

PENGENALAN POLA TULISAN TANGAN AKSARA SASAK MENGUNAKAN METODE LINEAR DISCRIMINANT ANALYSIS DAN JARINGAN SYARAF TIRUAN JENIS BACKPROPAGATION

(*Handwritten Sasak Script Recognition Using Linear Discriminant Analysis And
Backpropagation Artificial Neural Network*)

A.A.Sg.Mas Karunia Maharani*, Fitri Bimantoro

Program Studi Teknik Informatika, Fakultas Teknik, Universitas Mataram

Jl. Majapahit 62, Mataram, Lombok NTB, INDONESIA

Email: karuniamaharani27@gmail.com, bimo@unram.ac.id

Abstract

Sasak's script is thr one of Indonesia's cultural heritages which is endangered language category and it needs to be preserved so its not become extinct script. If there is no preservation, Sasak's script will be close to extinction. Given the importance of conservation efforts, a reserch was carried out to recognize the Sasak's script handwriting patterns. This reserch aims to see the optimal model for recognizing Sasak script patterns and to see how accurate the Linear Discriminant Analysis method as extraction features and Backpropagation Neural Network as a classification method for recognizing Sasak's script patterns. The dataset in this reserch consisted of three variations amount of data which are 13500, 10800 and 2700. According to the final results, the best accuracy of the proper model was obtained in DCT coefficient test parameter 64, eigen value are 17, number of neurons are 128, the number of hidden layers are 2 and learning rate 0.003 on the 10800 dataset with 92.20% accuracy, 92% precision and 92% recall with 250,691s computing time. This shows that the LDA and ANN-BP methods can work optimally to recognize Sasak's script pattern.

Keywords: *Pattern Recognition, Handwriting, Sasak Script, LDA, Backpropagation.*

*Penulis Korespondensi

1. PENDAHULUAN

Aksara Sasak merupakan salah satu warisan budaya Indonesia yang perlu dilestarikan agar tidak punah. Namun penggunaan aksara Sasak makin berkurang belakangan ini. Jika tidak ada upaya untuk mengenalkan kembali aksara ini, aksara Sasak akan masuk kedalam kategori *endangered language*. *Endangered languages* adalah bahasa-bahasa yang terancam punah yang tidak memiliki generasi muda sebagai penuturnya dan penutur yang fasih hanyalah kelompok generasi menengah atau dewasa [1]. Ini artinya bahwa aksara Sasak akan kehilangan penggunaannya karena yang hanya bisa menulis dan membacanya hanyalah kelompok generasi. Sistem pendidikan juga menjadi faktor yang berpengaruh terhadap pelestarian budaya. Salah satu aksara di Indonesia yaitu aksara Sasak sudah jarang digunakan karena tidak diajarkan lagi di sekolah-sekolah. Sehingga dikhawatirkan aksara Sasak akan mengalami kepunahan. Mengingat betapa pentingnya upaya pelestarian, maka dilakukan penelitian untuk mengenali aksara Sasak dengan menggunakan metode *Linear Discriminant Analysis* (LDA) dan Jaringan Syaraf

Tiruan *Backpropagation* (JST-BP). Aplikasi ini selanjutnya dapat digunakan sebagai media pembelajaran untuk mendukung pengenalan aksara Sasak.

Penelitian sebelumnya mengenai aksara Sasak menggunakan metode *moment invariant* dan SVM memiliki tingkat akurasi 89.76% untuk 63 fitur dan 92.52% untuk 112 fitur [2]. Penelitian lainnya menggunakan metode *integral projection* untuk ekstraksi fitur dan metode PCA untuk reduksi dimensi dan *neural network* sebagai metode klasifikasi aksara dengan tingkat akurasi sistem sebesar 41,38% [3]. Penelitian [3] membandingkan kinerja metode PCA dengan LDA, hasil meunjukkan bahwa akurasi metode LDA lebih tinggi. LDA merupakan metode yang digunakan untuk mengatasi kekurangan PCA. Penelitian pengenalan pola wajah menggunakan metode LDA memiliki tingkat akurasi sebesar 80% [4]. Penelitian mengenai pengenalan garis telapak tangan dengan menggunakan metode LDA memiliki akurasi sebesar 93% [5]. Selain metode LDA, penelitian ini juga menggunakan metode *neural network* dalam pengklasifikasiannya. Penelitian mengenai pengenalan pola aksara Jawa dengan menggunakan metode

backpropagation memiliki tingkat akurasi 99.8% untuk data latih dan 95.81% untuk data uji [6]. Penelitian lainnya pola aksara Jawa menggunakan metode JST-BP memiliki tingkat keakuratan yaitu sebesar 99.56% 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 [7]. Penelitian pengenalan pola aksara Jawa menerapkan metode *thinning* serta pembagian citra menjadi 16 subcitra dengan metode perambatan-balik memiliki akurasi 75% [8].

Berdasarkan uraian diatas, peneliti merancang model pembelajaran mesin untuk mengenali pola tulisan tangan aksara Sasak dengan menggunakan metode LDA dan JST-BP. Diharapkan kedua metode ini dapat diterapkan untuk mengenali pola tulisan tangan aksara Sasak sehingga ke depannya kedua metode ini dapat dikembangkan untuk sistem pembelajaran aksara Sasak.

2. TINJAUAN PUSTAKA

Penelitian sebelumnya mengenai pengenalan pola aksara Sasak yaitu menggunakan metode *support vector machine* (SVM) dengan akurasi 92.52% [2] dan penelitian lainnya menggunakan ekstraksi fitur *integral projection* dan klasifikasi NN dengan akurasi 41.38% [3]. Kedua penelitian menggunakan *dataset* sejumlah 2700 data. Penelitian tentang klasifikasi menggunakan metode JST-BP sudah pernah dilakukan beberapa kali pada interval tahun 2015-2019. Penelitian-penelitian tersebut antara lain tentang klasifikasi wajah [9], klasifikasi huruf vokal [10], identifikasi otentifikasi citra tanda tangan [11], klasifikasi pola sidik jari [12], klasifikasi aksara jawa [13]. Penelitian terkait metode LDA untuk ekstraksi fitur sebelumnya telah dilakukan yaitu untuk pengenalan wajah [4], angka meteran air otomatis dengan menggunakan 10 angka 0-9 dan *dataset* yang digunakan berjumlah 900 citra, akurasi yang dihasilkan yaitu 98% [14] dan pengenalan telapak tangan [5].

Penelitian lainnya mengenai *handwriting recognition* dengan menggunakan metode JST-BP terdapat 26 kelas dengan jumlah *dataset* 910 citra, akurasi tertinggi dari pengujian tersebut adalah 91% [15]. Penelitian lainnya yaitu pengenalan pola Aksara Jawa dengan menggunakan beberapa metode ANN yaitu diantaranya CPN, ENN 1 layer, ENN 2 layer dan JST-BP dengan akurasi secara berurutan yaitu 71.45%, 94.19%, 92.26% dan 98.71%. *Dataset* dari penelitian tersebut berjumlah 620 data dengan *output* sebanyak 20 *neuron*. Metode JST-BP merupakan metode dengan akurasi tertinggi pada penelitian ini [16]. Penelitian

terkait dengan metode JST-BP juga ada pada penelitian klasifikasi pola huruf vokal dengan jumlah kelas sebanyak 5 kelas dan jumlah *dataset* yang digunakan yaitu berjumlah 450 citra. Skenario training dan testing menghasilkan akurasi sebesar 84% dan 76% [17]. Pada penelitian *Farsi Character*, metode yang digunakan yaitu JST-BP, akurasi yang dihasilkan yaitu 85% dengan jumlah *dataset* sebanyak 250 citra dan terdapat 32 kelas sesuai dengan jumlah karakter [18]. Penelitian lainnya yaitu *optical character recognition* (OCR), dengan menggunakan 62 karakter dan *dataset* sebanyak 558 citra menghasilkan akurasi sebesar 99% untuk citra digit numerik (0~9), huruf kapital (A~Z) menghasilkan akurasi sebesar 97%, huruf kecil (a~z) menghasilkan akurasi sebesar 96% dan untuk *alphanumeric character* akurasi yang dihasilkan sebesar 93% [19].

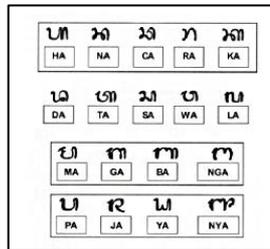
Berdasarkan penelitian [20], metode LDA memiliki beberapa keunggulan dibandingkan metode yang lain seperti relatif mudah diimplementasikan karena hanya memiliki *co-occurrent* dan nilai rata-rata global (μ) dari *Eigen Analysis*. Pada ekstraksi fitur dengan LDA, *dataset* lokasinya tetap, namun kelas yang dibentuk menjadi lebih terpisah, kondisi ini disebabkan oleh jarak antar data pelatihan dalam satu kelas menjadi lebih kecil [21].

Berdasarkan penelitian-penelitian yang telah dilakukan sebelumnya, dapat dilihat bahwa JST-BP dapat bekerja dengan baik untuk pengklasifikasian citra. Kelebihan metode ini mampu memformulasikan pengalaman dan sangat fleksibel dalam perubahan aturan perkiraan [22]. LDA juga baik digunakan untuk ekstraksi fitur karena meminimalkan persebaran dalam kelas (*within class*) dan memaksimalkan persebaran antar kelas (*between class*), hal ini dapat memudahkan persebaran data yang nantinya akan diolah [4]. Oleh karena itu, penulis bermaksud untuk menggunakan metode LDA sebagai metode ekstraksi fitur dan JST-BP sebagai metode klasifikasi citra aksara sasak.

2.1 Aksara Sasak

Aksara merupakan suatu simbol visual yang tertera pada suatu media (kertas, kain) untuk mengungkapkan unsur-unsur yang ekspresif dalam suatu bahasa. Aksara digunakan untuk secara khusus menuliskan bahasa daerah tertentu. Salah satu bahasa daerah nusantara yang digunakan di Lombok adalah bahasa Sasak dan ditulis dengan menggunakan aksara Sasak [23]. Aksara berfungsi sebagai media penulisan untuk membaca karya sastra kuno berbahasa daerah. Huruf sasak merupakan salah satu aksara tradisional nusantara yang digunakan oleh masyarakat suku Sasak

di Lombok, Indonesia untuk menulis bahasa daerah yaitu bahasa sasak, yang terdiri dari 18 karakter dasar [19]. Semua karakter huruf sasak dapat dilihat pada Gambar 1.

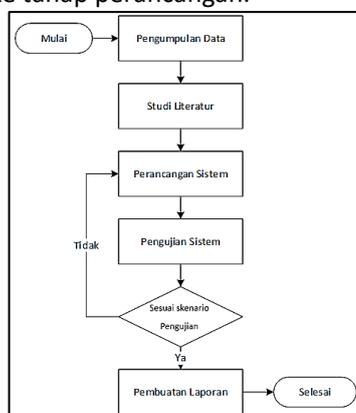


Gambar 1 Karakter aksara Sasak [2]

3. METODE PENELITIAN

3.1 Diagram Alir Penelitian

Diagram alir proses penelitian yang diawali dengan proses pengumpulan data, seperti yang digambarkan pada Gambar 2. Data yang dikumpulkan pada proses pengumpulan citra aksara berjumlah 18 karakter. Proses pengambilan citra aksara berdasarkan empat kategori, yaitu berdasarkan tulisan tangan anak-anak SD, SMP dan SMA serta tulisan tangan Mahasiswa. Langkah kedua yakni studi literatur untuk mempelajari cara membangun sistem sesuai dengan metode yang digunakan. Langkah ketiga adalah tahap perancangan system dari penelitian yang akan dilakukan seperti tahap analisa dan *coding*. Setelah melalui tahap perancangan sistem, dilakukan tahap pengujian sistem untuk menguji apakah sistem yang telah dibuat sudah berfungsi sesuai dengan tujuan. Sistem dikatakan sesuai jika sistem sudah mampu melakukan *training* dan klasifikasi data uji kedalam kelas tertentu pada suatu data uji, dan mengklasifikasikan data baru tersebut ke suatu kelas tertentu. Jika sesuai maka proses dilanjutkan ke tahap selanjutnya yaitu pembuatan laporan, jika tidak maka proses akan kembali ke tahap perancangan.



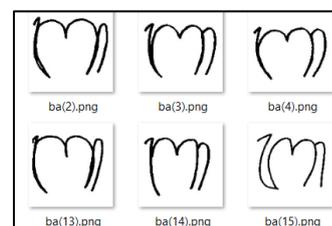
Gambar 2. Diagram alir proses penelitian.

3.2 Pengambilan Dataset

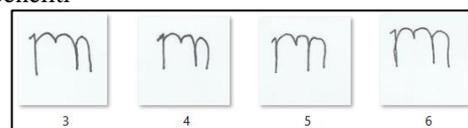
Pengambilan citra aksara sasak menggunakan *template* dari kolom tabel dengan tinggi dan lebar yang di *setting* 4cmx4cm dengan banyak kotak sebanyak 18 kotak atau tabel sesuai dengan jumlah karakter aksara sasak yaitu 18 buah. Sumber pengambilan data tulisan tangan aksara sasak dibagi menjadi empat kategori yaitu SD, SMP, SMA dan Perguruan Tinggi. Pembagian kategori pada pengambilan data dilakukan untuk memperoleh keberagaman data dari sisi pendidikan dan pernah mempelajari aksara sasak sebelumnya. Keempat kategori dibagi lagi menjadi orang yang pernah mempelajari penulisan aksara sasak dan orang yang tidak pernah belajar penulisan aksara Sasak. Masing-masing kategori melibatkan sepuluh orang dan masing-masing orang menulis 18 karakter sebanyak 15 kali. Data yang terkumpul sebanyak 10800 citra. Data yang terkumpul lalu di *scan* dengan dengan resolusi tinggi. Pada penelitian sebelumnya terdapat 2700 data yang terkumpul dengan sistematis pengambilan data yang sama yaitu dengan menggunakan tulisan tangan manual. Data ini akan digunakan sebagai data pembandingan pada saat proses pelatihan dan klasifikasi citra. Tabel I menunjukkan perbedaan kedua *dataset*.

TABEL I. PERBANDINGAN KEDUA DATASET

Dataset 10800	Dataset 2700
<ul style="list-style-type: none"> • Pengambilan data berdasarkan empat kategori yaitu SD,SMP, SMA dan Perguruan Tinggi 	<ul style="list-style-type: none"> • Pengambilan data hanya melibatkan satu kategori yaitu Perguruan Tinggi.
<ul style="list-style-type: none"> • Responden sebanyak 40 orang di mana masing-masing kategori terdiri atas 10 orang dan menulis karakter aksara sebanyak 15 kali. Sehingga jumlah <i>dataset</i> yaitu 10800. 	<ul style="list-style-type: none"> • Responden sebanyak 15 orang di mana masing-masing orang menulis karakter aksara sebanyak 10 kali. Sehingga jumlah <i>dataset</i> yaitu 2700.
<ul style="list-style-type: none"> • Format citra yaitu JPG 	<ul style="list-style-type: none"> • Format citra yaitu PNG
<ul style="list-style-type: none"> • Resolusi 600 dpi 	<ul style="list-style-type: none"> • Resolusi 2400 dpi
<ul style="list-style-type: none"> • Dibedakan menjadi dua kategori yaitu pernah belajar Aksara Sasak dan belum pernah belajar Aksara Sasak. 	<ul style="list-style-type: none"> • Tidak diberlakukan pembedaan kategori pernah dan belum pernah mempelajari Aksara Sasak.
<ul style="list-style-type: none"> • Size awal antara 176-200 pixel 	<ul style="list-style-type: none"> • Size awal antara 190-230 pixel

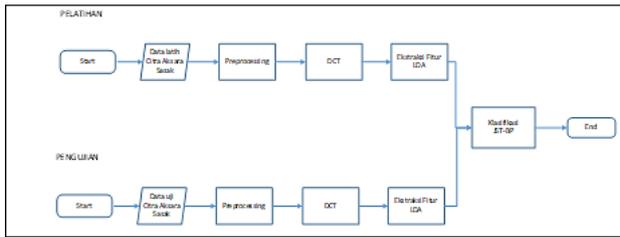


Gambar 3. Contoh citra Aksara Sasak yang diambil oleh peneliti



Gambar 4. Contoh citra Aksara Sasak dari penelitian sebelumnya[2]

3.3 Perancangan Algoritma



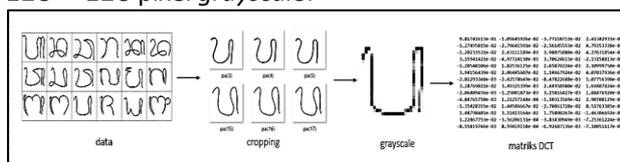
Gambar 5. Blok diagram sistem

Secara sederhana algoritma pengenalan aksara Sasak ditunjukkan pada Gambar 5. Alur untuk proses pelatihan dan pengujian adalah sama yang dimulai dengan mempersiapkan data, kemudian dilanjutkan dengan *preprocessing*. Selanjutnya masuk ke proses ekstraksi fitur dengan menggunakan metode LDA, setelah fitur didapatkan maka masuk ke proses klasifikasi dengan menggunakan metode JST-BP. Setelah proses klasifikasi menghasilkan akurasi yang baik maka prosesnya akan berhenti.

3.4 Preprocessing

Pada penelitian ini proses *preprocessing* yang dilakukan yaitu *cropping*, *resize*, *grayscale*, reduksi dimensi dan melibatkan proses DCT. Ukuran dari citra yang digunakan yaitu 128x128, 64x64 dan 32x32 pixel. Penggunaan metode DCT untuk menghindari matriks *scatter* S_b dan S_w pada proses ekstraksi fitur LDA bersifat singular.

Tahap *preprocessing* pada citra aksara sasak yaitu dilakukan proses *cropping*, *grayscale*, *resizing*, normalisasi dan reduksi dimensi seperti pada Gambar 6. Tahapan *cropping* dilakukan dengan menggunakan *coding* yang disesuaikan dengan *template* saat pengambilan data. Penggunaan *coding* dikarenakan jumlah *dataset* yang banyak dan untuk meminimalisir waktu saat proses *preprocessing*. Selanjutnya yaitu *load dataset* menggunakan *method* "imread" dari *library* "opencv". Selanjutnya, citra masuk ke proses *grayscale* dan *resize* untuk mendapatkan ukuran citra yang diinginkan. Adapun variasi ukuran citra yang digunakan dalam pengujian yaitu 32×32 , 64×64 , 128×128 pixel *grayscale*.



Gambar 6. Proses *preprocessing dataset* Aksara Sasak

Tambahan DCT sebagai *preprocessing* untuk menghindari kesingularan dari matriks *scatter* S_b dan S_w pada proses ekstraksi fitur LDA. Matriks singular adalah matriks yang tidak mempunyai determinan

yang mengakibatkan *eigen analysis* pada proses LDA tidak dapat dilakukan. Kesingularan terjadi karena *mean* atau rata-rata dari citra yang diproses menghasilkan nilai satu. Akibat dari citra pada aksara sasak memiliki dominan *background* putih dibandingkan dengan pola aksara yang tertulis. Berdasarkan penelitian[34] DCT dipilih karena memiliki kemampuan yang baik untuk mengumpulkan informasi fitur penting pada frekuensi rendahnya. DCT memiliki kelebihan yaitu fitur pada citra dapat dibangun menggunakan persentase kecil dari koefisien DCT pada komponen frekuensi rendahnya. Pemilihan koefisien pada penelitian ini yaitu dengan rentang 64~256 koefisien, karena dengan jumlah tersebut citra hasil rekonstruksi masih mengandung informasinya. Tahapan normalisasi citra dan reduksi dimensi dari dua dimensi menjadi satu dimensi diproses dalam *metode* DCT.

3.5 Ekstraksi Fitur LDA

LDA bekerja berdasarkan analisa matrik sebaran (*scatter matrix analysis*) yang bertujuan menemukan suatu proyeksi optimal yang dapat memaksimalkan sebaran antar kelas dan meminimumkan sebaran dalam kelas data. Pada LDA terjadi perbedaan yang minimum dari citra dalam kelas. Perbedaan antar kelas direpresentasikan oleh matriks S_b (*scatter between class*) dan perbedaan dalam kelas direpresentasikan oleh matriks S_w (*scatter within class*). Perhitungan kedua matriks ini berfungsi untuk memaksimalkan jarak antar kelas dan meminimumkan jarak dalam kelas. Berikut merupakan langkah-langkah ekstraksi ciri menggunakan LDA [25].

- a. Mengubah matriks dua dimensi yang didapat dari *pixel* setiap *dataset* menjadi matriks satu dimensi atau mengubahnya ke dalam bentuk vektor baris atau kolom.
- b. Data *training* yang didapat kemudian dikelompokkan ke dalam matriks sejumlah kelas (x_i).
- c. Menghitung nilai rata – rata (*mean*) dari tiap – tiap kelas (μ_i) dengan Persamaan (1) dan jika tiap data *training* diubah ke bentuk vektor baris, maka perhitungan *mean* dimensi dihitung berdasarkan kolom. Jika data *training* diubah ke bentuk vektor kolom, maka perhitungan *mean* dimensi dihitung berdasarkan baris, sehingga nantinya jumlah *mean* dimensi yang dihasilkan akan sama dengan jumlah dimensi satu data *training* dan

bukan dimensi jumlah dari *dataset*.

$$\mu_i = \frac{1}{N_i} \sum_{x \in \omega_i} x \quad (1)$$

- d. Menghitung nilai rata – rata (*mean*) total dari semua kelas (μ) dengan Persamaan (2):

$$\mu = \frac{1}{x_1 + \dots + x_c} \sum_{x \in \omega_i} x \quad (2)$$

- e. Menghitung matriks sebaran antar kelas (S_b) dengan Persamaan (3) dan matriks sebaran dalam kelas (S_w) dengan Persamaan (4):

$$S_b = \sum_{i=1}^c N_i (\mu_i - \mu)(\mu_i - \mu)^T \quad (3)$$

$$S_w = \sum_{i=1}^c \sum_{j=1}^{N_i} ((x_j - \mu_i)(x_j - \mu_i)^T) \quad (4)$$

- f. Menghitung nilai *covariance* matriks (C) menggunakan nilai S_b dan S_w yang telah didapat dengan Persamaan (5):

$$C = (S_w)^{-1} * S_b \quad (5)$$

- g. Menghitung *eigen value* (λ) dan *eigen vector* (v) dengan Persamaan (6):

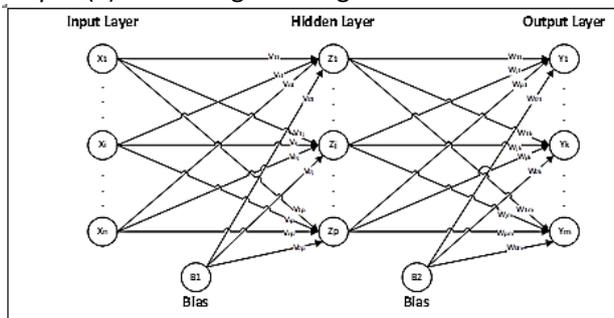
$$S_b v = \lambda S_w v \quad (6)$$

- h. Memproyeksi citra asal dengan *eigen vector* terpilih menggunakan Persamaan (7):

$$p = v^T x^i \quad (7)$$

3.6 Klasifikasi Backpropagation

JST-BP memiliki beberapa unit (*neuron*) yang ada dalam satu atau lebih *layer* tersembunyi. Gambar 8 adalah arsitektur *backpropagation multilayer* dengan 1 *hidden layer*. Pada Gambar 7 tampak bahwa, unit *input* dilambangkan dengan X, unit *hidden* dilambangkan dengan Z, dan unit *output* dilambangkan dengan Y. Bobot antara unit *input* (X) dan unit *hidden* (Z) dilambangkan dengan V, sedangkan bobot antara unit *hidden* (Z) dan unit *output* (Y) dilambangkan dengan W.



Gambar 7. Arsitektur jaringan JST-BP

Dalam JST-BP, fungsi aktivasi yang dipakai harus memenuhi beberapa syarat yaitu: kontinu, terdiferensial dengan mudah, dan merupakan fungsi

yang tidak turun. Salah satu fungsi yang memenuhi ketiga syarat tersebut sehingga sering dipakai adalah fungsi *sigmoid biner* yang memiliki range (0, 1) [26]. Persamaan fungsi aktivasi *sigmoid biner* yaitu sebagai berikut:

$$f(x) = \frac{1}{1+e^{-x}} \quad (8)$$

$$f'(x) = f(x)(1 - f(x)) \quad (9)$$

Algoritma pelatihan JST-BP terdiri dari proses *feedforward* dan *backward*. Algoritma tersebut yaitu sebagai berikut [27]:

Langkah 0: Inisialisasi bobot (ambil bobot awal dengan nilai acak yang cukup kecil)

Langkah 1: Jika kondisi penghentian belum terpenuhi, lakukan langkah 2 sampai 9

Langkah 2: Untuk setiap pasang data pelatihan, lakukan langkah 3 sampai 8

Fase I: Feedforward

Langkah 3: Tiap unit masukan ($x_i, i = 1, 2, \dots, n$) menerima sinyal dan meneruskannya ke unit selanjutnya, yaitu lapisan tersembunyi

Langkah 4 : Hitung semua keluaran pada lapisan tersembunyi ($Z_j, j = 1, 2, \dots, p$)

$$Z_net_j = v_{0j} + \sum_{i=1}^n x_i v_{ij} \quad (10)$$

Gunakan fungsi aktivasi untuk menghitung sinyal *output*-nya:

$$Z_j = (Z_net_j) \quad (11)$$

Dan kirimkan sinyal tersebut ke semua unit lapisan atasnya (unit-unit *output*). Langkah ini dilakukan sebanyak jumlah lapisan tersembunyi.

Langkah 5 : Hitung semua keluaran jaringan di lapisan *output* ($Y_k, k = 1, 2, \dots, m$)

$$Y_net_k = w_{0k} + \sum_{j=1}^p z_j w_{jk} \quad (12)$$

Gunakan fungsi aktivasi untuk menghitung sinyal *output*-nya:

$$Y_k = f(y_net_k) \quad (13)$$

Fase II: Backpropagation

Langkah 6: Hitung faktor δ unit keluaran berdasarkan kesalahan di setiap unit keluaran ($y_k, k = 1, 2, \dots, m$)

$$\delta_k = (t_k - y_k) f'(y_net_k) \quad (14)$$

δ merupakan unit kesalahan yang akan dipakai dalam perubahan bobot *layer* di bawahnya (langkah 7). $f'(y_net_k)$ merupakan fungsi turunan dari fungsi aktivasi *sigmoid biner*.

Kemudian hitung koreksi bobot (yang nantinya akan digunakan untuk memperbaiki w_{jk}) dengan laju percepatan α

$$\Delta w_{jk} = \alpha \cdot \delta \cdot z_j \quad (15)$$

Kemudian hitung juga koreksi bias (yang nantinya akan digunakan untuk memperbaiki nilai w_{0k})

$$\Delta w_{0k} = \alpha \cdot \delta_k \quad (16)$$

Langkah 7: Hitung faktor δ unit tersembunyi berdasarkan kesalahan di setiap unit tersembunyi ($z_j, j = 1, 2, \dots, p$)

$$\delta_{net_j} = \sum_{k=1}^m \delta_k \cdot w_{jk} \quad (17)$$

Faktor δ unit tersembunyi:

$$\delta_j = \delta_{net_j} f'(z_{net_j}) \quad (18)$$

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

$$\Delta v_{ij} = \alpha \cdot \delta_j \cdot x_i \quad (19)$$

Kemudian hitung juga koreksi bias (yang nantinya akan digunakan untuk memperbaiki nilai v_{0j})

$$\Delta v_{0j} = \alpha \cdot \delta_j \quad (20)$$

Fase III: Perubahan Bobot

Langkah 8: Tiap-tiap unit *output* ($Y, k = 1, 2, \dots, m$) memperbaiki bobotnya ($j = 0, 1, 2, \dots, p$)

$$w_{jk} (baru) = w_{jk} (lama) + \Delta w_{jk} \quad (21)$$

Tiap-tiap unit tersembunyi ($Z_j, j = 1, 2, 3, \dots, p$) memperbaiki bobotnya ($j = 0, 1, 2, 3, \dots, n$)

$$v_{ij} (baru) = v_{ij} (lama) + \Delta v_{ij} \quad (22)$$

Langkah 9: Kondisi pelatihan berhenti

Ketiga fase tersebut diulang terus menerus hingga kondisi penghentian dipenuhi. Umumnya kondisi penghentian yang sering dipakai adalah jumlah iterasi atau kesalahan. Iterasi akan dihentikan jika jumlah iterasi yang dilakukan sudah melebihi jumlah maksimum iterasi yang ditetapkan, atau jika kesalahan yang terjadi sudah lebih kecil dari batas toleransi yang diijinkan [28].

3.7 Teknik Pengujian

Tahapan pengujian sistem merupakan tahapan untuk mencoba sistem apakah berjalan dengan baik dan benar serta untuk mengetahui kekurangannya. Dalam pengujian sistem dilakukan perhitungan akurasi dengan rumus sebagai berikut :

$$Akurasi = \frac{jumlah\ data\ sesuai\ target}{Total\ keseluruhan\ data} \quad (23)$$

$$Presisi = \frac{jml\ data\ yang\ sesuai\ target\ di\ satu\ kelas}{jml\ seluruh\ data\ yang\ sesuai\ target} \quad (24)$$

$$Recall = \frac{jml\ data\ yang\ sesuai\ target\ di\ satu\ kelas}{jml\ data\ di\ satu\ kelas} \quad (25)$$

4. HASIL DAN PEMBAHASAN

4.1 Skenario Pengujian

Ada beberapa tahapan yang dilakukan pada penelitian ini yaitu yang pertama adalah pencarian model terbaik untuk tahap ekstraksi dan klasifikasi menggunakan *dataset* berisi 10800 data. Selanjutnya dari model terbaik tersebut dilakukan pengujian terhadap tiga jenis yaitu *dataset* 10800, *dataset* 2700

dan *dataset* gabungan dari data 10800 dan 2700. Pengujian ini dilakukan untuk melihat pengaruh perlakuan data yang berbeda pada tulisan aksara sasak dengan karakter yang sama. Perbedaan dari kedua *dataset* yaitu berdasarkan kertas yang digunakan, *template* pengambilan data serta perbedaan jenis *scanner* dan resolusi *scanner* yang digunakan. Pada penelitian ini, *scanner* yang digunakan yaitu *Canon MP 287* dengan resolusi 600 dpi, sementara *dataset* pada penelitian sebelumnya menggunakan *scanner Canon LiDE 120* dengan resolusi 2400 dpi. Pengujian terhadap model JST-BP yang dibangun dilakukan dengan berbagai parameter dengan urutan pengujian ditunjukkan pada Tabel II.

TABEL II. PARAMETER PENGUJIAN PADA PENELITIAN INI

No	Parameter Pengujian
1.	Nilai <i>eigen value</i> pada proses ekstraksi ciri di LDA (100%,75% dan 50%)
2.	Jumlah <i>hidden layer neural network</i> (1HL, 2HL dan 3HL)
3.	Ukuran <i>neuron hidden layer</i> (32 <i>neuron</i> , 64 <i>neuron</i> dan 96 <i>neuron</i>)
4.	Ukuran citra masukan (32x32, 64x64 dan 128x128)
5.	<i>Learning rates</i> (0.1~0.5)

4.2 Pengaruh DCT Terhadap Akurasi

Untuk pengujian ini, citra yang diujikan sebanyak 10800 citra dengan menggunakan citra ukuran 128x128 dengan koefisien awal 256. Citra hasil *preprocessing* selanjutnya masuk ke tahap ekstraksi fitur LDA. Setelah fitur dari citra didapatkan maka akan masuk ke tahap klasifikasi. Rasio citra perbandingan *training* dan *testing* adalah 70:30 dengan cara *split* data. Setelah itu citra masuk ke tahap klasifikasi dengan menggunakan metode klasifikasi KNN, hal ini dilakukan untuk mempermudah mengetahui akurasi dari koefisien DCT dan *eigen value* sebelum masuk ke metode JST-BP. Hasil pengujian ditunjukkan pada Tabel III yang merupakan perbandingan jumlah koefisien DCT yang dipilih terhadap akurasi yang diberikan untuk 3 jenis variasi ukuran citra.

TABEL III. PERBANDINGAN AKURASI TERHADAP JUMLAH KOEFISIEN DCT UNTUK 3 VARIASI UKURAN CITRA

DCT koef	Size	Akurasi Testing(%)		
		128	64	32
64		90.92	90.24	85.92
81		90.49	90.58	86.14
100		90.86	89.84	85.77
121		90.37	90.06	83.98
144		84.72	89.07	84.07
169		89.04	88.42	83.58
196		89.38	88.76	82.87
256		86.57	86.32	84.07

Berdasarkan Tabel III, citra dengan ukuran 128 x 128 memiliki akurasi tertinggi dengan koefisien DCT

64, yaitu 90.92%. Untuk citra dengan ukuran 64×64 dan 32×32 koefisien DCT 81 yang menghasilkan akurasi tertinggi yaitu pada dengan akurasi 90.58% dan 86.14%, %, hal ini sangat sejalan dengan hasil penelitian [29], bahwa dengan DCT citra dapat direpresentasikan dengan sebagian kecil koefisiennya yang mengandung informasi dominannya Berdasarkan hasil akurasi tertinggi dari koefisien DCT terhadap masing-masing citra, maka koefisien dengan akurasi tertinggi dipilih sebagai ukuran koefisien untuk skenario pengujian selanjutnya.

4.3 Hasil Pengujian

4.3.1 Pengujian Terhadap Eigen Value

Pengujian terhadap *eigen value* pada penelitian ini bertujuan untuk mengetahui jumlah *eigen* untuk proyeksi data *input* menjadi ciri LDA. Caranya adalah dengan memilih *eigen vector* yang diperoleh dari *eigen analysis* yang bersesuaian dengan *eigen value* terbesar, kemudian memproyeksikannya ke dalam ruang *eigen* untuk setiap data masukan. Ukuran citra awal yang digunakan yaitu 128×128 dengan koefisien DCT 64, sesuai dengan hasil akurasi koefisien DCT tertinggi. Selanjutnya variasi beberapa pemilihan jumlah *eigen* pada tiga ukuran citra yang berbeda disimulasikan dan diperoleh hasil uji seperti pada Tabel IV.

TABEL IV. PENGUJIAN NILAI EIGEN TERHADAP SIZE CITRA

Eigen value	Size	Akurasi Testing(%)		
		128	64	32
17		90.92	90.58	86.14
12		89.87	90.09	84.81
9		88.51	87.80	81.88

Pada Tabel IV, nilai jumlah *eigen value* terbaik untuk klasifikasi Aksara Sasak adalah 17 untuk ketiga ukuran citra yang diujikan yaitu 128×128 , 64×64 dan 32×32 dengan akurasi sebesar 90.92%, 90.58% dan 86.14%. Hal ini menunjukkan bahwa *eigen value* terbaik merupakan $n - 1$ dari kelas. Pada penelitian ini terdapat 18 kelas sesuai dengan jumlah karakter Aksara Sasak. Berdasarkan akurasi tertinggi yang didapatkan pada *eigen value* 17, maka skenario pengujian selanjutnya menggunakan *eigen value* dengan nilai 17.

4.3.2 Pengujian Terhadap Neuron dan Hidden Layer

Tujuan dari pengujian ini untuk mengetahui pengaruh jumlah *neuron* berbeda terhadap performa proses klasifikasi pada *hidden layer* satu, *hidden layer* dua dan *hidden layer* tiga. Pada pengujian ini bentuk

dari jaringan Neural Network yang dibangun yakni terdiri dari jumlah neuron 128, 96 dan 64. Ukuran citra yang digunakan yaitu 128×128 dengan koefisien DCT 64, *eigen value* 17 dan *learning rate* 0.001. Ketiga *neuron* dikombinasikan dan diujikan sesuai dengan Tabel V.

TABEL V. PENGARUH JUMLAH NEURON BERBEDA PADA HIDDEN LAYER

Percobaan ke -n	Jumlah Neuron			Akurasi (%)	Presisi (%)	Recall (%)	Time (s)
	Hidden Layer 1	Hidden Layer 2	Hidden Layer 3				
tahap 1	128	128	96	90.23	90.22	90.23	377.690
tahap 2	128	96	96	89.90	89.92	89.91	357.723
tahap 3	128	128	64	90.18	90.17	90.18	354.095
tahap 4	128	64	64	89.69	89.67	89.67	237.610
tahap 5	128	96	64	89.11	89.10	89.10	311.608
tahap 6	128	96	32	88.78	88.79	88.79	246.026
tahap 7	96	96	64	89.77	89.77	89.78	236.715
tahap 8	96	64	64	89.04	89.05	89.05	221.264
tahap 9	96	64	32	86.53	86.54	86.53	194.34
tahap 10	64	64	64	91.37	91.38	91.38	246.968
tahap 11	96	96	96	91.42	91.43	91.43	250.042
tahap 12	128	128	128	91.72	91.73	91.73	382.725

Pada Tabel V, hasil dari kombinasi tiap *neuron* dengan ukuran berbeda menghasilkan akurasi yang berbeda. Untuk *neuron* berukuran besar memiliki akurasi yang lebih baik dibandingkan dengan *neuron* dengan ukuran yang lebih kecil, namun waktu komputasi yang dibutuhkan *neuron* ukuran besar lebih lama dibandingkan dengan jumlah *neuron* yang lebih sedikit.

4.3.3 Pengujian Terhadap Jumlah Hidden Layer

Jumlah *hidden layer* dalam arsitektur *neural network* tidak memiliki ketetapan yang pasti. Semakin sedikit atau semakin banyaknya jumlah *hidden layer* yang digunakan dalam suatu model belum tentu menambah akurasi dari model yang dibangun. Pengujian menggunakan citra dengan ukuran 128×128 dengan koefisien DCT 64, *eigen value* 17 dan *learning rate* 0.001. Oleh karena itu, pada pengujian ini penelitian menggunakan 3 variasi dengan jumlah *neuronnya* tetap yaitu sebagai parameter uji dengan hasil seperti pada Tabel VI.

TABEL VI. PERFORMA PENGUJIAN JUMLAH HIDDEN LAYER

Jumlah HL	Akurasi Testing (%)	Presisi (%)	Recall (%)	Waktu komputasi (s)
1	81.23	81.24	81.23	175.109
2	91.72	91.73	91.73	247.941
3	91.45	91.46	91.45	323.919

Berdasarkan hasil pengujian Tabel VI, didapatkan bahwa akurasi *testing* model dengan 2 *hidden layer* memiliki performa yang lebih baik dibandingkan dengan 1 atau 3 *hidden layer*. Model dengan arsitektur 2 *hidden layer* mampu memberikan performa akurasi

sebesar 91.72% dengan waktu komputasi 247.941 s sedangkan model dengan arsitektur 1 dan 3 *hidden layer* menunjukkan performa akurasi pada angka 81.23% dengan waktu komputasi 175.109 s dan 91.45% dengan waktu komputasi 323.919s. Maka dari itu dipilih 2 *hidden layer* karena memiliki akurasi tertinggi dengan waktu komputasi relatif lebih cepat dibandingkan dengan tiga *hidden layer*. Selanjutnya akan diujikan arsitektur 2 *hidden layer* untuk mencari ukuran *neuron* terbaiknya.

4.3.4 Pengujian Terhadap Learning Rate

Learning rate merupakan salah satu parameter *training* untuk menghitung nilai koreksi bobot pada waktu proses training. Dengan nilai α ini berada pada *range* nol sampai satu. Pada penelitian ini nilai dari *learning rate* yang diujikan yaitu 0.001, 0.002, 0.003, 0.004 dan 0.005. Pada pengujian ini arsitektur yang digunakan adalah terbaik dari pengujian sebelumnya. Yaitu menggunakan 2 HL dengan jumlah *neuron* 128 dengan ukuran citra 128x128, koefisien DCT 64 dan *eigen value* 17. Pengujian ini bertujuan untuk mengetahui performa dari *learning rate* dengan nilai berbeda dan akurasi terbaik nantinya digunakan sebagai paramter pengujian pada skenario selanjutnya. Tabel VII menunjukkan hasil dari performa pengujian *learning rate*.

TABEL VII. PERFORMA PENGUJIAN *LEARNING RATE*

Nilai Learning Rate	Akurasi Testing (%)	Presisi (%)	Recall (%)	Waktu komputasi (s)
0.001	91.72	91.73	91.73	258.964
0.002	91.82	91.83	91.83	249.725
0.003	92.11	92.12	92.13	240.670
0.004	91.59	91.60	91.60	239.446
0.005	91.66	91.67	91.67	237.350

Berdasarkan Tabel VII, *learning rate* dengan nilai 0.003 memiliki akurasi tertinggi yaitu 92.11% dengan waktu komputasi 240.670s. *Learning rate* lainnya yaitu 0.001 menghasilkan akurasi 91.72% dengan waktu komputasi 258.964s. *Learning rate* 0.002 menghasilkan akurasi 91.82% dengan waktu komputasi 249.725s. *Learning rate* 0.004 menghasilkan akurasi 91.59% dengan waktu komputasi 239.446s. Dan yang terakhir yaitu *learning rate* 0.005 menghasilkan akurasi 91.66% dengan waktu komputasi 237.350s. Berdasarkan hasil pengujian diatas dapat disimpulkan bahwa semakin besar nilai *learning rate* yang digunakan maka semakin cepat waktu yang dibutuhkan pada proses *training*.

4.3.5 Pengujian Terhadap Ukuran Citra Masukan

Ukuran citra merupakan bagian yang diatur di dalam persiapan sebelum melakukan proses *training* atau bisa disebut sebagai bagian yang ditentukan pada

tahap *pre-processing*. Semakin kecil ukuran citra maka detail dari citra itu sendiri semakin tidak terlihat sedangkan semakin besar ukuran citra semakin banyak informasi fitur yang didapatkan. Pada penelitian ini *size* citra akan uji pada 3 ukuran citra yaitu 32x32, 64x64 dan 128x128 dengan menggunakan arsitektur *neuron* 128 2 *hidden layer*, *learning rate* 0.003. Hasil dari pengujian dapat dilihat pada Tabel VIII.

TABEL VIII. PERFORMA PENGUJIAN UKURAN CITRA

Ukuran Citra	Akurasi Testing (%)	Presisi (%)	Recall (%)	Waktu komputasi (s)
32X32	85.70	85.72	85.72	254.880
64x64	91.32	91.33	91.33	251.864
128X128	92.11	92.12	92.13	240.670

Berdasarkan hasil pengujian Tabel VIII, ukuran citra 128x128 memiliki performa tertinggi dengan akurasi 92.11% dengan waktu komputasi 240.670 s, sementara citra dengan ukuran 32x32 dan 64x64 memiliki akurasi sebesar 85.70% dengan waktu komputasi 254.880 s dan 91.32% dengan waktu komputasi 251.864 s . Sehingga arsitektur terbaik pada pengujian terhadap ukuran citra masukan adalah citra 128x128 . Untuk waktu komputasi ukuran citra 128x128 memiliki waktu komputasi tercepat dibandingkan dengan ukuran citra 64x64 dan 32x32. Hal ini dikarenakan pengaruh koefisien DCT pada citra ukuran 128x128 lebih kecil dibandingkan ukuran citra lainnya. Koefisien DCT dengan akurasi tertinggi pada ukuran citra 128x128 yaitu 64, sementara untuk ukuran citra 64x64 dan 32x32 akurasi koefisien DCT tertinggi yaitu ada pada koefisien 81. Sehingga pemilihan hal ini dapat mempengaruhi waktu komputasi pada saat proses *training*.

4.4 Pengujian Model

4.4.1 Pengujian Model Dataset 10800 Citra

Pada Tabel IX dapat dilihat performa pengujian *dataset* 10800 dengan menggunakan *cross validation* 10 *fold*. Pembagian data dalam tiap *fold* 1800 data per *fold* baik untuk tiap data *training* dan data *testing*. Pengujian ini menggunakan model terbaik yang telah diujikan sebelumnya seusi dengan parameter yang telah diujikan sebelumnya. Model terbaik yang didapatkan saat proses pengujian sebelumnya yaitu *size* 128x128 dengan koefisien DCT 64, *learning rate* 0.003, 2 *hidden layer* dengan jumlah *neuron* 128. Berdasarkan model terbaik yang digunakan untuk pengujian *dataset* 10800, didapatkan rata-rata akurasi yaitu 92.20% dengan rata-rata waktu komputasi adalah 250.691s.

TABEL IX PERFORMA PENGUJIAN DATASET 10800 DENGAN CROSS VALIDATION

cros validation	Akurasi Training (%)	Akurasi Testing (%)	Presisi (%)	Recall (%)	Waktu komputasi (s)
1	100	92.37	92.38	92.38	249.676
2	100	91.60	91.61	91.61	250.148
3	100	92.43	92.44	92.44	244.177
4	100	92.25	92.26	92.26	256.924
5	100	91.35	91.36	91.36	256.266
6	100	92.71	92.72	92.72	260.197
7	100	92.38	92.39	92.39	243.798
8	100	92.10	92.11	92.11	251.292
9	100	92.63	92.64	92.64	242.354
10	100	92.01	92.02	92.02	252.115
Rata-rata	100	92.20	92.21	92.21	250.691

4.4.2 Pengujian Model Dataset 2700 Citra

Pada Tabel X dapat dilihat performa pengujian dataset 2700 dengan menggunakan cross validation 10 fold. Pembagian data dalam tiap fold 270 data per fold baik untuk tiap data training dan data testing. Pengujian ini menggunakan model terbaik yang telah diujikan sebelumnya seusi dengan parameter yang telah diujikan sebelumnya. Model terbaik yang didapatkan saat proses pengujian sebelumnya yaitu size 128x128 dengan koefisien DCT 64, learning rate 0.003, 2HL dengan jumlah node 128. Berdasarkan model terbaik yang digunakan untuk pengujian dataset 2700, didapatkan rata-rata akurasi yaitu 71.73% dengan rata-rata waktu komputasi adalah 65.446%.

TABEL X. PERFORMA PENGUJIAN DATASET 2700 DENGAN CROSS VALIDATION

cros validation	Akurasi Training (%)	Akurasi Testing (%)	Presisi (%)	Recall (%)	Waktu komputasi (s)
1	99.84	73.82	73.81	73.82	85.908
2	99.78	71.11	71.12	71.11	62.919
3	99.89	75.16	75.17	75.16	62.097
4	99.94	74.93	74.94	74.93	64.133
5	99.52	72.71	72.72	72.71	65.575
6	99.57	72.22	72.23	72.22	78.552
7	99.78	73.45	73.46	73.45	62.235
8	99.89	75.18	75.19	75.18	57.833
9	99.89	72.71	72.72	72.71	57.314
10	99.84	74.07	74.08	74.07	57.890
Rata-rata	99.79	71.73	71.74	71.74	65.446

4.4.3 Pengujian Model Dataset 13500 Citra

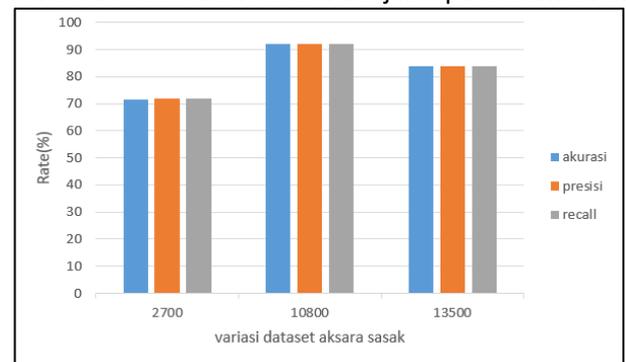
Pada Tabel XI dapat dilihat performa pengujian dataset 13500 dengan menggunakan cross validation 10 fold. Pembagian data dalam tiap fold 1350 data per fold baik untuk tiap data training dan data testing. Pengujian ini menggunakan model terbaik yang telah diujikan sebelumnya seusi dengan parameter yang telah diujikan sebelumnya. Model terbaik yang didapatkan saat proses pengujian sebelumnya yaitu size 128x128 dengan koefisien DCT 64, learning rate 0.003, 2HL dengan jumlah node 128. Berdasarkan

model terbaik yang digunakan untuk pengujian dataset 13500, didapatkan rata-rata akurasi yaitu 83.92% dengan rata-rata waktu komputasi adalah 361.275%.

TABEL XI. PERFORMA PENGUJIAN DATASET 13500 DENGAN CROSS VALIDATION

cros validation	Akurasi Training (%)	Akurasi Testing (%)	Presisi (%)	Recall (%)	Waktu komputasi (s)
1	99.88	85.75	86	86	348.126
2	99.57	83.80	84	84	360.224
3	98.22	83.38	83	83	369.242
4	99.43	83.43	84	84	365.426
5	99.75	84.14	84	84	337.357
6	99.88	84.49	85	85	363.858
7	99.91	84.69	85	85	367.107
8	99.31	83.23	83	83	367.031
9	99.95	83.72	84	84	364.443
10	99.88	82.46	83	83	370.213
Rata-rata	99.58	83.92	84	84	361.275

Berdasarkan pengujian yang telah dilakukan, dapat dibandingkan hasil untuk pengujian variasi dataset Aksara Sasak. Nilai akurasi, presisi, dan recall dari 3 variasi dataset Aksara Sasak disajikan pada Gambar 8.



Gambar 8. Diagram perbandingan nilai akurasi, presisi, dan recall pengujian dengan keseluruhan variasi dataset

Gambar 8 menunjukkan bahwa performa terbaik ditunjukkan ketika melakukan pengujian menggunakan dataset aksara sasak sejumlah 10800 citra. Hal ini dikarenakan adanya perbedaan proses pengambilan data dan preprocessing data yang dilakukan pada dataset 10800 dan dataset 2700. Pengambilan data dan preprocessing data sangat mempengaruhi akurasi dari suatu citra. Berdasarkan kedua jenis dataset, terlihat perbedaan ketebalan citra, citra pada dataset 10800 ditulis lebih tebal dibandingkan dengan citra pada dataset 2700. Proses scanning yang dilakukan juga berbeda. Hal ini dapat mempengaruhi fitur-fitur yang ada pada suatu citra. Banyaknya variasi penulisan aksara pada dataset 10800 berdasarkan empat kategori pendidikan menunjukkan hasil yang lebih baik. Sementara untuk data 2700 memiliki akurasi yang paling rendah diantara ketiga data, hal ini dikarenakan jumlah data yang lebih

sedikit dan variasi tulisan aksara yang lebih sedikit pula sehingga proses pengenalan citra tidak maksimal. Dan untuk data gabungan antara 10800 dan 2700 memiliki akurasi dibawah aksara yang 10800 dan memiliki akurasi yang lebih tinggi daripada *dataset* 2700. Hal ini dikarenakan data perbedaan dalam mengolah *dataset* pada *dataset* 2700. Jadi kesimpulannya adalah proses pengambilan data dan *preprocessing* data sangat mempengaruhi akurasi dari suatu citra. Banyaknya data dan variasi sumber penulisan beragam menghasilkan akurasi yang lebih baik dibandingkan dengan jumlah data yang sedikit. Semakin banyak ragam tulisan atau jenis aksara yang ada maka program dapat belajar lebih banyak sehingga menghasilkan akurasi yang lebih tinggi.

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Berdasarkan penelitian yang sudah dilakukan, terdapat beberapa hal yang bisa penulis simpulkan antara lain sebagai berikut.

- a. Model yang dihasilkan pada penelitian ini menggunakan *input* citra *grayscale* dengan ukuran 128x128 pixel, *hidden layer* berjumlah 2 dengan *neuron* masing-masing sebanyak 128 dengan koefisien DCT 64.
- b. Model terbaik yang dihasilkan pada penelitian ini menggunakan input citra grayscale dengan ukuran 128x128 pixel, dengan koefisien DCT 64, eigen value bernilai 17, hidden layer berjumlah 2 dengan neuron masing-masing sebanyak 128 dan learning rate 0.003.
- c. Persentasi tertinggi pada pengujian ini yaitu 92.20% pada dataset 10800 berdasarkan arsitektur dan parameter uji terbaik dari pengujian sebelumnya. Hal ini membuktikan bahwa metode usulan memiliki potensi yang baik untuk dikembangkan sebagai media pembelajaran Aksara Sasak.
- d. Learning rate terbaik pada pengujian ini yaitu 0.003, dimana semakin besar nilai learning rate maka dapat mempercepat proses komputasi.
- e. Eigen value terbaik yaitu bernilai $n - 1$ dari kelas yaitu 17 eigen, hal ini dikarenakan citra dibagi menjadi 18 kelas sesuai dengan jumlah karakter Aksara Sasak.
- f. Akurasi tertinggi dari perbandingan ketiga jenis

dataset ditunjukkan pada dataset 10800, hal ini disebabkan oleh adanya perbedaan proses pengambilan data dan preprocessing dengan dataset 2700. Hal ini menunjukkan pengambilan data dan persiapan data sebelum masuk ke tahap selanjutnya sangat mempengaruhi akurasi.

5.2. Saran

Ada beberapa saran yang dapat penulis berikan apabila penelitian ini akan dikembangkan kembali antara lain sebagai berikut.

- a. Persiapan sebelum penelitian seperti pengambilan data harus diperhatikan untuk mendapatkan data penelitian yang bagus dan kategori sumber pengambilan data dapat ditambahkan seperti penutur langsung Aksara Sasak.
- b. Pembuatan media pembelajaran aksara sasak dalam platform *android* atau *web*.
- c. Dapat mencoba model arsitektur lain pada pengujian *neural network* seperti menambahkan *hidden layer* dan jumlah *neuron* yang lebih beragam.
- d. Menguji parameter terbaik pada tiap *dataset* yaitu *dataset* 10800, 2700 dan 13500.

UCAPAN TERIMA KASIH

Terima kasih saya ucapkan kepada seluruh yang terlibat dalam penelitian ini, baik dosen pembimbing yang telah membimbing dalam penelitian ini hingga selesai maupun teman-teman yang telah membantu dan berkontribusi baik berupa moril maupun materil.

DAFTAR PUSTAKA

- [1] F. H. Tondo, "Kepunahan Bahasa-Bahasa Daerah: Faktor Penyebab Dan Implikasi Etnolinguistik," *J. Masy. Budaya*, vol. 11, no. 2, pp. 277–296, 2009.
- [2] Riska Yulianti, "Pengenalan Pola Tulisan Tangan Suku Kata Aksara Sasak Menggunakan Metode Moment Invariant dan Support Vector Machine," Mataram: Universitas Mataram, 2018.
- [3] Eka Dina Juliani Utari, "Pengenalan Pola Tulisan Tangan Huruf Sasak Menggunakan Metode *Integral Projection* dan *Neural Network*," *J-COSINE*, Vol. 3, No. 1, 2019.
- [4] R. Lim, Raymond dan K. Gunadi, "Face Recognition Menggunakan Metode Linear Discriminant Analysis (LDA)," *J. KOMMIT*, Vol.12, pp. 134-142, 2002.

- [5] B. Pradinta, Ernawati dan E. P. Purwandari, "Identifikasi Citra Garis Telapak Tangan Menggunakan Metode Linear Discriminant Analysis Dengan Probabilitas Naïve Bayesian," *J. Pseudocode*, Vol 4, No.2, pp.156-167, 2017
- [6] Farida Asriani, "Handwritten Javanese Character Recognition System Using Artificial Network Backpropagation," vol. 5, 2009.
- [7] Nazla Nurmila, "Algoritma Back Propagation Neural Network Untuk Pengenalan Pola Karakter Huruf Jawa," vol. 1, no. 1, 2015.
- [8] B. Isnawati, "Analisis Implementasi Jaringan Syaraf Tiruan Back Propagation Untuk Klasifikasi Huruf Dasar Aksara Jawa" Yogyakarta: Universitas Gadjah Mada, 2015.
- [9] D. Wulansari, "Identifikasi Gender Berdasarkan Citra Wajah Menggunakan Deteksi Tepi dan Backpropagation," *J. SNATi*, 2017.
- [10] D. A. Pancorowati dan M. A. Bustomi, "Klasifikasi Pola Huruf Vokal dengan Menggunakan Jaringan Saraf Tiruan," *J. TEKNIK POMITS*, pp. 1-7, 2017.
- [11] R. Abdilah, "Identifikasi Otentifikasi Citra Tanda Tangan Menggunakan Wavelet dan Backpropagation," *J. Seminar Nasional Aplikasi Teknologi sInformasi*, pp. 5-9, 2017.
- [12] Sulistiyasni, "Klasifikasi Pola Sidik Jari Menggunakan Jaringan Syaraf Tiruan Backpropagation," *J. Ilmiah Teknik Informatika*, vol.10, pp. 215-224, 2016.
- [13] F. Asriani, "Pengenalan Pola Aksara Jawa Tulisan Tangan dengan Jaringan Syaraf Tiruan Perambatan-Balik," *J. Ilmiah Dinamika Rekayasa*, vol. 5, no. 2, pp. 34-36, 2009.
- [14] B. Widoyono, T. Agung Budi Wirayuda dan B. Purnama, "Implementasi Linear Discriminant Analysis Dan Jaringan Syaraf Tiruan Backpropagation Untuk Membaca Angkat Pada Meteran Air Secara Otomatis," Telkom University, 2013.
- [15] P. Chakraborty and P. Sarkar, "An Approach to Handwriting Recognition using Back-Propagation Neural Network," *Int. J. Comput. Appl.*, vol. 68, no. 13, pp. 6-12, 2013.
- [16] G. S. Budhi and R. Adipranata, "Handwritten javanese character recognition using several artificial neural network methods," *J. ICT Res. Appl.*, vol. 8, no. 3, pp. 195-212, 2015.
- [17] D. A. Pancorowati and M. A. Bustomi, "Klasifikasi Pola Huruf Vokal dengan Menggunakan Jaringan Saraf Tiruan," pp. 1-7, 2005.
- [18] R. G. Ahangar and M. F. Ahangar, "Handwritten Farsi Character Recognition using Artificial Neural Network," no. August 2009, 2009.
- [19] S. Afroge, B. Ahmed, and F. Mahmud, "Optical character recognition using back propagation neural network," *ICECTE 2016 - 2nd Int. Conf. Electr. Comput. Telecommun. Eng.*, no. November 2017, 2017.
- [20] Fandiansyah, J. Y. Sari, and I. P. Ningrum, "Pengenalan Wajah Menggunakan Metode Linear Discriminant Analysis dan k Nearest Neighbor," *J. Ultim.*, vol. 9, no. 1, pp. 1-9, 2018.
- [21] M. C. Wijaya and A. Prijono, Pengolahan Citra Digital Menggunakan MatLAB Image Processing Toolbox, Bandung: Informatika, 2007.
- [22] I. M. Mataram, "Peramalan Beban Hari Libur Menggunakan Artificial Neural Network," *J. Tek. Elektro*, pp. 53-56, 2008.
- [23] Bahrie., H. Sudirman & Lalu Ratmaja, "Bahan Ajar Muatan Lokal, Gumi Sasak, KSU 'Prima Guna' ". Lombok Timur, 2013.
- [24] H. Yasri, "Cara Cepat Belajar Aksara Sasak". Mataram: Pustaka Widiya, 2010
- [25] S. Cahyani, R. Wiryasaputra, and R. Gustriansyah, "Identifikasi Huruf Kapital Tulisan Tangan Menggunakan Linear Discriminant Analysis dan Euclidean Distance," vol. 01, pp. 57-67, 2018.
- [26] Aloysius Tanto Wibowo, "Pengenalan Pola Tulisan Tangan Aksara Jawa Dengan Algoritma Backpropagation". Yogyakarta, 2018.
- [27] Shabrina, M., "Pengenalan Iris Mata Menggunakan Metode Analisis Komponen Utama (Principal Component Analysis – PCA) dan Jaringan Saraf Tiruan Perambatan Balik". Skripsi. Universitas Diponegoro, 2012.
- [28] J. J. Siang, *Jaringan Syaraf Tiruan dan Pemrogramannya Menggunakan Matlab*, 1st ed. Yogyakarta: ANDI, 2005.
- [29] I. G. P. S. Wijaya, K. Uchimura, and Z. Hu, "Face Recognition Based on Dominant Frequency Features and Multiresolution Metric," *Int. J. Innovative Comput. Inf. Control*, vol. 5, no. 1349-4198, pp. 641-651, 2009.