *output layer* diaktifkan.

Bagian di atas berkaitan dengan nilai yang tidak berubah sepanjang pelatihan JST. Nilai *input* dan target *output* akan diganti-ganti sepanjang pelatihan tetapi dengan nilai yang sudah ditentukan berdasarkan citra biner dan kelas kepribadian manusia. Ada bagian JST yang akan berubah selama pelatihan yaitu nilai bobot JST.

Bobot adalah penghubung antar neuron. Fungsinya seperti dendrit di jaringan syaraf manusia, yaitu membawa informasi dari satu neuron ke neuron lainnya. Pada awalnya bobot diberi nilai acak. Kemudian berdasarkan proses belajar JST bobot akan berubah

Cara belajar JST adalah seperti berikut. Pertama data *input* dan target *output* sudah dimiliki. Dalam investigasi ini ada sebanyak 176 pasangan *input* dan target *output* yang dipakai untuk pelatihan. Data ini akan dimasukkan ke dalam JST per pasangan. Setiap kolom *input* dimasukkan ke *input layer*, kemudian dibawa informasinya ke *hidden layer* melalui bobot yang menghubungkan *input layer* dan *hidden layer*. Setiap neuron di *hidden layer* akan menerima perkalian setiap *input* terhadap bobot dan kemudian dijumlahkan seperti pada (4).

$$S_i = \sum_{j=1}^k w_{ji} \quad (4)$$

S_i merupakan penjumlahan dari hasil perkalian bobot w dengan *input* i . Penjumlahan dari hasil perkalian bobot dengan *input* yang bersesuaian akan dimasukkan ke dalam (5) yaitu fungsi aktivasi.

$$o_i = f(S_i) \quad (5)$$

Output dari setiap neuron seperti dari (5) akan menjadi *input* bagi *layer* berikutnya. Persamaan yang digunakan tetap saja, yang berubah hanya indeks yang menunjukkan tempat *layer* yang bersesuaian.

Sesampainya di bagian *output*, hasil keluaran akan dibandingkan dengan target. *Output* ini akan berbeda dengan target, yang menunjukkan adanya *error*. Informasi *error* ini disimpan karena proses *feedforward* belum selesai. Proses *feedforward* akan memasukkan pasangan *input* dan *output* target satu per satu yaitu sampai 176 pasang. *Error* dari setiap pasang dicatat untuk kemudian dihitung *root mean square error*-nya (RMSE). Selesaiannya semua pasang dimasukkan ke dalam *feedforward* menyatakan satu *epoch*. RMSE dari satu *epoch* akan digunakan untuk memperbarui nilai bobot.

Cara memperbarui nilai bobot ada banyak. Yang terkenal dengan sebutan *back propagation*, misalnya Levenberg Marquadt (LM), *Gradient Descent* (GD), *Gradient Descent with Momentum* (GDM), dan masih banyak lagi. Intinya adalah mengirimkan balik informasi RMSE ini untuk memperbarui nilai bobot, terutama sekali nilai bobot yang berada di *layer* sebelum *hidden layer* karena tidak ada target yang menjadi acuan berapa besar *error* yang terjadi.

Pengiriman balik nilai *error* ini mempertimbangkan persamaan yang digunakan untuk memperbarui bobot. Pada LM ada turunan orde satu yang digunakan untuk memperbarui nilai bobot. Cara ini cukup efektif tetapi juga bisa membutuhkan waktu sangat lama. Besarnya matriks *input*,

seperti yang telah disebutkan pada awal bab, juga menyebabkan lambatnya nilai *error* mengecil.

Sedangkan pada GD diperlukan perhitungan gradien fungsi pada titik berdasarkan *error* yang didapat. Gradien ini menentukan jumlah langkah meloncat untuk mempercepat proses memperbarui bobot.

Kedua algoritme di atas memiliki kelemahan yaitu kadang terjebak pada *local minima*. Hal ini akan menyebabkan perhitungan berhenti. Karena itu pada algoritme GD ditambahkan lagi momentum yang dapat menendang perhitungan bila terjebak pada *local minima*. Ketiga algoritme ini akan diuji coba semua.

IV. HASIL DAN ANALISIS

Simulasi dilakukan menggunakan konfigurasi yang telah dijelaskan pada bab sebelumnya. Berikut ini perilaku yang dapat diambil dari simulasi.

Simulasi pertama dilakukan dengan delapan buah neuron di *hidden layer* dan LM sebagai algoritme belajar untuk memperbarui bobot. Simulasi berjalan lambat dan memerlukan belasan ribu *epoch* untuk mencapai RMSE yang cukup kecil. Ketika dimulai, secara rata-rata RMSE berada pada kisaran 1 kemudian berkurang sedikit demi sedikit sampai pada nilai yang diinginkan. Hasil pelatihan memberikan *output* yang membawa kisaran 0 – 0,3 dan 0,73 sampai 1. Untuk mendapatkan hasil klasifikasi yang bagus maka perlu diberikan nilai batas. Saat nilai batas di atas 0,75 maka ada beberapa kelas yang salah klasifikasinya. Baru bila diberi batas nilai 0,7 hasil klasifikasi memberikan akurasi 100% untuk semua kelas dan keseluruhan 176 data. Sayangnya ketika bobot yang didapat ini dipakai untuk data pengujian hasilnya menurun cukup besar. Akurasi 65% didapatkan secara keseluruhan.

Sebenarnya diharapkan akurasi data pengujian bisa ditingkatkan, misalnya dengan memperbesar dimensi citra biner. Tetapi saat diujicobakan dimensi citra biner tidak bisa dibuat lebih besar dari 15 x 15. Artinya terdapat 225 buah *input* maksimum yang bisa dijalankan oleh komputer. Pesan yang ditangkap dari program yang digunakan adalah kehabisan memori. Memang menurut beberapa hasil percobaan, algoritme LM ini sangat boros menggunakan memori komputer.

Selanjutnya simulasi dilakukan dengan menggunakan GD dan GDM. Ternyata nilai awal RMSE rata-rata sama seperti LM. Hanya saja untuk *epoch* berikutnya RMSE langsung ke nilai setengah dari nilai awal. Setelah mencapai nilai setengah ini tidak ada lagi perbaikan dari GD dan GDM bahkan sampai jumlah *epoch* maksimum tercapai.

Padahal ada kelebihan dari kedua algoritme ini. Keduanya sanggup mengatasi kekurangan memori yang ada di LM. Tidak ada masalah sampai *epoch* maksimum tercapai, tidak ada peringatan memori habis.

Setelah mendapatkan algoritme belajar yang memberikan hasil yang cukup baik, simulasi dilakukan dengan konfigurasi yang lain. Konfigurasi yang digunakan adalah mengubah jumlah neuron di *hidden layer*. Percobaan pertama menggunakan delapan neuron diubah satu per satu sampai dua

belas neuron. Berdasarkan hasil simulasi tidak ada perubahan signifikan terhadap hasil sehingga disimpulkan dengan memakai delapan buah neuron dirasa mencukupi untuk keperluan klasifikasi kepribadian manusia ini.

Kemudian konfigurasi diubah lagi dengan memakai fungsi aktivasi yang berbeda. Menurut pengamatan tidak ada perubahan akurasi, jumlah *epoch*, dan kecepatan proses dengan perubahan fungsi aktivasi. Artinya fungsi sigmoid dan fungsi linier dapat digunakan untuk keperluan klasifikasi ini.

V. KESIMPULAN

Berdasarkan hasil simulasi bisa dilaporkan bahwa JST bisa digunakan untuk membuat klasifikasi kepribadian manusia berdasarkan huruf “t”. Klasifikasi ini dilakukan dengan konfigurasi JST *Multi Layer Perceptron*, satu buah *hidden layer* dengan delapan buah neuron. Konfigurasi seperti ini memberikan akurasi 100% untuk 176 data pelatihan dan 65% untuk data pengujian.

Perubahan jumlah neuron di *hidden layer* dan fungsi aktivasi tidak memberikan perubahan signifikan terhadap hasil simulasi. Algoritme GD dan GDM juga tidak bisa dicari

hasilnya karena tidak bisa konvergen ke satu nilai meskipun jumlah *epoch* maksimum sudah tercapai.

Penelitian selanjutnya adalah dengan menggunakan fitur perbandingan tinggi, kemiringan garis, dan perpotongan garis huruf “t”. Fitur ini memungkinkan jumlah *input* menjadi lebih kecil sehingga bisa menggunakan data citra yang lebih besar. Penelitian juga bisa dilakukan pada contoh huruf lain seperti “d” dan “i”.

REFERENSI

- [1] B. Lutfianto, *Analisis Tulisan Tangan: Grapho for Succ*. Gramedia Pustaka Utama, 2011.
- [2] F. Widiastuti, W. Kaswidjanti, and H. C. Rustamaji, “JARINGAN SARAF TIRUAN BACKPROPAGATION UNTUK APLIKASI PENGENALAN TANDA TANGAN,” *Telematika*, vol. 11, no. 1, 2015.
- [3] S. Dang and M. K. Mahesh, “Handwriting Analysis of Human Behavior Based on Neural Network,” *Int. J. Adv. Res. Comput. Sci. Softw. Eng.*, vol. 4, no. 9, Sep. 2014.
- [4] H. N. Champa and K. R. AnandaKumar, “Artificial Neural Network for Human Behavior Prediction through Handwriting Analysis,” *Int. J. Comput. Appl. IJCA*, vol. 2, no. 2, pp. 36–41, 2010.
- [5] M. Grundland and N. A. Dodgson, “The decolorize algorithm for contrast enhancing, color to grayscale conversion,” University of Cambridge, Technical Report UCAM-CL-TR-649, 2005.