Implementasi Jaringan Syaraf Tiruan *Back Propagation* dalam Pengembangan Aplikasi untuk Mengidentifikasi Aksara *Katakana*

M. Fakhrurrozi Bimo A.[1], Ipung Permadi, S.Si., M.Cs.[2] dan Arief Kelik N., S.Kom., M.Cs[3]
[1,2,3]*Jurusan Teknik Informatika, Fakultas Teknik, Universitas Jenderal Soedirman*
Jalan Mayjend. Sungkono KM.5 Desa Blater, Kecamatan Kalimanah, Kabupaten Purbalingga
Jawa Tengah 53371
E-mail : fakhrurroziarfianto@gmail.com[1], iipunk@yahoo.co.id[2], ariefkelikn@gmail.com[3]

Abstract—Japanese has some letter's writing systems. Katakana letter is one of those. It used to write some absorption words from foreign language in Japanese. Katakana has several forms that has to learned by someone who's learning Japanese. It increases learning difficulties of Japanese. According to that, an application will build to help its user identify Katakana. It developed on desktop platform and it uses one of Artificial Intellegence's subdivision. It is Artificial Neural Network. This will be implemented in application and this is using Back Propagation learning algorithm model. Also, this will be built and trained to identify three shapes of Katakana, gojūon, dakuon, and handakuon. Various of learning parameters is applied and some learning processes is performed in this research. Those learning processess, which have been performed, obtain accuracy's percentage of artificial neural network in identifying these characters. Those are: 44,06% accuracy for gojūon; 48,75% accuracy for dakuon; and 82,5% accuracy for handakuon.

Abstrak—Bahasa Jepang mempunyai beberapa sistem aksara yang digunakan. Salah satunya adalah aksara *Katakana*, yang digunakan untuk menuliskan istilah bahasa Asing ke dalam bahasa Jepang. Aksara ini mempunyai beberapa bentuk yang harus dipelajari oleh seseorang yang sedang mempelajari bahasa Jepang. Hal ini menyebabkan tingkat kesulitan dalam mempelajari bahasa Jepang semakin tinggi. Maka, dibuatlah sebuah aplikasi yang akan membantu penggunanya mengidentifikasi aksara *Katakana*. Aplikasi ini dikembangkan dalam *platform desktop* dan menggunakan salah satu cabang ilmu dalam Kecerdasan Buatan, yaitu Jaringan Syaraf Tiruan (*Artificial Neural Network*). Jaringan syaraf tiruan yang diimplementasikan dalam aplikasi tersebut menggunakan model algoritma pembelajaran *Back Propagation*. Jaringan syaraf tiruan tersebut dibuat dan dilatih untuk mengidentifikasi tiga jenis aksara *Katakana*, yaitu aksara *gojūon*, *dakuon*, dan *handakuon*. Beberapa parameter pembelajaran diterapkan dan beberapa proses pembelajaran dilakukan pada penelitian ini. Pembelajaran jaringan syaraf tiruan yang telah dilakukan menghasilkan tingkat akurasi jaringan syaraf tiruan dalam mengidentifikasi aksara-aksara tersebut. Tingkat akurasi tersebut adalah: 44,06% untuk aksara *gojūon*; 48,75% untuk aksara *dakuon*; dan 82,5% untuk aksara *handakuon*.

Kata Kunci—back propagation, citra, jaringan syaraf tiruan, *katakana*.

I. PENDAHULUAN

Bahasa Jepang mempunyai beberapa sistem penulisan aksara. Sistem penulisan aksara ini terbagi atas dua sistem penulisan, yaitu aksara *Kana*dan aksara *Kanji*. Aksara *Kana* merupakan aksara asli bahasa Jepang yang merupakan simplifikasi dari bentuk aksara *Kanji* bahasa Mandarin/Cina. Sedangkan, aksara *Kanji* yang dipakai dalam bahasa Jepang merupakan aksara *Kanji* yang sama dengan aksara *Kanji* pada bahasa Mandarin/Cina. [1]

Aksara *Kana* dibagi lagi menjadi dua kelompok aksara, yaitu aksara *Hiragana* dan aksara *Katakana*. Aksara-aksara tersebut sama-sama digunakan untuk melambangkan bunyi/lafal, berbeda dengan sistem alfabet pada umumnya. Bunyi/lafal tersebut merupakan kombinasi dari satu huruf konsonan dan satu huruf vokal (contohnya *na*, *ha*, *ta*, dsb). Hanya terdapat satu aksara yang melambangkan bunyi huruf konsonan, yaitu konsonan *n*. [1]

Aksara *Katakana* merupakan salah satu jenis aksara *Kana*. Aksara ini digunakan untuk menuliskan kata-kata serapan yang berasal dari bahasa selain bahasa Mandarin/Cina. Sebagai contoh, kata *english* dalam bahasa Inggris akan diserap ke dalam bahasa Jepang menjadi エギリシ (baca: *egirisu*). Begitu pula dengan kata "helikopter" (inggris: *helicopter*) akan diserap ke dalam bahasa Jepang menjadi ヘリ (baca: *heri*).

Sistem penulisan aksara dalam bahasa Jepang, termasuk aksara *Katakana* yang sudah dijelaskan sebelumnya, harus dipelajari oleh setiap orang yang ingin belajar berbahasa Jepang. Tentu saja sistem penulisan aksara yang banyak ini akan menambah tingkat kesulitan dalam mempelajari bahasa Jepang. Oleh karena itu, dibutuhkan alat bantu yang dapat mengurangi tingkat kesulitan dalam mempelajari bahasa Jepang.

Salah satu alat bantu yang dapat dikembangkan untuk membantu para pembelajar bahasa Jepang yang sedang mempelajari aksara *Katakana* adalah alat bantu berbentuk aplikasi. Aplikasi yang dikembangkan akan

menggunakan salah satu bidang studi dalam ilmu komputer, yaitu Kecerdasan Buatan. Sehingga, aplikasi ini akan mampu mengidentifikasi dan mengenali aksara *Katakana* yang dimasukkan ke dalam aplikasi.

Kecerdasan Buatan/*Artificial Intellegence* merupakan metode pendekatan yang diterapkan terhadap komputer, sehingga komputer dapat berpikir dan bertingkah layaknya manusia (*thinking humanly* dan *acting humanly*) serta berpikir dan bertikah secara rasional (*thinking rationally* dan *acting rationally*). [2] Dengan demikian, Kecerdasan Buatan dapat dibagi menjadi beberapa cabang studi. Salah satunya adalah Jaringan Syaraf Tiruan.

Jaringan Syaraf Tiruan/*Artificial Neural Network* merupakan suatu sistem pemroses informasi yang mempunyai karakteristik yang hampir sama dengan jaringan syaraf biologis dan dibentuk sebagai generalisasi model matematika dari jaringan syarav biologis. Sehingga, pembentukan jaringan syaraf tiruan ditentukan oleh 3 hal, yaitu: [3]

1. Pola hubungan antar *node* atau *neuron* jaringan syaraf tiruan, atau dikenal pula sebagai arsitektur.
2. Metode untuk menentukan bobot penghubung (*weight*), atau dikenal pula sebagai algoritma pembelajaran.
3. Fungsi aktivvasi yang digunakan pada setiap *node* atau *neuron* dalam arsitektur jaringan syaraf tiruan.

II. METODE PENELITIAN

Penelitian ini dilakukan dengan prosedur-prosedur yang telah ditetapkan. Prosedur-prosedur tersebut dilakukan secara sekuensial (terurut). Dengan kata lain, hasil dari prosedur penelitian yang telah selesai akan digunakan pada prosedur penelitian selanjutnya. Prosedur-prosedur penelitian tersebut terdiri dari:

1. Prosedur pengumpulan data.
2. Prosedur *preprocessing* data
3. Prosedur pembentukan dan pembelajaran jaringan.
4. Prosedur pengembangan aplikasi

A. Prosedur Pengumpulan Data

Prosedur ini merupakan permulaan dari penelitian. Data-data dibutuhkan dalam penelitian akan dikumpulkan pada prosedur ini. Data yang digunakan pada penelitian ini adalah citra/gambar dari aksara *Katakana*. Aksara-aksara tersebut terdiri dari:

1. Aksara *Katakana* jenis *gojūon*, yaitu bentuk dasar dari aksara *Katakana*. Aksara ini terdiri atas 46 bentuk.
2. Aksara *Katakana* jenis *dakuon*, yaitu bentuk modifikasi dari aksara *Katakana gojūon* yang merepresentasikan bunyi/lafal dengan huruf konsonan *b*, *d*, *g*, *j*, dan *z*. Aksara ini terdiri atas 20 bentuk.
3. Aksara *Katakana* jenis *handakuon*, yaitu bentuk modifikasi dari aksara *Katakana* jenis *gojūon* yang merepresentasikan bunyi/lafal dengan huruf konsonan *p*. Aksara ini terdiri atas 5 bentuk.

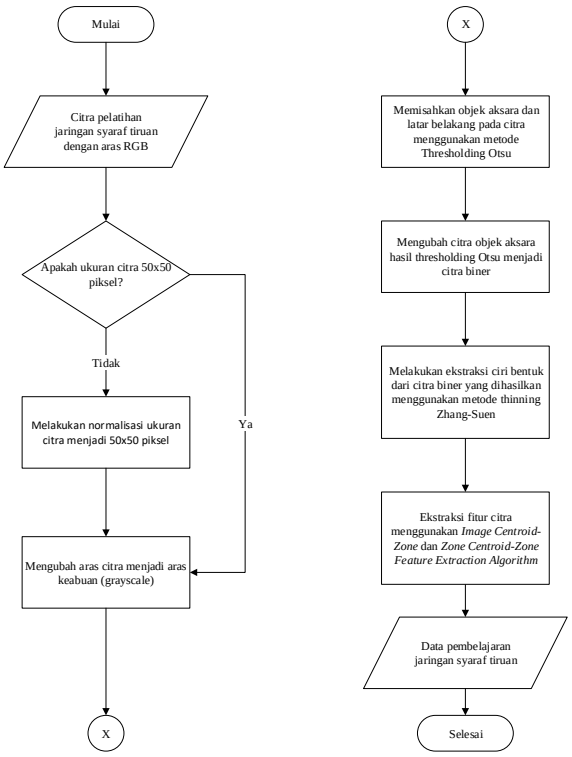
Masing-masing bentuk aksara *Katakana* tersebut mempunyai 20 buah citra yang dikumpulkan sebagai data penelitian. Citra tersebut terdiri dari 5 buah citra aksara tulisan cetak dan 15 buah citra aksara tulisan tangan.

B. Prosedur Preprocessing Data

Prosedur ini merupakan prosedur lanjutan setelah prosedur sebelumnya telah menghimpun data-data penelitian yang akan digunakan. Prosedur ini akan melakukan penyesuaian data penelitian, sehingga data penelitian tersebut dapat digunakan sepenuhnya dalam prosedur penelitian selanjutnya.

Prosedur penelitian ini terdiri dari beberapa proses yang akan dijalankan. Proses-proses tersebut adalah sebagai berikut (gambar 1 mencantumkan *flowchart* dari prosedur penelitian ini) .

1. Normalisasi ukuran citra aksara menjadi 50x50 piksel.
2. Pengubahan aras citra aksara menjadi aras keabuan (*grayscale*).
3. Pemisahan objek aksara dan latar belakang pada citra aksara menggunakan metode *thresholding Otsu*.
4. Pengubahan aras citra aksara menjadi aras biner.
5. Ekstraksi ciri bentuk dari citra aksara dengan menggunakan metode *thinning Zhang-Suen*.
6. Ekstrasi fitur dari hasil ekstrasi ciri bentuk citra aksara menggunakan algoritma ekstrasi fitur *Image Centroid and Zone* dan *Zone Centroid and Zone*.



Gambar 1. *Flowchart* prosedur *preprocessing* data

C. Prosedur Pembentukan dan Pembelajaran Jaringan

Prosedur penelitian ini dilakukan setelah dilakukan *preprocessing* citra menjadi data yang siap digunakan. Prosedur ini akan melakukan pembentukan dan pembelajaran jaringan syaraf tiruan yang akan digunakan untuk mengidentifikasi aksara *Katakana*. Prosedur ini akan menetapkan parameter-parameter yang dibutuhkan untuk membentuk dan melatih jaringan syaraf tiruan. Parameter-parameter tersebut adalah:

1. Jumlah *layer* pada jaringan syaraf tiruan.
2. Jumlah *neuron* pada tiap *layer* jaringan syaraf tiruan.
3. Nilai laju pembelajaran (*learning rate*).
4. Nilai momentum.
5. Nilai iterasi maksimum.

Jaringan syaraf tiruan akan dilatih menggunakan algoritma pembelajaran *Back Propagation*. Pada algoritma pembelajaran ini, arsitektur jaringan syaraf tiruan terbagi atas tiga *layer* utama, yaitu *layer input*/masukan, *layer hidden*/tersembunyi, dan *layer output*/keluaran. Urutan langkah-langkah dalam algoritma pembelajaran ini adalah sebagai berikut. [4]

1. Inisialisasi bobot (Bobot diberi nilai acak yang bernilai kecil).
2. Jika kondisi berhenti salah, kerjakan 2-9.
3. Untuk tiap pasangan pola masukan, kerjakan 3-8.
4. Tiap unit masukan (x_i , i = 1, … , n) menerima sinyal masukan x_i dan meneruskan sinyal ini ke semua unit pada lapisan diatasnya (lapisan tersembunyi).

$$z_{in\_j} = v_{0j} + \sum_{i=1}^{n} x_i v_{ij} \quad (1)$$

z_{in_j} : sinyal masukan x_i menuju *hidden neuron j*
v_{0j} : bias pada *hidden neuron j*
x_i : masukan dari *input neuron i*
v_{ij} : bobot masukan x_i terhadap *hidden neuron j*

5. Tiap unit tersembunyi (z_j, j = 1, … , p) hitung sinyal masukan

$$z_j = f(z_{in\_j}) \quad (2)$$

z_{in_j} : sinyal masukan x_i menuju *hidden neuron j*
z_j : masukan dari *hidden neuron j*

Fungsi aktivasi yang digunakan adalah sigmoid biner dan kirim sinyal ini ke semua unit pada lapisan diatasnya (unit keluaran).

$$y_{in\_k} = w_{0k} + \sum_{j=1}^{p} z_j w_{jk} \quad (3)$$

y_{in_k} : sinyal masukan dari z_j menuju *output neuron k*
w_{0k} : bias pada *output neuron k*
z_j : masukan dari *hidden neuron j*
w_{jk} : bobot masukan z_j terhadap *output neuron k*

6. Tiap unit keluaran (y_k, k = 1, … , m) hitung sinyal masukan berbobotnya, dan terapkan fungsi aktivasinya untuk menghitung sinyal keluaran dengan persamaan (4).

$$y_k = f(y_{in_k}) \quad (4)$$

y_{in_k} : sinyal masukan z_j menuju *output neuron k*
y_k : keluaran dari *output neuron k*

7. Tiap unit keluaran (y_k , k = 1, … , m) menerima pola target yang bersesuaian dengan pola masukan, dihitung parameter informasi galat dengan persamaan (5).

$$\delta_k = (t_k - y_k)\, f'(y_{in\_k}) \qquad (5)$$

δ_k : tingkat galat keluaran *output neuron k*
t_k : target keluaran dari *output neuron k*
y_k : keluaran dari *output neuron k*
y_{in_k} : sinyal masukan dari z_j menuju *output neuron k*

Parameter koreksi bobotnya (dipakai untuk mengubah w_{jk}) dihitung dengan persamaan (6) dan persamaan (7).

$$\Delta w_{jk}(t) = \beta \delta_k z_j \qquad (6)$$

$$\Delta w_{jk}(t+1) = \beta \delta_k z_j + \alpha \Delta w_{jk}(t) \qquad (7)$$

$\Delta w_{jk}(t)$: perubahan bobot pada iterasi sebelumnya
$\Delta w_{jk}(t+1)$: perubahan bobot pada iterasi saat ini
δ_k : tingkat galat keluaran *output neuron k*
β : laju pembelajaran/*learning rate*
α : momentum
z_j : masukan dari *hidden neuron j*

Sedangkan, parameter koreksi bias (dipakai untuk mengubah w_{0k}), dihitung dengan persamaan (8) dan persamaan (9),

$$\Delta w_{0k} = \beta \delta_k \qquad (8)$$

$$\Delta w_{0k}(t+1) = \beta \delta_k + \alpha \Delta w_{0k}(t) \qquad (9)$$

$\Delta w_{jk}(t)$: perubahan bias pada iterasi sebelumnya
$\Delta w_{jk}(t+1)$: perubahan bias pada iterasi saat ini
δ_k : tingkat galat keluaran *output neuron k*
β : laju pembelajaran/*learning rate*
α : momentum

z_j : masukan dari *hidden neuron j*

8. Pada tiap unit tersembunyi (z_j , j = 1, … , p) input δ_{in_j} (dari unit pada lapisan dibawahnya) dihitung dengan persamaan (10),

$$\delta_{in\_j} = \sum_{k=1}^{m} \delta_k w_{jk} \qquad (10)$$

δ_{in_j} : masukan galat pada *hidden neuron k*
δ_k : tingkat galat keluaran *output neuron k*
w_{jk} : bobot masukan z_j terhadap *output neuron k*

Parameter informasi galat dihitung dengan persamaan (11).

$$\delta_j = \delta_{in\_j}\, f'(z_{in\_j}) \qquad (11)$$

δ_j : tingkat galat keluaran *hidden neuron j*
δ_{in_j} : masukan galat pada *hidden neuron j*
z_{in_j} : sinyal masukan x_i menuju *hidden neuron j*

Koreksi bobot yang dipakai untuk mengubah v_{ij} dihitung dengan persamaan (12) dan (13)

$$\Delta v_{ij}(t) = \beta \delta_j x_i \qquad (12)$$

$$\Delta v_{ij}(t+1) = \beta \delta_j x_i + \alpha \Delta v_{ij}(t) \qquad (13)$$

$\Delta v_{ij}(t)$: perubahan bobot pada iterasi sebelumnya
$\Delta v_{ij}(t+1)$: perubahan bobot pada iterasi saat ini
δ_j : tingkat galat keluaran *hidden neuron j*
β : laju pembelajaran/*learning rate*
α : momentum
x_i : masukan dari *input neuron i*

Parameter koreksi bias yang dipakai untuk mengubah v_{0j} dihitung dengan persamaan (14) dan perbaruan bias dihitung dengan persamaan (15)

$$\Delta v_{0j}(t) = \beta \delta_j \qquad (14)$$

$$\Delta v_{0j}(t+1) = \beta \delta_j + \alpha \Delta v_{0j}(t) \qquad (15)$$

$\Delta v_{0j}(t)$: perubahan bias pada iterasi sebelumnya
$\Delta v_{0j}(t+1)$: perubahan bias pada iterasi saat ini
δ_j : tingkat galat keluaran *hidden neuron j*
β : laju pembelajaran/*learning rate*
α : momentum

9. Tiap unit keluaran (y_k ,k = 1, … , m) perubahan bobot dan bias dilakukan berdasarkan persamaan (16) dan persamaan (17) (j= 0, … , p).

$$w_{jk}(t+1) = w_{jk}(t) + \Delta w_{jk}(t+1) \quad (16)$$

$$w_{0k}(t+1) = w_{0k}(t) + \Delta w_{0k}(t+1) \quad (17)$$

$w_{jk}(t+1)$: nilai bobot baru untuk iterasi selanjutnya
$w_{jk}(t)$: nilai bobot lama pada iterasi saat ini
$\Delta w_{jk}(t+1)$: perubahan bobot pada iterasi saat ini
$w_{0k}(t+1)$: nilai bias baru untuk iterasi selanjutnya
$w_{0k}(t)$: nilai bias lama pada iterasi saat ini
$\Delta w_{0k}(t+1)$: perubahan bias pada iterasi saat ini

Pada tiap unit tersembunyi (z_j,j= 1, … ,p) bias dan bobotnya diperbaharui dengan persamaan (17) dan persamaan (18) (i = 0, … , n).

$$v_{ij}(t+1) = v_{ij}(t) + \Delta v_{ij}(t+1) \quad (17)$$
$$v_{0j}(t+1) = v_{0j}(t) + \Delta v_{0j}(t+1) \quad (18)$$

$v_{ij}(t+1)$: nilai bobot baru untuk iterasi selanjutnya
$v_{ij}(t)$: nilai bobot lama pada iterasi saat ini
$\Delta v_{ij}(t+1)$: perubahan bobot pada iterasi saat ini
$v_{0j}(t+1)$: nilai bias baru untuk iterasi selanjutnya
$v_{0j}(t)$: nilai bias lama pada iterasi saat ini
$\Delta v_{0j}(t+1)$: perubahan bias pada iterasi saat ini

10. Uji kondisi berhenti

Pada urutan tahap inisialisasi bobot, digunakanlah algoritma inisialisasi bobot *Nguyen-Widrow*. Algoritma ini menggunakan koefisien *n* yang melambangkan jumlah *neuron* pada *layer input*, koefisien *p* yang melambangkan jumlah *neuron* pada *layer hidden*, dan koefisien β yang melambangkan faktor skala yang dihitung menggunakan persamaan $\beta = 0.7\sqrt[n]{p}$. Urutan langkah dalam algoritma *Nguyen-Widrow* ini adalah sebagai berikut. [3]

a. Inisialisasi bobot awal v_{ji} dengan bilangan acak dalam rentang bilangan -0.5 sampai 0.5
b. Menghitung nilai $\|v_j\|$ dengan persamaan $\|v_j\| = \sqrt{v_{j1}^2 + v_{j2}^2 + \cdots + v_{jn}^2}$.
c. Menghitung bobot awal yang akan digunakan pada pelatihan jaringan syaraf tiruan dengan persamaan $v_{ji} = \frac{\beta\, v_{ji\,(lama)}}{\|v_j\|}$.
d. Menginisialisasi nilai bias awal v_{j0} yang digunakan pada pelatihan dengan bilangan acak diantara -β dan β.

D. Prosedur Pengembangan Aplikasi

Prosedur ini dilakukan untuk mengimplementasikan hasil pembentukan dan pembelajaran jaringan syaraf tiruan yang dilakukan pada prosedur sebelumnya ke dalam suatu aplikasi. Aplikasi yang akan dikembangkan menggunakan metode pengembangan sistem *prototyping*. Metode pengembangan sistem ini terdiri atas beberapa tahap, yaitu *communication, quick plan, modelling, construction of prototype,* dan *deployment, delivery, and feedback*. [5]

III. HASIL DAN PEMBAHASAN

A. Data Penelitian

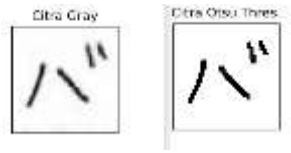
Data penelitian yang dikumpulkan sebanyak 1420 buah citra aksara *Katakana*. Citra aksara tersebut terdiri dari 920 buah citra aksara *Katakana* jenis *gojūon*, 400 buah citra aksara *Katakana* jenis *dakuon*, dan 100 buah citra aksara *Katakana* jenis *handakuon*.

B. Preprocessing Data

Prosedur *preprocessing* citra aksara diawali dengan normalisasi ukuran citra aksara. Ukuran citra aksara akan dibuat seragam menjadi 50 x 50 piksel. Selanjutnya, prosedur *preprocessing* dilanjutkan dengan mengubah aras citra (dengan aras RGB –*Red, Green, Blue*-) menjadi aras keabuan. Hal ini dikarenakan sebagian besar pemrosesan citra dimulai dari aras keabuan (*grayscale*). Gambar 2 menunjukkan perubahan dari citra awal menjadi citra beraras keabuan (*grayscale*)

Gambar 2. *Preprocessing* citra RGB → citra *grayscale*

Citra beraras keabuan/*grayscale* akan diproses kembali untuk memisahkan objek citra dan latar belakang citra. Tahap ini akan menggunakan metode pengambangan atau *thresholding Otsu*. Metode ini akan menghitung nilai ambang atau *threshold* intensitas pada piksel-piksel citra, sehingga nilai ambang atau *threshold* tersebut memisahkan objek aksara dari latar belakangnya ketika diterapkan pengambangan. [6] Gambar 3 menunjukkan hasil pengambangan menggunakan metode *Otsu*.



Gambar 3. Hasil *preprocessing* citra *grayscale* dengan *Otsu thresholding*

Prosedur *preprocessing* akan dilanjutkan kembali dengan pengubahan aras citra *grayscale* menjadi aras biner. Tahap *preprocessing* ini akan mengubah nilai intensiteas piksel dari nilai intensitas 0-255 (gradasi hitam ke putih) menjadi nilai intensitas 0-1 (hitam dan putih). Nilai intensitas 1 (warna putih) menunjukkan objek aksara pada citra.

Tahap *preprocessing* yang dilakukan berikutnya adalah mengekstrak bentuk dari objek aksara. Tahap ini dapat dilakukan dengan melakukan skeletonisasi, yaitu mengambil kerangka dari objek. *Skeleton* atau kerangka tersebut adalah bentuk unik dari suatu objek yang menyerupai rangka yang mempunyai karakteristik, ketebalannya 1 mm, melewati tengah objek, dan menyatakan topologi dari objek. [6]

Tahap *preprocessing* ini menggunakan metode *thinning* atau penipisan. Metode ini digunakan untuk mengurangi nilai piksel-piksel pada objek, sehingga menghasilkan bentuk objek paling sederhana. [7]

Metode *thinning* yang akan diterapkan ini menggunakan algoritma *Zhang-Suen*. Algoritma ini akan melakukan dua kali pemeriksaan untuk menghilangkan piksel objek, yaitu piksel pada bagian kanan bawah (tenggara) objek untuk pemeriksaan pertama dan piksel pada kiri atas (barat laut) pada pemeriksaan kedua. [8] Gambar 4 menunjukkan hasil *thinning* menggunakan algoritma ini pada citra aras biner.

Gambar 4. Hasil *preprocessing* citra *grayscale Otsu* →citra *skeleton* dengan algoritma *Zhang-Suen*

Tahap *preprocessing* terakhir adalah mengekstrasi fitur objek pada citra. Fitur ini diekstrak dari citra *skeleton* objek aksara menggunakan algoritma ekstraksi fitur *Image Centroid and Zone* dan *Zone Centroid and Zone*. Algortima ini berbasis pada zona (*zone-based*), sehingga citra aksara akan dibagi menjadi beberapa zona dengan ukuran tiap zona yang seragam. Algoritma ini akan menghasilkan fitur citra yang mempunyai *robustness* terhadap variasi yang sempit, implementasi algoritma yang mudah, dan memberikan hasil pengenalan yang lebih baik. [9]

Algoritma ekstraksi yang digunakan pada permulaan ektraksi fitur adalah algoritma *Image Centroid and Zone*. Algoritma ini akan mencari pusat massa objek pada citra terlebih dahulu. Setelah itu, algoritma ini akan menghitung jarak pusat massa objek aksara dan potongan objek aksara pada tiap zona yang ditentukan. [9]

Algoritma ekstraksi yang digunakan selanjutnya adalah algoritma *Zone Centroid and Zone*. Algoritma ini akan mencari pusat massa potongan objek aksara pada tiap zona terlebih dahulu. Kemudian, algoritma ini akan menghitung jarak pusat massa potongan objek aksara dengan piksel objek aksara pada zona yang sama. [9]

Tahap *preprocessing* ini akan menghasilkan dua macam fitur. Fitur tersebut adalah fitur hasil operasi algoritma *Image Centroid and Zone* dan algoritma *Zone Centroid and Zone*. Sedangkan, pada tahap *preprocessing* ini, tiap citra objek aksara akan dibagi menjadi 25 zona (ukuran tiap zona sebesar 10 x 10 piksel) dan 50 zona (ukuran tiap zona sebesar 5 x 10 piksel). Sehingga, fitur yang dihasilkan sebanyak 50 fitur (25 fitur *Image Centroid and Zone* dan 25 fitur *Zone Centroid and Zone*) dan 100 fitur (50 fitur *Image Centroid and Zone* dan 50 fitur *Zone Centroid and Zone*). Kedua himpunan fitur ini akan digunakan pada prosedur pembentukan dan pembelajaran jaringan syaraf tiruan.

C. Prosedur Pembentukan dan Pembelajaran Jaringan

Prosedur ini dilakukan untuk membentuk dan melatih jaringan syaraf tiruan yang akan digunakan untuk mengidentifikasi aksara *Katakana* (*gojūon, dakuon, handakuon*). Prosedur ini diawali dengan membentuk arsitektur jaringan syaraf tiruan, kemudian melatih dan menguji jaringan syaraf tiruan tersebut.

Jaringan syaraf tiruan yang akan dibentuk terbagi atas tiga macam. Jaringan syaraf tiruan tersebut terbagi berdasarkan jenis aksara *Katakana* yang akan diidentifikasi. Sehingga, setiap jenis aksara *Katakana* diidentifikasi dengan jaringan syaraf tiruan yang berbeda.

Arsitektur dari masing-masing jaringan syaraf tiruan yang dibuat terdiri dari satu lapisan *input*, satu lapisan *hidden*, dan satu lapisan *output*. Sedangkan, jumlah *neuron* pada tiap lapisan yang akan dilatih dan diuji tercantum pada tabel 1. Sedangkan

Tabel 1. Jumlah *neuron* pada tiap lapisan jaringan

Pembelajar	JST	Neuron		
		Input	hidden	output
1 dan 2	*Gojūon*	50	100	46
	Dakuon		80	20
	Handakuon		60	5
3	*Gojūon*	100	170	46
	Dakuon		130	20
	Handakuon		80	5

Tabel 2. Parameter pembelajaran pada tiap jaringan

Pembelajar	JST	Parameter		
		β	α	Iter_max
1	*Gojūon*	0,003	0,05	50000
	Dakuon			
	Handakuon			
2	*Gojūon*	0.01	0,05	
	Dakuon			
	Handakuon			
3	*Gojūon*	0,003	0,05	
	Dakuon			
	Handakuon			

Dari beberapa kali pembelajaran yang dilakukan dengan arsitektur dan parameter pembelajaran pada tabel 1 dan tabel 2, maka dihasilkan nilai akurasi jaringan syaraf tiruan dalam mengidentifikasi aksara *Katakana* yang diinginkan. Nilai akurasi tersebut dihitung dengan persamaan $\%\,akurasi = \frac{aksara\ yang\ dikenali}{jumlah\ aksara\ yang\ diidentifikasi}\ x\ 100$. Nilai akurasi pada tiap pembelajaran jaringan tersebut tercantum pada tabel 3.

Tabel 3. Nilai akurasi (%) pada tiap jaringan

Pembelajar	JST	Nilai akurasi	
		Pelatihan (%)	Pengujian (%)
1	*Gojūon*	18,49%	21,73%
	Dakuon	72,5%	38,75%
	Handakuon	98,75%	85%
2	*Gojūon*	13,72%	8,152%
	Dakuon	77,5%	46,25%
	Handakuon	100%	82,5%
3	*Gojūon*	78,26%	44,02%
	Dakuon	96,25%	48,75%
	Handakuon	100%	82,5%

D. Prosedur Pengembangan Aplikasi

Prosedur penelitian terakhir ini akan menggunakan jaringan syaraf tiruan yang telah dilatih dan diuji pada prosedur sebelumnya. Jaringan syaraf tiruan yang akan digunakan adalah jaringan syaraf tiruan hasil pembelajaran ketiga.

Hasil identifikasi aksara pada aplikasi berkaitan secara langsung dengan nilai akurasi jaringan syaraf tiruan. Tabel 4 menampilkan beberapa hasil identifikasi aksara pada aplikasi.

Tabel 4. Contoh hasil identifikasi aksara *Katakana* pada aplikasi

Aksara	Lafal sebenarnya	Lafal hasil identifikasi
パピプペポ	PAPIPUPEPO	PAPIPUPEPO
ドイツ	DOITSU	IITSU
アメリカ	AMERIKA	ANOWOKI
エギリス	EGIRISU	BIZERINO

IV. KESIMPULAN

Berdasarkan prosedur-prosedur penelitian yang telah dilakukan, kesimpulan yang dapat didapatkan adalah:

1) Jaringan syaraf tiruan yang telah dibuat, dilatih, dan diuji masih belum mengenali dan mengidentifikasi aksara *Katakana* dengan baik. Hal ini dibuktikan dengan nilai akurasi jaringan *gojūon* dan jaringan *dakuon* belum mencapai nilai lebih dari 75%. Hanya jaringan *handakuon* yang mencapai nilai akurasi lebih dari 75%.
2) Kemungkinan jaringan syaraf tiruan yang belum mengenali aksara *Katakana* dengan baik pada penelitian ini dikarenakan:
   a) Keterbatasan citra pembelajaran yang digunakan.

b) Fitur yang diekstraksi masih belum merepresentasi keunikan masing-masing citra.
c) Parameter-parameter pembelajaran jaringan syaraf tiruan yang masih belum optimum.

DAFTAR PUSTAKA

[1] Henshall, K. G. & Takagaki, T., *Learning Japanese Hiragana and Katakana: Workbook and Practice Sheets*. 2nd ed. North Clrendon, Vermont: Tuttle Publishing (2006) .

[2] Suyanto, *Artificial Intellegence: Searching, Reasoning, Planning, dan Learning*. Edisi kedua. Bandung: Penerbit Informatika (2014) .

[3] Siang, J. J., *Jaringan Syaraf Tiruan dan Pemrogramannya menggunakan MATLAB*. Yogyakarta: Penerbit Andi (2009) .

[4] Asriani, F. & Nugraha, A. W. W., "Ekstraksi Ciri Batang untuk Pengenalan Nomor Rekening Tulisan Tangan dengan Jaringan Syaraf Tiruan Perambatan Balik," *Dinamika Rekayasa*, Vol. 8 ED-2 (2012, Aug.) 36–41.

[5] Pressman, R. S., *Software Engineering: A Practitoner's Approach*. 7th ed. New York: McGraw-Hill (2001) .

[6] Kadir, A. & Susanto, A., *Teori dan Aplikasi Pengolahan Citra*. Yogyakarta: Penerbit Andi (2013) .

[7] Fisher, R., Prekins, S., Walker, A. & Wolfart, E. (2018, July). *Morphology-Thinning*. [Online]. Available: https://homepages.inf.ed.ac.uk/rbf/HIPR2/thin.htm

[8] Anonymous. (2018, August). *Zhang-Suen Thinning Algorithm*. [Online]. Available: http://rstudio-pubs-static.s3.amazonaws.com/302782_e337cfbc5ad24922bae96ca5977f4da8.html

[9] Rajashekararadhya, R. S. & Ranjan, P. V. ., "Efficient Zone Based Feature Extraction Algorithm for Handwritten Numeral Recognition of Four Popular South Indian Scripts," *Journal of Theoretical ad Applied Information Technology*, Vol. IV (2008) 1171–1181.