
PENERAPAN METODE BACKPROPAGATION UNTUK IDENTIFIKASI HURUF HIJAIYAH TULISAN TANGAN

Ariesta Damayanti¹⁾, Pujiatus Syahara²⁾

^{1,2}Teknik Informatika, STMIK AKAKOM Yogyakarta

e-mail: iest_ayanthi@akakom.ac.id¹⁾, pujiatus.syahara@gmail.com²⁾

ABSTRACT

Hijaiyah letters are Arabic spelling letters that are the original language of the Qur'an. Just like other types of letters, the hijaiyah has certain shapes and characteristics that will form a certain pattern. By using the concept of artificial neural networks, can dibangun a system that can recognize the pattern by doing the previous training. One of the most commonly used meotodes in artificial neural network paradigms is the crawling or backpropagation buffer. This hijaiyah letters identification system is built using the handwritten hijaiyah image data of 150 images. The feature or feature taken from the image is the binary value of the letter pattern and the number of objects contained in the letters. Prior to the feature extraction process, the image first passes the preprocessing stage consisting of color binerization, object widening, cropping, and resizing. The result obtained by backpropagation method is the system is able to recognize handwriting hijaiyah pattern well. All training data have been correctly identified, while as many as 150 test data can be identified as 77 letters with an accuracy of 51.33%. This accuracy value is obtained with the architectural arrangement of the number of hidden layer neurons = 60, minimum error = 0.001 and maximum iteration = 10000.

Keywords: *backpropagation, biner, hijaiyah, , pattern, preprocessing*

ABSTRAK

Huruf hijaiyah adalah huruf-huruf ejaan bahasa Arab yang merupakan bahasa asli Al-Qur'an. Sama seperti jenis huruf yang lain, huruf hijaiyah memiliki bentuk dan karakteristik tertentu yang akan membentuk suatu pola tertentu. Dengan menggunakan konsep jaringan syaraf tiruan, dapat dibangun suatu sistem yang dapat mengenali pola tersebut dengan melakukan pelatihan sebelumnya. Salah satu meotode yang sering digunakan dalam paradigma jaringan syaraf tiruan adalah perambantan galat mundur atau backpropagation. Sistem identifikasi huruf hijaiyah ini dibangun dengan menggunakan data citra huruf hijaiyah yang ditulis tangan sebanyak 150 citra. Fitur atau ciri yang diambil dari citra adalah nilai biner dari pola huruf dan jumlah objek yang terdapat pada huruf. Sebelum dilakukan proses ekstraksi fitur, citra terlebih dahulu melewati tahap preprocessing yang terdiri dari binerisasi warna, pelebaran objek, cropping, dan resizing. Hasil yang diperoleh dengan metode backpropagation adalah sistem mampu mengenali pola huruf hijaiyah tulisan tangan dengan baik. Seluruh data latih berhasil diidentifikasi dengan benar, sedangkan sebanyak 150 data uji mampu diidentifikasi sebanyak 77 huruf dengan menghasilkan akurasi sebesar 51.33%. Nilai akurasi ini didapatkan dengan susunan arsitektur jumlah neuron hidden layer = 60, minimal error = 0.001 dan maksimal iterasi = 10000.

Kata kunci: *backpropagation, biner, hijaiyah, , pola, preprocessing*

1. PENDAHULUAN

Al-Qur'an sebagai kitab suci umat Islam mengandung ayat-ayat berbahasa Arab, dan yang harus dipahami pertama kali saat belajar membaca Al-Qur'an adalah huruf-hurufnya yang biasa kita sebut sebagai huruf Hijaiyah. Huruf Hijaiyah adalah huruf-huruf ejaan bahasa Arab sebagai bahasa asli Al-Qur'an. Seperti halnya huruf latin (alfabet) yang menyusun kata dan kalimat dalam bahasa Indonesia dan bahasa Inggris, huruf hijaiyah juga memiliki peran yang sama untuk bahasa Arab. Baik huruf latin maupun huruf hijaiyah, semua jenis huruf yang ada di seluruh dunia memiliki bentuk dan karakteristik yang berbeda-beda sehingga dapat ditentukan pola dari masing-masing jenis huruf tersebut.

Jaringan Saraf Tiruan (JST) adalah salah satu cabang ilmu dari bidang ilmu kecerdasan buatan. Pemodelan dengan JST merupakan pembelajaran dan penyesuaian dari suatu objek. Metode *backpropagation* melatih jaringan untuk mendapatkan keseimbangan antara kemampuan jaringan mengenali pola yang digunakan selama pelatihan serta kemampuan jaringan untuk memberikan respon yang benar terhadap pola masukan yang serupa (tetapi tidak sama) dengan pola yang dipakai selama pelatihan [1]. *Backpropagation* merupakan metode pembelajaran yang terawasi dan biasanya digunakan oleh model jaringan syaraf tiruan *perceptron* dengan banyak lapisan (*multi-layered perceptron*) untuk mengubah bobot-bobot yang terhubung dengan neuron-neuron yang ada pada lapisan tersembunyinya [2]. Ciri dari metode *backpropagation* adalah meminimalkan *error* pada *output* yang dihasilkan oleh jaringan. Arsitektur jaringan syaraf tiruan dengan metode *backpropagation* terdiri dari satu layer input, satu atau lebih layer tersembunyi (*hidden layer*) dan satu layer *output*. Jumlah *neuron* pada lapisan masukan sama dengan jumlah ciri atau fitur pada pola yang akan dikenali, sedangkan jumlah *neuron* pada lapisan keluaran sama dengan jumlah kelas pola.

Beberapa penelitian mengenai pengenalan pola huruf tertentu telah dilakukan sebelumnya. Dalam [3] dikembangkan aplikasi untuk membaca huruf Hijaiyah dan angka arab dengan *neural network*. Pada penelitian tersebut hasil pelatihan belum dapat mengenali huruf Hijaiyah dengan baik, sedangkan untuk angka arab dapat dikenali dengan nilai akurasi 90%. Sedangkan pada [4] metode *backpropagation* digunakan untuk mengenali pola karakter huruf Jawa. Untuk mengetahui karakteristik *backpropagation* yang digunakan dilakukan pelatihan dengan menggunakan data sampel yang terbagi atas 6 jenis, yaitu 1 sampel, 3 sampel, 5 sampel, 8 sampel, 10 sampel, dan 15 sampel. Hasil dari pengujian sistem menunjukkan rata-rata keakuratan dengan metode *backpropagation* dalam mengenali pola karakter huruf Jawa adalah sebesar 99,563% untuk data sampel berupa data pelatihan, 61,359% untuk data sampel diluar data pelatihan, dan 75% untuk data sampel data pelatihan dan di luar data pelatihan.

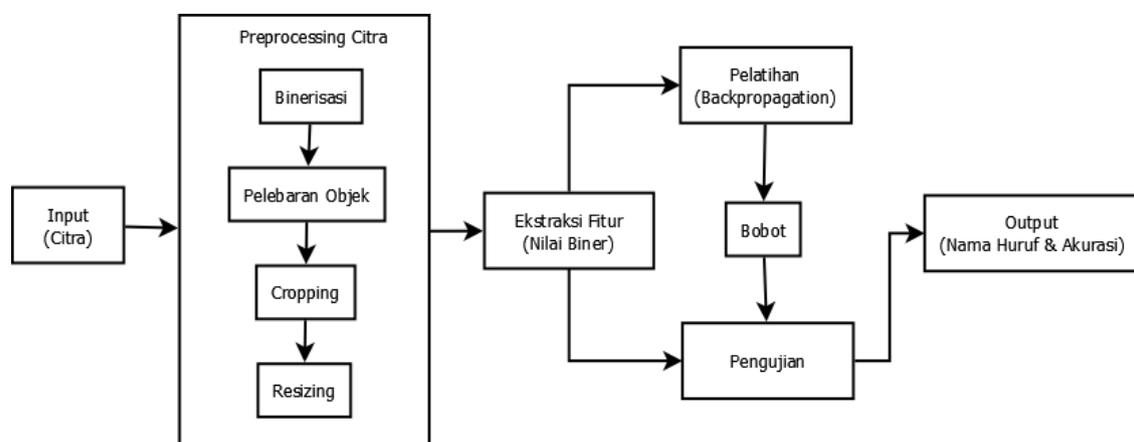
Penelitian yang dilakukan pada [5] juga melakukan pengujian untuk mengidentifikasi aksara Jawa dengan metode *backpropagation*. Identifikasi aksara Jawa yang dilakukan pada

penelitian ini juga akan melakukan proses *text to speech* yaitu merubah teks ke format suara. Data yang diuji berupa huruf vokal aksara Jawa atau yang disebut Aksara Swara. Dari pengujian yang dilakukan didapatkan hasil dari 5 huruf vokal yang diuji dapat terbaca dan dikonversikan sebanyak 5 karakter.

Penelitian mengenai penggunaan *backpropagation* yang telah dilakukan untuk mengenali pola huruf-huruf tertentu mendorong dilakukannya penelitian mengenai identifikasi huruf Hijaiyah ini dengan memanfaatkan metode *backpropagation*. Huruf Hijaiyah yang akan diidentifikasi adalah huruf hijaiyah lepas yang terdiri dari 30 macam huruf, mulai dari Alif (ا) sampai dengan Ya (ي) dengan menggunakan citra tulisan tangan. Jumlah data sampel yang dikumpulkan sebanyak 10 citra untuk tiap macam huruf sehingga total data keseluruhan sebanyak 300 data. Dengan penerapan metode *backpropagation* ini hasil yang diperoleh diharapkan dapat mengidentifikasi huruf Hijaiyah dengan benar dengan nilai akurasi yang baik.

2. METODE PENELITIAN

Arsitektur sistem digunakan untuk menjelaskan proses-proses yang terjadi dan alur kerja dalam sistem secara keseluruhan. Gambar 1 menunjukkan rancangan arsitektur sistem identifikasi huruf Hijaiyah yang dibangun.



Gambar 1. Arsitektur Sistem Identifikasi Huruf Hijaiyah

Dari Gambar 1 dapat dijelaskan bahwa masukan berupa citra tulisan tangan huruf Hijaiyah melalui proses pemrosesan awal (*preprocessing*). Preprocessing citra mencakup proses binerisasi citra, pelebaran objek, *cropping* objek, *resizing* citra. Binerisasi citra adalah proses mengubah citra berwarna atau citra keabuan menjadi citra biner (hitam-putih). Proses ini biasanya menggunakan konsep *thresholding* (pengambangan). Proses pengambangan ini mengambil nilai warna pada tiap piksel citra lalu dibandingkan dengan nilai *threshold* (ambang

batas). Tiap piksel citra lalu diubah warnanya menjadi warna putih jika nilai warna atau keabuannya diatas ambang batas. Sebaliknya jika nilai warna atau keabuannya dibawah ambang batas maka akan diubah menjadi warna hitam.

Pelebaran objek dilakukan untuk menebalkan tulisan huruf, supaya huruf yang ditulis terlalu tipis tidak hilang ketika nantinya dilakukan proses *resizing*. *Cropping* objek adalah proses memotong area citra sehingga hanya tersisa area citra yang berisi objek saja. Proses ini dilakukan untuk membuang area kosong yang tidak berguna di sekeliling objek dan menghindari kesalahan pembelajaran karena letak objek (tulisan huruf) yang berbeda-beda posisinya. *Resizing* citra adalah proses mengecilkan ukuran citra berdasarkan jumlah pikselnya. Misalnya citra dengan ukuran 480 x 640 piksel dikecilkan menjadi 1/32 ukuran asli menjadi 15 x 20 piksel. Proses *resizing* citra diperlukan untuk mengurangi jumlah piksel citra namun tetap dengan mempertimbangkan bentuk objek supaya tidak mengalami perubahan signifikan. Jumlah piksel citra dikurangi agar jumlah *node* atau *neuron* pada *layer input* jaringan syaraf tiruan yang digunakan tidak terlalu banyak sehingga mengakibatkan waktu pemrosesan terlalu lama.

Fitur citra yang akan digunakan untuk mengidentifikasi huruf Hijaiyah adalah nilai biner (1 dan 0) dalam bentuk matriks yang diambil dari warna hitam dan warna putih pada piksel citra yang sudah diubah menjadi citra biner. Proses ekstraksi fitur disini berarti proses merepresentasikan citra biner kedalam matriks piksel dengan ukuran tertentu (x kali y) dan mengambil nilai 1 dan 0 dari warna hitam (objek) dan putih (*background*). Secara *default* program Matlab menilai warna hitam sebagai 0 dan warna putih sebagai 1. Namun karena objek yang kita identifikasi disini adalah tulisan huruf yang berwarna hitam sedangkan *background* nya berwarna putih, maka nilai ini kita balik sehingga objeknya menjadi berwarna putih (bernilai 1) dan *background* nya menjadi berwarna hitam (bernilai 0). Nilai-nilai ini lalu dijadikan vektor input bagi jaringan syaraf tiruan.

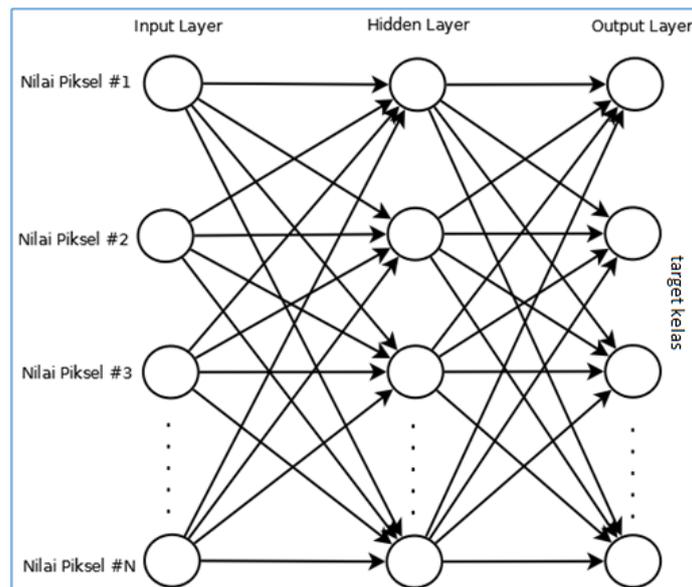
Selain nilai biner yang didapatkan dari representasi warna piksel citra, fitur lain yang diekstraksi adalah jumlah objek dari huruf yang terdapat di dalam citra. Penulisan huruf Hijaiyah terdiri dari garis lengkung dan titik, dua hal inilah yang dimaksud dengan objek. Misalnya pada huruf *Tsa*, terdapat 1 garis lengkung dan 3 titik, artinya jumlah objek pada huruf *Tsa* adalah sebanyak 4 objek. Jumlah *neuron* pada *output layer* harus sesuai dengan jumlah kelas yang akan diidentifikasi. Pada kasus ini ada 30 macam huruf hijaiyah yang akan diidentifikasi, maka jumlah *neuron* pada *output layer* adalah 30. Nilai yang ada pada *output layer* pada dasarnya adalah nilai target yang ingin dicapai oleh jaringan. Maka dari itu kita buat nilai target yang berbeda-beda untuk masing-masing kelas. Misalnya untuk huruf pertama (Alif) maka *node* pertama diisi nilai 1 sedangkan *node* lainnya diisi nilai 0. Lalu untuk huruf kedua (Ba) maka *node* kedua diisi nilai 1 sedangkan *node* lainnya diisi nilai 0. Begitu seterusnya sampai huruf terakhir. Nilai target yang digunakan sebagai nilai *neuron* pada *output layer* untuk mengidentifikasi huruf Hijaiyah dapat dilihat pada Tabel 1.

Tabel 1. Nilai Target untuk *Output Layer*

No.	Alif	Ba	Ta	Tsa	Jim	Ha	Kho	Dal	Dzal	Ro	...	Ya
1	1	0	0	0	0	0	0	0	0	0	...	0
2	0	1	0	0	0	0	0	0	0	0	...	0
3	0	0	1	0	0	0	0	0	0	0	...	0
4	0	0	0	1	0	0	0	0	0	0	...	0
5	0	0	0	0	1	0	0	0	0	0	...	0
6	0	0	0	0	0	1	0	0	0	0	...	0
7	0	0	0	0	0	0	1	0	0	0	...	0
8	0	0	0	0	0	0	0	1	0	0	...	0
9	0	0	0	0	0	0	0	0	1	0	...	0
10	0	0	0	0	0	0	0	0	0	1	...	0
11	0	0	0	0	0	0	0	0	0	0	...	0
12	0	0	0	0	0	0	0	0	0	0	...	0
13	0	0	0	0	0	0	0	0	0	0	...	0
14	0	0	0	0	0	0	0	0	0	0	...	0
15	0	0	0	0	0	0	0	0	0	0	...	0
16	0	0	0	0	0	0	0	0	0	0	...	0
17	0	0	0	0	0	0	0	0	0	0	...	0
18	0	0	0	0	0	0	0	0	0	0	...	0
19	0	0	0	0	0	0	0	0	0	0	...	0
20	0	0	0	0	0	0	0	0	0	0	...	0
21	0	0	0	0	0	0	0	0	0	0	...	0
22	0	0	0	0	0	0	0	0	0	0	...	0
23	0	0	0	0	0	0	0	0	0	0	...	0
24	0	0	0	0	0	0	0	0	0	0	...	0
25	0	0	0	0	0	0	0	0	0	0	...	0
26	0	0	0	0	0	0	0	0	0	0	...	0
27	0	0	0	0	0	0	0	0	0	0	...	0
28	0	0	0	0	0	0	0	0	0	0	...	0
29	0	0	0	0	0	0	0	0	0	0	...	0
30	0	0	0	0	0	0	0	0	0	0	...	1

Pada penelitian ini, proses pelatihan dan pengujian dilakukan dengan menggunakan jaringan syaraf tiruan metode *backpropagation*. Jaringan menerima input dari hasil ekstraksi fitur citra berupa nilai-nilai biner (0 atau 1). Jika ukuran akhir citra setelah melewati tahap *preprocessing* adalah 15 x 20 piksel, maka akan terbentuk matriks berukuran 15 x 20. Sehingga jumlah neuron atau node pada *input layer* adalah sebanyak 300. Nilai (0 atau 1) pada baris pertama kolom pertama matriks[1,1] akan menjadi Nilai Piksel #1, kemudian nilai

pada baris pertama kolom kedua matriks[2,1] akan menjadi Nilai Piksel #2, begitu seterusnya sampai didapatkan Nilai Piksel #300. Struktur jaringan syaraf tiruan *backpropagation* bisa dilihat pada Gambar 2.

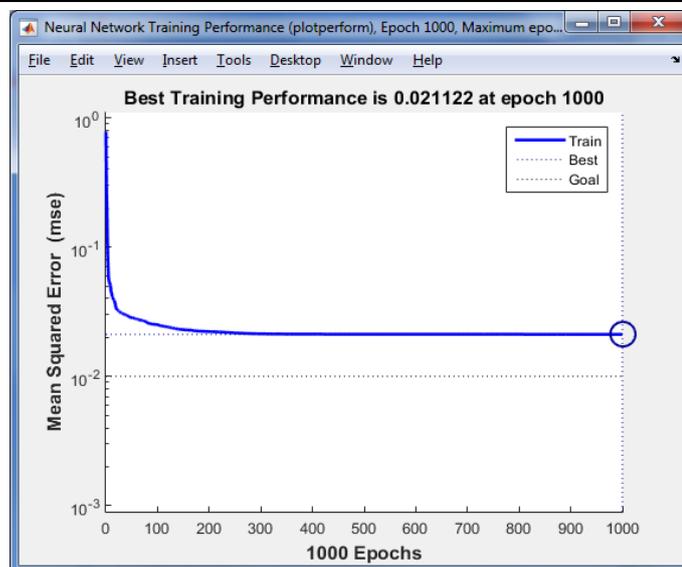


Gambar 2. Struktur JST *Backpropagation* untuk Identifikasi Huruf Hijaiyah

3. HASIL DAN PEMBAHASAN

3.1 Pelatihan

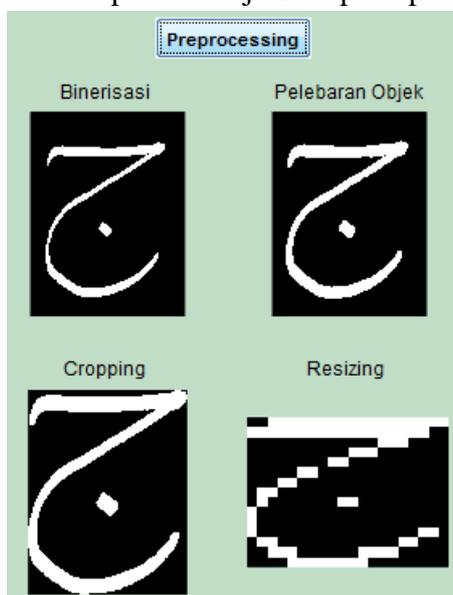
Proses pelatihan dimulai dengan memasukkan jumlah *neuron hidden layer*, maksimum iterasi dan nilai *error* minimum. Pada pelatihan ini digunakan data-data berupa *hidden layer* = 10, max iterasi = 1000 dan min *error* = 0.01. Sebelum proses pelatihan dilakukan dilakukan *preprocessing* citra terlebih dahulu. Proses pelatihan akan berhenti ketika maksimum iterasi atau minimum *error* telah tercapai. Hasil *performance* jaringan sseperti ditampilkan pada Gambar 3.



Gambar 3. Performance Struktur Jaringan Syaraf Tiruan

3.2 Pemrosesan Citra

Proses pertama yang dilakukan pada pemrosesan citra adalah memilih dan menampilkan citra.. Kemudian citra melewati tahap *preprocessing* dimana sistem akan melakukan *preprocessing* tersebut seperti ditunjukkan pada pada Gambar 4.

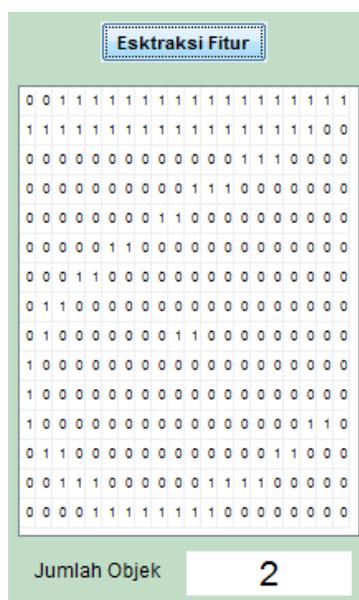


Gambar 4. Tampilan Citra Hasil *Preprocessing*

Citra pada sebelah kiri atas adalah hasil *preprocessing* pertama yaitu mengubah citra warna menjadi citra biner atau biasa disebut juga citra hitam-putih. Citra sebelah kanan atas adalah hasil *preprocessing* kedua yaitu melebarkan objek atau mempertebal tulisan huruf. Citra sebelah kiri bawah adalah hasil *preprocessing* ketiga yaitu *cropping*. Citra dipotong sampai

dengan batas tepi objek untuk membuang ruang kosong yang tidak dibutuhkan di sekeliling objek. Citra sebelah kanan bawah adalah hasil *preprocessing* terakhir yaitu *resizing*. Ukuran citra diperkecil berdasarkan dimensi pikselnya menjadi 15 x 20 piksel.

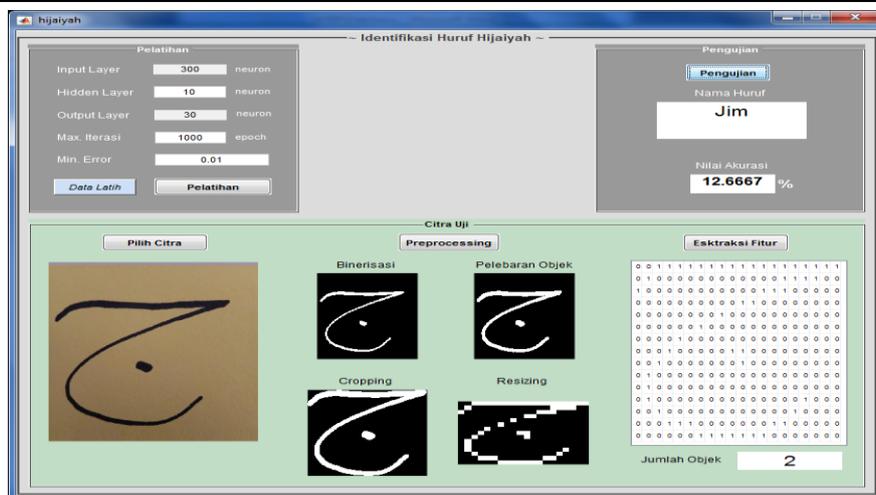
Selanjutnya sistem akan melakukan proses ekstraksi fitur dimana sistem akan mengambil nilai biner dari citra hasil *preprocessing* terakhir dan mengubahnya menjadi matriks. Sistem juga akan mendeteksi jumlah objek yang terdapat pada citra. Misalnya pada uji coba ini huruf Jim memiliki jumlah objek 2, berupa garis dan titik. Matriks nilai biner dan jumlah objek inilah yang dijadikan fitur untuk mengidentifikasi huruf Hijaiyah. Matriks hasil ekstraksi fitur seperti terlihat pada Gambar 5. Nilai 0 mewakili warna hitam pada citra yaitu *background* nya, sedangkan nilai 1 mewakili warna putih pada citra yaitu objek berupa tulisan huruf Hijaiyah.



Gambar 5. Tampilan Hasil Ekstraksi Fitur

3.3 Pengujian

Hasil dari pengujian untuk salah satu citra yang diambil sebagai data uji adalah huruf pada citra teridentifikasi sebagai huruf Jim (benar) dan nilai akurasi yang didapat adalah sebesar 12.66%. Nilai akurasi ini dihitung dari keseluruhan data uji sebanyak 150 citra. Tampilan sistem setelah pengujian bisa dilihat pada Gambar 6.



Gambar 6. Hasil Pengujian Salah Satu Citra

3.4 Mencari Performa Terbaik Sistem

Berdasarkan uji coba yang telah dilakukan, didapatkan hasil berupa nilai akurasi sebesar 12.66%. Nilai akurasi ini mewakili performa sistem, atau bisa dikatakan seberapa baik sistem bisa mengidentifikasi huruf Hijaiyah tulisan tangan. Nilai akurasi 12.66% masih sangat rendah, dari total 150 data uji sistem hanya berhasil mengidentifikasi 19 huruf dengan benar. Sisanya sebanyak 131 huruf tidak berhasil diidentifikasi dengan benar. Pengujian terhadap performa terbaik sistem juga dilakukan dengan melakukan pengujian untuk mencari nilai *hidden layer* terbaik, nilai maksimal iterasi terbaik dan nilai minimal *error* terbaik. Tabel 2 menyajikan hasil pengujian yang dilakukan untuk mencari nilai *hidden layer* terbaik.

Tabel 2. Hasil Pengujian Mencari Nilai Hidden Layer Terbaik

No	Input Layer	Hidden Layer	Output Layer	Max. Iterasi	Min. Error	Akurasi
1	300	10	30	1000	0.01	12.66
2	300	20	30	1000	0.01	23.33
3	300	30	30	1000	0.01	30.66
4	300	40	30	1000	0.01	27.33
5	300	50	30	1000	0.01	36.66
6	300	60	30	1000	0.01	38.00
7	300	70	30	1000	0.01	34.66
8	300	80	30	1000	0.01	36.66
9	300	90	30	1000	0.01	29.33
10	300	100	30	1000	0.01	32.00

Berdasarkan Tabel 2, terlihat bahwa jumlah *neuron hidden layer* berpengaruh terhadap akurasi sistem. Jumlah *neuron hidden layer* yang terlalu sedikit bisa menyebabkan nilai

akurasi rendah. Begitu pula jika jumlahnya terlalu banyak maka nilai akurasi juga ada kemungkinan semakin turun. Oleh karena itu perlu dicari jumlah neuron yang pas yang bisa memberikan nilai akurasi paling baik. Nilai akurasi tertinggi didapatkan pada jumlah *neuron* 60, yaitu sebesar 38%. Jumlah *neuron* ini digunakan untuk uji coba selanjutnya.

Tabel 3. Hasil Pengujian Mencari Nilai Max. Iterasi Terbaik

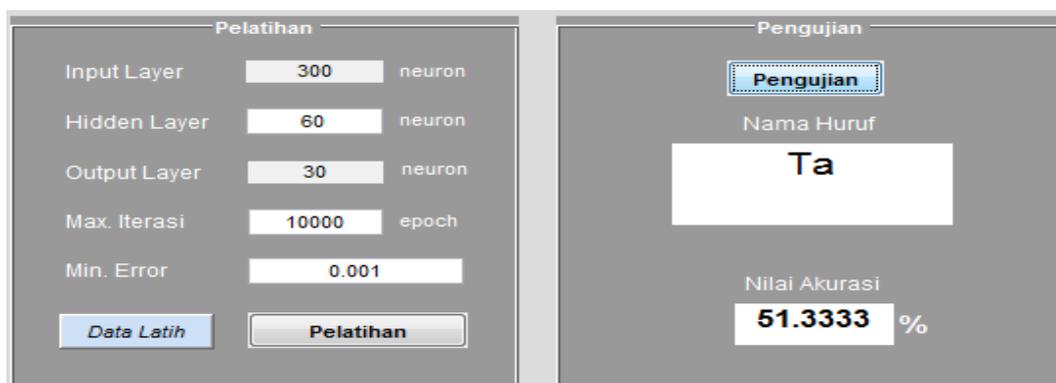
No	Input Layer	Hidden Layer	Output Layer	Max. Iterasi	Min. Error	Akurasi
1	300	60	30	1000	0.01	38.00
2	300	60	30	2000	0.01	35.33
3	300	60	30	3000	0.01	34.66
4	300	60	30	4000	0.01	35.33
5	300	60	30	5000	0.01	38.66
6	300	60	30	6000	0.01	34.66
7	300	60	30	7000	0.01	37.33
8	300	60	30	8000	0.01	36.00
9	300	60	30	9000	0.01	37.33
10	300	60	30	10000	0.01	38.66

Hasil pengujian untuk mencari nilai maksimal iterasi dapat dilihat pada Tabel 3. Maksimal iterasi pada pengujian ini tidak memberikan pengaruh signifikan dikarenakan proses pelatihan selalu berhenti atau selesai sebelum batas iterasi tercapai. Proses pelatihan berhenti karena target *performance* atau *min. error* tercapai lebih dulu. Meskipun selisihnya tidak signifikan, namun akurasi terbaik sebesar 38,66% didapatkan ketika pelatihan menggunakan max iterasi 10000. Oleh karena itu, pengujian selanjutnya akan menggunakan max iterasi 10000. Uji coba terakhir adalah mencari nilai minimal *error* yang paling baik. Tabel 4 menunjukkan hasil pengujian pencarian nilai minimal *error*.

Tabel 4. Hasil Pengujian Mencari Nilai Min. Error Terbaik

No	Input Layer	Hidden Layer	Output Layer	Max. Iterasi	Min. Error	Akurasi
1	300	60	30	10000	0.01	38.66
2	300	60	30	10000	0.001	51.33
3	300	60	30	10000	0.0001	44.66
4	300	60	30	10000	0.00001	40.00
5	300	60	30	10000	0.000001	36.66
6	300	60	30	10000	0.0000001	33.33
7	300	60	30	10000	0.00000001	28.66

Nilai akurasi tertinggi didapatkan pada min. *error* 0.001 yaitu sebesar 51,33%. Artinya sistem berhasil mengidentifikasi 77 huruf hijaiyah dengan benar dari total 150 data uji. Jika nilai min. *error* dibuat lebih rendah dari 0.001, nilai akurasi justru semakin turun. Berdasarkan hasil uji coba yang telah dilakukan diperoleh hasil akurasi terbaik dengan pengaturan struktur jaringan, yaitu jumlah *neuron hidden layer* = 60, maksimal iterasi = 10000 dan minimal *error* = 0.001. Struktur jaringan yang didapatkan dari hasil uji coba memberikan hasil pengujian dengan nilai akurasi sebesar 51,33%, seperti ditunjukkan pada Gambar 7.



Gambar 7. Hasil Performa Terbaik Sistem

4. KESIMPULAN

Berdasarkan pembahasan yang telah dilakukan kesimpulan yang didapatkan adalah Metode pelatihan *backpropagation* dan ekstraksi fitur binerisasi pola huruf terbukti mampu melatih sistem untuk mengenali pola huruf Hijaiyah tulisan tangan tangan. Pada pengujian yang dilakukan sistem berhasil 100% mengidentifikasi data latih dengan benar. Sedangkan untuk data uji, sistem berhasil mendapatkan nilai akurasi tertinggi sebesar 51,33%. Jumlah *neuron* pada *hidden layer* jaringan akan mempengaruhi akurasi sistem. Jumlah *neuron* yang terlalu sedikit atau terlalu banyak tidak bisa menghasilkan akurasi yang tinggi. Dari hasil uji coba, nilai akurasi tertinggi didapatkan ketika jumlah *neuron hidden layer* sebanyak 60.

DAFTAR PUSTAKA

- [1] Siang, Jong Jek, *Jaringan Syaraf Tiruan dan Pemrogramannya Menggunakan Matlab*. Yogyakarta: Andi. 2005.
- [2] Kusumadewi, Sri, Kusumadewi, Sri, 2004, *Membangun Jaringan Syaraf Tiruan Menggunakan Matlab dan Excel Link*. Yogyakarta: Graha Ilmu. 2004
- [3] M. Anif *et al.*, *Pengembangan Aplikasi Text Recognition Dengan Klasifikasi Neural*

- Network. BOT ISSN 1693-9166. 2013;vol. 10, no. 1:pp. 59–67.*
- [4] N. Nurmila, A. Sugiharto, and E. A. Sarwoko, *Algoritma Back Propagation Neural Network untuk Pengenalan Karakter Huruf Jawa. J. Masy. Inform. ISSN 2086-4930. 2005; vol. 1, no. 1:pp. 1–10*
- [5] I. Prihandi and S. Zubair. *Prototipe Text Recognition dengan Kalsifikasi Neural Network dan Text-to-Speech pada Huruf Aksara Jawa.1st Informatics Conf. 2015; vol. 2015:pp. 14–17.*