

Implementasi Jaringan Saraf Tiruan Backpropagation untuk Pengenalan Karakter pada Dokumen Tercetak

Dian Puspita Chandra¹, Alethea Suryadibrata²

Program Studi Informatika, Universitas Multimedia Nusantara, Tangerang, Indonesia

Diterima 30 Oktober 2019

Disetujui 20 Desember 2019

Abstract--Digital documents from the scanner device cannot be edited. To be able to edit digital documents, Optical Character Recognition (OCR) technology is needed. This research was conducted with the aim of implementing backpropagation artificial neural networks in printed documents and to find out how the accuracy of the implementation of backpropagation artificial neural networks in printed documents. This research uses multilayer networks with three layers. The input layer consists of 225 nodes with 15×15 pixels digital image as input, hidden layer consists of 110 nodes, and the output layer consists of 54 nodes representing A-Z, a-z, point punctuation (.), and comma punctuation (,). The learning rate used in this research is 0,29. The average accuracy level obtained from the implementation of backpropagation artificial neural networks in this research was 94 % for Ms Arial Unicode font type, 96,6 % for Tahoma font type, and 94 % for Times New Roman font type.

Index Terms--Artificial Neural Network, Backpropagation, Image Processing, Optical Character Recognition

I. PENDAHULUAN

Perkembangan ilmu pengetahuan dan teknologi telah memberikan banyak perubahan dalam kehidupan manusia. Salah satu perubahan yang terjadi adalah diterapkannya *paperless office* pada suatu perusahaan atau instansi. *Paperless office* merupakan pengurangan pemakaian kertas dengan mendigitalkan dokumen yang dimiliki suatu perusahaan atau instansi sehingga mengurangi pemakaian kertas secara fisik. Menurut Susilawati [1], konsep *paperless office* dapat menunjang kinerja suatu lembaga pada masa kini dan yang akan datang, karena konsep digitalisasinya sangat menunjang dalam penyampaian arus informasi secara cepat, tepat, lengkap, dan yang paling penting efisien.

Untuk dapat mendigitalkan dokumen tercetak, suatu perusahaan atau instansi dapat menggunakan alat pemindai (*scanner*). Akan tetapi, dokumen digital yang dihasilkan dari alat pemindai tidak dapat diedit. Sehingga, untuk mengubah atau memperbaiki dokumen digital yang dihasilkan dari alat pemindai, suatu perusahaan atau instansi membutuhkan

teknologi *Optical Character Recognition* (OCR) untuk dapat mengedit dokumen digital tersebut.

OCR dapat digunakan untuk mengenali teks yang ada pada dokumen tercetak. Menurut Mithe dkk. [2], OCR adalah teknologi yang memungkinkan berbagai jenis dokumen dikonversi menjadi data yang dapat diedit dan dicari. OCR telah muncul sebagai area penelitian utama sejak tahun 1950 [3]. Sampai saat ini, OCR telah banyak diteliti dengan berbagai metode. Li dkk. [4] melakukan penelitian mengenai OCR untuk pengenalan karakter pada naskah bahasa tradisional yang dinamai New Tai Lue menggunakan *backpropagation neural network* dengan akurasi akhir sebesar 98,04%. Selain penelitian yang dilakukan Li dkk. [4], Radhiah dkk. [5] melakukan penelitian untuk membandingkan tingkat akurasi pengenalan karakter pada huruf arab tercetak dari dua metode yang berbeda yaitu *backpropagation neural network* dengan *hidden markov model*. Hasil akhir penelitian tersebut menunjukkan bahwa tingkat akurasi dari pengenalan karakter arab terisolasi dengan metode *backpropagation neural network* lebih tinggi yaitu sebesar 100% untuk *font* Arial Unicode Ms, Tahoma, dan Times New Roman. Sementara, tingkat akurasi dari pengenalan karakter arab terisolasi dengan metode *hidden markov model* adalah sebesar 74% untuk *font* Arial Unicode Ms, 61% untuk *font* Tahoma, dan 77% untuk *font* Times New Roman.

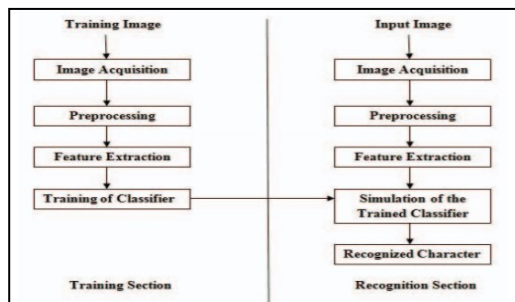
Berdasarkan latar belakang tersebut, diimplementasikan jaringan saraf tiruan *backpropagation* untuk pengenalan karakter dalam penelitian ini. Pengenalan karakter akan dilakukan pada dokumen tercetak yang sudah berupa citra digital. Citra digital akan diproses dengan operasi morfologi dilasi dan erosi, teknik segmentasi *thresholding*, deteksi tepi *canny*, dan penghalusan citra dengan filter *gaussian*.

II. DASAR TEORI

A. *Optical Character Recognition* (OCR)

Optical Character Recognition (OCR) merupakan teknologi untuk pengenalan karakter yang biasa digunakan untuk mengubah teks dalam citra digital

menjadi format teks yang dapat diolah komputer. Dalam penelitian Afroge dkk. [6], OCR terdiri dari dua bagian yaitu bagian pelatihan dan pengenalan. Selain itu, Afroge dkk. [6] menunjukkan bahwa sistem OCR terdiri dari image acquisition, preprocessing, dan feature extraction, serta training of the classifier pada bagian pelatihan dan simulation of the trained classifier pada bagian pengenalan. Ilustrasi sistem OCR dapat dilihat pada Gambar 1.



Gambar 1. Sistem *Optical Character Recognition*

Dalam penelitian Afroge dkk. [6], *feature extraction* digunakan untuk mengekstrak ciri numerik dari data asli. Menurut penelitian yang dilakukan oleh Nugroho [7], *feature extraction* dapat tidak dilakukan namun mengakibatkan proses pelatihan membutuhkan waktu yang lebih lama.

B. Thresholding

Metode *thresholding* adalah proses mengubah citra berderajat keabuan menjadi citra biner atau hitam putih sehingga dapat diketahui daerah mana yang termasuk objek dan *background* dari citra secara jelas [8]. Menurut Hietania dkk. [9], *thresholding* merupakan teknik yang sederhana dan efektif untuk segmentasi citra. *Thresholding* diklasifikasikan menjadi dua yaitu *global thresholding* dan *local thresholding (adaptive)*.

- *Global Thresholding*

Global thresholding merupakan salah satu teknik *thresholding* yang paling sederhana. Apabila nilai ambang t bergantung hanya pada satu nilai aras keabuan $f(y, x)$, *thresholding* disebut global [10].

- *Local Thresholding*

Local thresholding merupakan salah satu teknik *thresholding* yang menghitung nilai *threshold* secara lokal. *Thresholding* disebut lokal jika nilai ambang t bergantung pada $f(y, x)$ dan $g(y, x)$ dengan $g(y, x)$ menyatakan properti citra lokal pada titik (y, x) [10].

C. Operasi Morfologi

Menurut Kadir dan Susanto [10], operasi morfologi merupakan operasi yang umum dikenakan pada citra biner (hitam-putih) untuk mengubah struktur bentuk objek yang terkandung dalam citra.

- *Dilasi*

Dilasi merupakan salah satu operasi morfologi yang digunakan untuk mendapatkan efek pelebaran dengan menambah piksel pada citra digital.

- *Erosi*

Erosi merupakan salah satu operasi morfologi yang digunakan untuk mendapatkan efek penipisan dengan mengurangi piksel pada citra digital.

D. Operator Canny

Operator *Canny* merupakan operator deteksi tepi yang dikemukakan pada tahun 1986 oleh John F. Canny. Menurut Hermawati [11], operator ini diformulasikan dengan tiga tujuan utama yaitu *optimal detection* dengan tidak adanya respon yang salah (palsu), *good localization* dengan jarak minimal antara posisi tepi sebenarnya dengan posisi yang terdeteksi, dan *single response* dengan mengeliminasi *multiple response* menjadi satu garis tepi saja.

E. Gaussian Blur

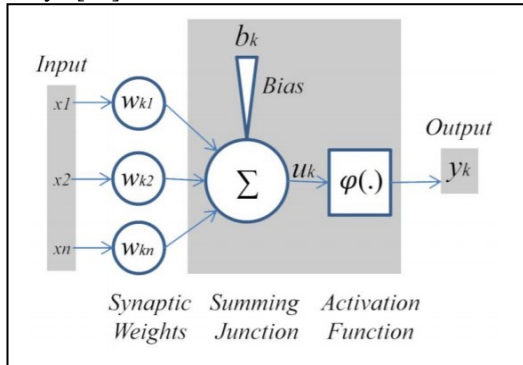
Gaussian blur adalah filter blur yang menempatkan warna transisi yang signifikan dalam sebuah *image*, kemudian membuat warna-warna pertengahan untuk menciptakan efek lembut pada sisi-sisi sebuah *image* [12]. *Gaussian blur* memberikan efek blur dengan menggunakan filter *gaussian*. Filter *gaussian* merupakan filter linear dengan nilai pembobotan pada piksel-piksel yang dipilih berdasarkan bentuk fungsi *gaussian*. Filter *Gaussian* biasanya digunakan dalam pengolahan citra digital untuk penghalusan citra atau penghilangan derau (*noise*).

F. Jaringan Saraf Tiruan

Jaringan saraf tiruan adalah paradigma pengolahan informasi yang terinspirasi oleh sistem saraf secara biologis, seperti informasi pada otak manusia [13]. Jaringan saraf tiruan merupakan salah satu upaya manusia untuk memodelkan fungsi dari sistem saraf manusia dalam memproses informasi untuk melaksanakan suatu tugas. Jaringan saraf tiruan tersusun atas sejumlah neuron yang dihubungkan oleh *link*. Setiap *link* memiliki bobot yang digunakan untuk mengirimkan informasi yang diterima dari satu neuron ke neuron lain yang terhubung. Bobot tersebut akan selalu diperbaharui selama proses pembelajaran.

Satu sel saraf dalam jaringan saraf tiruan terdiri dari tiga bagian yaitu *summing function*, fungsi aktivasi, dan *ouput*. *Summing function* adalah fungsi yang digunakan untuk menjumlahkan nilai bobot yang ada pada setiap *link* yang terhubung dengan neuron. Sedangkan, fungsi aktivasi adalah fungsi yang digunakan untuk mengaktifkan neuron. Informasi input pada jaringan saraf tiruan akan dikirimkan ke setiap neuron dengan nilai bobot tertentu. Kemudian, nilai bobot akan dijumlahkan

dengan *summing function* dan hasil dari penjumlahan akan dibandingkan dengan suatu nilai ambang (*threshold*) tertentu melalui fungsi aktivasi setiap neuron. Apabila input tersebut melewati suatu nilai ambang (*threshold*) tertentu, maka neuron tersebut akan diaktifkan. Jika tidak, maka neuron tersebut tidak akan diaktifkan. Apabila neuron tersebut diaktifkan, maka neuron tersebut akan mengirimkan *output* melalui bobot-bobot *outputnya* ke semua neuron yang berhubungan dengannya, demikian seterusnya [14].



Gambar 2. Model Saraf

Jaringan saraf tiruan memiliki beberapa lapisan yaitu lapisan input, lapisan tersembunyi (*hidden layer*), dan lapisan *output*. Pada lapisan input, setiap unit menerima pola inputan data dari luar. Pada lapisan *output*, *output* yang dihasilkan dari lapisan ini akan menjadi solusi jaringan saraf tiruan terhadap suatu tugas. Sementara, lapisan tersembunyi (*hidden layer*) adalah lapisan yang menghubungkan lapisan input dan lapisan *output*. *Output* yang dihasilkan lapisan tersembunyi (*hidden layer*) tidak dapat diamati secara langsung.

G. Backpropagation

Backpropagation merupakan algoritma pembelajaran yang terawasi dan biasanya digunakan oleh perceptron dengan banyak lapisan untuk mengubah bobot-bobot yang terhubung dengan neuron-neuron yang ada pada lapisan tersembunyinya [15]. Proses pelatihan jaringan dengan algoritma ini terdiri dari tiga tahap yaitu tahap perambatan maju (*forward propagation*), tahap perambatan balik (*backpropagation*), dan tahap perubahan bobot dan bias. *Forward propagation* digunakan untuk membawa data pada input sampai ke *output*. Sedangkan, *backpropagation* digunakan untuk menyesuaikan bobot-bobot dan bias berdasarkan nilai *error* yang didapatkan dengan arah mundur.

Langkah-langkah yang dilakukan pada pada proses perubahan bobot adalah sebagai berikut :

1. Inisialisasi nilai bobot dengan menetapkan nilai bobot awal secara acak.
2. Lakukan *forward propagation* selama kondisi berhenti bernilai salah atau nilai *epoch* (set

pelatihan pada jaringan) lebih kecil dari nilai maksimum *epoch*.

Langkah-langkah yang dilakukan pada tahap *forward propagation* :

1. Setiap unit input (X_i , $i = 1, 2, \dots, n$) menerima sinyal X_i dan mengirim sinyal tersebut ke semua unit pada lapisan tersembunyi (*hidden layer*).
2. Setiap unit tersembunyi (Z_j , $j = 1, 2, \dots, p$) menjumlahkan bobot sinyal input dengan persamaan :

$$Zin_j = V_{0j} + \sum_{i=1}^n X_i V_{ij} \quad (1)$$

gunakan fungsi aktivasi untuk menghitung sinyal *output* :

$$Z_j = f(Zin_j) \quad (2)$$

3. Setiap unit tersembunyi (Y_k , $k = 1, 2, \dots, m$) menjumlahkan bobot sinyal input dengan persamaan :

$$Yin_k = W_{0k} + \sum_{j=1}^p Z_j W_{jk} \quad (3)$$

gunakan fungsi aktivasi untuk menghitung sinyal *output* :

$$Y_k = f(Yin_k) \quad (4)$$

setelah itu, kirim sinyal ke lapisan *output*.

Langkah-langkah yang dilakukan pada tahap *backpropagation* :

1. Setiap unit *output* (Y_k , $k = 1, 2, \dots, m$) menerima pola target yang sesuai pada pola input pelatihan, hitung informasi *error* dengan persamaan :

$$\delta_k = (t_k - Y_k) f'(Yin_k) \quad (5)$$

hitung koreksi nilai bobot yang akan digunakan untuk memperbaharui bobotnya dengan persamaan :

$$\Delta W_{jk}(t) = \alpha \delta_k Z_j \quad (6)$$

hitung koreksi nilai bias yang akan digunakan untuk memperbaharui biasnya dengan persamaan :

$$\Delta W_{0k}(t) = \alpha \delta_k \quad (7)$$

Kirimkan nilai δ_k ke setiap unit yang berada pada lapisan dibawahnya.

2. Setiap unit pada lapisan tersembunyi (Z_j , $j = 1, 2, \dots, p$) jumlahkan hasil perubahan input dari setiap unit pada lapisan diatasnya dengan persamaan :

$$\delta in_j = \sum_{k=1}^m \delta_k W_{jk} \quad (8)$$

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

$$\delta_j = \delta in_j f'(Zin_j) \quad (9)$$

hitung koreksi nilai bobot dengan persamaan :

$$\Delta V_{jk}(t) = \alpha \delta_j X_i \quad (10)$$

hitung koreksi nilai bias dengan persamaan :

$$\Delta V_{0k}(t) = \alpha \delta_j \quad (11)$$

Langkah-langkah yang dilakukan pada tahap perubahan bobot dan bias :

1. Setiap unit *output* (Y_k , $k = 1, 2, \dots, m$) perbaharui nilai bias dan nilai bobot ($j = 0, \dots, p$) dengan persamaan :

$W_{jk}(baru) = W_{jk}(lama) + \Delta W_{jk}$ (12)
 untuk setiap unit lapisan tersembunyi ($Z_j, j = 1, 2, \dots, p$) perbaharui nilai bias dan nilai bobot ($i = 0, \dots, n$) dengan persamaan :

$$V_{ij}(baru) = V_{ij}(lama) + \Delta V_{ij} \quad (13)$$

2. Tes kondisi berhenti.

H. Fungsi Aktivasi

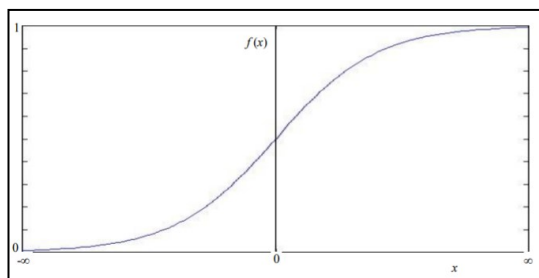
Fungsi aktivasi adalah fungsi yang digunakan pada jaringan saraf tiruan untuk mengaktifkan neuron. Menurut Suhardi [16], karakteristik yang harus dimiliki oleh fungsi aktivasi jaringan backpropagation antara lain harus kontinu, terdiferensialkan, dan tidak menurun secara monotonis (*monotonically non-decreasing*). Beberapa fungsi aktivasi yang biasa digunakan dalam algoritma backpropagation diantaranya :

1. Fungsi Sigmoid

Fungsi ini memiliki jangkauan nilai antara 0 sampai 1. Fungsi sigmoid biner diilustrasikan pada Gambar 3. Fungsi sigmoid didefinisikan dengan persamaan pada Persamaan (14) dan fungsi turunan sigmoid didefinisikan dengan persamaan pada Persamaan (15).

$$y = f(x) = \frac{1}{(1 + e^{-x})} \quad (14)$$

$$y = f'(x) = \frac{1}{(1 + e^{-x})} \times (1 - \frac{1}{(1 + e^{-x})}) \quad (15)$$



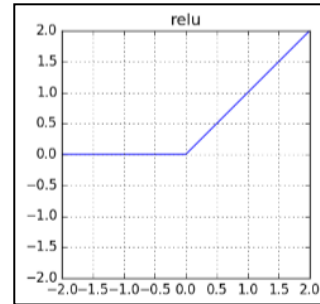
Gambar 3. Fungsi Aktivasi Sigmoid

2. Fungsi Rectified Linear Unit (ReLU)

Rectified linear unit (ReLU) adalah salah satu fungsi aktivasi yang memiliki perhitungan yang sederhana. Menurut Wibawa [17], kelebihan ReLU akan muncul saat berhadapan dengan jaringan yang memiliki neuron yang banyak sehingga dapat mengurangi waktu training dan testing dengan signifikan. Fungsi ReLU diilustrasikan pada Gambar 4. Fungsi ReLU didefinisikan dengan persamaan pada Persamaan (16) dan fungsi turunan ReLU didefinisikan dengan persamaan pada Persamaan (17).

$$y = f(x) = \max(0, x) \quad (16)$$

$$y = f'(x) = \begin{cases} 1, & x > 0 \\ 0, & x \leq 0 \end{cases} \quad (17)$$



Gambar 4. Fungsi Aktivasi ReLU

III. PERANCANGAN SISTEM

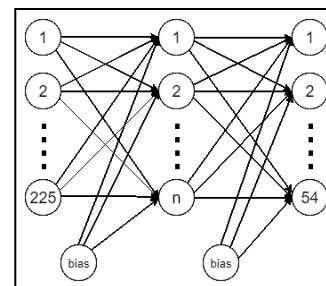
A. Perancangan Jaringan Saraf Tiruan

Arsitektur jaringan saraf tiruan yang digunakan dalam penelitian adalah jaringan lapisan banyak (*multilayer network*). Jaringan ini terdiri dari tiga lapisan yaitu lapisan input, lapisan tersembunyi, dan lapisan *output*. Lapisan input terdiri dari 225 node yang merupakan piksel-piksel dari citra input yang sudah dinormalisasi menjadi 15×15 piksel. Lapisan *output* terdiri dari 54 node yang merepresentasikan karakter A-Z, a-z, serta tanda baca titik (.) dan koma (,). Berdasarkan Persamaan (18), jumlah node yang digunakan pada lapisan tersembunyi adalah 110. Sementara, learning rate yang digunakan berdasarkan pada Persamaan (19) adalah 0,29. Perhitungan jumlah node pada lapisan tersembunyi dan learning rate mengacu pada penelitian yang dilakukan dalam penelitian [18].

$$N^{(h)} = \sqrt{N^{(i)}N^{(o)}} \quad (18)$$

$$\eta = 32/\sqrt{N^{(i)}N^{(o)}} \quad (19)$$

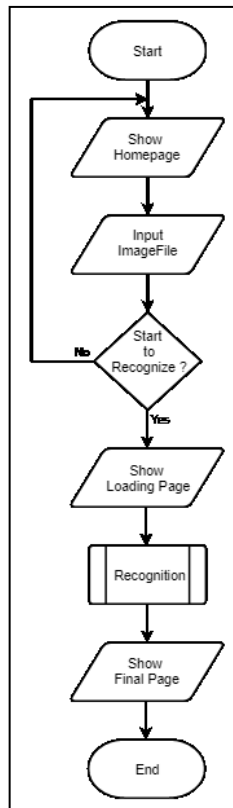
Fungsi aktivasi yang digunakan pada lapisan tersembunyi adalah ReLU dan fungsi aktivasi yang digunakan pada lapisan *output* adalah sigmoid. Pada tahap pelatihan, digunakan seluruh algoritma *backpropagation* untuk mendapatkan bobot dan bias optimal. Sedangkan, pada tahap pengenalan hanya dilakukan *forward propagation* dengan bobot dan bias yang sudah didapat pada tahap pelatihan. Arsitektur jaringan saraf tiruan yang digunakan dalam penelitian ini dapat dilihat pada Gambar 5.



Gambar 5. Arsitektur Jaringan Saraf Tiruan

B. Flowchart

Gambar 6. menunjukkan alur proses aplikasi secara keseluruhan. Ketika aplikasi dijalankan, halaman utama ditampilkan. Ketika *image file* sudah dipilih dan tombol *start to recognize* ditekan maka halaman *loading* ditampilkan dan proses *recognition* dijalankan. Setelah proses *recognition* selesai dijalankan maka halaman akhir ditampilkan.



Gambar 6. Flowchart Aplikasi

Pada proses *recognition* akan dilakukan pemrosesan gambar yang dimulai dengan pengubahan gambar menjadi *grayscale image*. Setelah input diubah menjadi *grayscale image*, proses *line extraction* dijalankan. Proses *line extraction* mengembalikan jumlah baris dan *array* yang berisi *image file* yang sudah dipotong per baris. Setelah proses *line extraction* dilakukan, dijalankan iterasi sebanyak jumlah baris. Selama iterasi dilakukan, proses *word extraction* dijalankan. Proses *word extraction* mengembalikan jumlah kata dan *array* yang berisi *image file* yang sudah dipotong per kata. Selanjutnya, dijalankan iterasi sebanyak jumlah kata. Selama iterasi dilakukan, proses *letter extraction* dijalankan. Proses *letter extraction* mengembalikan jumlah huruf dan *array* yang berisi *image file* yang sudah dipotong per huruf. Setelah itu, dijalankan iterasi sebanyak jumlah kata. Selama iterasi dilakukan, dijalankan proses *normalize* dan *forward propagation*. Proses *normalize* akan mengembalikan *image file* yang sudah dinormalisasi menjadi 15×15 piksel. *Image file* yang dikembalikan setelah proses

normalize dijadikan input pada proses *forward propagation* untuk dikenali. Proses *forward propagation*, mengembalikan *array* yang berisi *output* dari lapisan *output* dan lapisan tersembunyi. Setelah proses *forward propagation* selesai dilakukan, nilai maksimum dari *array* yang berisi *output* dari lapisan *output* dijadikan indeks untuk diklasifikasikan sebagai karakter.

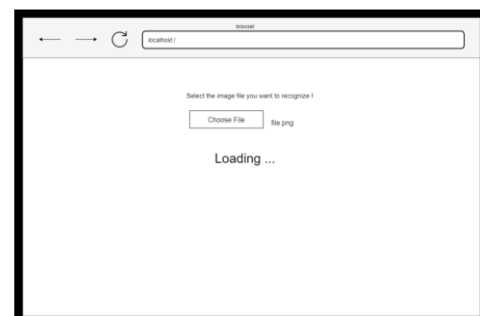
C. Rancangan Antarmuka Aplikasi

Rancangan antarmuka untuk implementasi jaringan saraf tiruan *backpropagation* untuk pengenalan karakter pada dokumen tercetak terdiri dari halaman awal, halaman *loading*, dan halaman akhir. Berikut ini merupakan rancangan antarmuka untuk implementasi jaringan saraf tiruan *backpropagation* untuk pengenalan karakter pada dokumen tercetak.



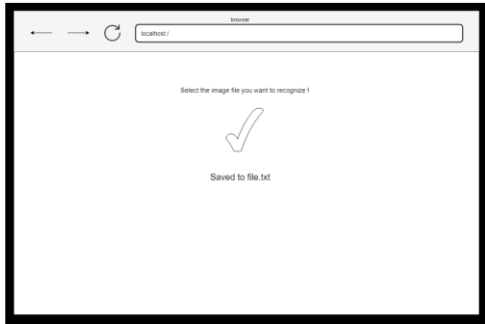
Gambar 7. Rancangan Halaman Awal

Gambar 7. merupakan rancangan antarmuka halaman awal aplikasi. Pada halaman ini terdapat tombol *choose file* dan *start to recognize*. Tombol *choose file* digunakan untuk memilih *image file* yang ingin dikenali. Sedangkan, tombol *start to recognize* digunakan untuk memulai pengenalan pada *image file*.



Gambar 8. Rancangan Halaman Loading

Gambar 8. merupakan rancangan antarmuka halaman *loading* aplikasi. Halaman ini ditampilkan saat proses pengenalan sedang dilakukan.



Gambar 9. Rancangan Halaman Akhir

Gambar 9. merupakan rancangan antarmuka halaman akhir aplikasi. Halaman ini ditampilkan saat proses pengenalan sudah selesai dilakukan.

Format persamaan merupakan suatu pengecualian di dalam spesifikasi *template* ini. Anda harus menentukan apakah akan menggunakan jenis tulisan Times New Roman atau Symbol (jangan jenis tulisan yang lain). Bila Anda membuat beberapa persamaan berbeda, akan lebih baik bila Anda mempersiapkan persamaan tersebut sebagai gambar dan menyisipkannya ke dalam artikel Anda setelah diberi *style*.

Beri penomoran untuk persamaan Anda secara berurutan. Nomor persamaan berada dalam tanda kurung seperti (1), dan diletakkan pada bagian kanan dengan menggunakan suatu *right tab stop*.

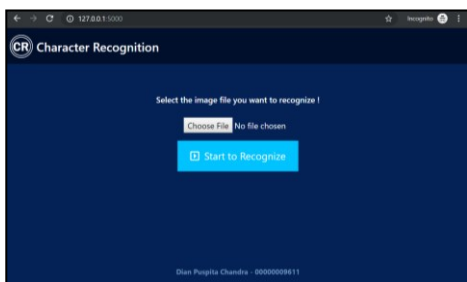
$$\int_0^{r_2} F(r, \phi) dr d\phi = [\sigma r_2 / (2\mu_0)] \quad (1)$$

Perhatikan bahwa persamaan di atas diposisikan di bagian tengah dengan menggunakan suatu *center tab stop*. Pastikan bahwa simbol-simbol yang digunakan dalam persamaan Anda didefinisikan sebelum atau sesudah persamaan. Gunakan "(1)," bukan "Persamaan (1)," kecuali pada awal sebuah kalimat, seperti "Persamaan (1) merupakan"

IV. IMPLEMENTASI DAN UJI COBA

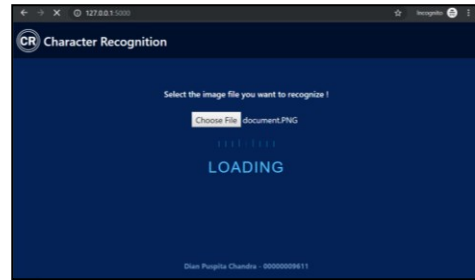
A. Implementasi Antarmuka

Gambar 10. menunjukkan hasil implementasi halaman awal. Pada halaman ini terdapat dua tombol yaitu *choose file* dan *start to recognize*. Tombol *choose file* digunakan untuk memilih *image file* yang ingin dikenali. Sedangkan, tombol *start to recognize* digunakan untuk memulai pengenalan pada *image file*.



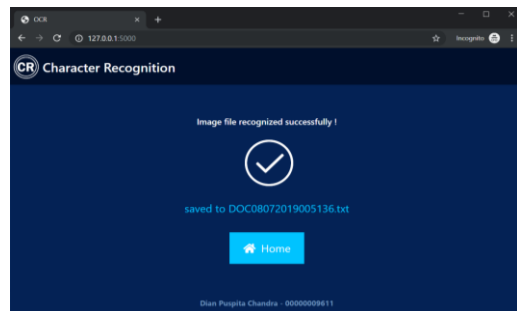
Gambar 10. Hasil Implementasi Halaman Awal

Gambar 11. menunjukkan hasil implementasi halaman *loading*. Halaman ini ditampilkan ketika tombol *start to recognize* pada halaman utama ditekan. Halaman ini ditampilkan sampai proses pengenalan selesai dilakukan.



Gambar 11. Hasil Implementasi Halaman Loading

Gambar 12. merupakan hasil implementasi halaman akhir. Halaman ini menampilkan nama *file* yang digunakan untuk menyimpan hasil pengenalan dan tombol *home* yang akan menampilkan halaman awal jika ditekan. Halaman ini ditampilkan setelah halaman *loading* setelah proses pengenalan selesai dilakukan.



Gambar 12. Hasil Implementasi Halaman Akhir

B. Uji Coba Jaringan Saraf Tiruan

Uji coba jaringan saraf tiruan digunakan untuk melihat kebenaran dari implementasi kode jaringan saraf tiruan *backpropagation* yang telah dibuat. Jaringan saraf tiruan diuji dengan menjalankan logika AND, OR, dan XOR. Arsitektur jaringan yang digunakan untuk uji coba adalah jaringan lapisan banyak (*multilayer network*) dengan tiga lapisan yaitu lapisan input dengan dua node, lapisan tersembunyi dengan tiga node, dan lapisan *output* dengan dua node yang merepresentasikan 0 dan 1. *Learning rate* yang digunakan adalah 0,1 dengan target *error* 0,01. Jaringan akan dilatih sampai total *error* lebih kecil dari target *error*.

| X1 | X2 | T |
|----|----|---|
| 0 | 0 | 0 |
| 1 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 1 | 1 |

Gambar 13. Tabel Kebenaran Logika OR

Gambar 13. merupakan gambar tabel kebenaran logika OR. Pada logika OR, *output* bernilai nol jika kedua input bernilai nol. Pelatihan jaringan saraf tiruan dengan logika OR dilakukan dengan *epoch* sebanyak 161 dengan semua hasil uji benar. Hasil uji logika OR dapat dilihat pada Gambar 14.

| Input | Output | Output pada Lapisan Output |
|---------|--------|---|
| [0. 0.] | 0 | [0.6980956521367753, 0.5521530076446559] |
| [0. 1.] | 1 | [0.20347556695954708, 0.7530506102134477] |
| [1. 0.] | 1 | [0.13256140609376277, 0.8243350577763973] |
| [1. 1.] | 1 | [0.03692253339567626, 0.9082554483297861] |

Gambar 14. Hasil Uji Logika OR

| X ₁ | X ₂ | T |
|----------------|----------------|---|
| 0 | 0 | 0 |
| 1 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 1 | 0 |

Gambar 15. Tabel Kebenaran Logika XOR

Gambar 15. merupakan gambar tabel kebenaran logika XOR. Pada logika XOR, *output* bernilai nol jika kedua input bernilai sama. Pelatihan jaringan saraf tiruan dengan logika XOR dilakukan dengan *epoch* sebanyak 1794 dengan semua hasil uji benar. Hasil uji logika XOR dapat dilihat pada Gambar 16.

| Input | Output | Output pada Lapisan Output |
|---------|--------|---|
| [0. 0.] | 0 | [0.91498314366016, 0.08824166371790268] |
| [0. 1.] | 1 | [0.08708269277816834, 0.9099110503987154] |
| [1. 0.] | 1 | [0.06800521021736916, 0.9285870375298074] |
| [1. 1.] | 0 | [0.9311800476349783, 0.07166826871819867] |

Gambar 16. Hasil Uji Logika XOR

Gambar 17. merupakan gambar tabel kebenaran logika AND. Pada logika AND, *output* bernilai satu jika kedua input bernilai satu. Pelatihan jaringan saraf tiruan dengan logika AND dilakukan dengan *epoch* sebanyak 2200 dengan semua hasil uji benar. Hasil uji logika AND dapat dilihat pada Gambar 18.

| X ₁ | X ₂ | T |
|----------------|----------------|---|
| 0 | 0 | 0 |
| 1 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 1 | 1 |

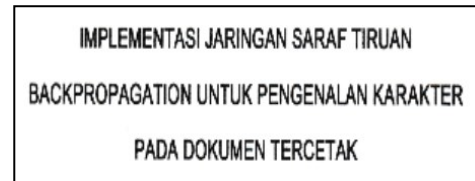
Gambar 17. Tabel Kebenaran Logika AND

| Input | Output | Output pada Lapisan Output |
|---------|--------|--|
| [0. 0.] | 0 | [0.9997257418370317, 0.00030755987632060414] |
| [0. 1.] | 0 | [0.9468622708217166, 0.053811517802038585] |
| [1. 0.] | 0 | [0.9380910725429703, 0.06540355640372532] |
| [1. 1.] | 1 | [0.06896374307797372, 0.9282451695412202] |

Gambar 18. Hasil Uji Logika AND

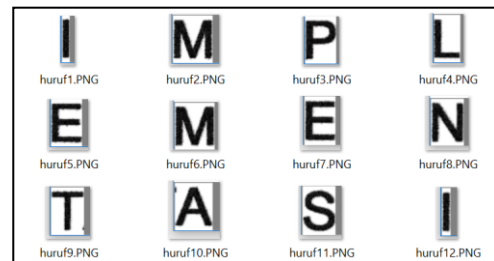
C. Evaluasi Hasil

Pengujian aplikasi dilakukan dengan menggunakan dokumen tercetak yang sudah dijadikan citra digital. Proses perubahan dokumen tercetak menjadi citra digital dilakukan menggunakan alat pemindai (*scanner*). Gambar 19. merupakan salah satu citra digital dengan *font* Arial Unicode Ms yang dijadikan sebagai input.



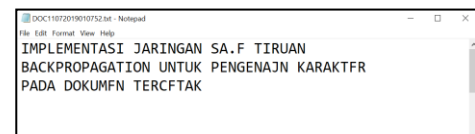
Gambar 19. Citra dengan Font Arial Unicode Ms

Citra akan diproses dengan proses *line extraction*, *word extraction*, dan *letter extraction* untuk mendapatkan potongan huruf dari citra. Citra yang sudah dipotong disimpan ke dalam *array* kemudian citra dinormalisasi dan dijadikan input pada proses *forward propagation* untuk dikenali sebagai karakter. Gambar 20. Merupakan hasil dari proses *letter extraction* dari citra kata pertama dari proses *word extraction* dan citra baris pertama dari proses *line extraction*.



Gambar 20. Hasil Proses Letter Extraction

Setelah proses pengenalan selesai dilakukan, hasil dari pengenalan karakter ditulis ke dalam *file*. *File* tersebut dijadikan sebagai *output* dari aplikasi dengan ekstensi .txt yang dapat dilihat pada Gambar 21.



Gambar 21. Hasil Output Aplikasi

Jumlah data *training* yang digunakan untuk pelatihan jaringan saraf tiruan adalah 486 data. Jumlah data *training font* Arial Unicode Ms adalah 162 data dengan 54 karakter yang masing-masing berjumlah tiga data. Jumlah data *training font* Tahoma adalah 162 data dengan 54 karakter yang masing-masing berjumlah tiga data. Jumlah data *training font* Times New Roman adalah 162 data dengan 54 karakter yang masing-masing berjumlah tiga data.

Pengujian dilakukan dengan 15 dokumen tercetak yang sudah berupa citra digital dengan jenis *font* Arial Unicode Ms, Tahoma, dan Times New Roman masing-masing sebanyak lima. Seluruh citra digital dihitung tingkat akurasi dengan membagi jumlah karakter uji yang dikenali dengan jumlah karakter uji keseluruhan. Akurasi yang dihasilkan dari hasil penelitian dapat dilihat pada Tabel 1.

Tabel 1. Hasil Uji Coba

| Nama | Font | Persentase Tingkat Akurasi |
|----------------|------------------|----------------------------|
| Citra Input 1 | Arial Unicode Ms | 92 % |
| Citra Input 2 | Arial Unicode Ms | 91 % |
| Citra Input 3 | Arial Unicode Ms | 98 % |
| Citra Input 4 | Arial Unicode Ms | 94 % |
| Citra Input 5 | Arial Unicode Ms | 97 % |
| Citra Input 6 | Tahoma | 96 % |
| Citra Input 7 | Tahoma | 97 % |
| Citra Input 8 | Tahoma | 98 % |
| Citra Input 9 | Tahoma | 96 % |
| Citra Input 10 | Tahoma | 96 % |
| Citra Input 11 | Times New Roman | 95 % |
| Citra Input 12 | Times New Roman | 94 % |
| Citra Input 13 | Times New Roman | 93 % |
| Citra Input 14 | Times New Roman | 95 % |
| Citra Input 15 | Times New Roman | 93 % |

Nilai rata-rata tingkat akurasi dari setiap *font* didapat dengan perhitungan sebagai berikut.

$$\text{Akurasi Arial Unicode Ms} = \frac{(0,92 + 0,91 + 0,98 + 0,94 + 0,97)}{5} \times 100 \%$$

$$\text{Akurasi Arial Unicode Ms} = 94 \%$$

$$\text{Akurasi Tahoma} = \frac{(0,96 + 0,97 + 0,98 + 0,96 + 0,96)}{5} \times 100 \%$$

$$\text{Akurasi Tahoma} = 96,6 \%$$

$$\text{Akurasi Times New Roman} = \frac{(0,95 + 0,94 + 0,93 + 0,95 + 0,93)}{5} \times 100 \%$$

$$\text{Akurasi Times New Roman} = 94 \%$$

Jenis *font* Tahoma menghasilkan rata-rata tingkat akurasi yang paling tinggi yaitu sebesar 96,6 % disusul dengan rata-rata tingkat akurasi jenis *font* Arial Unicode Ms sebesar 94 % dan Jenis *font* Times New Roman sebesar 94 %.

V. SIMPULAN

Aplikasi pengenalan karakter pada dokumen tercetak dibuat dengan bahasa pemrograman python

dan OpenCV *library* untuk bagian pengolahan citra. Arsitektur jaringan saraf tiruan yang digunakan adalah jaringan lapisan banyak (*multilayer network*) dengan tiga lapisan. Lapisan input terdiri dari 225 node dengan masukan berupa citra digital berukuran 15 × 15 piksel, lapisan tersembunyi terdiri dari 110 node, dan lapisan *output* terdiri dari 54 node yang merepresentasikan karakter A-Z, a-z, serta tanda baca titik (.) dan koma (,). *Learning rate* yang digunakan dalam penelitian ini adalah 0,29. Jaringan dibuat agar bisa mengenali karakter dengan jenis *font* Arial Unicode Ms, Tahoma, dan Times New Roman. Rata-rata tingkat akurasi yang didapat adalah 94 % untuk jenis *font* Arial Unicode Ms, 96,6 % untuk jenis *font* Tahoma, dan 94 % untuk jenis *font* Times New Roman.

DAFTAR PUSTAKA

- [1] R. Susilawati. (2012). Strategi Paperless Office di STIE Nusa Megarkencana. *Jurnal Ekonomi dan Kewirausahaan*, [online] 8(16), hal.61-69. Tersedia di: <http://ejournal.stienusa.ac.id/index.php/ekowir/article/view/35>
- [2] R. Mithe, S. Indalkar, dan N. Divekar. (2013). Optical Character Recognition. *International Journal of Recent Technology and Engineering*, [online] 2(1), hal.72-75. Tersedia di: <https://pdfs.semanticscholar.org/6a4b/4f04d5ce3c3592832eb40c23cc8fc5a9131e.pdf>
- [3] P. Bairagi. (2018). Optical Character Recognition for Hindi. *International Research Journal of Engineering and Technology*, [online] 5(5), hal.3968-3973. Tersedia di: <https://www.irjet.net/archives/V5/I5/IRJET-V5I5846.pdf>
- [4] D. Li, P. Yu, H. Li, dan G. Peng, "Printed New Tai Lue Character Recognition Based on BP Neural Network," dalam *IEEE International Conference on Signal and Image Processing*, 2016.
- [5] A. Radhiah, C. Machbub, E. Hidayat, dan A. Prihatmanto, "Printed Arabic Letter Recognition Based On Image," dalam *International Conference on Signals and Systems*, 2018.
- [6] S. Afroge, B. Ahmed, dan F. Mahmud, "Optical Character Recognition using Back Propagation Neural Network," dalam *International Conference on Electrical, Computer & Telecommunication Engineering*, 2016.
- [7] M. Nugroho, "Perbandingan Beberapa Metode Praproses Pada Pengenalan Karakter Tulisan Tangan Menggunakan Jaringan Syaraf Tiruan Propagasi Balik," Skripsi, Jurusan Ilmu Komputer, Institut Pertanian Bogor. Indonesia, 2003.
- [8] I. Malik, "Pembuatan Aplikasi Pengolahan Citra Teks Inggris Dan Penerjemahannya Ke Dalam Bahasa Indonesia Berbasis Sistem Operasi Android," Skripsi, Universitas Komputer Indonesia. Indonesia, 2014.
- [9] F. Hietania, W. Santiyasa, dan I. Dwidasmara. (2012). Implementasi Backpropagation Dalam Pengolahan Citra Teks Tulisan Tangan Menjadi Teks Digital. *Jurnal Elektronik Ilmu Komputer Udayana*, [online] 1(1), hal.1-10. Tersedia di: <https://ojs.unud.ac.id/index.php/JLK/article/view/2789>
- [10] A. Kadir, dan A. Susanto, *Teori dan Aplikasi Pengolahan Citra*. Yogyakarta: ANDI, 2013.
- [11] F. Hermawati, *Pengolahan Citra Digital Konsep dan Teori*. Yogyakarta: ANDI, 2013.
- [12] A. Wedianto, H. Sari, dan Y. Suzantri. (2016). Analisa Perbandingan Metode Filter Gaussian, Mean dan Median Terhadap Reduksi Noise. *Jurnal Media Infotama*, [online]

- 12(1), hal.21-30. Tersedia di: <https://jurnal.unived.ac.id/index.php/jmi/article/view/269>
- [13] T. Sutojo, E. Mulyanto, dan V. Suhartono, *Kecerdasan Buatan*. Yogyakarta: ANDI, 2011.
- [14] E. Handoyo, dan L. Susanto. (2011). Penerapan Jaringan Syaraf Tiruan metode Propagasi Balik Dalam Pengenalan Tulisan Tangan Huruf Jepang Jenis Hiragana dan Katakana. *Jurnal Informatika*, [online] 7(1), hal.39-55. Tersedia di : <https://repository.maranatha.edu/373/>
- [15] I. Wirawan, *Metode Penalaran dalam Kecerdasan Buatan*. Depok: Rajawali Pers, 2017, hal.135-186.
- [16] I. Suhardi. (2007). Evaluasi Pengaruh Fungsi Aktivasi Dan Parameter Kemiringannya Terhadap Unjuk Kerja Pengenalan Jaringan Syaraf Tiruan. *Jurnal Ilmiah Semesta Teknika*, [online] 10(1), hal.53-68. Tersedia di: <http://journal.umy.ac.id/index.php/st/article/view/854>
- [17] M. Wibawa. (2017). Pengaruh Fungsi Aktivasi, Optimisasi dan Jumlah Epoch Terhadap Performa Jaringan Saraf Tiruan. *Jurnal Sistem Dan Informatika (JSI)*, [online] 11(2), hal.167-174. Tersedia di: <https://jsi.stikombali.ac.id/index.php/jsi/article/view/129>
- [18] K. Shibata, dan Y. Ikeda, "Effect of number of hidden neurons on learning in large-scale layered neural networks," dalam *ICROS-SICE International Joint Conference*, 2009.