

**NASKAH PUBLIKASI**

**IMPLEMENTASI METODE BACKPROPAGATION  
NEURAL NETWORK DALAM PENGENALAN KARAKTER  
TULISAN HIRAGANA JEPANG**



Disusun oleh:

**DIAN AGUS SETIANI**

**5160411024**

**PROGRAM STUDI INFORMATIKA  
FAKULTAS TEKNOLOGI INFORMASI DAN ELEKTRO  
UNIVERSITAS TEKNOLOGI YOGYAKARTA  
2020**

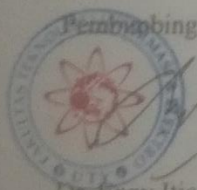
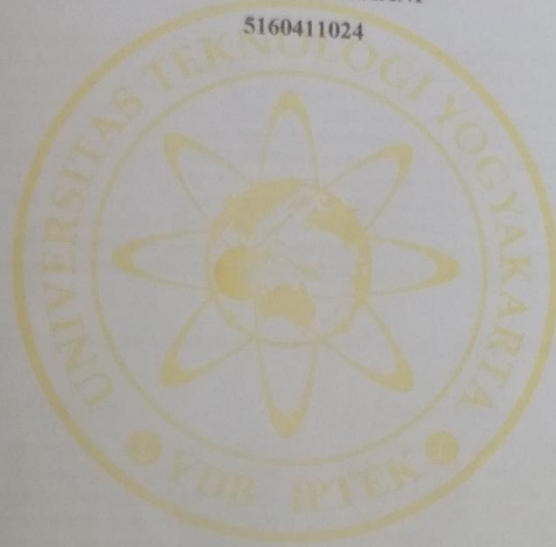
NASKAH PUBLIKASI

IMPLEMENTASI METODE BACKPROPAGATION  
NEURAL NETWORK DALAM PENGENALAN KARAKTER  
TULISAN HIRAGANA JEPANG

Disusun oleh

DIAN AGUS SETIANI

5160411024



Dr. Enny Itje Sela, S.Si., M.Kom.

Tanggal: 09-09-2020

# IMPLEMENTASI METODE BACKPROPAGATION NEURAL NETWORK DALAM PENGENALAN KARAKTER TULISAN HIRAGANA JEPANG

<sup>1</sup>DIAN AGUS SETANI, <sup>2</sup>ENNY ITJE SELA  
Program Studi Informatika, Fakultas Teknologi Informasi & Elektro  
Universitas Teknologi Yogyakarta  
Jl. Ringroad Utara Jombor Sleman Yogyakarta  
Email: [dianagussetiani11@gmail.com](mailto:dianagussetiani11@gmail.com)

## ABSTRAK

Berkaitan dengan proses pembelajaran bahasa Jepang, penguasaan akan Huruf Hiragana merupakan langkah paling awal yang harus dilewati. Sering terjadi hambatan dalam mempelajari Huruf Hiragana dikarenakan karakter Huruf Hiragana yang berbeda dengan huruf latin yang sering digunakan oleh masyarakat. Terlebih lagi jika pembelajar tersebut kesulitan mengalokasikan waktunya jika harus mengikuti sebuah sesi kursus bahasa Jepang. Untuk itu peneliti membangun sistem pengenalan pola karakter Huruf Hiragana menggunakan metode *Backpropagation Neural Network* dan ekstraksi ciri *Zoning*. Inputan yang digunakan berupa citra tulisan tangan Huruf Hiragana dengan format .jpg. Tulisan tangan Huruf Hiragana tersebut ditulis oleh 6 orang dimana masing-masing menuliskan 46 Huruf Hiragana. Pada proses pelatihan, data latih yang digunakan sebanyak 184 data sedangkan pada proses pengujian, data uji yang digunakan sebanyak 92 data. Pencapaian tingkat akurasi maksimum pada pengujian sebanyak 69.56 % dengan akurasi pelatihan sebanyak 91.30 %.

**Kata Kunci :** Huruf Hiragana, *Backpropagation Neural Network*, *Zoning*, Pengenalan Pola.

## 1. PENDAHULUAN

Jepang merupakan salah satu negara paling maju didunia. Selain memiliki inovasi teknologi yang canggih, Jepang juga memiliki kebudayaan yang menarik serta banyak diminati masyarakat indonesia, beberapa diantaranya adalah manga, anime, film, musik, fashion, make up dan lain-lain. Semakin meningkatnya jumlah penggemar anime Jepang di Indonesia menyebabkan semakin banyak pula orang Indonesia yang ingin mempelajari bahasa Jepang. Mempelajari bahasa Jepang harus mengenali hurufnya terlebih dahulu karena berbeda dengan huruf latin.

Ada 3 jenis huruf yang dipakai untuk menulis dalam bahasa jepang. Huruf-huruf tersebut disebut Kanji, Hiragana dan Katakana (Mulyono and Nuryadin, E. H., 2011)<sup>[3]</sup>. Huruf abjad Hiragana yang ada di Jepang ini merupakan salah satu huruf yang cukup sering digunakan di Negara Jepang.

### 1.1 Latar Belakang

Bahasa Jepang termasuk bahasa yang memiliki bentuk bahasa yang berbeda dengan bahasa asing lainnya. Bentuk bahasa tersebut dapat dilihat dari pelafalan, kosakata, gramatikal, tata bahasa, cara-cara pengucapan, dan gaya bahasa yang digunakan. gaya bahasa Jepang sangat dipengaruhi oleh faktor sosial budaya seperti, wilayah atau daerah, kelas sosial, perbedaan jenis kelamin, dan usia (Drs. Sudjianto, M. H., 2010)<sup>[1]</sup>.

Bagi otak manusia yang memiliki kemampuan luar biasa dapat dengan mudah membedakan huruf-huruf Jepang tersebut. Mengenali huruf Jepang menggunakan komputer merupakan salah satu penerapan dari pengenalan karakter. Pengenalan karakter adalah sub topik dalam pengenalan pola di mana hasil dari pengenalan pola tersebut telah banyak dimanfaatkan di kehidupan nyata seperti proses translasi tulisan tangan menjadi huruf digital pada

mobile touchscreen dan berbagai hal lainnya. Salah satu ilmu yang dapat digunakan untuk pengenalan karakter huruf Hiragana adalah Jaringan Syaraf Tiruan (JST).

## 1.2 Batasan Masalah

Penelitian implementasi metode *Backpropagation Neural Network* dalam pengenalan karakter tulisan Hiragana Jepang ini mencakup berbagai hal, sebagai berikut :

- a. Sistem menggunakan metode *Backpropagation Neural Network*.
- b. Data yang digunakan adalah citra tulisan tangan huruf Hiragana sebanyak 276 data yang ditulis oleh 6 orang mahasiswa Universitas Teknologi Yogyakarta, dimana masing-masing menuliskan 46 huruf Hiragana.
- c. Pengenalan karakter huruf Jepang hanya pada huruf Hiragana dasar (seion) yang terdiri dari 46 huruf.
- d. Data latih yang digunakan sebanyak 184 data dan data uji sebanyak 92 data.
- e. Huruf Hiragana yang dikenali dalam bentuk citra digital dengan format jpg.
- f. Citra karakter tulisan huruf Hiragana yang digunakan memiliki latar belakang yang bersih (warna putih).
- g. Ekstraksi fitur yang digunakan adalah metode *Zoning* gabungan ICZ (*Image Centroid and Zone*) dan ZCZ (*Zone Centroid and Zone*).
- h. Pembuatan sistem dilakukan dengan menggunakan bahasa pemrograman Java, serta menggunakan basisdata MySQL.

## 1.3 Tujuan Penelitian

Tujuan dari penelitian implementasi metode *Backpropagation Neural Network* dalam pengenalan karakter tulisan Hiragana Jepang ini antara lain :

- a. Mengembangkan sistem untuk mengenali pola huruf hiragana.
- b. Mengetahui performansi sistem yang diukur dalam bentuk akurasi.

## 2. LANDASAN TEORI

### 2.1 Huruf Hiragana

Menurut <sup>[11]</sup>Sunarti et al., (2016) Bahasa Jepang atau Nihongo memiliki tiga macam huruf atau kana yaitu Hiragana, Katakana, serta Kanji. Masing-masing dari huruf

tersebut memiliki fungsi tersendiri. Hiragana digunakan pada bahasa Jepang asli dan tata bahasa, sedangkan Katakana digunakan pada bahasa serapan dan terutama penulisan nama. Kanji atau Hanzi sendiri merupakan serapan dari bangsa Cina ketika dibawah kekuasaan Dinasti Han yang umumnya digunakan pada kata benda.

Banyak yang telah melakukan penelitian seperti ini salah satunya <sup>[7]</sup>R, W. K. et al., (2015) mengatakan bahwa “Masih banyak berbagai kalangan mengalami kesulitan mempelajari bahasa Jepang karena bentuk tulisan yang rumit dan tata bahasa sangat jauh berbeda dengan bahasa Indonesia. Oleh karena itu penulis mengambil tema aplikasi pembelajaran huruf Jepang hiragana dan katakana berbasis *platform android*”.

### 2.2 Image Preprocessing

*Image Processing* adalah suatu metode untuk mengolah gambar (*Image*) ke dalam bentuk digital untuk tujuan tertentu. Berikut macam – macam *preprocessing* image yang digunakan :

#### a. Grayscale

*Grayscale* adalah sebuah proses yang mengolah citra dengan mengubah nilai piksel pada citra menjadi citra keabuan. Transformasi ini bertujuan untuk meningkatkan kontras pada citra, sehingga informasi-informasi pada citra itu bisa lebih terlihat. Warna yang merupakan tingkatan dari warna abu-abu hanya digunakan oleh citra keabuan (*grayscale*). Tingkat keabuan disini yaitu warna abu-abu dari berbagai tingkat seperti dari hitam hingga yang mendekati putih (Sukemi and Pratama, Y. T., 2018)<sup>[10]</sup>.

Pengubahan dari citra berwarna ke bentuk *grayscale* biasanya mengikuti aturan sebagai berikut :

$$I(i,j) = \frac{R(i,j) + G(i,j) + B(i,j)}{3} \quad (2.1)$$

dimana  $I(i,j)$  adalah nilai intensitas citra *grayscale*, sedangkan  $R(i,j)$ ,  $G(i,j)$ , dan  $B(i,j)$  masing-masing adalah nilai intensitas warna merah, hijau, dan biru dari citra asal.

#### b. Gaussian Blur

Menurut <sup>[12]</sup>Wibowo, S. H. and Susanto, F., (2016), dengan menggunakan metode filter *Gaussian* maka *noise* yang terdapat pada citra digital sedapat mungkin bisa

berkurang. Hal ini juga bertujuan untuk menghasilkan tepian citra yang sesungguhnya. Apabila proses ini tidak digunakan maka pada pendeteksian garis-garis yang halus juga akan terdeteksi sebagai tepian. *Gaussian* filter yang digunakan adalah filter 2 dimensi dengan persamaan berikut:

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (2.2)$$

dimana  $G(x,y)$  elemen matriks *gauss* diposisi  $(x,y)$ ,  $\pi = 22/7$ ,  $e = 2.71828182846$ , dan  $\sigma$  adalah nilai standar deviasi ( $\sigma = 1$ ). Pada *Gaussian* dilakukan perhitungan konvolusi dimana *pixel* di ambil dari gambar hasil *grayscale* dan dikali dengan karnel pengali.

#### c. Binerisasi

Binerisasi citra merupakan proses merubah citra ke dalam bentuk biner (0 dan 1). Dengan merubah ke bentuk biner, citra hanya akan mempunyai 2 warna yakni hitam dan putih. Mengubah gambar ke biner prosesnya hampir sama dengan mengubah gambar ke grayscale, bedanya warna rata-rata akan dikelompokkan menjadi dua, jika intensitas warna dimulai dari 0 sampai dengan 255 maka diambil nilai tengahnya yaitu 128, jika dibawah 128 maka warna akan cenderung hitam dan diatas 128 warna akan cenderung putih. Binerisasi cira dapat dilakukan menggunakan persamaan berikut:

$$g(x,y) = \begin{cases} 1 & \text{if } f(x,y) \geq T \\ 0 & \text{if } f(x,y) < T \end{cases} \quad (2.3)$$

#### d. Thinning Zhang-Suen

Menurut <sup>[2]</sup>Fakhrina, F. A. et al., (2016) algoritma *Thinning Zhang-Suen* merupakan algoritma iteratif dan menghapus semua titik luar gambar kecuali ini menjadi kerangka. Metode ini digunakan untuk ekstraksi kerangka citra terdiri dari menghapus semua poin kontur gambar kecuali poin milik kerangka. Untuk mempertahankan kerangka citra maka T. Zhang dan C. Suen menyatakan bahwa proses ini akan membagi setiap iterasi menjadi dua-subiterasi.

Langkah pertama dari sebuah penelusuran adalah menandai semua titik object untuk dihapus, jika titik object tersebut memenuhi syarat-syarat (Rahayu, A., 2020)<sup>[7]</sup>:

$$1) \quad 2 \leq B(P_1) \leq 6 \quad (2.4)$$

$B(P_1)$  adalah banyaknya titik tetangga dari titik  $P_1$  yang bukan nol, yaitu:  $B(P_1) = P_2 + P_3 + P_4 + P_5 + \dots + P_8 + P_9$ .

$$2) \quad A(P_1) = 1 \quad (2.5)$$

$A(P_1)$  adalah banyaknya transisi pola 0 ke 1 dari nilai piksel  $P_2, P_3, P_4, \dots, P_8, P_9, P_{10}$  ( $P_{10} = P_2$ ) secara berurutan, untuk operasi terhubung 8 (8 titik tetangga).

$$3) \quad P_2 \bullet P_4 \bullet P_6 = 0 \quad (2.6)$$

$$P_2 \bullet P_6 \bullet P_8 = 0 \quad (2.7)$$

Setelah subiterasi pertama selesai, hapus seluruh titik yang telah ditandai, kemudian dilanjutkan dengan subiterasi kedua, yang diterapkan terhadap titik kontur dari hasil subiterasi pertama, dimana hanya kondisi 3) dan 4) yang akan diubah, sedangkan kondisi 1) dan 2) sama seperti pada subiterasi pertama

$$3') \quad P_2 \bullet P_4 \bullet P_8 = 0 \quad (2.8)$$

$$4') \quad P_2 \bullet P_6 \bullet P_8 = 0 \quad (2.9)$$

#### e. Cropping

*Cropping* merupakan salah satu pengolahan citra yang bertujuan untuk memotong bagian citra yang tidak diperlukan. Proses *cropping* atau pemotongan pada gambar atau citra dilakukan agar mempersempit batasannya pada bagian gambar yang ingin dideteksi.

### 2.3 Ekstraksi Ciri Zoning

*Zoning* merupakan salah satu metode ekstraksi ciri pada citra karakter. Secara umum, dengan metode ekstraksi ciri zoning citra akan dibagi menjadi beberapa zona yang berukuran sama, untuk kemudian dari setiap zona akan diambil cirinya (Pitoyo, C. R. et al., n.d.)<sup>[6]</sup>.

Ekstraksi ciri *Zoning* membagi citra pada sejumlah zona yang sama untuk dikenali ciri dari setiap karakter huruf yang selanjutnya menghasilkan sebuah nilai untuk diproses pada tahapan klasifikasi. Variasi metode ekstraksi ciri *zoning* ada 3 macam yaitu metode ekstraksi ciri jarak metrik ICZ (*Image Centroid and Zone*), ZCZ (*Zone Centroid and Zone*), dan gabungan (ICZ + ZCZ). Dalam penelitian ini variasi metode ekstraksi ciri *zoning* yang digunakan yaitu gabungan (ICZ + ZCZ). Adapun algoritma dari metode ekstraksi ciri *zoning* gabungan (ICZ+ZCZ) adalah sebagai berikut (Pitoyo, C. R. et al., n.d.)<sup>[6]</sup>:

a. Hitung *centroid* pusat dari citra masukan dengan rumus :

$$x_c = \frac{\sum_{i=1}^x f(x,y) x_i}{\sum f(x,y)} \quad y_c = \frac{\sum_{i=1}^y f(x,y) y_i}{\sum f(x,y)} \quad (2.10)$$

dimana :

$x_c$  = koordinat x centroid pusat

$y_c$  = koordinat y centroid pusat

$x_i$  = koordinat x ke i pada citra

$y_i$  = koordinat y ke i pada citra

$f(x,y)$  = nilai piksel pada koordinat ke x,y

b. Bagi citra dengan n zona yang sama.

c. Hitung jarak antara *centroid* citra dengan setiap *pixel* yang ada dalam zona dengan

$$\text{rumus : Jarak} = \sqrt{(x-x_c)^2 + (y-y_c)^2} \quad (2.11)$$

$x_c$  = koordinat x centroid pusat

$y_c$  = koordinat y centroid pusat

$x$  = koordinat x pada piksel

$y$  = koordinat y pada piksel

d. Ulangi langkah c untuk semua *pixel* yang ada dalam zona.

e. Hitung jarak rata-rata antara titik-titik tersebut dengan rumus :

$$\overline{\text{Jarak}} = \frac{1}{n} (\text{jarak1} + \dots + \text{jarakn}) \quad (2.12)$$

dimana :

$\overline{\text{Jarak}}$  = jarak rata-rata

$n$  = banyaknya piksel yang bernilai 1

Jarakn = jarak pada zona ke-n

f. Hitung *centroid* tiap zona menggunakan persamaan 2.10.

g. Hitung jarak antara *centroid* zona dan setiap *pixel* yang ada dalam zona menggunakan persamaan 2.11.

h. Ulangi langkah g untuk semua *pixel* yang ada dalam zona.

i. Hitung jarak rata-rata antara titik – titik tersebut menggunakan persamaan 2.12.

j. Ulangi langkah c – i untuk semua zona secara berurutan.

Akan didapatkan 2n ciri untuk klasifikasi dan pengenalan.

#### 2.4 Normalisasi Min Maks

Min-Max normalization merupakan metode normalisasi dengan melakukan transformasi linier terhadap data asli sehingga menghasilkan keseimbangan nilai perbandingan antar data saat sebelum dan sesudah proses (Nasution, D. A. et al., 2019)<sup>[4]</sup>. Metode ini dapat menggunakan rumus sebagai berikut :

$$\text{nor}(x) = \frac{(x - \text{min})}{\text{max} - \text{min}} (\text{maxNew} - \text{minNew}) + \text{minNew} \quad (2.13)$$

#### 2.5 Jaringan Syaraf Tiruan Backpropagation

*Backpropagation* merupakan salah satu model yang terdapat pada JST yang menggunakan *supervised learning*. Algoritma ini sering digunakan untuk menyelesaikan suatu masalah yang rumit. Hal ini dikarenakan algoritma ini dilatih menggunakan metode pembelajaran. Pada jaringan ini diberikan pola yang sepasang yang terdiri atas pola masukan dan pola yang dikehendaki (Suhartanto, R. S. et al., 2017)<sup>[9]</sup>.

Terdapat 3 fase dalam pelatihan BPNN (*Backpropagation Neural Network*), yaitu fase maju (*feed forward*), fase mundur (*back propagation*), dan fase modifikasi bobot. Dalam fase *feed forward*, pola masukan dihitung maju dimulai dari lapisan input hingga lapisan output. Dalam fase *back propagation*, tiap-tiap unit output menerima target pola yang berhubungan dengan pola input untuk dihitung nilai kesalahan. Kesalahan tersebut akan dipropagasikan mundur. Sedangkan fase modifikasi bobot bertujuan untuk menurunkan kesalahan yang terjadi. Ketiga fase tersebut diulang secara terus menerus hingga kondisi penghentian dipenuhi (Nurmila, N. et al., n.d.)<sup>[5]</sup>.

Adapun algoritma dari jaringan syaraf tiruan *Backpropagation* adalah sebagai berikut (Suhartanto, R. S. et al., 2017)<sup>[9]</sup> :

1. Inisialisasi bobot (ambil bobot awal dengan nilai random yang cukup kecil).
2. Tetapkan nilai *maksimum* epoch, *target error*, dan *learning rate* ( $\alpha$ ).
3. Inisialisasi nilai epoch = 0 dan MSE = 1.
4. Melakukan langkah-langkah berikut selama (epoch < maksimum epoch) dan (MSE > target error) :
  - a. Epoch = epoch + 1.
  - b. Untuk tiap-tiap pasangan elemen yang akan dilakukan pembelajaran, kerjakan :

*Feedforward* :

- Tiap-tiap unit input ( $x_i$ ,  $i=1,2,3,4,\dots,n$ ) menerima sinyal  $x_i$  dan meneruskan sinyal tersebut ke semua unit pada lapisan yang ada di atasnya (lapisan tersembunyi).
- Tiap-tiap unit pada suatu lapisan tersembunyi ( $z_j$ ,  $j=1,2,3,4,\dots,n$ ) menjumlahkan sinyal-sinyal input terbobot dengan rumus :
$$z_{in_j} = b1_j + \sum_{i=1}^n x_i v_{ij} \quad (2.14)$$

gunakan fungsi aktivasi untuk menghitung sinyal outputnya dengan rumus fungsi aktivasi *sigmoid* :

$$y=f(x)=\frac{1}{1+e^{-\alpha x}} \quad z_j=f(z_{in_j}) \quad (2.15)$$

ulangi langkah ini sebanyak jumlah lapisan tersembunyi dan kirimkan sinyal tersebut ke semua unit di lapisan atasnya (unit-unit output).

- Tiap-tiap unit output ( $y_k$ ,  $k=1,2,3,4,\dots,m$ ) menjumlahkan sinyal-sinyal input terbobot dengan rumus :

$$y_{in_k}=b2_k+\sum_{i=1}^p z_i w_{jk} \quad (2.16)$$

Gunakan fungsi aktivasi untuk menghitung sinyal outputnya dengan rumus :

$$y_k=f(y_{in_k}) \quad (2.17)$$

*Backpropagation* :

- Tiap-tiap unit output ( $y_k$ ,  $k = 1,2,3,4,\dots,m$ ) menerima target pola yang berhubungan dengan pola input pembelajaran, hitung informasi errornya:

$$\delta 2_k=(t_k-y_k)f'(y_{in_k}) \quad (2.18)$$

$$\varphi 2_{jk}=\delta_k z_j \quad (2.19)$$

$$\beta 2_k=\delta_k \quad (2.20)$$

Kemudian hitung koreksi bobot (yang nantinya akan digunakan untuk memperbaiki nilai  $w_{jk}$ ) dengan rumus :

$$\Delta w_{jk}=\alpha \varphi 2_{jk} \quad (2.21)$$

hitung juga koreksi bias (yang nantinya akan digunakan untuk memperbaiki nilai  $b2_k$ ) dengan rumus :

$$\Delta b2_k=\alpha \beta 2_k \quad (2.22)$$

Langkah ini juga dilakukan sebanyak jumlah lapisan tersembunyi, yaitu menghitung informasi *error* dari suatu lapisan tersembunyi ke lapisan tersembunyi sebelumnya.

- Tiap-tiap unit tersembunyi ( $z_j, j=1,2,3,\dots,p$ ) menjumlahkan delta inputnya (dari unit-unit yang berada pada lapisan di atasnya) dengan rumus:

$$\delta_{in_j}=\sum_{k=1}^m \delta 2_k w_{jk} \quad (2.23)$$

kalikan nilai ini dengan turunan dari fungsi aktivasinya untuk menghitung informasi *error* dengan rumus :

$$\delta 1_j=\delta_{in_j} f'(z_{in_j}) \quad (2.24)$$

$$\varphi 1_{ij}=\delta 1_j x_j \quad (2.25)$$

$$\beta 1_j=\delta 1_j \quad (2.26)$$

Kemudian hitung koreksi bobot (yang nantinya akan digunakan untuk memperbaiki nilai  $v_{ij}$ ) dengan rumus :

$$\Delta v_{ij}=\alpha \varphi 1_{ij} \quad (2.27)$$

Hitung juga koreksi bias (yang nantinya akan digunakan untuk memperbaiki nilai  $b_{ij}$ ) dengan rumus :

$$\Delta b1_j=\alpha \beta 1_j \quad (2.28)$$

- Tiap-tiap unit output ( $y_k$ ,  $k=1,2,3,\dots,m$ ) memperbaiki bias dan bobotnya ( $j = 0,1,2,\dots,p$ ) dengan rumus :

$$w_{jk}(\text{baru})=w_{jk}(\text{lama})+\Delta w_{jk} \quad (2.29)$$

Tiap-tiap unit tersembunyi ( $z_j$ ,  $j = 1,2,3,\dots,p$ ) memperbaiki bias dan bobotnya ( $i = 0,1,2,\dots,n$ ) dengan rumus :

$$v_{ij}(\text{baru})=v_{ij}(\text{lama})+\Delta v_{ij} \quad (2.30)$$

$$b1_j(\text{baru})=b1_j(\text{lama})+\Delta b1_j \quad (2.31)$$

- Hitung MSE

### 3. METODOLOGI PENELITIAN

#### 3.1 Tahapan Penelitian

Dalam penelitian implementasi metode *Backpropagation Neural Network* dalam pengenalan karakter tulisan Hiragana Jepang terdiri dari lima tahapan penelitian, yaitu pengumpulan data, pra proses *image*, ekstraksi ciri *zoning*, normalisasi data menggunakan metode min maks, pelatihan dan pengujian jaringan syaraf tiruan, analisis hasil penelitian dan kesimpulan. Tahapan penelitian ini dapat dilihat pada Gambar 3.1.

##### Pengumpulan Data

Pengumpulan data yang dilakukan adalah dengan mengisikan form tulisan tangan huruf Hiragana. Data yang digunakan merupakan citra tulisan huruf Hiragana sebanyak 276 data yang ditulis oleh 6 orang mahasiswa Universitas Teknologi Yogyakarta, dimana masing-masing menuliskan 46 huruf Hiragana.

##### Pre Processing Image

*Pre processing image* digunakan untuk mengolah citra untuk kemudian diambil cirinya. Pada penelitian ini, pra proses *image* yang digunakan meliputi *resize* piksel citra inputan menjadi ukuran 100x100, konversi citra ke *grayscale*, menghilangkan *noise* dengan *gaussian blur*, ubah citra ke dalam citra biner, tipiskan citra menggunakan penipisan *zhang suen*, pemotongan citra atau *cropping*, *resize* piksel menjadi ukuran 40x42, ubah citra ke citra biner dan yang

terakhir tipiskan citra menggunakan penipisan *zhang suen*.

### Ekstraksi Ciri Zoning

Setelah melewati pra proses image, tahap selanjutnya adalah ekstraksi ciri. Ekstraksi ciri pada sistem ini menggunakan metode *zoning* gabungan ICZ (*Image Centroid and Zone*) dan ZCZ (*Zone Centroid and Zone*). Citra yang diinputkan dibagi ke dalam 6 zona yang sama kemudian diambil cirinya. Hasil dari ekstraksi ciri ini menghasilkan ciri sebanyak 12 yang kemudian akan menjadi inputan jaringan syaraf tiruan. Hasil dari ekstraksi ciri kemudian akan disimpan ke dalam tabel.

### Normalisasi Min-Maks

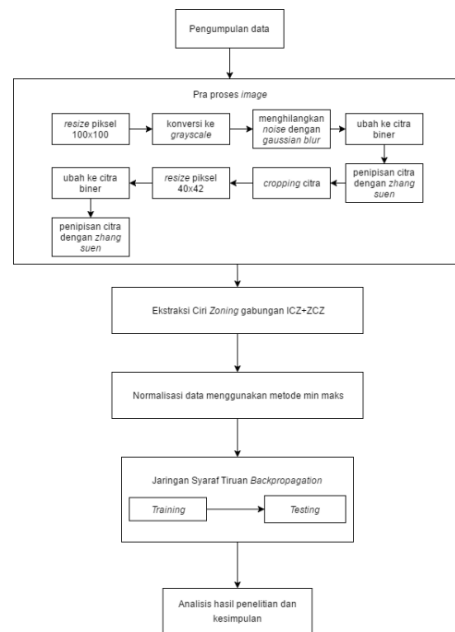
Data hasil ekstraksi ciri yang diambil dari tabel kemudian akan dinormalisasi menggunakan metode min maks dengan range antara 0.1 sampai 0.9 sebelum masuk ke proses pelatihan dan pengujian.

### Pelatihan dan Pengujian Data

Data yang sudah dinormalisasi sebelumnya, kemudian akan digunakan untuk pelatihan dan pengujian menggunakan metode *Backpropagation Neural Network*. Dalam penelitian ini arsitektur jaringan syaraf tiruan yang digunakan yaitu 1 *input layer*, 1 *hidden layer*, dan 1 *output layer*. Data akan melalui proses pelatihan berdasarkan parameter yang diinputkan untuk mendapatkan bobot yang optimal. Setelah proses pelatihan selesai, selanjutnya dilakukan pengujian terhadap data uji menggunakan bobot yang dihasilkan pada proses pelatihan.

### Analisis Hasil dan Kesimpulan

Setelah hasil dari penelitian atau obyek sudah didapatkan melalui proses *training* dan *testing* selanjutnya akan dianalisa berdasarkan tingkat akurasi pelatihan dan pengujian dari sistem pengenalan karakter tulisan tangan hiragana.

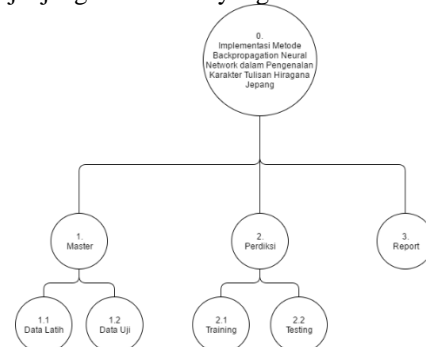


Gambar 3. 1 Tahapan penelitian

## 3.2 Metode Penelitian

### a. Perancangan Sistem

Dalam proses perancangan sistem ini menggunakan Diagram Jenjang dan Data Flow Diagram (DFD) yang bertujuan sebagai gambaran bagaimana sistem ini akan dibuat. Berikut diagram jenjang dari sistem yang dibuat :



Gambar 3. 2 Diagram Jenjang

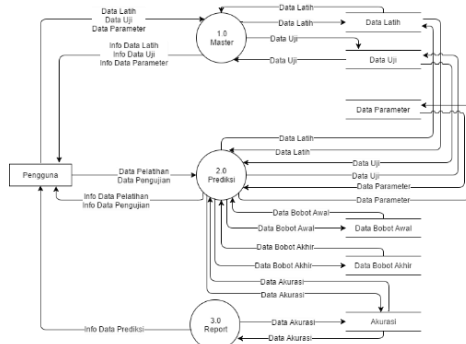
Diagram jenjang pada sistem ini memiliki 2 level. Pada level yang pertama terdapat 3 proses, yaitu proses master, prediksi, dan report. Sedangkan untuk level 2 pada proses 1 terdiri dari proses master data latih dan data uji. Kemudian untuk level 2 proses 3 terdapat proses *training* dan *testing*.

### Data Flow Diagram (DFD)

DFD Level 1 dari penelitian ini, seperti pada Gambar 3.3 terdiri dari 1 pengguna dan

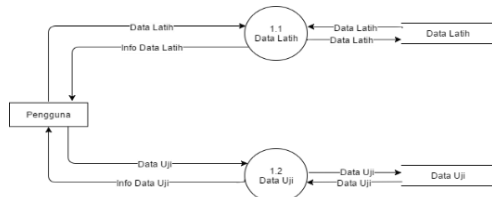


3 proses yaitu proses master, prediksi dan report. Pada proses master melibatkan penyimpanan data latih dan data uji, kemudian pada proses prediksi melibatkan penyimpanan data latih, data uji, data parameter, data bobot awal, data bobot akhir, dan akurasi. Sedangkan pada proses report melibatkan penyimpanan data hasil testing.



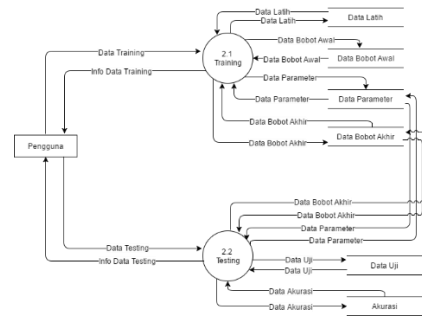
Gambar 3. 3 DFD Level 1

DFD Level 2 proses 1 pada penelitian ini terdapat pada Gambar 3.4. Pada DFD Level 2 proses 1 terdiri dari 1 pengguna dan 2 proses yaitu data latih dan data uji. Pada proses data latih melibatkan penyimpanan data latih sedangkan pada proses data uji melibatkan penyimpanan data uji.



Gambar 3. 4 DFD Level 2 Proses 1

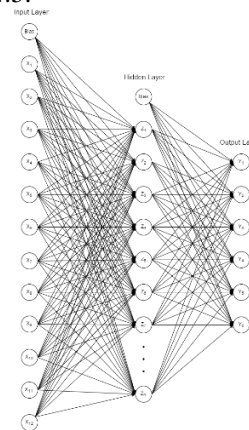
DFD Level 2 proses 2 seperti pada Gambar 4.4 terdiri dari 1 pengguna dan 2 proses yaitu proses training dan testing. Pada proses training melibatkan penyimpanan data latih, data bobot awal, data parameter, dan data bobot akhir. Sedangkan pada proses testing melibatkan penyimpanan data uji, data bobot akhir dan akurasi.



Gambar 3. 5 DFD Level 2 Proses 2

### Arsitektur Jaringan Syaraf Tiruan

Arsitektur jaringan syaraf yang dipakai pada penelitian ini terdiri dari 1 input layer, 1 hidden layer, dan 1 output layer. Pada input layer, digunakan node sebanyak 12 sedangkan pada output layer, node yang digunakan sebanyak 6. Bias digunakan pada input layer dan hidden layer. Arsitektur jaringan syaraf yang dipakai terdapat pada Gambar 4.5.



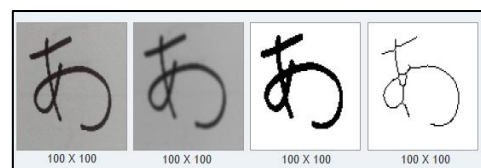
Gambar 3. 6 Arsitektur jaringan syaraf tiruan

## 4. HASIL DAN PEMBAHASAN

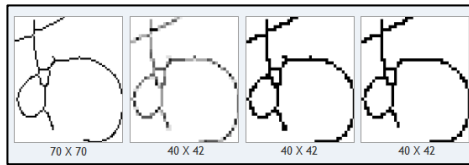
### 4.1 Hasil

#### a. Hasil Pengolahan Citra

Tahapan pengolahan citra pada penelitian ini diantaranya adalah *grayscale*, *gaussian blur*, *binerisasi*, *thinning*, *cropping*, *resizing* dan ekstraksi ciri menggunakan metode *Zoning*. Berikut ini hasil proses pengolahan citra :

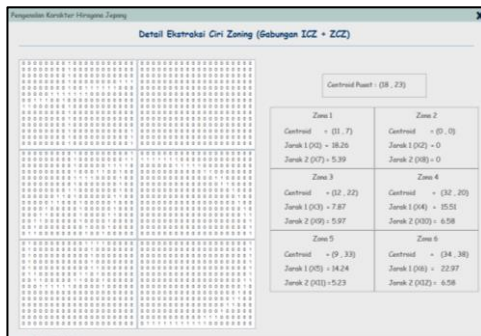


Gambar 4. 1 Citra hasil pengolahan citra 1



Gambar 4. 2 Citra hasil pengolahan citra 2

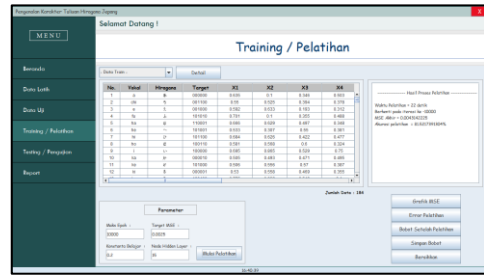
Dalam penelitian ini ekstraksi ciri yang digunakan yaitu metode ekstraksi ciri *zoning*. Citra dibagi menjadi 6 zona dengan ukuran yang sama, kemudian dicari nilai *centroid*nya lalu dihitung jarak *euclidean* dari masing-masing piksel ke *centroid*. Jarak tersebut yang digunakan sebagai ciri. Detail ekstrasi ciri dan pembagian zona pada citra hasil pemrosesan terdapat pada Gambar 4.3.



Gambar 4. 3 Detail ekstraksi ciri pada citra a

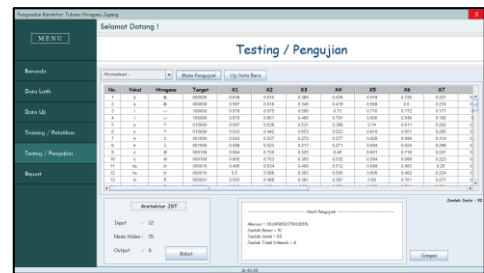
b. Hasil Pelatihan dan Pengujian  
 Pada proses pelatihan jaringan syaraf tiruan, parameter yang diperlukan diantaranya jumlah maks epoch, target *error*, konstanta belajar, dan jumlah node *hidden*. Halaman pelatihan pada sistem terdapat pada Gambar 4.4. Data latih yang digunakan sebanyak 184 data. Data tersebut dinormalisasi terlebih dahulu menggunakan normalisasi min maks. Data latih yang digunakan merupakan data hasil ekstraksi ciri dari citra yang kemudian dinormalisasi menggunakan normalisasi min-maks. Parameter-parameter yang digunakan antara lain :

- Node input = 12
- Node output = 6
- Konstanta belajar = 0.01, 0.02, 0.1, 0.2
- Target error = 0.0025
- Maksimal epoch = 500, 1000, 5000, 10000, 15000



Gambar 4. 4 Halaman Pelatihan

Setelah didapatkan bobot yang optimal kemudian dilakukan pengujian terhadap data uji. Data uji yang digunakan sebanyak 92 data. Seperti halnya proses peatihan, data uji dalam proses pengujian juga dinormalisasi terlebih dahulu menggunakan normalisasi min maks.



Gambar 4. 5 Halaman Pengujian

## 4.2 Pembahasan

Pelatihan data menggunakan model jaringan syaraf tiruan backpropagation yang dilakukan pada data latih sebanyak 184 data dan pengujian terhadap 92 data uji dilakukan sebanyak 100 kali percobaan. Hubungan jumlah epoch, jumlah node *hidden*, serta akurasi pada beberapa percobaan terdapat pada Tabel 4.1.

Tabel 4. 1 Tabel palatihan dan pengujian

| No | Iterasi | Hidden | K B | Akurasi Latih | Akurasi Uji |
|----|---------|--------|-----|---------------|-------------|
| 1  | 10000   | 15     | 0.1 | 77.17 %       | 38.04 %     |
| 2  | 10000   | 20     | 0.1 | 80.97 %       | 39.13 %     |
| 3  | 10000   | 25     | 0.1 | 83.15 %       | 43.47 %     |
| 4  | 10000   | 30     | 0.1 | 84.23 %       | 48.91 %     |
| 5  | 10000   | 35     | 0.1 | 84.23 %       | 42.39 %     |

|    |           |    |     |            |            |
|----|-----------|----|-----|------------|------------|
| 6  | 1000<br>0 | 15 | 0.2 | 81.52<br>% | 33.69<br>% |
| 7  | 1000<br>0 | 20 | 0.2 | 89.13<br>% | 43.37<br>% |
| 8  | 7167      | 25 | 0.2 | 91.3<br>%  | 69.56<br>% |
| 9  | 1000<br>0 | 30 | 0.2 | 86.95<br>% | 47.82<br>% |
| 10 | 1000<br>0 | 35 | 0.2 | 83.69<br>% | 43.37<br>% |

Nilai akurasi data uji terbesar terdapat pada node hidden 25 dengan konstanta belajar sebnyak 0.2 dan epoh sebnyak 10000 yaitu sebesar 69.56 %. Model jaringan syaraf tiruan yang digunakan berhasil mengenali data uji dengan jumlah benar sebanyak 64, jumlah salah sebanyak 22, dan jumlah tak dikenali sebanyak 6. Adapun akurasi data latih dengan jumlah node hidden 25 yaitu sebanyak 91.30 %.

## 5. PENUTUP

### 5.1 Kesimpulan

Dari hasil penelitian dan pembahasan yang telah dilakukan, dapat ditarik kesimpulan sebagai berikut :

- a. Sistem yang diusulkan mampu mengenali huruf Hiragana, hal ini dibuktikan pada proses pengujian data uji, huruf yang dikenali dengan benar lebih banyak dari pada huruf yang tidak dikenali dengan benar yaitu sebanyak 64 dari 92 data.
- b. Tingkat akurasi tertinggi pengujian yaitu sebanyak 69.56 % dengan akurasi pelatihan sebanyak 91.30 %.

### 5.2 Saran

Sistem pengenalan pola huruf Hiragana menggunakan metode *Backpropagation Neural Network* dengan ekstraksi ciri *Zoning* yang telah dibuat masih memiliki akurasi yang rendah. Sistem ini dapat dikembangkan dengan menambahkan proses pengolahan gambar pada citra input agar dapat memisahkan huruf dari beberapa huruf dalam satu citra.

## DAFTAR PUSTAKA

- [1] Drs. Sudjianto, M.H., (2010), *Metodologi Pembelajaran Keterampilan Berbahasa Jepang*, Kesaint Blanc Bekasi.
- [2] Fakhrina, F.A., Rahmadwati and Wijono, (2016), *THINNING ZHANG-SUEN DAN STENTIFORD UNTUK MENENTUKAN EKSTRAKSI CIRI (MINUTIAE)*
- [3] Mulyono and Nuryadin, E.H., (2011), *Tanoshii Nihongo 1 Buku Pelajaran Bahasa Jepang Untuk SMA dan MA Kelas X*, S. S. Liem, Ed.
- [4] Nasution, D.A., Khotimah, H.H. and Chamidah, N., (2019), *PERBANDINGAN NORMALISASI DATA UNTUK KLASIFIKASI WINE MENGGUNAKAN ALGORITMA K-NN, CESS (Journal of Computer Engineering System and Science)*, 4, 78–82.
- [5] Nurmila, N., Sugiharto, A. and Sarwoko, E.A., *ALGORITMA BACK PROPAGATION NEURAL NETWORK UNTUK PENGENALAN POLA KARAKTER HURUF JAWA*, *Jurnal Masyarakat Informatika*, 1, 1–9.
- [6] Pitoyo, C.R., Zuraiyah, T.A. and Qur'ania, A., *PENGENALAN HURUF TULISAN TANGAN MENGGUNAKAN METODE ZONING DAN SUPPORT VECTOR MACHINE*, .
- [7] R, W.K., Jalinus and S.P., E., (2015), *APLIKASI PEMBELAJARAN HURUF JEPANG HIRAGANA DAN KATAKANA BERBASIS SISTEM OPERASI ANDROID*, *JURNAL TEKNIK FTUP*, 28, 89–95.
- [8] Rahayu, A., (2020), *Analisa dan Implementasi Metode Zhang-Suen Dalam Pengerangkaan (Skeleton) Pada Citra Untuk Mengurangi Redundant*, *JURIKOM (Jurnal Riset Komputer)*, 7, 156–161.
- [9] Suhartanto, R.S., Dewi, C. and Muflikhah, L., (2017), *Implementasi Jaringan Syaraf Tiruan Backpropagation untuk Mendiagnosis Penyakit Kulit pada Anak*, *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 1, 555–562.
- [10] Sukemi and Pratama, Y.T., (2018), *Optimalisasi Image Analisis Noise Citra Menggunakan Algoritma Gaussian Filter*, *Prosiding Annual Research Seminar 2018 Computer Science and ICT*, 4, 278–282.
- [11] Sunarti, R.Y, R. and Damhudi, D., (2016), *Aplikasi Pembelajaran Huruf Hiragana dan Katakana Dilengkapi dengan Suara Berbasis Android*, *Jurnal Sains, Teknologi Dan Industri*, 14, 9–16

*SEBAGAI IDENTIFIKASI POLA SIDIK JARI WHORL DAN LOOP*, *Teknologi Elektro*, 15, 127–133.

- [12] Wibowo, S.H. and Susanto, F., (2016), *PENERAPAN METODE GAUSSIAN SHOOTING UNTUK MEREDUKSI NOISE PADA SITRA DIGITAL*, *Jurnal Media Infotama*, 12, 129-135.