

**ANALISIS PERBANDINGAN FUNGSI AKTIVASI SIGMOID
BIPOLAR DENGAN SIGMOID BINER DALAM PENGENALAN
AKSARA BATAK TOBA DENGAN ALGORITMA
*BACKPROPAGATION***

SKRIPSI

OLEH:

**ESRAYANTI SIMANJUNTAK
178160055**



**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS MEDAN AREA
MEDAN
2022**

UNIVERSITAS MEDAN AREA

© Hak Cipta Di Lindungi Undang-Undang

1. Dilarang Mengutip sebagian atau seluruh dokumen ini tanpa mencantumkan sumber
2. Pengutipan hanya untuk keperluan pendidikan, penelitian dan penulisan karya ilmiah
3. Dilarang memperbanyak sebagian atau seluruh karya ini dalam bentuk apapun tanpa izin Universitas Medan Area

Document Accepted 27/12/22

Access From (repository.uma.ac.id)27/12/22

**ANALISIS PERBANDINGAN FUNGSI AKTIVASI SIGMOID
BIPOLAR DENGAN SIGMOID BINER DALAM PENGENALAN
AKSARA BATAK TOBA DENGAN ALGORITMA
*BACKPROPAGATION***

SKRIPSI

Diajukan sebagai Salah Satu Syarat untuk Memperoleh

Gelar Sarjana di Fakultas Teknik

Universitas Medan Area

Oleh:

ESRAYANTI SIMANJUNTAK

178160055

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS MEDAN AREA
MEDAN
2022**

UNIVERSITAS MEDAN AREA

© Hak Cipta Di Lindungi Undang-Undang

1. Dilarang Mengutip sebagian atau seluruh dokumen ini tanpa mencantumkan sumber
2. Pengutipan hanya untuk keperluan pendidikan, penelitian dan penulisan karya ilmiah
3. Dilarang memperbanyak sebagian atau seluruh karya ini dalam bentuk apapun tanpa izin Universitas Medan Area

LEMBAR PENGESAHAN

Judul Skripsi : Analisis Perbandingan Fungsi Aktivasi Sigmoid Bipolar Dengan Sigmoid Biner Dalam Pengenalan Aksara Batak Toba Dengan Algoritma *Backpropagation*


Nama : Esrayanti Simanjuntak

NPM : 178160055

Fakultas : Teknik

Disetujui Oleh
Komisi Pembimbing


Nurul Khairina, S.Kom, M.Kom
Pembimbing I


Zulfikar Sembiring, S.Kom, M.Kom
Pembimbing II

Diketahui :


Dr. Kurniadi Syah, S.Kom, M.Kom
NIDN : 0105058804


Rizki Maulana, S.Kom, M.Kom
NIDN : 0109038902

Tanggal Lulus : 26-09-2022

HALAMAN PERNYATAAN

Dengan ini saya menyatakan dengan sejujurnya bahwa skripsi ini adalah hasil dari penelitian dan presentasi saya. Oleh sebab itu, saya tidak mencantumkan hasil karya orang lain yang telah diterbitkan sebelumnya, atau bahan yang pernah diajukan untuk gelar sarjana di Universitas Medan Area ataupun perguruan tinggi yang lainnya. Apabila suatu saat nanti terdapat kejanggalan dalam pernyataan saya ini, maka saya bersedia menerima sanksi akademik yang berlaku sesuai dengan ketentuan di Universitas Medan Area. Demikian pernyataan ini saya buat.

Medan, 26/09/2022

Yang membuat pernyataan,


Esrayanti Simanjuntak

178160055



HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI
TUGAS AKHIR/SKRIPSI/TESIS UNTUK KEPENTINGAN AKADEMIS

Sebagaimana civitas akademika Universitas Medan Area, bahwa saya yang bertanda tangan di bawah ini:

Nama : Esrayanti Simanjuntak
NPM : 178160055
Program Studi : Teknik Informatika
Fakultas : Teknik
Jenis Karya : Skripsi

Dalam pengembangan ilmu pengetahuan, setuju untuk memberikan kepada Universitas Medan **Hak Bebas Royalti Non-eksklusif (*Non-exclusve Royalty-Free Right*)** atas karya ilmiah saya dengan judul:

Analisis Perbandingan Fungsi Aktivasi Sigmoid Bipolar Dengan Sigmoid Biner Dalam Pengenalan Aksara Batak Toba Dengan Algoritma Backpropagation

Bersama dengan perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti yang bersifat *non-eksklusif* ini, Universitas Medan Area berhak menyimpan, mengalihkan media/format, mengelola dalam bentuk database, memelihara dan mempublikasikan tugas akhir/tesis/skripsi saya selama saya tetap menyebut nama saya sebagai pencipta/penulis dan sebagai pemilik hak cipta. Demikian Surat pernyataan ini saya buat dengan sebenarnya.

Dibuat di: Medan

Pada tanggal: 26-09-2022

Yang menyatakan



(Esrayanti Simanjuntak)

ABSTRAK

Pengenalan aksara kuno umumnya masih dilakukan secara manual dan belum ada sistem yang dapat mengenali dengan tingkat akurasi yang baik. Oleh sebab itu, diperlukan sebuah teknologi pengenalan pola untuk membantu mempercepat proses pengenalan dengan cara menginput data berupa citra digital tanpa harus memasukkan data tersebut tanpa mengetik. Pada penelitian ini dilakukan analisis perbandingan penggunaan fungsi aktivasi Sigmoid Bipolar dengan Sigmoid Biner pada algoritma *Backpropagation* untuk pengenalan aksara Batak Toba dimana pada awal dilakukan *praprosesing* file-file aksara Batak Toba yaitu *resizing* dengan ukuran yang seragam serta pengolahan citra menjadi citra biner. Setelah dilakukan pengujian pengenalan dengan algoritma *Backpropagation* menggunakan 190 data *training* yang terdiri dari 19 aksara Batak Toba serta 8 kali pengujian dengan menggunakan *epoch* 150, 300, 450, 600, 750, 900, 1050 dan 1200 maka diperoleh bahwa nilai akurasi yang terbaik pada penggunaan fungsi aktivasi Sigmoid Bipolar pada *epoch* ke 1050 sebesar 80.53 % dan pada Sigmoid Biner pada *epoch* ke 1050 sebesar 78.95 %.

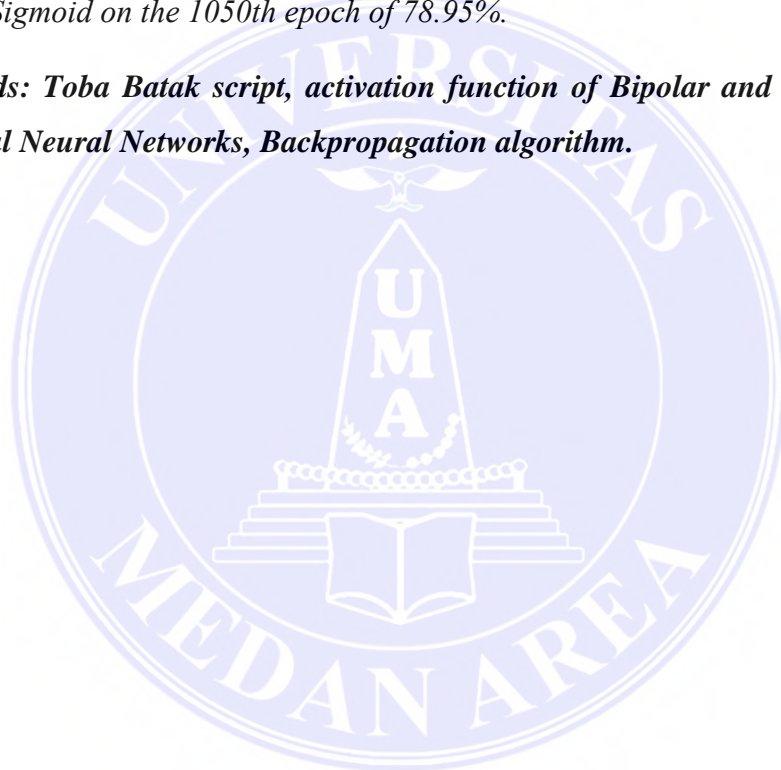
Kata kunci: Aksara Batak Toba, fungsi Aktivasi Sigmoid Biner, Fungsi Aktivasi Sigmoid Bipolar, Jaringan Saraf Tiruan, Algoritma *Backpropagation*.



ABSTRACT

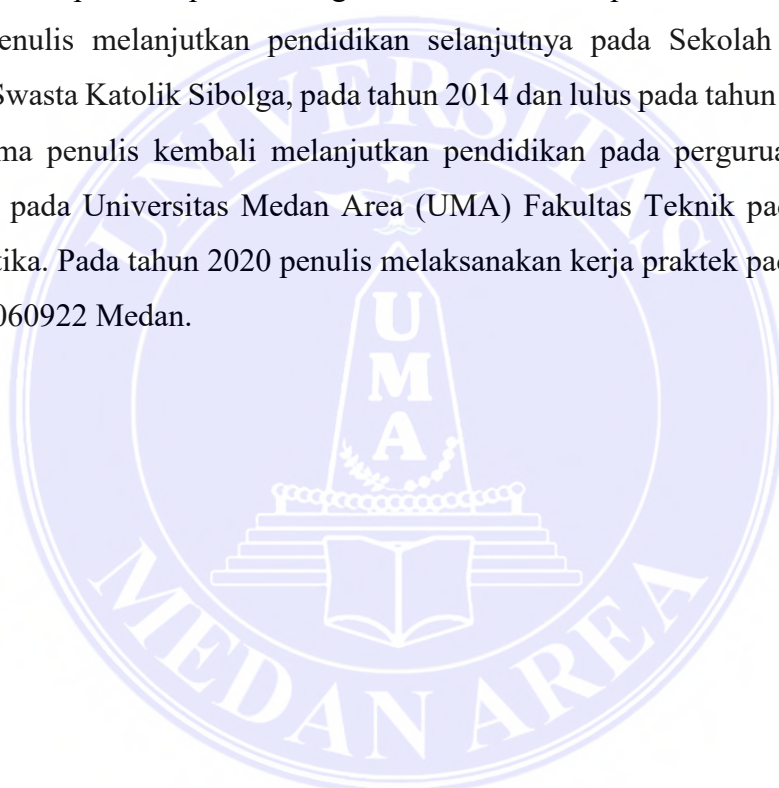
The introduction of ancient characters is generally still done manually and there is no system that can recognize it with a good level of accuracy. Therefore, we need a pattern recognition technology to help speed up the recognition process by inputting data in the form of digital images without having to enter the data without typing. In this study, a comparative analysis of the use of the Sigmoid Bipolar with Binary activation function was carried out on the Backpropagation pada algorithm for the introduction of the Toba Batak script where at the beginning the Toba Batak script files were preprocessed, namely resizing, which is a uniform size and processing the image into a binary image. After testing the recognition using the Backpropagation algorithm using 190 training data consisting of 19 Toba Batak characters and 8 times testing using epochs 150, 300, 450, 600, 750, 900, 1050 and 1200, it was obtained that the best accuracy value on the use of the Bipolar Sigmoid activation function at the 1050th epoch of 80.53% and the Binary Sigmoid on the 1050th epoch of 78.95%.

Keywords: *Toba Batak script, activation function of Bipolar and Sigmoid Binary, Artificial Neural Networks, Backpropagation algorithm.*



RIWAYAT HIDUP

Esrayanti Simanjuntak, dilahirkan di Sitahan Barat tanggal 15 Maret 1999. Anak terakhir dari lima bersaudara yaitu pasangan dari Henri Simanjuntak dan Noderia Tanjung. Penulis menyelesaikan pendidikan sekolah dasar di Sekolah Dasar (SD) N 153001 Kolang Nauli 2, Kecamatan Kolang, Kabupaten Tapanuli Tengah pada tahun 2011. Pada tahun yang sama penulis melanjutkan pendidikan Sekolah Menengah Pertama (SMP) pada Sekolah Menengah Pertama (SMP) Negeri 1 Kolang, Kecamatan Kolang, Kabupaten Tapanuli Tengah, selama 3 tahun penuh dan selesai pada tahun 2014. Penulis melanjutkan pendidikan selanjutnya pada Sekolah Menengah Atas (SMA) Swasta Katolik Sibolga, pada tahun 2014 dan lulus pada tahun 2017. Pada tahun yang sama penulis kembali melanjutkan pendidikan pada perguruan tinggi swasta, tepatnya pada Universitas Medan Area (UMA) Fakultas Teknik pada program studi Informatika. Pada tahun 2020 penulis melaksanakan kerja praktek pada Sekolah Dasar (SD) N 060922 Medan.



KATA PENGANTAR

Puji syukur atas kehadiran Tuhan Yang Maha Kuasa atas rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan penyusunan skripsi dengan judul “Analisis Perbandingan Fungsi Aktivasi Sigmoid Bipolar Dengan Sigmoid Biner Dalam Pengenalan Aksara Batak Toba Dengan Algoritma Backpropagation”. Penulis sangat menyadari bahwa pada penulisan skripsi ini dapat terselesaikan secara baik berkat dorongan, motivasi, bimbingan, kerjasama dari berbagai pihak tertentu. Oleh sebab itu, dimana pada kesempatan kali ini penulis menyampaikan banyak terima kasih kepada:

1. Kedua Orang Tua dan saudara-saudara penulis yang telah mendukung, memberi semangat, motivasi, dan banyak perhatian serta memenuhi segala kebutuhan yang dibutuhkan penulis selama masa penyusunan skripsi ini.
2. Bapak Prof. Dr. Dadan Ramdan, M.Eng, M.Sc, selaku Rektor Universitas Medan Area.
3. Bapak Dr. Rahmad Syah, S.Kom, M.Kom selaku Dekan Fakultas Teknik Universitas Medan Area.
4. Ibu Susilawati, S.Kom M.Kom, selaku Wakil Dekan Bidang Pendidikan, Penelitian Dan Pengabdian Universitas Medan Area.
5. Bapak Rizki Muliono, S.Kom, M.Kom, selaku Ketua Prodi Teknik Informatika Universitas Medan Area.
6. Ibu Nurul Khairina, S.Kom, M.Kom, selaku dosen pembimbing I yang telah memberikan banyak masukan, kritik, saran dan motivasi kepada penulis serta membimbing penulis dalam menyelesaikan skripsi ini terselesaikan.
7. Bapak Zulfikar Sembiring, S.Kom, M.Kom, selaku dosen pembimbing II, yang telah memberikan arahan, bimbingan, semangat, motivasi dan dorongan kepada penulis hingga penyusunan skripsi ini terselesaikan.
8. Serta semua pihak yang telah membantu penulis dalam menyelesaikan skripsi ini, yang namanya tidak bisa disebutkan satu persatu. Terima kasih banyak.

Sebagai manusia, penulis tidak pernah luput dari kesalahan, penulis menyadari bahwa Skripsi ini masih jauh dari kata sempurna. Oleh karena itu, kritik dan saran yang membangun sangat diharapkan untuk pengembangan selanjutnya.

Medan, 26 September 2022

Esrayanti Simanjuntak

NPM. 178160055



DAFTAR ISI

HALAMAN PENGESAHAN	iii
HALAMAN PERNYATAAN.....	iv
HALAMAN PERSTUJUAN PUBLIKASI	v
ABSTRACT.....	vi
RIWAYAT HIDUP.....	viii
KATA PENGANTAR.....	ix
DAFTAR ISI.....	xi
DAFTAR TABEL.....	xiv
DAFTAR GAMBAR.....	xv
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	4
1.3 Tujuan Penelitian.....	4
1.4 Manfaat Penelitian.....	4
1.5 Batasan Masalah.....	4
1.6 Sistematika Penulisan.....	5
BAB II TINJAUAN PUSTAKA.....	6
2.1 Aksara Batak	6
2.1.1 Ibu Surat.....	7
2.2 Huruf Batak	9
2.3 Aplikasi <i>Optical Character Recognition</i> (OCR).....	9
2.4 Citra.....	9
2.5 Gambaran Sebuah Citra	10
2.5.1 Gambaran Analog Sebuah Citra.....	11
2.5.2 Citra Digital.....	11
2.5.3 Jenis Citra Digital	11
2.5.4 Format Citra Digital	13
2.5.5 Format Data Bitmap	13
2.5.6 Format JPEG	14
2.6 Metode Algoritma <i>Backpropagation</i>	14

2.6.1 Tahap Propagasi Maju	15
2.6.2 Tahap Propagasi Balik	15
2.6.3 Tahap Modifikasi Bobot	16
2.6.4 Tahap Pembelajaran	16
2.7 Fungsi Aktivasi	19
2.8 Algoritma Mark Direction dan Intensity of Character	20
2.9 Metode Normalisasi Data	20
2.10 Penelitian Terkait	21
BAB III ANALISIS DAN PERANCANGAN	22
3.1 Analisis	22
3.1.1 Teknik pengumpulan Data	22
3.1.2 Menangkap Gambar	23
3.1.3 Data Pelatihan	23
3.1.4 Data Pengujian	25
3.1.5 Diagram Penelitian	25
3.1.6 <i>Preprocessing</i>	26
3.1.7 <i>Feature Extraction</i>	29
3.1.8 Normalisasi Data	34
3.1.9 <i>Flowchart</i> Pelatihan	38
3.1.10 Arsitektur Jaringan	40
3.1.11 Proses Pelatihan	41
3.1.12 Proses Pengenalan Aksara Batak	49
3.1.13 Perancangan Aplikasi	51
BAB IV HASIL DAN PEMBAHASAN	53
4.1 Hasil	53
4.1.1 Tampilan Aplikasi	53
4.1.2 Pengujian Data	55
4.2 Pembahasan	73
BAB V KESIMPULAN DAN SARAN	73
5.1 Kesimpulan	74
5.2 Saran	74
DAFTAR PUSTAKA	75

DAFTAR GAMBAR

Gambar 2.1. Silsilah aksara.....	6
Gambar 2.2. Koordinat pada grafik matematik	10
Gambar 2.3. Koordinat pada citra.....	10
Gambar 2.4. Proses sampling dan kuantisasi.....	11
Gambar 2.5. Citra Warna	13
Gambar 2.6. Arsitektur Jaringan	15
Gambar 2.7 Fungsi Aktivasi Sigmoid Biner	19
Gambar 2.8. Fungsi Aktivasi Sigmoid Bipolar.....	19
Gambar 3.1. Aksara Batak Toba.....	23
Gambar 3.2. Data Pelatihan Aksara Batak Toba	23
Gambar 3.3. Data Pengujian Aksara Batak Toba	24
Gambar 3.4. <i>Flowchart</i> Proses Pengenalan Aksara Batak Toba	25
Gambar 3.5. <i>Resize</i> Citra Menjadi 3x3 Piksel	26
Gambar 3.6. Nilai RGB	27
Gambar 3.7. Nilai Piksel <i>Grayscale</i>	27
Gambar 3.8. Matriks Citra Biner	28
Gambar 3.9. Citra Biner Pada Aksara.....	29
Gambar 3.10. Citra Dengan 6 Bagian.....	29
Gambar 3.11 Ciri Piksel 1 <i>Black</i> Bagian 1	30
Gambar 3.12. <i>Masking Left Diagonal</i> Citra Diagonal Bagian 1	30
Gambar 3.13. <i>Masking Diagonal</i> Citra Aksara.....	31
Gambar 3.14. Citra Aksara Dengan <i>Masking Horizontal</i>	31
Gambar 3.15. Citra Aksara Dengan <i>Masking</i> Bagian 1	32
Gambar 3.16 <i>Flowchart</i> Pelatihan	39
Gambar 3.17 <i>Flowchart</i> Proses Pengenalan	40
Gambar 3.18 Rancangan Aplikasi	51
Gambar 4.1 Tampilan Menu Utama	53
Gambar 4.2 Tampilan <i>Training</i>	54
Gambar 4.3 Tampilan Pengenalan.....	54
Gambar 4.4 Hasil Pengujian Dengan Fungsi Aktivasi Sigmoid Bipolar	63
Gambar 4.5 Hasil Pengujian Dengan Fungsi Aktivasi Sigmoid Biner.....	72

DAFTAR TABEL

Tabel 2.1. Huruf-huruf ina ni surat	7
Tabel 2.2 Anak Ni Surat	8
Tabel 2.3 Jumlah Bit Per Piksel	14
Tabel 2.4. Penelitian Terdahulu	21
Tabel 3.1 Ciri Hitam	32
Tabel 3.2 Ekstraksi Ciri Pada Diagonal Kiri.....	33
Tabel 3.3. Ekstraksi Ciri Pada Diagonal Kanan.....	33
Tabel 3.4. Ekstraksi Ciri Untuk <i>Vertical</i>	34
Tabel 3.5 Ekstraksi Ciri Posisi <i>Horizontal</i>	34
Tabel 3.6 Ekstraksi Ciri <i>Black Normal</i>	35
Tabel 3.7 Ekstraksi Ciri Pada Diagonal Kiri Normal	36
Tabel 3.8 Ciri Ekstraksi Bagian Diagonal Kanan	36
Tabel 3.9 Ekstraksi Ciri <i>Vertical Normal</i>	37
Tabel 3.10 Ekstraksi Ciri <i>Horizontal Normal</i>	37
Tabel 3.11 Ekstraksi Ciri	41
Tabel 3.12 Data <i>Input Proses Training Backpropagation</i>	42
Tabel 3.13 Biner Target	43
Tabel 4.1 Hasil Percobaan dengan nilai 150 <i>epoch</i>	55
Tabel 4.2 Hasil Percobaan dengan nilai 300 <i>epoch</i>	56
Tabel 4.3 Hasil Percobaan dengan nilai 450 <i>epoch</i>	57
Tabel 4.4 Hasil Percobaan dengan nilai 600 <i>epoch</i>	58
Tabel 4.5 Hasil Percobaan dengan nilai 750 <i>epoch</i>	59
Tabel 4.6 Hasil Percobaan dengan nilai 900 <i>epoch</i>	60
Tabel 4.7 Hasil Percobaan dengan nilai 1050 <i>epoch</i>	61
Tabel 4.8 Hasil Pengujian dengan nilai 1200 <i>epoch</i>	62
Tabel 4.9 Hasil Akurasi Pengujian Fungsi Aktivasi Sigmoid Bipolar	63
Tabel 4.18 Hasil Akurasi Pengujian Keseluruhan Aktivasi Sigmoid Biner	72

BAB I

PENDAHULUAN

1.1 Latar Belakang

Aksara Batak ialah suatu keanekaragaman seni dan budaya yang ada di Indonesia antara lain kesenian adalah tarian, bahasa, musik dan masih banyak lagi. Suku Batak itu terdiri dari lima jenis suku yaitu Mandailing, Simalungun, Toba serta Pakpak. Sejarahnya aksara Batak termasuk kedalam rumpun tulisan India dimana aksara India yang tertua yaitu tulisan Brahmi yang dimana menurunkan dua kelompok yakni India Selatan dan India Utara (Kozok, 2009).

Terdapat dua jenis perangkat tulisan aksara batak yaitu ibu surat (huruf utama) dan anak surat (huruf turunan). Aksara Batak juga bisa diklasifikasikan menjadi abugida (jenis huruf fonetis yang setiap suara bahasanya bisa dilambangkan secara akurat). Indonesia merupakan negara kepulauan mempunyai banyak warisan budaya yang sangat beragam, diantaranya adalah aksara masing-masing wilayah yang dianggap aksara Nusantara. Aksara Batak adalah rumpun menurut aksara nusantara ini perlu mendapat perhatian dampak keterbatasan data dan informasi (Winardi, 2015).

Sistem JST seperti layaknya manusia, akan belajar dari suatu contoh ataupun kasus, dimana JST dibentuk untuk memecahkan suatu masalah tertentu seperti pengenalan pola atau klasifikasi karena proses pembelajaran dan salah satu algoritma yang populer pada JST adalah *Backpropagation*

Fungsi aktivasi ialah salah satu parameter yang sangat terpenting dalam jaringan syaraf tiruan yang dimana fungsi ini tidak hanya menentukan keputusan akhir dan juga fungsi aktivasi ini mampu menunjukkan total signal dari simpul inputan data. Oleh sebab itu, pemilihan sebuah fungsi aktivasi tidak dapat secara sembarangan dipilih karena sangat besar berdampak pada performa jaringan (Kusumadewi, 2018).

Pengenalan dokumen aksara batak selama ini masih dilakukan secara manual. Dr. Uli Kozok sudah menciptakan tulisan aksara Batak telah di jalankan pada Windows, Linux, dan Apple OSX namun belum terdapat sistem yang bisa

mengenali dokumen aksara Batak dengan cara yang baik. Oleh sebab itu, penerapan sebuah teknologi diharapkan agar bisa mengenalkan dokumen aksara Batak dan juga dapat membantu mempercepat *user* agar lebih mudah mengolah informasi dari citra tanpa harus memasukkan data dengan cara mengetik.

Penelitian sebelumnya yang dilakukan oleh (Adinugroho & Sari 2017) tentang Pengenalan Tulisan Tangan Digit Menggunakan *Learning Vector Quantization*. Dalam tulisan ini digunakan sepuluh fitur struktural untuk pengenalan tulisan tangan. Sistem pengenalan digit tergantung pada jenis fitur yang digunakan dan data yang digunakan pada sistem ini sebanyak 50000 gambar data pelatihan dan 5000 data Pengenalan. Penelitian berikutnya yaitu perbandingan pengenalan tulisan tangan dengan aksara Batak yang telah dicetak menggunakan metode *Discrete Hidden Markov Models* (HMM). Dalam penelitian ini, setelah langkah *preprocessing*, dilanjutkan proses segmentasi menggunakan Algoritma *Backpropagation*. Hasil dari penelitian ini menunjukkan tingkat pengenalan 93,40%.

Selanjutnya untuk penelitian ini, penulis memakai metode *Backpropagation* dimana pada metode ini dilakukan pelatihan untuk lapisan kompetitif yang terawasi. Pada lapisan yang kompetitif ini yang terdiri dari lapisan *Input*, *Hidden*. Dimana *output* agar dapat mengklasifikasi vektor *input* dan nilai *output* berdasarkan proses pelatihan dalam lapisan kompetitif dimana akan diperoleh sesuai dengan waktu antara vektor *input*. Maka dua vektor masukan kompetitif akan meletakkan ke dua vektor *input* tersebut kedalam contoh yang sama. Dengan adanya sistem pengenalan dokumen aksara batak diharapkan dapat mempermudah pengguna untuk mengenali dokumen batak dengan baik.

Pada penelitian sebelumnya yang dilakukan oleh (Sry Redjeki, 2013) Tentang Analisis Fungsi Aktivasi Sigmoid Dengan Algoritma *Backpropagation* Pada Perediksi Data. Dari hasil penelitian yang dilakukan sebelumnya tingkat akurasi sigmoid biner lebih unggul dibandingkan dengan sigmoid bipolar akan tetapi dari segi kecepatan pembelajaran tersebut sigmoid bipolar jauh lebih cepat dibandingkan sigmoid biner. Dapat dilihat dari segi tingkat rata-rata akurasi yang dihasilkan fungsi aktivasi sigmoid biner yaitu 60-70% sedangkan sigmoid bipolar yaitu 50-60%.

Dari hasil penelitian yang dilakukan oleh (Iwan Suhardi, 2018) jaringan saraf tiruan pada pengenalan pola karakter angka memberikan pengenalan yang sangat cukup baik, tingkat pengenalan jaringan menggunakan fungsi aktivasi sigmoid biner dengan Sigmoid Bipolar yang dimana tingkat akurasi yang lebih baik yaitu dengan menggunakan Sigmoid Bipolar mencapai sekitar 81% pada asitektur jaringan 16 neuron.

Metode *Backpropagation* adalah suatu metode dalam menangani kasus pengenalan pola yang sangat kompleks. Jika unit yang terdapat pada lapisan *input* telah terhubung dengan unit yang ada di lapisan tersembunyi maka sebaliknya *output* juga akan terhubung pada lapisan tersembunyi, pada jaringan saraf tiruan terdapat beberapa fungsi aktivasi yang dimana fungsi dari aktivasi ini yaitu untuk mengaktifkan dan menonaktifkan neuron. Sehingga untuk metode *Backpropagation* ini digunakan fungsi aktivasi sigmoid bipolar dan sigmoid biner yang dimana karakteristik yang dimiliki fungsi aktivasi sigmoid bipolar dan sigmoid biner adalah kontinyu dan tidak menurun secara monoton (Julpan, Nababan, E. B. & Zarlis, M., 2016). Yang dimana untuk fungsi aktivasi sigmoid bipolar, pada fungsi ini *output* yang dihasilkan yaitu 1, 0 atau -1 sedangkan untuk sigmoid biner nilai yang dihasilkan yaitu rentang nilai 0 dan 1.

Fungsi aktivasi yang tepat pada sebuah aplikasi berbasis jaringan saraf tiruan akan sangat mempengaruhi performnya baik dari segi kecepatan pemrosesan data maupun dari segi tingkat akurasi yang diperoleh. Hal inilah yang menjadikan sebuah alasan perlu melakukan analisa hasil dari penggunaan fungsi aktivasi sigmoid bipolar dengan sigmoid biner pada algoritma *Backpropagation* untuk pengenalan Aksara Batak.

Berdasarkan latar belakang diatas, penulis memaparkan proses pengenalan dokumen aksara batak dengan teknik jaringan syaraf tiruan selanjutnya penulis akan memberi judul penelitian ini dengan “Analisis Perbandingan Fungsi Aktivasi Sigmoid Bipolar Dengan Sigmoid Biner Dalam Pengenalan Aksara Batak Toba Dengan Algoritma *Backpropagation*”.

1.2 Rumusan Masalah

Bagaimana menganalisis hasil perbandingan tingkat akurasi Sigmoid Bipolar dengan Sigmoid Biner pada pengenalan Aksara Batak Toba.

1.3 Tujuan Penelitian

Adapun tujuan penelitian ini yaitu untuk menganalisis hasil perbandingan tingkat akurasi antara fungsi aktivasi Sigmoid Bipolar dan Sigmoid Biner pada algoritma *Backpropagation* untuk pengenalan Aksara Batak Toba.

1.4 Manfaat Penelitian

Adapun manfaat dari sebuah penelitian ini adalah:

1. Mengetahui perbandingan nilai akurasi pada penerapan fungsi aktivasi Sigmoid Bipolar dan Biner pada Algoritma *Backpropagation*.
2. Membantu dan mempercepat pengguna dalam mengenalkan Aksara Batak Toba.
3. Menjadi referensi selanjutnya dalam pengembangan dibidang pengenalan pola.
4. Diperolehnya sebuah perangkat lunak yang dapat melakukan pengenalan aksara Batak Toba.

1.5 Batasan Masalah

Berdasarkan perumusan masalah diatas, maka didapat batasan masalah pada penelitian ini sebagai berikut:

1. Algoritma yang digunakan adalah *Backpropagation* dengan fungsi aktivasi Sigmoid Bipolar dan Biner.
2. File citra aksara Batak Toba yang di *input* adalah berupa dokumen aksara hasil scan digital yang berformat .jpg.
3. Dimensi dokumen aksara adalah sebesar 24 x 24 piksel.
4. Hasil pengenalan berupa tampilan citra aksara.
5. Bahasa pemrograman menggunakan Matlab versi 2015a.
6. Jumlah aksara yang dilatih oleh jaringan adalah 19 jenis dan 190 file citra dimana masing-masing aksara ada 10 file.

1.6 Sistematika Dalam Penulisan

Adapun susunan penulisan dalam sistematika pada skripsi ini terdapat lima bagian terpenting yaitu:

Bab I : Pendahuluan

Pada bagian pendahuluan ini berisi sebuah latar belakang penelitian, tujuan penelitian, manfaat penelitian, rumusan masalah, batasan masalah, hipotesis dan sistematika penulisan.

Bab II : Landasan Teori

Pada bagian ini berisikan konsep-konsep yang dibutuhkan agar memahami apa saja permasalahan yang akan dibahas pada penelitian ini. Pada teori ini dibahas yang berhubungan tentang aksara batak toba, gambar, *grayscale*, ekstraksi citra, *thresholding*, Algoritma *Backpropagation* serta fungsi aktivasi Sigmoid biner dan Sigmoid Bipolar.

Bab III : Analisis dan Perancangan

Pada bagian ini terdapat suatu analisis arsitektur untuk langkah setiap *pre-processing image*, biner, fitur, ekstraksi ciri, *flowchart*, serta algoritma *Backpropagation* untuk mengenalkan aksara Batak Toba.

Bab IV : Implementasi dan Pengujian

Untuk tahap ini dipaparkan pembahasan suatu implementasi yang dapat dibangun untuk pengujian dari data sampel yang dibuat.

Bab V : Kesimpulan Dan Saran

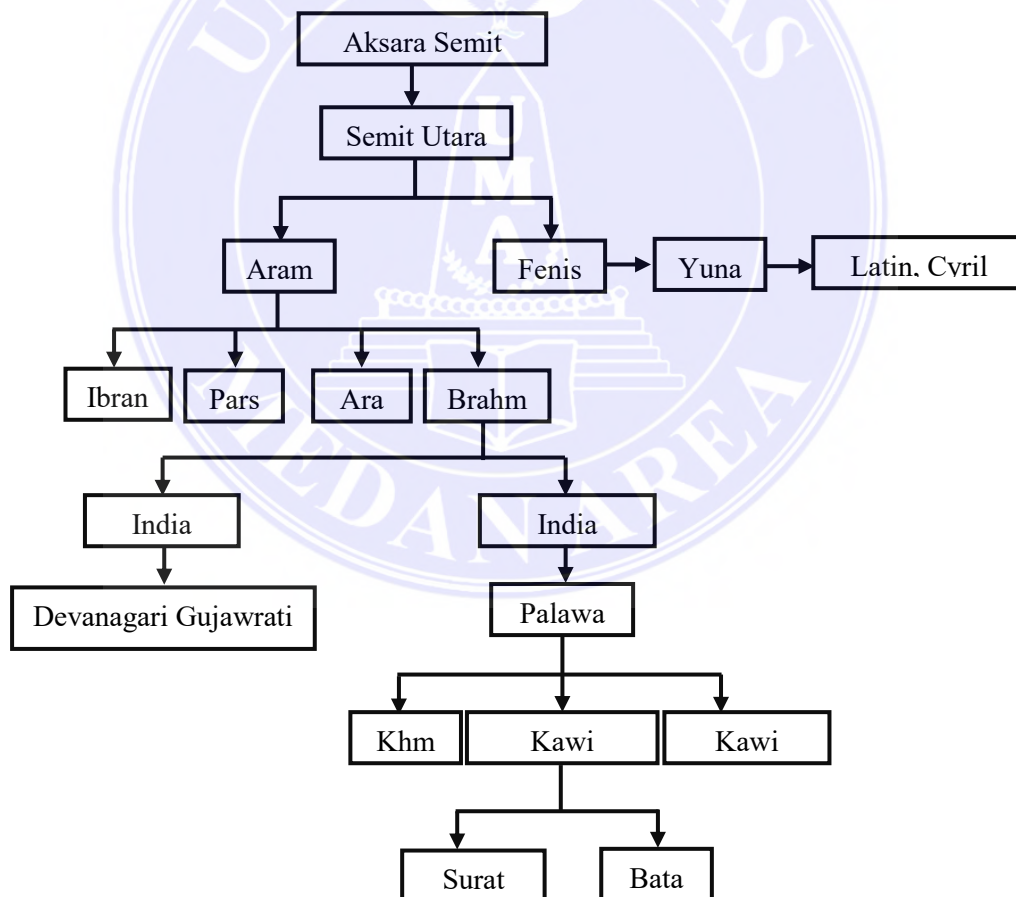
Untuk tahap ini diambil beberapa kesimpulan dan saran yang diajukan dalam pengembangan untuk penelitian yang akan di bahas selanjutnya.

BAB II

TINJAUAN PUSTAKA

2.1 Aksara Batak

Ilmu Paleografi adalah sebuah ilmu pengetahuan tentang tulisan-tulisan kuno, para ahli umumnya mengatakan bahwa surat Batak ialah salah satu turunan dari aksara Brahmi india yang dimana melalui perantara aksara Kawi, surat batak secara tradisional dapat ditulis di sejumlah media, diantaranya yang dibuat dari bambu, tulang dan kulit kayu. Berikut ini terdapat susunan silsilah sebuah aksara yang sebagian besar yang terdapat di Afrika, Eropa, yang berasal dari atau sumber yaitu Semit Kuno yang menjadi nenek moyang tulisan Asia. (Latin, Yunani).



Gambar 2.1 Silsilah Aksara Batak

Sumber (Kozok, 2009)

Pada Gambar 2.1 diatas terlihat dimana secara garis besar posisi aksara Batak pada sebuah silsilah tulisan sedunia (Kozok, 2009). Adapun aksara Nusantara asli dapat atas lima kelompok besar yaitu:

1. Aksara Hanacaraka yang ada pada suku Sunda, Jawa dan Bali.
2. Aksara atau surat Ulu dijumpai pada daerah Kerinci, Rejang, Lampung, Lembak, Pasemah dan Serawi.
3. Aksara atau surat Batak terdapat di daerah Toba, Simalungun, Angkola-Mandailing serta Pakpak Dairi.
4. Aksara Sulawesi yang terdapat pada daerah Bugis, Makassar dan Bima.
5. Aksara Filipina yang terdapat pada daerah Bisaya, Tagalog, Tagbanwa serta Mangyan.

2.1.1 Ibu Surat

Ibu surat pada dasarnya berakiran dengan /a/ (selain alfabet dari i dan u) dapat dilihat pada Tabel 2.1.

Tabel 2.1 Huruf-Huruf Ina Ni Surat

Alfabet Latin	Surat Batak					Alfabet Latin	Surat Batak				
	Karo	Toba	Dairi	Simalungun/Timur	Mandailing		Karo	Toba	Dairi	Simalungun/Timur	Mandailing
a	ᐃ	ᐃ	ᐃ	ᐃ	ᐃ	ta	ᐃ	ᐃ	ᐃ	ᐃ	ᐃ
ha	ᐃ	ᐃ	ᐃ	ᐃ	ᐃ	sa	ᐃ	ᐃ	ᐃ	ᐃ	ᐃ
ka	ᐃ	ᐃ	ᐃ	ᐃ	ᐃ	ya	ᐃ	ᐃ	ᐃ	ᐃ	ᐃ
ba	ᐃ	ᐃ	ᐃ	ᐃ	ᐃ	nga	ᐃ	ᐃ	ᐃ	ᐃ	ᐃ
pa	ᐃ	ᐃ	ᐃ	ᐃ	ᐃ	la	ᐃ	ᐃ	ᐃ	ᐃ	ᐃ
na	ᐃ	ᐃ	ᐃ	ᐃ	ᐃ	nya	ᐃ	ᐃ	ᐃ	ᐃ	ᐃ
wa	ᐃ	ᐃ	ᐃ	ᐃ	ᐃ	ca	ᐃ	ᐃ	ᐃ	ᐃ	ᐃ
ga	ᐃ	ᐃ	ᐃ	ᐃ	ᐃ	nda	ᐃ	ᐃ	ᐃ	ᐃ	ᐃ
ja	ᐃ	ᐃ	ᐃ	ᐃ	ᐃ	mba	ᐃ	ᐃ	ᐃ	ᐃ	ᐃ
da	ᐃ	ᐃ	ᐃ	ᐃ	ᐃ	i	ᐃ	ᐃ	ᐃ	ᐃ	ᐃ
ra	ᐃ	ᐃ	ᐃ	ᐃ	ᐃ	u	ᐃ	ᐃ	ᐃ	ᐃ	ᐃ
ma	ᐃ	ᐃ	ᐃ	ᐃ	ᐃ						

Sumber (Kozok, 2009)

Jenis aksara Nda dan Mba hanya ditemukan dalam bahasa Batak Toba yang berjenis konsonan rangkap, Banyak Aksara yang dipakai dalam bahasa Mandailing dan Dimasukkan pula pada huruf aksara Batak Toba walaupun tidak digunakan. Adapun aksara Ca hanya terdapat dalam bahasa Toba dan dalam bahasa Angkola Mandailing aksara Ca ditulis menggunakan alfabet Sa dimana masih ada satu perindikasi diakritik

yang bernama tompi pada atasnya. Komponen fonetis pada aksara Batak berupa Anak ni surat kebetulan disisipkan pada ina ni surat (perindikasi diakritik) berfungsi menjadi pengubah cara pengucapan berdasarkan ina ni surat misalnya yang bisa ditinjau dalam Tabel berikut.

Tabel 2.2 Anak Ni Surat

Ina ni surat Induk huruf berbunyi [a]	Induk huruf (ina ni surat) berbunyi a,i,u,e,o, dan h				
	Haluaan berbunyi [i]	Haborotan berbunyi [u]	Singkora berbunyi [e]	Hatadingan berbunyi [o]	Sikorjan berbunyi [h]
ang	ing	ung	eng	ong	hah
hang	hing	hung	heng	hong	mah
gang	ging	gung	geng	gong	nah
nang	ning	nung	neng	nong	rah
mang	ming	mung	meng	mong	tah
rang	ring	rung	reng	rong	lah
tang	ting	tung	teng	tong	pah
lang	ling	lung	leng	long	sah
sang	sing	sung	seng	song	dah
dang	ding	dung	deng	dong	gah
pang	ping	pung	peng	pong	jah
bang	bing	bung	beng	bong	bah
ngang	nging	ngung	ngeng	ngong	wah
jang	jing	jung	jeng	jong	ngah
yang	ying	yung	yeng	yong	nyah
wang	wing	wung	weng	wong	yah
nyang	nying	nyung	nyeng	nyong	

Sumber (Kozok, 2009)

1. Suara [e] (*hatadingan*)
2. Suara [ŋ] (*haminsaran*)
3. Suara [u] (*haborotan*)
4. Suara [i] (*haluaan*)
5. Suara i [o] (*sihora*)
6. *Pangolat* (tanda untuk menghilangkann suara [a] pada ina ni surat)

Nama dan petunjuk diakritik diatas hanya pada bahasa Batak Toba dan pada bahasa Batak lainnya banyak sekali variasi nama diantaranya terdapat alfabet ina ni surat, contohnya Pangolet pada bahasa Toba disebut “penengen”. Demikian juga halnya Ina ni surat Aksara Batak Toba (Kozok, 2009) menggunakan ina nisurat, dimana anak surat merupakan alfabet pada aksara Batak yang tersusun dari berdasarkan tradisinya masing-masing, yaitu: [e], [i], [o], [u], [ŋ], [x].

Dalam pertanda diakritik itu sendiri, mempunyai bentuk varian antara suatu wilayah yang memakai karakter yang sama, contohnya penggunaan pertanda diakritik memakai huruf Ka dan varian pertanda pangolat (Kozok, 2009).

2.2 Huruf Batak

Dimana jenis huruf yang terdapat pada *computer Macintosh* dan *windows*. Buku yang berjudul tentang huruf batak, dalam silsilah perkembangan tulisan batak ini ditulis oleh Uli Kozok yang diterbitkan Kepustakaan Populer Gramedia Tahun 2009. Dalam bahasa batak sering disebut bahwa aksara batak itu berjumlah sampula sia yang dimana dalam arti aksara batak berjumlah 19. Peneliti memperoleh persyaratan pada pengetikan *font* aksara batak pada personal komputer menurut seluruh etnis batak (Kozok, 2009). Yang dimana hurufnya memiliki beberapa urutan, diantaranya sebagai berikut:

2.3 Aplikasi *Optical Character Recognition* (OCR)

Aplikasi OCR yaitu salah satu program komputer dimana berguna dalam pembacaan huruf dapat mengenali dengan baik huruf yang terdapat pada sebuah pencetak printer yakni yang berdasarkan tulisan tangan. Aplikasi ini memiliki 2 metode yaitu *offline* dan *online*. Metode *offline* bekerja dengan mengenali tulisan berupa inputan file citra digital hasil scan ataupun akuisisi citra. *Input* metode ini dimana gambar berupa teks terlebih dahulu di-scan oleh *scanner* kemudian dikenali oleh aplikasi. Metode *online* adalah bekerja dalam mengenali tulisan dengan tulisan tangan langsung di-*input* berupa coretan tulisan secara *real time* di media penulisan digital. Aplikasi OCR bekerja dalam menerjemahkan citra huruf kedalam format *text* dengan mencocokkan pola karakternya per baris dengan pola dalam data *base* pada aplikasi dimana hasilnya ialah berupa teks (Prasetaningtyas, 2016).

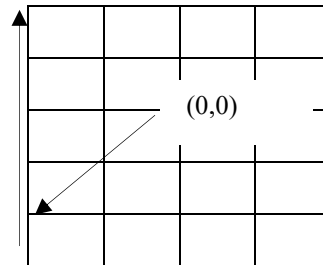
2.4 Citra

Citra (*image*) merupakan tampilan dari beberapa objek yang masih ada pada citra terhadap bidang 2 dimensi yang dimana biasanya dapat ditulis pada koordinat kartesius xy dimana pada setiap koordinat mewakili satu frekuensi terkecil dari objek itu (Putri, 2017), citra bisa berwujud gambar dua dimensi seperti lukisan dan foto.

2.5 Gambaran Sebuah Citra

Citra digital diperoleh dengan cara merekam kekuatan sinar yang dipantulkan oleh sebuah objek tertentu dimana citra juga diartikan sebagai gambaran objek pada bidang dwi matra. (Gonzalez, 2003).

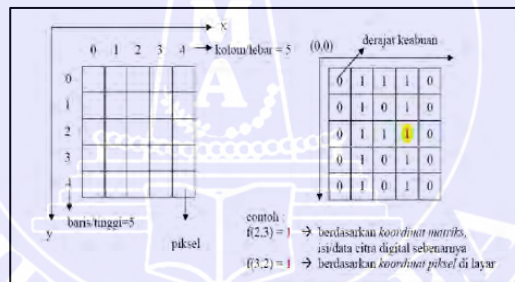
Piksel merupakan bayangan dalam bidang koordinat $(i,j)=0$ yang artinya nilai i dan j merupakan 0 dimana piksel berada pada pojok kiri atas citra. Berikut perbedaan yang memakai f =grafik pada nilai matematika dan koordinat yang berada pada sebuah citra.



Gambar 2.2 Koordinat Grafik Matematika

Sumber (Gonzalez, 2003)

Sebuah koordinat citra titik asal akan dilalui dari sudut kiri atas bidang pada grafik, dapat dilihat seperti gambar berikut:



Gambar 2.3 Koordinat Citra

Sumber (Gonzalez, 2003)

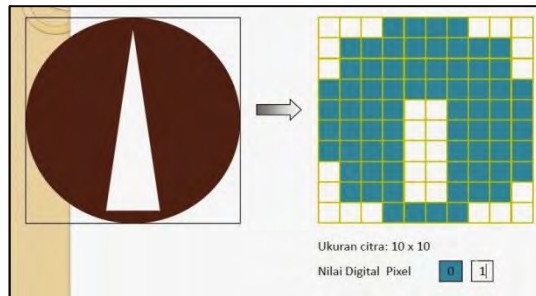
2.5.1 Gambaran Analog Sebuah Citra

Analog citra ini yaitu frekuensi yang bersifat konstan misalnya gambar dalam alat televisi analog, cetakan gambar pada kertas foto berupa gambar benda atau pemandangan, cetakan hasil lukisan pada kertas, sinar *rongen X-Ray* atau *output CT scan* dimana jenis *image* ini tidak bisa diolah komputer digital (Gonzalez, 2003).

2.5.2 Citra Digital

Tipe *image* ini merupakan pembentukan frekuensi contoh sinyal elektromagnet analog dan dimensi mode ganda kontinu yang diperoleh dalam proses sampling menurut gambar analog. Untuk memilih nilai warna piksel dalam gambar analog dalam proses

sampling, maka dilakukan menggunakan cara pembulatan nilai berdasarkan gambaran analognya dimana proses sampling bisa kita lihat yakni sebagai berikut:



Gambar 2.4 Proses Sampling Citra Analog

Sumber (Gonzalez, 2003)

Penjelasan dari sebuah citra digital berupa piksel-piksel yang menyatakan banyaknya titik dalam satu citra, sementara kuantisasi memperlihatkan taraf nilai pada piksel yang dimana (0-256).

2.5.3 Jenis-jenis Citra Digital

Ada beberapa cara yang dapat digunakan untuk menyimpan digital didalam memori. Cara penyimpanan menentukan citra digital yang terbentuk, ada beberapa jenis-jenis citra digital yang sangat sering digunakan yaitu citra warna, citra *grayscale* dan citra biner. (Sutoyo, 2010):

1. Citra Biner (Monokrom)

Citra biner (monokrom) hanya memiliki 2 warna yaitu hitam dan putih saja. Yang dimana dibutuhkan 1 bit di memori untuk menyimpan kedua warna tersebut.

Gradasi warna :



Bit 0 = warna hitam

Bit 1 = warna putih

Pada sebuah citra biner yang pikselnya terdiri atas dua intensitas warna yang dimana hitam dengan nilai 0 sedangkan putih dengan nilai 1. Jika sebuah *threshold* yang dipakai dengan nilai 128 dimana piksel tersebut kurang dari 128 maka perubahan terjadi yaitu dengan memberi nilai 0. Jika lebih dari satu atau nilainya sama dengan 128 maka perubahan akan dilakukan dengan memberi nilai 1. Citra biner ini diperoleh dari citra RGB (warna) dengan menggunakan rumus:

$$f(x) = (R+G+B)/3 \dots\dots\dots (2.1)$$

Nilai $f(x)$ dilakukan *threshold* 128, jika nilai $f(x) <$ dari 128 dan $f(x) = 0$ dan jika nilai $f(x) >$ dari 128, dan $f(x) = 1$.

$$f(x) = \begin{cases} 0 & \text{if } x < T \\ 1 & \text{if } x > T \end{cases} \dots\dots\dots (2.2)$$

2. Citra Grayscale (Skala Keabuan)

Sebuah warna sangat mempengaruhi pada suatu banyaknya jumlah bit yang dimana telah tersedia di dalam memori agar dapat menampung kebutuhan warna tersebut. Citra 2 sangat mewakili 4 warna sekaligus:



3 bit citra mewakili 8 warna pada suatu gradasi warna:



Jika jumlah bit suatu warna yang telah tersedia di memori maka gradasi warna citra akan semakin halus tampilannya.

3. Citra Warna (True Color)

Piksel yang terdapat di suatu citra warna sangat mewakili warna yang dimana kombinasi antara 3 warna tersebut yaitu (RGB= Red Green Blue). Untuk warna dasar yaitu dimana penyimpanannya 8 bit = 1 byte yang berarti terdapat pada warna yang mempunyai gradasi warna sebanyak 255. Dimana pada piksel mempunyai kombinasi antara warna yaitu: sebanyak $2^8 \cdot 2^8 \cdot 2^8 = 2^{24} = 16$ juta warna lebih. Itulah sebabnya mengapa format ini dinamakan *true color* karena mempunyai jumlah warna yang sangat cukup besar.

Penyimpanan citra *true color* di dalam memori sangat berbeda jauh dengan citra *grayscale*. Dimana setiap piksel dari citra *grayscale* 256 gradasi warna diwakili oleh 1 byte. Sedangkan 1 piksel citra *true color* diwakili oleh 3 byte, dimana masing- masing byte tersebut mampu merepresentasikan warna merah (*red*), hijau (*green*) dan biru (*blue*).



Gambar 2.5 Contoh Citra Warna
Sumber (Gonzalez, 2003)

2.5.4 Format Citra Digital

Format dari suatu citra memiliki karakteristik masing-masing, format dari suatu citra yang baik juga sangat tergantung pada setiap citranya. Dan setiap format data citra memiliki kekurangan dan kelebihan tersendiri pada saat penyimpanan suatu citra. Oleh karena itu, untuk penyimpanan suatu format data citra harus diperhatikan citra apa dan format file yang sesuai dan format ini terdiri dari beberapa jenis yang setiap jenisnya ditentukan dengan jumlah bit. (Gonzalez, 2003).

2.5.5 Format Data Bitmap

Bitmap adalah sebuah bahan matriks yang menjelaskan individualitas dari titik-titik dalam sebuah gambar yang biasanya disebut dengan piksel. Bitmap digunakan dari berbasis vektor yang memiliki resolusi dan tidak akan rusak pada saat gambar akan diperbesar ataupun diaplikasikan ke media cetak, setiap piksel yang membentuk bitmap memiliki warna tertentu yang memiliki nilai. Dan jika kita memperbesar gambar, kita dapat melihat dengan jelas setiap piksel tersebut, semakin tinggi jumlah piksel per gambar maka kualitas gambarnya semakin tinggi dan format gambar bitmap terdiri dari JPG, JPEG, PNG, BMP, Dan TIFF (Sutoyo, 2010).

Tabel 2.3 Jumlah Bit Per Piksel

No	Banyak Bit Per piksel	Banyak Warna Maksimum
1	1	2
2	4	16
3	8	256
4	16	65536
5	24	16777216

2.5.6 Format JPEG

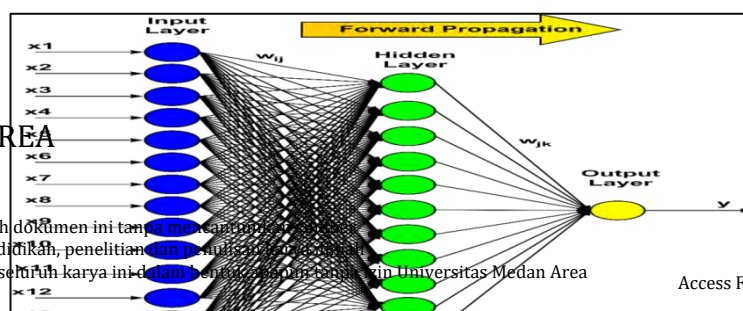
Citra dengan format JPEG dirancang untuk pemampatan citra, baik citra keabuan maupun citra warna (*true color*). Citra menggunakan format JPEG ini tidak hanya menangani data pada bentuk *image* saja, namun juga bisa berkaitan dengan *image* yang diklaim menggunakan *motion picture* (MPEG). Format JPEG yaitu format sebuah file yang terkompresi, hal ini ini sangat berguna untuk menyimpan foto pada ukuran lebih kecil dari BMP. JPEG adalah pilihan umum untuk digunakan di web karena di kompresi, namun JPEG tidak disarankan sebagai gambar arsip kemudian format JPEG tidak cocok digunakan untuk penyimpanan gambar pajang (Sutoyo, 2010).

2.6 Algoritma Backpropagation

Pada algoritma *Backpropagation* terdapat dua proses yaitu proses pelatihan dan pengenalannya dimana pada proses pelatihan berfungsi untuk memperoleh nilai bobot akhir dengan parameter sebagai berikut:

1. Banyak neuron jaringan untuk setiap aksara yaitu representasi dari ekstraksi ciri yang dipilih.
2. Banyak neuron pada setiap *hidden layer*.
3. Banyak lapisan tersembunyi atau *hidden layer*.
4. Banyak proses *epoch* yaitu 150, 300, 450, 600, 750, 900, 1050 dan 1200.
5. *Learning rate* 0,2.

Adapun banyaknya neuron pada hasil *layer* ialah hasil jaringan sebesar jenis aksara yang bisa dikenalkan. Arsitektur jaringan Algoritma *Backpropagation* yang digunakan bisa dilihat pada berikut:



Gambar 2.6 Arsitektur Jaringan

Sumber (Adinugroho, 2017)

Arsitektur pada gambar diatas, nilai bobot w_{ji} ialah bobot garis masukkan *layer* dari suatu bobot garis yang mampu menghubungkan bias pada *input layer* ke hidden *layer* z_j). Algoritma ini ialah termasuk prosedur pemecahan pembelajaran terbimbing (*supervised learning*) dimana nilai input dan hasil dipakai pada pemrosesan jaringan sehingga diperoleh nilai bobot yang diperlukan (Syafria, 2016):

2.6.1 Proses Pada Propagasi Maju

Proses propagasi maju ini, terdapat sinyal masukan pada (x_1) akan di propagasikan terhadap *layer* tersembunyi yang dimana menggunakan fungsi aktivasi yang ditentukan untuk keluaran unit tersembunyi Z_1 akan di propagasi maju terhadap *layer* tersembunyi lalu menggunakan fungsi aktivasi yang akan ditentukan dan dimana akan menghasilkan suatu keluaran yang terdapat pada jaringan y_k dan kemudian jaringan y_k tersebut akan dibandingkan terhadap target yang telah ditentukan (Adinugroho, 2017).

2.6.2 Tahap Propagasi Balik

Berdasarkan kesalahan $tk-y_k$ dihitung faktor $k=1$ dalam menstribusikan kesalahan yang terdapat di unit y untuk seluruh unit tersembunyi yang dimana terhubung dengan y . Dimana k juga akan dipakai dalam pengubahan bobot garis yang terhubung terhadap unit keluaran lalu hitung faktor j disetiap *layer* tersembunyi.

2.6.3 Tahap Modifikasi Bobot

Dimana *Weighting Modified* ini bobot seluruh jalur neuron di-*update* sekaligus dimana nilai bobot yang diubah akan di-*update* pada suatu garisan neuronnya dihitung dari nilai faktor δ neuron diatas lapisannya. Proses memodif bobot dalam jalur yang terhubung pada lapisan *output*-nya dari lapisan dalam unit keluaran (*output*) telah

berakhir dan selanjutnya kerjakan pelaksanaan sampai syarat terhenti tercapai yaitu jumlah maksimum *epoch* atau tidak terdeteksi (Isnanto, 2017).

2.6.4 Proses Pembelajaran

Seperti jaringan saraf lainnya, pada jaringan *feed-forward* (propagasi maju) *training* dilakukan agar tujuan memilih nilai bobot akhir menjadi hasilnya di akhir *training*, bobot optimal akan diperoleh. Rata-rata kesalahan yang akan dijadikan sebagai dasar perhitungan fungsi aktivasi, yang dimana pelatihan terhadap umpan maju *gradient* fungsi akitvasi tersebut dalam menentukan sistem pengaturan terhadap bobot (Amrutha, 2018).

Tahap-tahap yang akan dibahas pada sebuah prosedur pelatihan yaitu:

- Tahap 0 : Percobaan bobot yang berhubungan pada neuron yaitu dimana akan menggunakan bilangan secara acak paling kecil contohnya (-0.4 sampai +0.4, -0.5 sampai +0.5 dan -1 sampai +1). (Puspitaningrum, 2006).
- Tahap 1 : Buatlah langkah dua hingga langkah 9 selama syarat berhenti.
- Tahap 2 : Kerjakanlah tahap 3 dan tahap 8 untuk mendapatkan nilai pelatihannya.

Propagasi Maju

Tahap 3 : Seluruh unit yang menjadi masukan dari ($x_i, i = 1, \dots, n$) mendapat frekuensi masukan x_i , lalu akan menyebarkannya ke semua unit dalam lapisan tersembunyi.

Tahap 4 : Seluruh unit tersembunyi ($x_i, I = 1, \dots, p$) yaitu akan dijumlahkan bobot frekuensi:

$$z_{in_j} = v o_j + \sum_{i=1}^n x_i v_{ij} \dots \dots \dots (2.5)$$

Tahap 5 : Untuk unit keluaran ($y_k, k = 1, \dots, m$) jumlahkan semua bobot frekuensi masukannya:

$$y_{in_k} = w o_k + \sum_{j=1}^p z_j w_{jk} \dots \dots \dots (2.6)$$

Propagasi Balik

Tahap 6 : Setiap unit keluaran ($y_k, k = 1, \dots, m$) mendapat pola yang saling berafiliasi dalam masukan pola pelatihan, lalu hitunglah kesalahan informasinya,

$$\delta_k = (t_k - y_k) f'(y - in_k) \dots \dots \dots (2.7)$$

menghitung koreksi biasnya (dipakai untuk melihat w_{jk} nantinya),

$$\Delta w_{jk} = \alpha \delta_k z_j \dots \dots \dots (2.8)$$

menghitung koreksi biasnya (digunakan untuk mempengaruhi w_{ok} nantinya)

$$\Delta w_{ok} = \alpha \delta_k \dots \dots \dots (2.9)$$

dan kirimkan δ_k ke unit-unit pada lapisan dibawahnya,

Tahap 7 : Setiap unit lapisan tersembunyi ($z_j, j = 1, \dots, p$) jumlah hasil untuk perubahan dari masukannya.

$$\delta - in_j = \sum_{k=1}^m \delta_k w_{jk} \dots \dots \dots (2.10)$$

kalikan dengan turunan kedua setiap fungsi aktivasinya untuk menghitung informasi kesalahannya,

$$\delta_j = \delta - in_j f'(z - in_j) \dots \dots \dots (2.11)$$

$$\Delta v_{ij} = \alpha \delta_j x_i \dots \dots \dots (2.12)$$

hitung koreksi bias

$$\Delta v_{oj} = \alpha \delta_j \dots \dots \dots (2.13)$$

Tahap 8 : Kemudian *update* setiap bobot dan bias dalam interaksi antar lapisan

$$w_{ij}(\text{baru}) = w_{ij}(\text{lama}) + \Delta w_{ij} \dots \dots \dots (2.15)$$

Tahap 9 : Melakukan percobaan pada kondisi saat berhenti

Pada umumnya metode *Backpropagation* terdiri atas dua fase yaitu propagasi maju dan propagasi mundur. Dimana selama propagasi maju akan memetakan nilai pada masukan agar dimana mendapatkan nilai keluaran yang diharapkan sehingga nilai yang terdapat pada sigmoid biner yaitu 0 sampai 1 besarnya *output* adalah:

$$o_j = \frac{1}{1 + e^{-a \text{net}_j}} \dots \dots \dots (2.16)$$

di mana,

$$a_{net,j} = (\sum_{i=1}^n w_{ij}o_i) + \theta_j \dots\dots\dots(2.17)$$

Yaitu:

- o_j : masukan dari j unit
- w_{ij} : bobot yang terhubung antara unit i dan unit j
- $a_{net,j}$: jaringan keluaran pada unit j
- θ_j : bias pada unit j

Pada propagasi balik, setiap pola keluaran merupakan nilai keluaran aktual dibandingkan dengan nilai *output* yang diinginkan dan nilai kesalahan dihitung untuk setiap *output*. Sinyal frekuensi lalu merambat mundur untuk menggunakan lapisan *output* ke setiap unit, pada lapisan transisi yang akan berkontribusi agar *output* dan bobot akan menyesuaikan jumlah iterasi/*epoch* selama proses pelatihan, maka nilai kesalahan diminimalkan selama proses perulangan. Adapun fungsi *error* dalam neuron yang didapatkan dilihat menjadi berikut:

$$E = \frac{1}{2} \sum_{k=1}^n (t_k - o_k)^2 \dots\dots\dots(2.18)$$

- n : angka pada modul *output* didalam lapisan *output*
- t_k : *output* yang dikendaki dari keluaran unit k
- o_k : *output* jaringan dari keluaran unit k

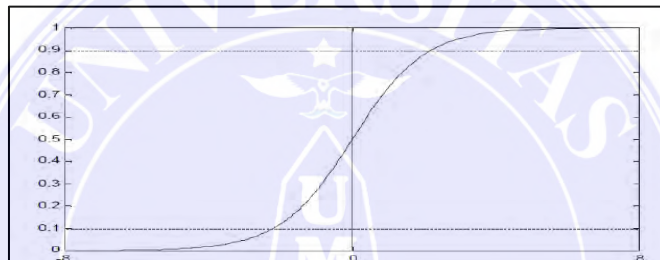
Apabila bobot awal terlalu besar maka masukan terhadap lapisan tersembunyi atau lapisan keluaran akan berada pada daerah dimana fungsi sigmoid tersebut akan sangat kecil. Jika bobot awal terlalu kecil, maka inputan ke setiap lapisan tersembunyi atau lapisan keluaran juga akan sangat kecil, dimana hal ini akan mempengaruhi proses pelatihan tersebut. Biasanya bobot awal akan dilakukan secara acak dimana dengan nilai -0.5 sampai 0.5, jika pelatihan telah selesai maka jaringan dapat digunakan dalam proses pengenalan. Adapun kelebihan dari algoritma *Backpropagaion* ini adalah memiliki beberpa unit yang ada antara satu atau lebih *layer* tersembunyi dibandingkan pada metode perceptron yang terdiri dari *layer* tunggal sedangkan kekurangannya yaitu lama proses pelatihan.

2.7 Fungsi Aktivasi

Fungsi aktivasi ialah sebuah operasi dasar pada neuron buatan yang meliputi penjumlahan bobot sinyal *input* akan mengaplikasikannya pada *output*. Untuk unit *input*.

Fungsi adalah fungsi indentitas. Pada dasarnya fungsi aktivasi yang sama digunakan juga untuk semua neuron pada *layer* tertentu dari jaringan saraf tiruan (Julpan, Nababan, E. B. & Zarlis, M. 2016). Fungsi aktivasi sigmoid biner memiliki nilai terhadap jangkauan 0 sampai 1 seperti yang ditunjukkan pada Gambar 9, dan rumusannya dapat didefenisikan sebagai persamaan berikut :

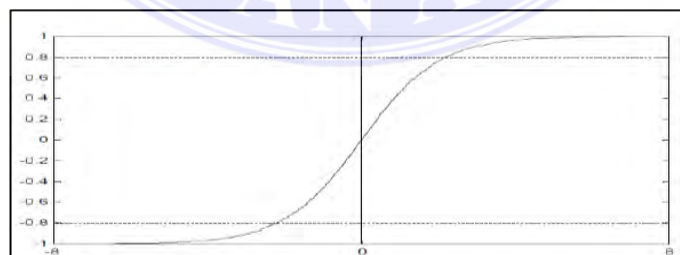
$$Y = f(x) = \frac{1}{(1+e^{-x})} \dots\dots\dots(2.19)$$



Gambar 2.7 Fungsi Aktivasi Sigmoid Biner

Fungsi aktivasi sigmoid bipolar yaitu *output range* antara 1 sampai -1 yang dimana secara umum sama dengan sigmoid biner. Fungsi aktivasi sigmoid Bipolar dapat ditunjukkan pada gambar 10 dan persamaannya didefenisikan sebagai berikut :

$$Y = f(x) = \frac{1-e^{-x}}{(1+e^{-x})} \dots\dots\dots(2.20)$$



Gambar 2.8 Fungsi Aktivasi Sigmoid Bipolar

2.8 Algoritma *Mark Direction* dan *Intency of Character*

Pada penelitian ini pengambilan citra dapat menggunakan dengan algoritma tanda arah (*Mark Direction*) dan intensitas karakter (*intency Of Character*). Algoritma tanda arah yaitu pencerian citra dengan menghitung banyaknya sebuah piksel yang

mempunyai tetangga yang berarah *horizontal*, *vertical*, diagonal ke kiri, dan diagonal ke kanan. Sedangkan pengertian intensitas karakter yaitu pencirian citra dengan cara menghitung berapa banyak jumlah piksel yang bernilai 0 (warna hitam). Jadi, tanda arah yang digunakan untuk menghitung berapa banyak piksel yang dapat memenuhi sebuah *masking horizontal*, *vertical*, diagonal kiri, dan diagonal kanan pada sebuah citra (Nicolaus Euclides Wahyu Nugroho, 2016)

2.9 Metode Normalisasi Data

(Nasution *et al.*, 2019)

Proses normalisasi nilai ekstraksi ciri dilakukan setelah diperoleh ekstraksi ciri citra aksara menggunakan metode *minmax* yang bertujuan agar memperbaharui total nilai ekstraksi citra pengujian dan pelatihan (Nasution *et al.*, 2019). Dalam data ekstraksi ciri seluruh bagian blok piksel dilakukan proses normalisasi supaya nilai ekstraksi ciri pada 0-1 menggunakan rumus:

$$X_n = \frac{(X_i - X_{min})}{(X_{max} - X_{min})} \dots\dots\dots(2.21)$$

X_i = Unit terhadap i

X_{min} = Unit minimum ke i

X_{max} = Unit maksimum ke i

2.10 Penelitian Terkait

Adapun penelitian yang telah ada sebelumnya adalah penelitian Putri (2017) dimana dilakukan pengenalan aksara Batak Karo dimana pada proses pelatihan jaringan (*training*) dengan akurasi 90% dan melakukan pengujian selama 500 kali percobaan.

Pada penelitian Riansyah *et al.* (2017) dilakukan sebuah perulangan gambaran medis yang ruang lingkup dalam penelitian ini terdapat beberapa bagian yaitu penentuan struktur jaringan, meningkatkan secara optimal prosedur pemecahan genetika dan prediksi jaringan. Pada penelitian ini, prosedur pemecahan genetika dipakai agar mengoptimalkan bobot (*weight*) setiap neuron dan bila sudah diperoleh bobot awal (*initial weight*) yang optimal, lalu dilakukanlah *prosessing* dan penetapan angka piksel-piksel baru keluaran (*output*).

Tabel 2.4 Penelitian Terdahulu

No	Nama Peneliti dan Tahun	Metode Yang Digunakan	Pembahasan
1	Putri, A. R, 2017	<i>Backpropagation</i>	Pada penelitian ini dilakukan suatu perulangan sebanyak 500 kali dan hasil akhir yang diperoleh yaitu 90%.
2	Riansyah, R. R. , Nurhasanah, Y. I. & Dewi, I. A. (2017)	<i>Metode Learning Vector Quantization</i>	Pada hasil Akurasi pada percobaan pengenalan adalah sebesar 78,67 %.
3	Faturrahman, I. Arini & Mintarsih, F. (2018)	<i>Metode Sobel dan Backpropagation</i>	Akurasi berhasil 100% dengan akurasi 10000 dengan memakai 28 data <i>training</i> .
4.	Winardi, S. dan Hamzah. (2015)	<i>Backpropagation</i>	Hasil akurasi pengenalan antara 5 % - 90 %.
5	Amalia, N., Hidayat, E. W. dan Aldya, A. P. (2020)	<i>Backpropagation Dan Deteksi Tepi Canny</i>	Data yang tidak terdeteksi yaitu sebanyak 5 kali yang terdeteksi sebanyak 16.

BAB III

ANALISIS DAN PERANCANGAN

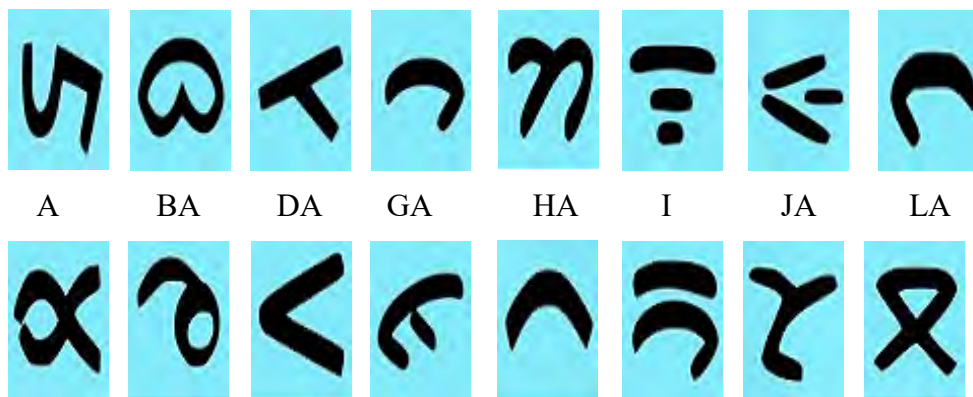
3.1 Analisis

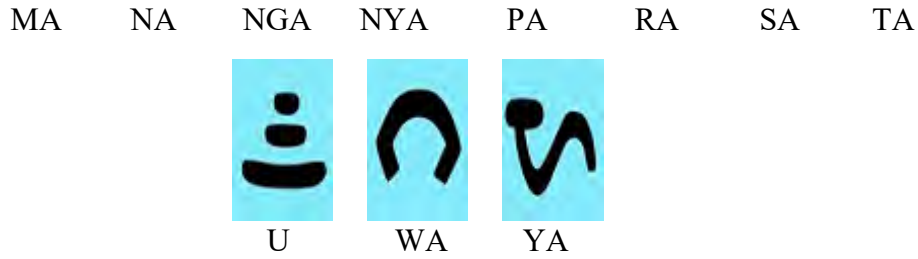
Pada bab ini yaitu membahas tentang bagaimana analisis fungsi aktivasi Sigmoid Biner dan Sigmoid Bipolar untuk proses pengenalan Aksara Batak Toba

Menggunakan Algoritma *Backpropagation*. Pada bab ini yang dibahas adalah tahap analisis dan tahap perancangan suatu sistem, yang dimana akan dilakukan analisa terhadap data yang diperoleh selanjutnya akan dilanjutkan untuk pemrosesan suatu data. Untuk tahap perancangan sistem akan dibahas juga tentang perancangan tampilan antar muka system yang diperuntukkan bagi pengguna.

3.1.1 Teknik Pengumpulan Data

Dalam penelitian ini data yang dipakai yaitu file citra tulisan Batak Toba yang berektensi .jpg. Dokumen aksara Batak Toba ini diperoleh dari Kepala Pusat kebudayaan Batak Medan, yang dimana dokumen semua aksara batak atau yang berhubungan tentang semua silsilah Batak hanya ada satu di Sumatera Utara yang tepatnya pada Universitas Nommensen. Data dibagi menjadi dua bagian yaitu data *training* dan data *testing*, yang dimana jumlah data *training* yaitu 190 aksara dan jumlah data *testing* yaitu 19 aksara. Untuk proses pengambilan data ini yaitu pertama- tama aksara batak di scan dengan menggunakan camera hp OPPO a53 dengan penyimpanan ram 8/128. Setelah seluruh aksara batak di scan satu persatu kemudian hasil scanan aksara Batak tersebut di simpan ke dalam laptop merk Hp dengan penyimpanan RAM 4/64.





Gambar 3.1 Aksara Batak Toba

3.1.2 Menangkap Gambar

Pada tahap ini citra yang dipakai yaitu citra tulisan Batak Toba yang dimana seluruh citra aksara Batak Toba berukuran 24x24 piksel agar *user* lebih mudah mengolah tulisan Batak tersebut. Masukan yang akan di proses adalah citra aksara batak toba yang telah di scan selanjutnya suatu sistem akan memproses citra agar dapat di proses ke tahap berikutnya.

3.1.3 Data Pelatihan

Pada tahap ini data pelatihan yang dipakai berjumlah 19 aksara batak toba, yang dimana aksara batak tersebut disimpan kedalam perhitungan matriks yang ada pada Pemrograman Matlab. Berikut contoh gambar aksara Batak Toba dalam proses pelatihan data.



Gambar 3.2 Data Pelatihan Aksara Batak Toba

3.1.4 Data Pengujian

Data pengujian adalah data yang dipakai agar memproses sistem aksara Batak Toba dimana untuk menggunakan data pengujian ini, maka sistem akan dapat mengenali aksara batak dengan baik yang dimana dapat dilihat pada gambar berikut:

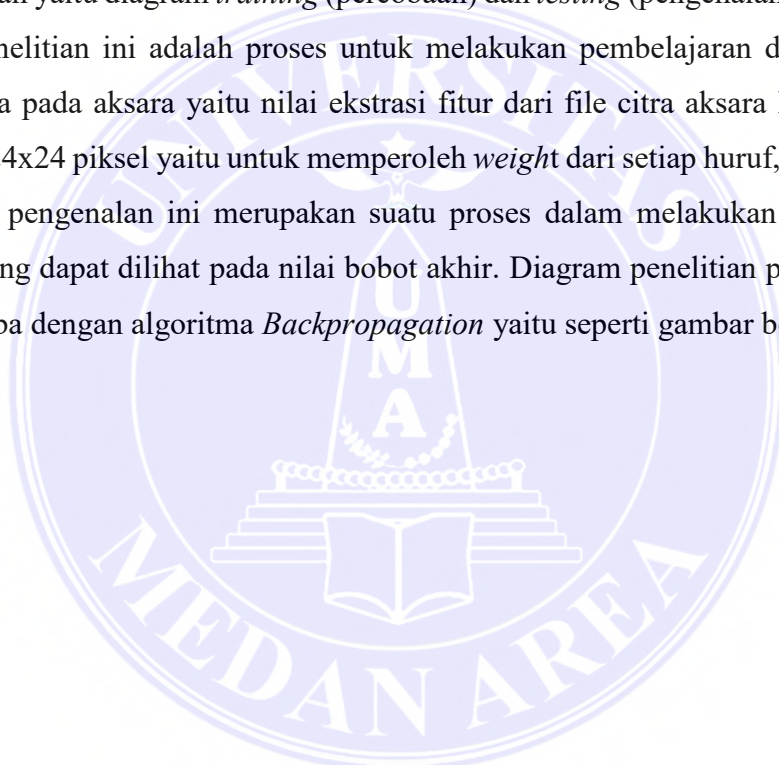


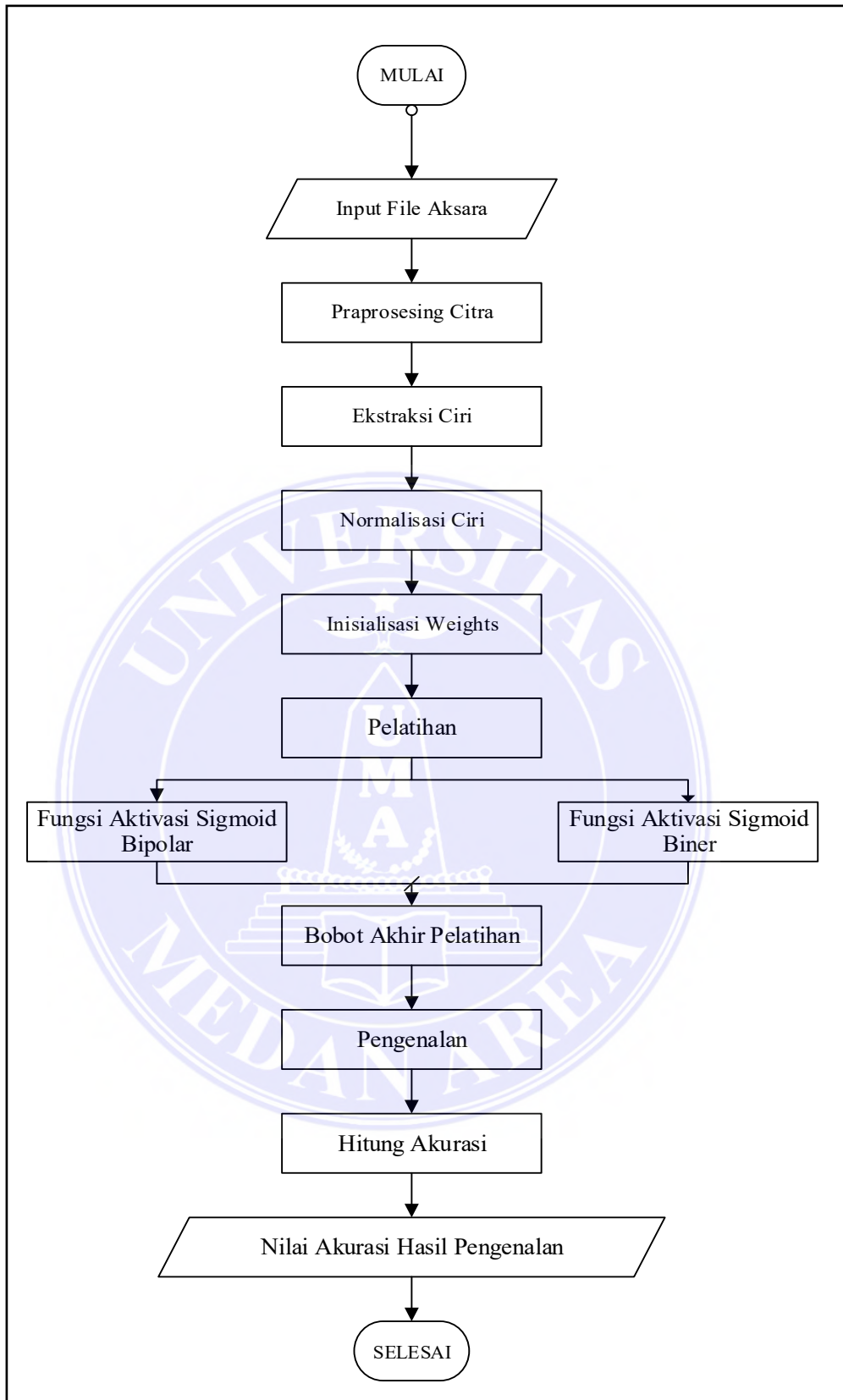


Gambar 3.3 Data Pengujian Aksara Batak Toba

3.1.5 Diagram Penelitian

Pada penelitian pengenalan aksara ini, adapun diagram penelitian ini dibuat dari dua bagian yaitu diagram *training* (percobaan) dan *testing* (pengenalan). Proses *training* pada penelitian ini adalah proses untuk melakukan pembelajaran dalam pengenalan pola-pola pada aksara yaitu nilai ekstrasi fitur dari file citra aksara Batak Toba yang ukuran 24x24 piksel yaitu untuk memperoleh *weight* dari setiap huruf, sedangkan untuk diagram pengenalan ini merupakan suatu proses dalam melakukan penentuan suatu huruf yang dapat dilihat pada nilai bobot akhir. Diagram penelitian pengenalan aksara batak toba dengan algoritma *Backpropagation* yaitu seperti gambar berikut ini.





Gambar 3.4 *Flowchart* Proses Pengenalan Aksara Batak Toba

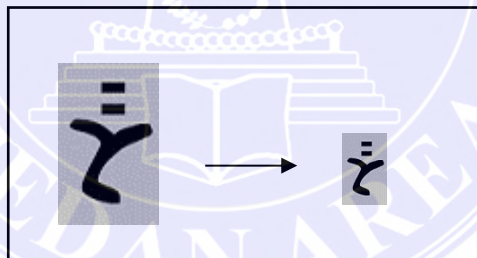
Terdapat pada Gambar 3.4 diatas pertama-tama dilakukan *input* data berupa file citra aksara Batak Toba, selanjutnya file tersebut diproses awal (*praprosesing*) untuk menjadi citra biner. Setelah citra diproses awal, maka proses lanjut adalah dilakukannya pengambilan nilai khas dari citra untuk mendapatkan nilai ciri berupa piksel-piksel setiap citra aksara Batak. Proses lanjutan adalah menormalisasi nilai ekstraksi ciri citra aksara Batak diatas agar berada dalam selang nilai yang seragam, dan selanjutnya dilakukan proses *training* menggunakan fungsi aktivasi agar semua aksara menerima nilai bobot akhir setiap aksaranya. Proses selanjutnya adalah aksara menggunakan cara perhitungan bobot akhir yang optimum tersebut menggunakan nilai ekstraksi karakteristik setiap aksara dalam proses *training* untuk menerima *output* hasil aksara.

3.1.6 Preprosesing

Pada proses *Preprosesing* citra ini adalah langkah awal sebelum melakukan ekstraksi ciri adalah *Preprosesing* citra aksara yaitu:

a. *Resize* citra

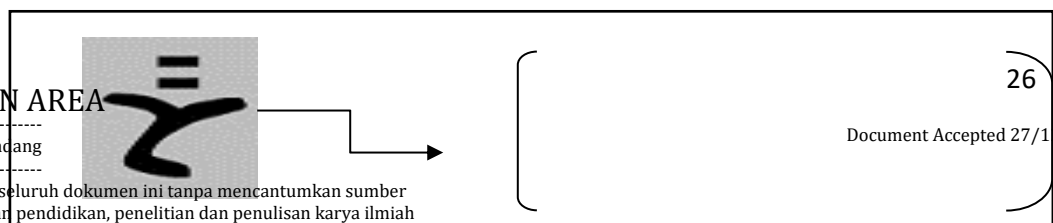
Tahap awal adalah membentuk dimensi citra aksara menjadi ukuran 3 x 3 piksel seperti pada gambar berikut:.



Gambar 3.5 *Resize* Citra Menjadi Ukuran 3 x 3 Piksel

1. Konversi ke *Grayscale*

Tahap selanjutnya merupakan gambaran aksara yang sudah di crop diubah skala warnanya sebagai citra menggunakan skala keabuan. Sebagai nilai RGB citra *input* dalam matriks berdimensi 3 x 3 piksel seperti pada gambar di bawah ini:



200,180,80	214,119,125	200,200,120
195,150,100	210,210,100	215,135,111
175,110,111	185,130,140	215,174,211

Gambar 3.6 Nilai RGB

Selanjutnya nilai komponen warna RGB per piksel pada gambar di atas di konversi ke nilai *grayscale* dengan cara mencari nilai rata-rata per piksel.

$$f_0(x,y) = \left(\frac{f^R(x,y) + f^G(x,y) + f^B(x,y)}{3} \right)$$

dimana f^R yaitu nilai komponen merah

f^G yaitu nilai komponen hijau

f^B yaitu nilai komponen biru

Nilai *grayscale* dihitung dengan menggunakan persamaan (3.4) sebagai berikut:

Piksel 1 = (200,180,80) = (200+180+80)/3 = 153

Piksel 2 = (214,119,125) = (214+119+125)/3 = 153

Piksel 3 = (200,200,120) = (200+200+120)/3 = 173

Piksel 4 = (195,150,100) = (195+150+100)/3 = 148

Piksel 5 = (210,210,100) = (210+210+100)/3 = 173

Piksel 6 = (215,135,111) = (215+135+111)/3 = 154

Piksel 7 = (175,110,111) = (175+110+111)/3 = 132

Piksel 8 = (185,130,140) = (185+130+140)/3 = 152

Piksel 9 = (215,174,211) = (215+174+211)/3 = 200

Pada nilai *grayscale* diatas semua piksel pada citra 9 piksel dimasukkan ke dalam matriks nilai piksel seperti pada Gambar 3.7.

153	153	173
148	173	154
132	152	200

Gambar 3.7 Nilai Piksel *Grayscale*

2. Proses Grayscale ke Biner

Perhitungan nilai biner berdasarkan aksara Batak Toba merupakan pengubahan intensitas nilai piksel citra berdasarkan *grayscale* yaitu 1 dan 0. Adapun nilai 0 warna *black* lalu nilai 1 warna *white* yang ada pada citra. Nilai ambang yang telah digunakan dapat diperoleh dengan rumus:

$$T = \frac{f_{maks} + f_{Min}}{2}$$

Agar mendapatkan nilai *threshold* dalam dilakukan dengan persamaan berikut:

$$T = \frac{200 + 132}{2} = 166$$

Nilai ambang 166 dapat digunakan jika persentase nilai piksel > 166 lebih besar dari 50 %, maka yang menjadi persentase nilai piksel > 166 yaitu < 50 %, maka gunakanlah rumus yaitu:

$$T = \frac{f_{BitMaks}}{2} = 255/2 = 128$$

Dari Gambar 14 di atas maka akan diperoleh persentase nilai piksel diatas 166 yaitu:

$3/9 \times 100 \% = 30 \%$, maka digunakanlah nilai ambang pada persamaan (3.3).

Piksel 1 = 153, Piksel 1 > 128 maka Piksel 1 = 1.

Piksel 2 = 153, Piksel 2 > 128 maka Piksel 2 = 1.

Pikesl 3 = 173, Piksel 3 > 128 maka Piksel 3 = 1.

Piksel 4 = 148, Piksel 4 > 128 maka Pksel 4 = 1.

Piksel 5 = 173, Piksel 5 > 128 maka Piksel 5 = 1.

Piksel 6 = 154 Piksel 6 > 128 maka Piksel 6 = 1.

Piksel 7 = 132, Piksel 7 > 128 maka Piksel 7 = 1.

Piksel 8 = 152, Piksel 8 > 128 maka Piksel 8 = 1.

Piksel 9 = 200, Piksel 9 > 128 maka Piksel 9 = 1.

Lalu nilai *threshold* yang terdapat piksel citra lalu dimasukkan kedalam matriks citra biner terlihat pada gambar 3.8.

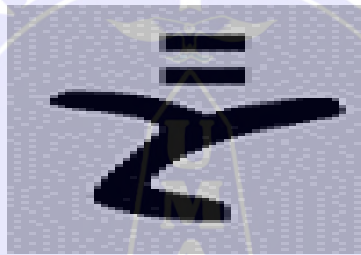
$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Gambar 3.8 Matriks Citra Biner

3.1.7 *Fiture Extraction*

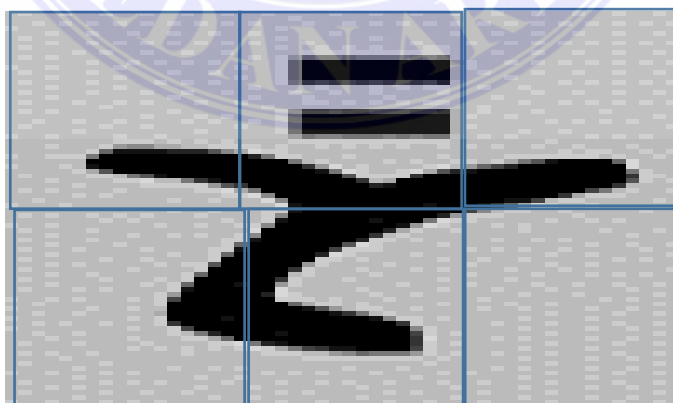
Proses ekstraksi fitur atau ciri dari citra aksara Batak Toba dapat berfungsi untuk menerima sebuah nilai setiap aksara batak yang selanjutnya akan dipakai dalam proses pelatihan. Pelatihan ini akan melakukan ekstraksi citra untuk memakai penyelesaian *Mark Direction* (MD) memakai Intensitas Karakter (IoC) (Nicolaus Euclides Wahyu Nugroho, 2016). Dimana mekanisme pemisahan MD diaktifkan lalu menghitung banyak piksel yang dimana prosesnya dapat kita lihat sebagai berikut:

- a. Pembagian gambar aksara dengan bagian dimensi 3x3 piksel.
- b. Temukan jenis-jenis citra aksara pada setiap bagian dengan menghitung sejumlah piksel dalam berposisi masking *vertical*, *horizontal* dan diagonal ke kanan serta diagonal kekiri.



Gambar 3.9 Citra Biner Pada Aksara

Gambar diatas diterapkan pembagian citra menjadi 6 bagian seperti terlihat dalam gambar dibawah ini:

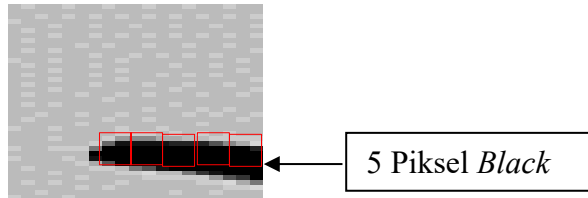


Gambar 3.10 Citra Dengan 6 Bagian

Sesudah citra hasil *thinning* dibagi menjadi 6 bagian, lalu proses yang akan dilakukan selanjutnya yaitu akan mencari nilai seluruh ekstraksi ciri pada citra dapat dibagi yaitu:

1. Eksraksi Ciri 1

Proses satu ini ialah menghitung nilai piksel hitam yang terdapat pada citra yang apabila bernilai 0.



Gambar 3.11 Ciri Piksel 1 *Black* Untuk Bagian 1

Seperti terlihat pada gambar diatas dimana proses pada suatu ciri ekstraksi 1 yaitu menghitung nilai yang terdapat pada piksel hitam yang ada pada bagian 1, apabila ditemukan maka akan bertambah satu *counter* yang dimana jumlah setiap piksel hitam yakni 5 piksel.

2. Ekstraksi Ciri 2

Untuk ciri yang kedua pada diagonal 1 adalah *masking* yang telah memenuhi piksel yang berwarna hitam dengan nilai 0 yang terletak di bagian diagonal kiri yang dapat dilihat pada gambar 19.

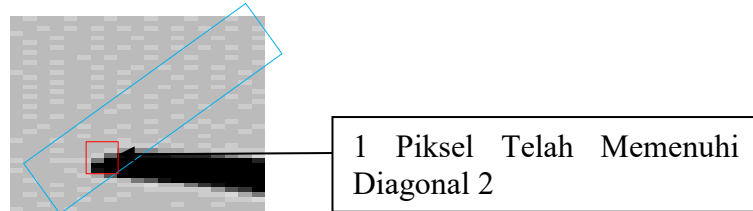


Gambar 3.12 *Masking Left Diagonal* Citra Aksara Bagian 1

Untuk gambar diatas jika suatu ekstraksi ciri dua merupakan ciri piksel berwarna hitam yang kemudian mendapatkan nilai pada piksel hitam yang terdapat pada citra pelatihan bagian satu yang sudah memenuhi piksel *black* yang dimana jumlah piksel hitam sebanyak 3.

3. Ekstraksi Ciri Tiga

Kemudian pada ciri tiga pada diagonal dua adalah dimana sebuah masking yang telah memenuhi jumlah sebuah piksel *black* 1 dapat dilihat pada bagian kanan diagonal seperti pada gambar berikut:.



Gambar 3.13 Masking Diagonal Citra Aksara

Jika pada gambar diatas adalah sebuah proses ekstraksi ciri 2 yaitu ciri pikselnya *vertical* yang kemudian mendapatkan nilai piksel yang berwarna hitam yang dapat dari citra pelatihan pada segment 1. Dan apabila piksel telah memenuhi piksel warna *black* maka angkanya akan bertambah 1 yang dimana jumlah piksel hitam itu berjumlah 1 piksel.

4. Ekstraksi Ciri 4

Pada ciri empat ini adalah posisinya *vertical* dimana masking yang telah memenuhi piksel hitam yang bernilai 0.

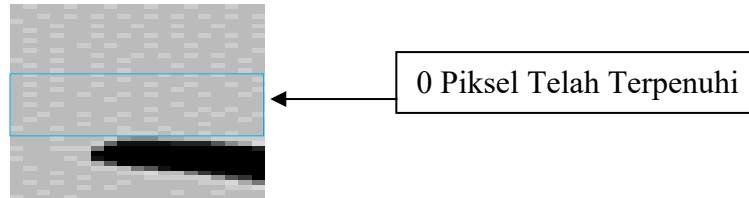


Gambar 3.14 Citra Aksara Dengan *Masking Horizontal*

Pada gambar 3.14 proses yang terdapat pada ekstraksi ciri 2 ini yaitu ciri piksel yang posisinya *vertical* yang kemudian mendapatkan nilai pada piksel hitam yang terdapat pada contoh citra bagian 1 yang dimana telah memenuhi piksel hitam yang bernilai 0 dan jumlah piksel pada warna hitam yaitu 2 piksel.

5. Ekstraksi Ciri 5

Kemudian pada ciri 5 dengan posisi *horizontal* adalah dengan *masking* telah memenuhi piksel *black* yang dimana posisi pikselnya mendatar dapat kita lihat pada gambar 3.15 berikut:



Gambar 3.15 Citra Aksara Dengan *Masking* Bagian 1

Seperti terlihat pada gambar di atas yang dimana proses ekstraksi ciri lima yaitu ciri piksel hitam dengan posisi *horizontal* yang dimana telah mendapatkan nilai piksel hitam dari contoh bagian satu dan telah mendapatkan piksel *black* yang nilainya adalah 0, selanjutnya nilai ekstraksi ciri yang ada pada bagian 1 yaitu $(5+3+1+2+0) = 11$ dan proses dilanjutkan ke 1 dan 6 dengan cara yang serupa kemudian seluruh jumlah data ekstraksi ciri yang terdapat diagonal 1 dan 2 yang posisinya *vertical* dan *horizontal*. Data ekstraksi ciri dapat dilihat pada table berikut.

Tabel 3.1 Ciri Hitam

Blok Piksel	Jumlah Ekstraksi Ciri
Bagian 1	11
Bagian 2	85
Bagian 3	25
Bagian 4	122
Bagian 5	44
Bagian 6	145

Berikut beberapa bagian yang terdapat pada diagonal kiri yaitu pada tabel berikut:

Tabel 3.2 Ekstraksi Ciri Pada Diagonal Kiri

Blok Piksel	Jumlah Ekstraksi Ciri
Bagian 1	55
Bagian 2	102
Bagian 3	1240
Bagian 4	541
Bagian 5	65
Bagian 6	650

Beberapa data ekstraksi ciri yang terdapat pada diagonal kanan seperti Tabel berikut:

Tabel 3.3 Ekstraksi Pada Ciri Diagonal Kanan

Blok Piksel	Jumlah Ekstraksi Ciri
Bagian 1	33
Bagian 2	420
Bagian 3	54
Bagian 4	122
Bagian 5	895
Bagian 6	745

Untuk data ekstraksi ciri dalam kondisi *vertical* yaitu seperti pada Tabel berikut:

Tabel 3.4 Ekstraksi Ciri Untuk *Vertical*

Blok Piksel	Jumlah Ekstraksi Ciri
Bagian 1	126
Bagian 2	451
Bagian 3	444
Bagian 4	452
Bagian 5	85
Bagian 6	41

Selanjutnya pada ekstraksi ciri posisi *horizontal* dapat yaitu seperti tabel berikut:

Tabel 3.5 Ekstraksi Ciri Posisi *Horizontal*

Blok Piksel	Jumlah Ekstraksi Ciri
Bagian 1	232
Bagian 2	222
Bagian 3	452
Bagian 4	542
Bagian 5	224
Bagian 6	66

Dengan semua jumlah yang terdapat pada semua blok piksel untuk seluruh citra aksara dengan jumlah yaitu 5 tabel ciri yaitu dengan jumlah piksel bagian yaitu 5x6 dimana semua berjumlah 30 kemudian akan dilakukan normalisasi.

3.1.8 Normalisasi Data

Proses normalisasi nilai ekstraksi ciri dilakukan setelah diperoleh ekstraksi ciri citra aksara yang bertujuan agar memperbaharui total nilai ekstraksi citra pengujian dan pelatihan. Dalam data ekstraksi ciri seluruh bagian blok piksel dilakukan proses normalisasi supaya nilai ekstraksi ciri pada 0-1 menggunakan rumus:

$$X_n = \frac{(X_i - X_{min})}{(X_{max} - X_{min})}$$

X_i = Unit terhadap i

X_{min} = Unit minimum ke i

X_{max} = Unit maksimum ke i

Sebuah ekstraksi ciri *black* dari citra yang terdapat pada seluruh blok piksel dapat dilihat seperti berikut: Bagian 1 = 11, Bagian 2 = 85, Bagian 3 = 25, Bagian 4 = 122, Bagian 5 = 44, Bagian 6 = 145.

$$\text{Bagian 1 } (x_1) = \frac{11-11}{145-11} = 0.000$$

$$\text{Bagian 2 } (x_2) = \frac{85-11}{145-11} = 0.5522$$

$$\text{Bagian 3 } (x_3) = \frac{25-11}{145-11} = 0.1044$$

$$\text{Bagian 4 } (x_4) = \frac{122-11}{145-11} = 0.8283$$

$$\text{Bagian 5 } (x_5) = \frac{44-11}{145-11} = 0.2462$$

$$\text{Bagian 6 } (x_6) = \frac{145-11}{145-11} = 1$$

Dari perhitungan normalisasi ekstraksi ciri diatas lalu dimasukkan ke sebuah tabel kemudian lakukan penjumlahan data yang sudah ternormalisasi dan lakukan dengan menggunakan cara yang serupa. Berikut hasil data yang telah ternormalisasi dari ekstraksi ciri *black* yaitu pada table berikut:

Tabel 3.6 Ekstraksi Ciri *Black* Normal

Blok Piksel	Jumlah Ekstraksi Ciri	Ekstraksi Ciri Yang Telah Ternormalisasi
Bagian 1	11	0
Bagian 2	85	0.55224
Bagian 3	25	0.10448
Bagian 4	122	0.82836
Bagian 5	44	0.24627
Bagian 6	145	1
Min	11	0
Mak	145	

Berikut hasil yang telah ternormalisasi seluruh data ekstraksi ciri pada bagian diagonal kiri yaitu sebagai berikut:

Tabel 3.7 Ekstraksi Ciri Pada Diagonal Kiri Normal

Blok Piksel	Jumlah Ekstraksi Ciri	Ekstraksi Ciri Yang Telah Ternormalisasi
Bagian 1	55	0
Bagian 2	102	0.03966
Bagian 3	1240	1
Bagian 4	541	0.41013
Bagian 5	65	0.00844
Bagian 6	650	0.50211
Min	55	
Mak	1240	

Berikut data yang telah ternormalisasi pada diagonal yakni sebagai berikut:

Tabel 3.8 Ciri Ekstraksi Bagian Diagonal Kanan

Blok Piksel	Jumlah	Data Yang Telah Ternormalisasi
Bagian 1	33	0
Bagian 2	420	0.44896
Bagian 3	54	0.02436
Bagian 4	122	0.10325
Bagian 5	895	1
Bagian 6	745	0.82599
Min	33	
Mak	895	

Berikut data yang telah ternormalisasi pada *vertical* yakni sebagai berikut:

Tabel 3.9 Ekstraksi Ciri *Vertical* Normal

Blok Piksel	Jumlah	Data Yang Ternormalisasi
Bagian 1	126	0.20681
Bagian 2	451	0.99757
Bagian 3	444	0.98054
Bagian 4	452	1
Bagian 5	85	0.10706
Bagian 6	41	0
Min	41	
Mak	452	

Berikut data yang telah ternormalisasi pada *horizontal* yakni sebagai berikut:

Tabel 3.10 Ekstraksi Ciri *Horizontal* Normal

Blok Piksel	Jumlah Ekstraksi Ciri	Ekstraksi Ciri Yang Telah Ternormalisasi
Bagian 1	232	0.34874
Bagian 2	222	0.32773
Bagian 3	452	0.81092
Bagian 4	542	1
Bagian 5	224	0.33193
Bagian 6	66	0
Min	66	
Mak	542	

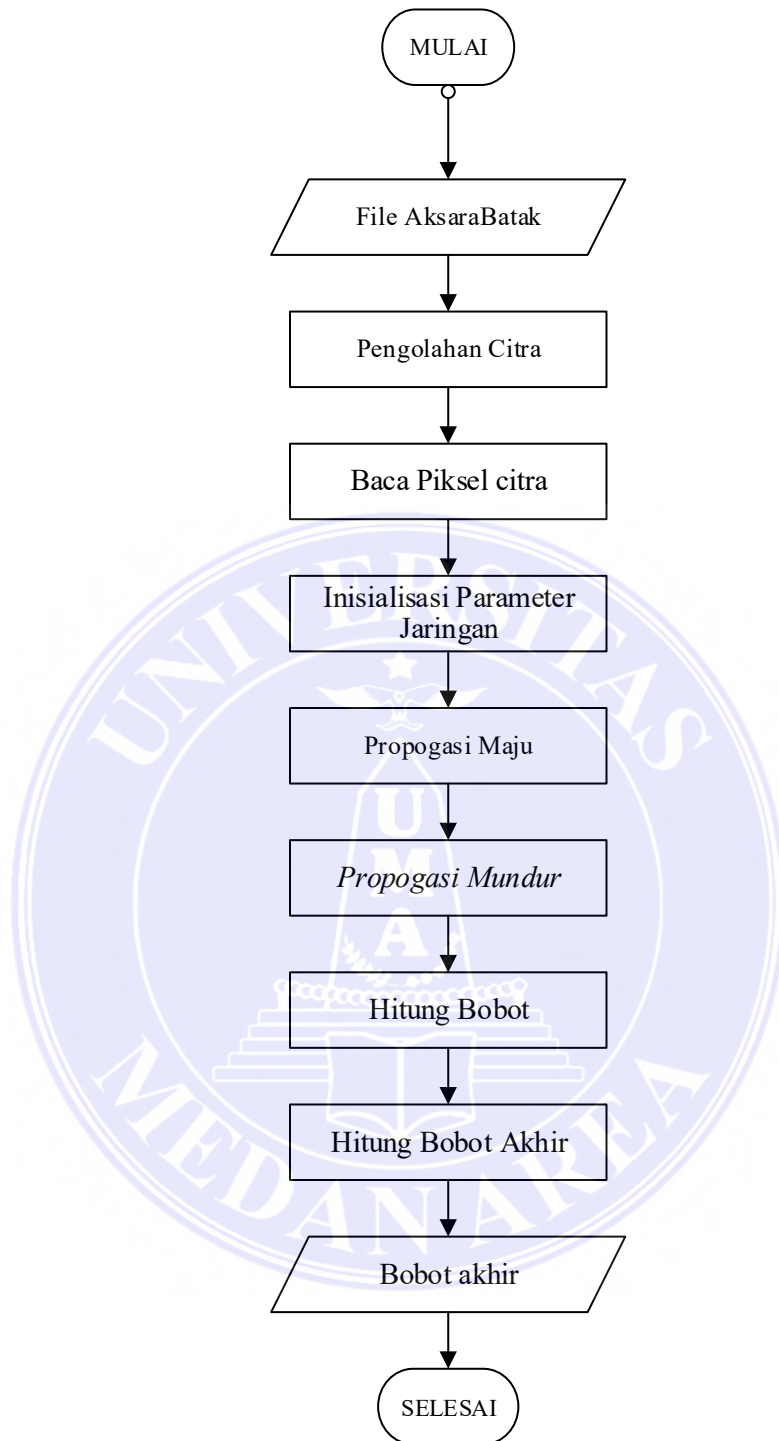
3.1.9 *Flowchart* Pelatihan

Pada penelitian ini dilakukan pengenalan aksara dengan algoritma *Backpropagation* dimulai dengan proses Pelatihan dan Pengenalan.

1. Proses Pelatihan

Proses pelatihan merupakan tahapan dimana dilakukan pengenalan semua data aksara Batak untuk memperoleh bobot akhir semua aksara. Tahapan ini terdiri dari proses maju (*feed forward*) dan proses mundur (*back forward*). Adapun *flowchart* Pelatihan dapat dilihat seperti pada gambar berikut.





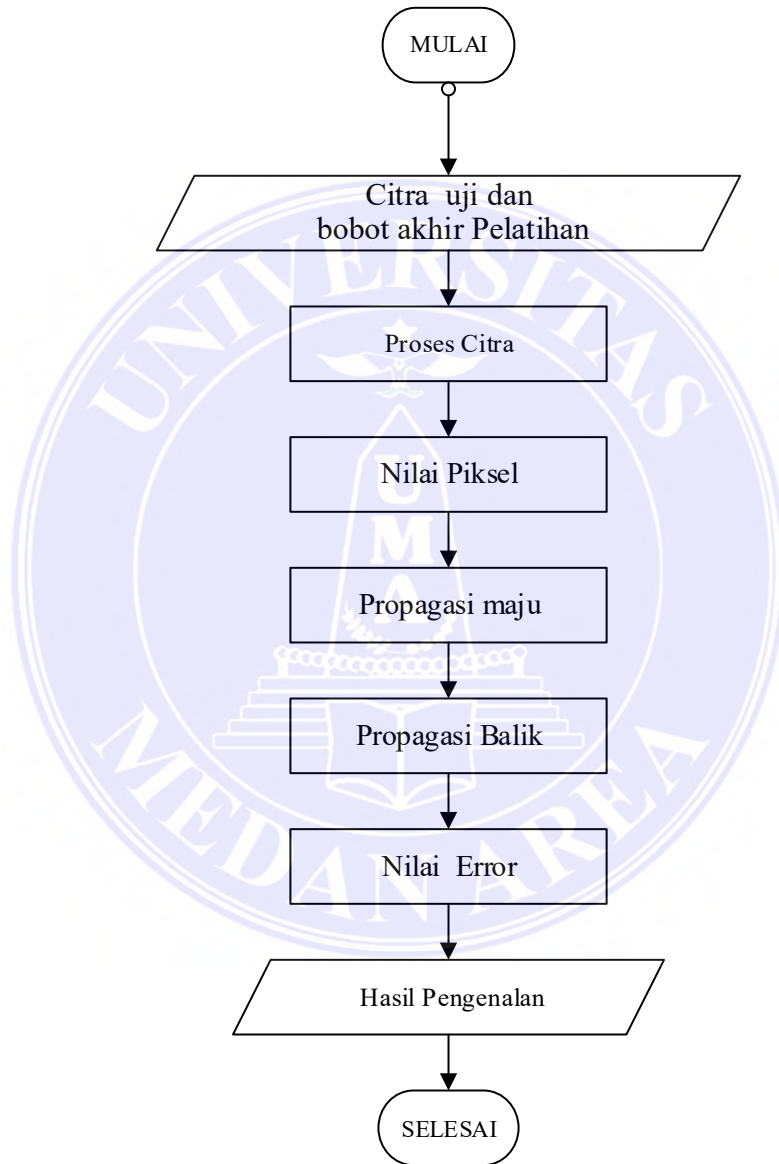
Gambar 3.16 *Flowchart* Pelatihan

Pada awalnya pengguna memasukkan *input* data sampel citra aksara selanjutnya dilakukan proses *preprocessing* citra guna mendapatkan hasil yang maksimal dimana dilakukan konversi citra dari citra warna ke biner. Setelah *preprocessing* selesai, proses

selanjutnya dilakukan proses *Feedforward* dan *Backpropagation* untuk mendapatkan nilai bobot.

2. Proses Pengenalan

Proses pengenalan merupakan suatu proses klasifikasi yang dimana bobot dan bias digunakan dalam proses pelatihan tersebut. Adapun *flowchart* proses Pengenalan dapat dilihat seperti berikut ini:



Gambar 3.17 *Flowchart* Proses Pengenalan

Keterangan:

h = jumlah maksimum *layer output* jaringan (aksara Batak)

a = jumlah maksimum data label citra (aksara Batak)

$error$ = jumlah gagal/ total label

3.20 Arsitektur Jaringan

Berikut ini parameter pada arsitektur jaringan yang digunakan:

- Input* jaringan sesuai dengan jumlah kombinasi ekstraksi ciri yaitu 30.
- Jumlah *hidden layer* yaitu 2 *hidden layer*.
- Jumlah neuron dalam *hidden layer* adalah 10.
- Jumlah lapisan keluaran yaitu 19, dimana harus sesuai jumlah huruf aksara Batak Toba.
- Nilai *epoch* yang digunakan adalah 150, 300, 450, 600, 750, 900, 1050 dan 1200.
- Nilai laju pembelajaran merupakan 0.2.
- Maksimum Kesalahan yaitu 0.001.

3.21 Proses Pelatihan

Saat melakukan pelatihan dipakai dua alur penjumlahan.

- Output*
- suatu ekstraksi ciri yang dimana aksara itu sendiri memakai 30 neuron dalam aksara Batak Toba yang dapat diliha pada tabel berikut.

Tabel 3.11 Ekstraksi Ciri

0	0	0	0.20681	0.34874
0.55224	0.03966	0.44896	0.99757	0.32773
0.10448	1	0.02436	0.98054	0.81092
0.82836	0.41013	0.10325	1	1
0.24627	0.00844	1	0.10706	0.33193
1	0.50211	0.82599	0	0

Dari Tabel diatas adalah vektor masukan aksara Ba dan seluruh vektor aksara diinputkan vektor misalnya dalam table dibawah ini.

Tabel 3.12 Data Input Proses *Training Backpropagation*.

No	Nilai Input	Nilai Output	Aksara
1	0.0000 0.0000 0.0000 0.2068 0.3487 0.5522 0.0397 0.4490 0.9976 0.3277 0.1045 1.0000 0.0244 0.9805 0.8109 0.8284 0.4101 0.1033 1.0000 1.0000 0.2463 0.0084 1.0000 0.1071 0.3319 1.0000 0.5021 0.8260 0.0000 0.0000	0.4402	A
2	0.1646 0.8057 0.5821 0.1969 0.6000 0.0492 0.9652 0.6995 0.5428 0.2216 0.5361 0.1103 0.6642 0.5958 0.2162 0.2274 0.8664 0.6922 0.8020 0.6545 0.4211 0.6610 0.9214 0.6903 0.1741 0.9179 0.1888 0.4765 0.6065 0.4451	0.4790	BA
3	0.2850 0.3573 0.3323 0.9581 0.0216 0.7248 0.7251 0.0662 0.7562 0.5097 0.5769 0.2983 0.8748 0.7769 0.7026 0.1868 0.5976 0.7341 0.3331 0.0598 0.3695 0.9381 0.2332 0.4390 0.7138 0.8225 0.6388 0.5981 0.3694 0.8975	0.5298	DA
4	0.2730 0.7943 0.1473 0.9565 0.1070 0.9650 0.2890 0.1450 0.5456 0.5504 0.4920 0.4133 0.6447 0.7548 0.9144 0.0532 0.8262 0.4709 0.3795 0.2226 0.3569 0.0881 0.9040 0.2745 0.7881 0.8266 0.8378 0.9840 0.2142 0.1029	0.5107	GA
5	0.6508 0.1074 0.3956 0.7256 0.8724 0.2968 0.0827 0.3591 0.4784 0.6774 0.4756 0.9511 0.2197 0.9472 0.5751 0.0338 0.8681 0.4676 0.2958 0.3455 0.5148 0.8003 0.4045 0.6729 0.3462 0.2932 0.1811 0.6091 0.9229 0.3435	0.4971	HA
6	0.7299 0.9896 0.5070 0.6117 0.6741 0.7647 0.4670 0.0504 0.3857 0.0109 0.3398 0.4652 0.1571 0.1680 0.2200 0.2534 0.7500 0.9504 0.6423 0.3205 0.2352 0.5604 0.3364 0.7305 0.8567 0.3174 0.2679 0.9012 0.6669 0.7889	0.5040	I
...
19	0.4978 0.1416 0.4149 0.3633 0.8379 0.2609 0.2587 0.3371 0.1127 0.5054 0.5539 0.8056 0.5474 0.7032 0.2446 0.9362 0.0262 0.9241 0.5563 0.5951 0.0459 0.3635 0.3694 0.7840 0.0212 0.1635 0.8746 0.6230 0.3180 0.9440	0.4710	YA

- c. Ubah nilai target *input* adalah nilai rata-rata ekstraksi ciri setiap bagian pada blok piksel pada table dibawah ini.

Tabel 3.13 Biner Target

No	Aksara	Vektor <i>Output</i>
1	A	0.4402
2	BA	0.4790
3	DA	0.5298
4	GA	0.5107
5	HA	0.4971
6	I	0.5040
7	JA	0.3874
8	LA	0.4215
9	MA	0.4662
10	NA	0.4494
11	NGA	0.4374
12	NYA	0.4435
13	PA	0.3409
14	RA	0.3709
15	SA	0.4103
16	TA	0.3955
17	U	0.3850
18	WA	0.3903
19	YA	0.4710

- d. Tentukan jumlah nilai *epoch* ataupun iterasi, yang dimana nilai kesalahan lalu berikan contoh suatu perhitungan yang dimana parameternya seperti dibawah ini.
1. Maksimal nilai *epoch* yaitu 150, 300, 450, 600, 750, 900, 1050 dan 1200.
 2. Nilai terendah untuk kesalahan yaitu 0,01.
 3. Nilai terhadap pembelajaran yaitu 0,05.
- e. Nilai pertama terhadap nilai acak.

1. Berikut nilai bobot acak yaitu:

$$V = [0.2542 \ 0.2541 \ 0.24850.0547 \ 0.6584 \ 0.4258 \ 0.5242 \ 0.5264 \ 0.4458 \ 0.3254 \\ 0.6654 \ 0.2542 \ 0.7759 \ 0.3254 \ 0.5524 \ 0.4587 \ 0.0254 \ 0.2541 \ 0.3254]$$

2. Nilai awal bias.

$$v_0 = \{0,0,0,0, \text{nilai } v_0 \text{ diperhitungkan sebagai angka } 0\};$$

3. Nilai awal untuk tersembunyi terhadap keluaran (w) yang dimana nilai acak [01]

$$w = [0.2542 \ 0.2541 \ 0.24850.0547 \ 0.6584 \ 0.4258 \ 0.5242 \ 0.5264 \ 0.4458 \ 0.3254 \\ 0.6654 \ 0.2542 \ 0.7759 \ 0.3254 \ 0.5524 \ 0.4587 \ 0.0254 \ 0.2541 \ 0.3254]$$

4. Nilai awal bias terhadap keluaran (w_0).

f. Lanjutkan perulangan hingga nilai suatu iterasi akan lebih kecil dari perulangan maksimum.

g. Proses propagasi maju:

Data = x_1 dengan perulangan = 1

$$\text{Data } x_1 = \{0.4402 \ 0.4790 \ 0.5298 \ 0.5107 \ 0.4971 \ 0.5040 \ 0.3874 \ 0.05264 \ 0.4662 \ 0.4494 \\ 0.4374 \ 0.4435 \ 0.3409 \ 0.3709 \ 0.4103 \ 0.3955 \ 0.38500.4710 \ 0.3951\}$$

Operasi pada *hidden layer* menggunakan fungsi aktivasi Sigmoid Biner:

$$Z_{in_j} = v_{0j} + \sum_{i=1}^n x_i v_{ij}$$

$$z_{in1} =$$

Data	Bobot	z_{in1}
0.44020	0.2542	0.0065
0.47900	0.2541	0.0065
0.52980	0.2485	0.0062
0.51070	0.0547	0.0003
0.49710	0.6584	0.0433
0.50400	0.4258	0.0181
0.38738	0.5242	0.0275
0.42152	0.5264	0.0277
0.46622	0.4458	0.0199
0.44942	0.3254	0.0106
0.43745	0.6654	0.0443
0.44352	0.2542	0.0065
0.34089	0.7759	0.0602
0.37094	0.3254	0.0106
0.41028	0.5524	0.0305
0.39549	0.4587	0.0210
0.38495	0.0254	0.0001
0.39030	0.2541	0.0065
0.44942	0.3254	0.0106
z_{in1}		3.16639

$$z_{in1} = 3.1663$$

$$z_{in2} = 2.4590$$

$$z_{in3} = 4.0215$$

$$z_{in4} = 3.2587$$

.....

$$z_{in30} = 3.1195$$

$$Z_1 = \frac{1}{1 + e^{-3.1663}} = 0.6524$$

$$Z_2 = \frac{1}{1 + e^{-2.4590}} = 0.5952$$

$$Z_3 = \frac{1}{1 + e^{-4.0215}} = 0.5333$$

$$Z_4 = \frac{1}{1 + e^{-3.2587}} = 0.5778$$

.....

$$Z_{30} = \frac{1}{1 + e^{-3.1195}} = 0.8014$$

Operasi pada *output layer*

$$y_{ink} = w_0 + \sum_{i=1}^n w_{jk} * z_j$$

$$y^k = \frac{1}{1 + e^{-y_{in,k}}}$$

$$y_{ink} = 0 + (0.2542 * 0.6524) + (0.2541 * 0.5952) + (0.2485 * 0.5333) + (0.0547 * 0.5778) + \dots + (0.3254 * 0.8014) = 15.2354$$

$$y^k = \frac{1}{1 + e^{-15.2354}} = 0.5774$$

Menghitung selisih dengan target kesalahan:

$$error = T_{ik} - y_k$$

$$= 1 - 0.5774$$

$$= 0.4226$$

h. Alur mundur

Hitung faktor unit kesalahan (δ):

$$\delta_k = (T_{ik} - y_k) * \left(\frac{1}{1 + e^{-y_{in}}}\right) * \left[1 - \left(\frac{1}{1 + e^{-y_{in}}}\right)\right]$$

$$\delta_k = (1 - 0.5774) * \left(\frac{1}{1 + e^{-0.5774}}\right) * \left[1 - \left(\frac{1}{1 + e^{-0.5774}}\right)\right]$$

$$\delta_k = 0.1521$$

Hitung suku perubahan bobot w (Δw_i):

$$\Delta w_{jk} = \alpha * \delta_k * z_j$$

α = rasio

δ = faktor unit kesalahan

$$\Delta w_{10} = (0.05 * 0.1521 * 0.6524) = 0.004961502$$

$$\Delta w_{20} = (0.05 * 0.1521 * 0.5952) = 0.004526496$$

$$\Delta w_{30} = (0.05 * 0.1521 * 0.5333) = 0.004055746$$

$$\Delta w_{40} = (0.05 * 0.1521 * 0.5778) = 0.004394169$$

Menghitung suatu unit kesalahan yang terdapat di lapisan tersembunyi:

$$\delta_{inj} = \delta * w_{jk}$$

$$\delta_{in1} = 0.1521 * 0.004961502$$

$$= 0.0007546444542$$

$$\delta_{in2} = 0.1521 * 0.004526496$$

$$= 0.0006884800416$$

$$\delta_{in3} = 0.1521 * 0.0040557465$$

$$= 0.00061687904265$$

$$\delta_{in4} = 0.1521 * 0.004394169$$

$$= 0.0006683531049$$

Yang menjadi faktor kesalahan pada unit tersembunyi:

$$\delta_i = \delta_{in1} * \left(\frac{1}{1+e^{-z_j}}\right) * \left[1 - \left(\frac{1}{1+e^{-z_j}}\right)\right]$$

$$\delta_1 = 0.0007546444542 * \left(\frac{1}{1+e^{-0.6524}}\right) * \left[1 - \left(\frac{1}{1+e^{-0.6524}}\right)\right] = 0.000033254$$

$$\delta_2 = 0.0006884800416 * \left(\frac{1}{1+e^{-0.5952}}\right) * \left[1 - \left(\frac{1}{1+e^{-0.5952}}\right)\right] = 0.000035241$$

$$\delta_3 = 0.004055746 * \left(\frac{1}{1+e^{-0.5333}}\right) * \left[1 - \left(\frac{1}{1+e^{-0.5333}}\right)\right] = 0.000037211$$

$$\delta_4 = 0.0006683531049 * \left(\frac{1}{1+e^{-0.5778}}\right) * \left[1 - \left(\frac{1}{1+e^{-0.5778}}\right)\right] = 0.000035241$$

Menghitung suku perubahan yang terjadi pada bobot v :

$$\Delta v_{i,j} = \alpha * \delta_j * x_{i,j}$$

$$\Delta v_{1,1} = 0.05 * 0.000033254 * 0.44020$$

$$= 0.00000073192054$$

$$\Delta v_{1,2} = 0.05 * 0.000035241 * 0.47900$$

$$= 0.00000084402195$$

$$\Delta v_{1,3} = 0.05 * 0.000037211 * 0.52980$$

$$= 0.00000098571939$$

$$\begin{aligned} \Delta v_{1,4} &= 0.05 * 0.000037211 * 0.51070 \\ &= 0.000000950182885 \end{aligned}$$

$$\begin{aligned} \Delta v_{1,5} &= 0.05 * 0.000037211 * 0.49710 \\ &= 0.000000924879405 \end{aligned}$$

$$\Delta v_{1,6} = 0.000000852412003$$

$$\Delta v_{1,7} = 0.000000852233003$$

$$\Delta v_{1,8} = 0.000000754221002$$

$$\Delta v_{1,9} = 0.000000663254101$$

$$\Delta v_{1,10} = 0.000000623541002$$

$$\Delta v_{...} = \dots\dots\dots$$

$$\Delta v_{1,19} = 0.000000584512003$$

i. Perubahan yang dilakukan terhadap bobot:

$$w_{ik} = W_{ik} + \Delta W_{ik}$$

$$\begin{aligned} w_1 &= 0.004961502 + 0.00000073192054 \\ &= 0.00496223392054 \end{aligned}$$

$$\begin{aligned} w_2 &= 0.004526496 + 0.00000084402195 \\ &= 0.00452734002195 \end{aligned}$$

$$\begin{aligned} w_3 &= 0.004055746 + 0.00000098571939 \\ &= 0.00405673171939 \end{aligned}$$

$$\begin{aligned} w_4 &= 0.004394169 + 0.000000950182885 \\ &= 0.004395119182885 \end{aligned}$$

w.....

w19

Berikut perhitungan pada bobot v baru:

$$v_{i,j} = v_{i,j} + \Delta v_{i,j}$$

$$v_{1,1} = 0.2542 + 7.3192054E-07 = 2.5420073E-01$$

$$v_{1,2} = 0.2541 + 8.44022E-07 = 2.5410084E-01$$

$$v_{1,3} = 0.2485 + 9.85719E-07 = 2.4850099E-01$$

$$v_{1,4} = 0.0547 + 9.50182E-07 = 5.4700950E-02$$

$$v_{1,5} = 0.6584 + 9.24879E-07 = 6.5840092E-01$$

$$v_{1,6} = 0.4258 + 8.52412E-07 = 4.2580085E-01$$

$$v_{1,7} = 0.5242 + 8.52233E-07 = 5.2420085E-01$$

$$\begin{aligned}
 v_{1,8} &= 0.5264+7.54221E-07= 5.2640075E-01 \\
 v_{1,9} &= 0.4458+6.63254E-07= 4.4580066E-01 \\
 v_{1,10} &= 0.3254+6.23541E-07= 3.2540062E-01 \\
 v_{1,11} &= 0.6654+5.84512E-07= 6.6540058E-01 \\
 v_{1,12} &= 0.2542+7.3064E-09= 2.5420001E-01 \\
 v_{1,13} &= 0.7759+9.133E-11= 7.7590000E-01 \\
 v_{1,14} &= 0.3254+1.14163E-12= 3.2540000E-01 \\
 v_{1,15} &= 0.5524+1.42703E-14= 5.5240000E-01 \\
 v_{1,16} &= 0.4587+1.78379E-16= 4.5870000E-01 \\
 v_{1,17} &= 0.0254+2.22974E-18=2.5400000E-02 \\
 v_{1,18} &= 0.25419+2.78717E-20= 2.5410000E-01 \\
 v_{1,19} &= 0.3254+0.000000584512003= 3.254000E-01
 \end{aligned}$$

Bobot V = 2.5420073E-01 2.5410084E-01 2.4850099E-01 5.4700950E-02
 6.5840092E-01 4.2580085E-01 5.2420085E-01 5.2640075E-01 4.4580066E-01
 3.2540062E-01 6.6540058E-01 2.5420001E-01 7.7590000E-01 3.2540000E-01
 5.5240000E-01 4.5870000E-01 2.5400000E-02 2.5410000E-01 3.2540000E-01

j. Hitung Setiap Bobot Akhir.

Bobot akhir akan dihitung dalam akhir perulangan hingga syarat berhenti terpenuhi, lalu dilanjutkan dalam data ke-2, apabila syarat berhenti telah terpenuhi, maka sebagai contoh bobot akhir yang diperoleh sebelumnya yaitu:

0.098987815
0.095701147
0.10269451
0.155977412
0.245866692
0.284631354
0.244590692
0.24443656
0.940736
0.292099325

$w =$

0.009899
0.009570
0.010269
0.015598
0.024587
0.028463
0.024459
0.024444
0.094074
0.029210

3.22 Proses Pengenalan Aksara Batak Toba

Proses pengenalan ini dilakukan yaitu menggunakan bobot akhir diperoleh pada proses pelatihan, adapun proses pengenalan pada aksara Batak Toba dengan contoh *input* citra aksara yakni sebagai berikut:

0.30504	0.39048	39744
0.30492	0.79848	
0.2982	0.30504	
0.06564	0.93108	
0.79008	0.39048	
0.51096	0.66288	
0.62904	0.55044	
0.63168	0.03048	
0.53496	0.30492	

Pada citra diatas hitung jarak data yang dimasukkan tersebut yang dimana akan memakai kedua bobot yakni sebagai berikut:

- a) Pada bobot data uji dihitung dimana aksara menggunakan penjumlahan dengan alur maju yakni dilakukannya satu persatu tiap bobot w dan v dalam ekstraksi karakteristik citra aksaranya apabila data *output* telah benar dengan menggunakan tujuan maka perhitungan dihentikan.

$$z_{inj} = v_{oj} + \sum_{i=1}^n x_i v_{ij}$$

Vektor *input*: [0.30504 0.30492 0.2982 0.06564 0.79008 0.51096 0.62904 0.63168 0.53496 0.39048 0.79848 0.30504 0.93108 0.39048 0.66288 0.55044 0.03048 0.30492 0.39744]

Misalkan bobot akhir pelatihan adalah: [0.167772 0.167706 0.16401 0.036102 0.434544 0.281028 0.345972 0.347424 0.294228 0.214764 0.439164 0.167772 0.512094 0.214764 0.364584 0.302742 0.016764 0.167706 0.218592]

$z_{in1} =$

0.30504	0.79848	X	0.167772	0.439164
0.30492	0.30504		0.167706	0.167772
0.2982	0.93108		0.16401	0.512094
0.06564	0.39048		0.036102	0.214764
0.79008	0.66288		0.434544	0.364584
0.51096	0.55044		0.281028	0.302742
0.62904	0.03048		0.345972	0.016764
0.63168	0.30492		0.347424	0.167706
0.53496	0.39744		0.294228	0.218592
0.39048			0.214764	

$= 6.854221$

$z_{in1} = 6.854221$

$z_{in2} = 7.365442$

$z_{in3} = 5.326580$

$z_{in4} = 4.111247$

$$Z1 = \frac{1}{1 + e^{-6.854221}} = 0.5485$$

$$Z2 = \frac{1}{1 + e^{-7.365442}} = 0.8854$$

$$Z3 = \frac{1}{1 + e^{-5.326580}} = 0.55241$$

$$Z4 = \frac{1}{1 + e^{-4.111247}} = 0.52411$$

....

$$Z30 = \frac{1}{1 + e^{-7.5421}} = 0.8978$$

Operasi untuk keluaran terhadap *layer*:

$$y_k = \frac{1}{1 + e^{-y_{in_k}}}$$

$$y_{in} = 0 + (0.9267 * 0.5485) + (0.2497 * 0.8854) + (0.8463 * 0.55241) +$$

$$(0.2970 * 0.52411) + \dots + (0.9563 * 0.5266) = 6.3224$$

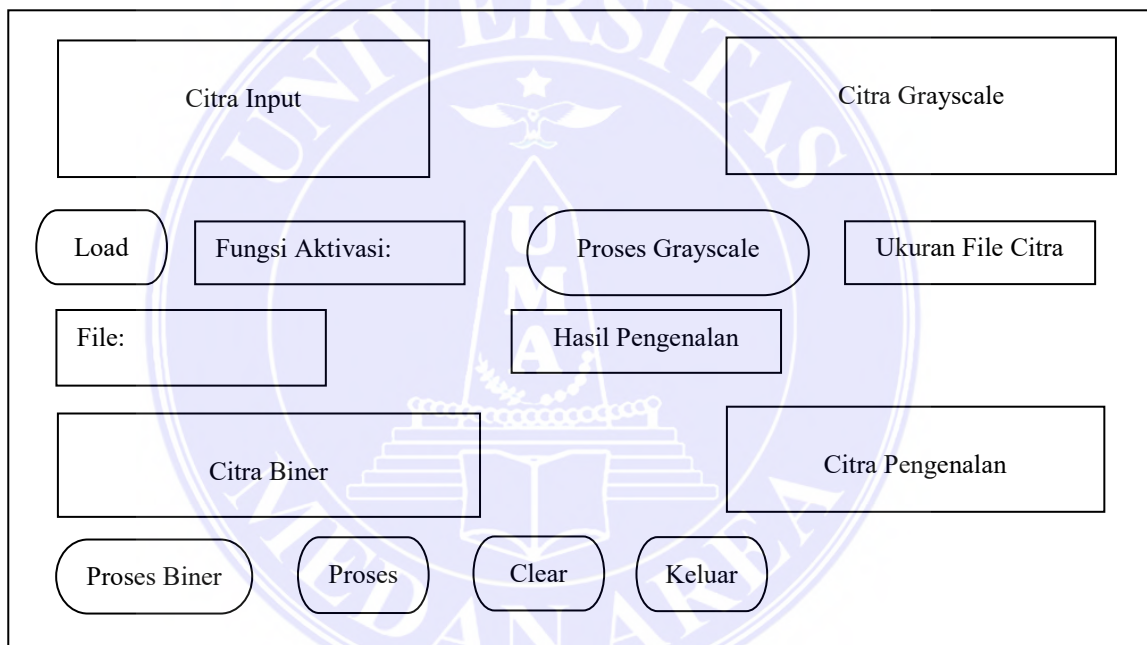
$$y = \frac{1}{1 + e^{-6.3224}}$$

$$y = 0.7222$$

- b) Pada perhitungan di atas didapatkan nilai $y = \{0.7222\}$ dimana $0.7222 \sim 0.5107$ dimana hasil ini mendekati nilai vektor *output* citra aksara GA [0.5107].

3.23 Perancangan Aplikasi

Berikut ini sebuah rancangan aplikasi pengenalan aksara Batak Toba dengan menggunakan metode *Backpropagation* dengan bentuk dialog yaitu:



Gambar 3.18 Rancangan Pengenalan

Pada Gambar 26 di atas merupakan gambar perancangan dialog yaitu gambaran program pengenalan aksara Batak dengan metode *Backpropagation*. Dimulai dari memasukkan citra aksara yang akan diuji dengan cara menekan *button* "Load". Setelah di *input* atau memasukkan citra tersebut, kemudian tekan *button* "Proses Grayscale", yang dimana ketika *button* tersebut ditekan akan muncul citra aksara secara otomatis berubah menjadi citra *grayscale*, selanjutnya tekan *button* "Proses Biner", yang dimana ketika *button* tersebut ditekan akan muncul citra aksara secara otomatis berubah menjadi citra biner dan pilih tombol Proses untuk melakukan proses pengenalan dengan hasilnya akan tampil dengan ukuran filenya. Setelah citra hasil yang diproses keluar,

secara otomatis akan menampilkan nama jenis aksara pada kolom hasil pengenalan. Kemudian tekan tombol *button* “Clear” untuk membersihkan hasil uji coba yang telah dilakukan atau jika ingin mengulangi dengan proses berikutnya, tekan *button* “Clear”.



BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Pengenalan aksara Batak Toba yang selama ini masih dilakukan secara manual. Pengenalan aksara pada penelitian ini dengan penerapan fungsi aktivasi Sigmoid Bipolar dan Sigmoid Biner pada algoritma *Backpropagation* serta untuk mendapatkan nilai akurasi masing-masing fungsi aktivasinya.

Setelah dilakukan pengujian pengenalan aksara Batak Toba dengan algoritma *Backpropagation* menggunakan 190 data *training* yang terdiri dari 19 aksara Batak Toba serta 10 kali pengujian dengan menggunakan *epoch* 150, 300, 450, 600, 750, 900, 1050 dan 1200 maka diperoleh bahwa nilai akurasi yang terbaik pada penggunaan fungsi aktivasi Sigmoid Bipolar pada *epoch* ke 1050 sebesar 80.53% dan pada Sigmoid Biner pada *epoch* ke 1050 sebesar 78.95%, dimana selisih nilai akurasi yaitu sebesar 1.59% dan nilai akurasi Sigmoid Bipolar lebih baik dibandingkan Sigmoid Biner.

5.2 Saran

Sebagai saran dari penulis agar hasil pengenalan aksara batak Toba lebih akurat adalah dengan menggunakan variasi ekstraksi fitur kemudian jumlah data untuk pelatihan diperbanyak dan penggunaan jumlah terhadap *hidden layer* secara optimum agar pelatihannya tidak terlalu lama karena *error* yang besar. Untuk penelitian selanjutnya agar menggunakan fungsi aktivasi lain ataupun menggunakan metode jaringan saraf tiruan yang lainnya.

DAFTAR PUSTAKA

- Adinugroho, S. &. (2017). Perbandingan Jaringan *Learning Vector Quantization* dan *Backpropagation* pada Klasifikasi Daun Berbasiskan Fitur Gabungan. *Jurnal Informatika & Multimedia*, Vol. 9, No. 02.
- Amalia, N. H. (2020). Pengenalan Aksara Sunda Menggunakan Metode Jaringan Saraf Tiruan *Backpropagation* Dan Deteksi Tepi Canny. *Journal of Computer Engineering System and Science* , Vol. 5 No. 1 Januari 2020.
- Amrutha, A. &. (2018). Performance analysis of Backpropagation Algorithm of Artificial Neural Networks in Verilog. *3rd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT-2018), MAY 18th, -*.
- Aritonang, M. &. (2019). *An Application of Backpropagation Neural Network for Sales Forecasting Rice Milling Unit. IEEE International Journal, -*.
- Faturrahman, I. A. (2018.). Pengenalan Pola Huruf Hijaiyah Khat Kufi Dengan Metode Deteksi Tepi Sobel Berbasis Jaringan Syaraf Tiruan *Backpropagation*.. *Jurnal Teknik Informatika* , Vol 11 NO. 1, April 2018.
- Gonzalez, R. &. (2003). *Digital Image Processing. Second edition, USA: Addison-Wesley Publishing Co, University of Tennessee., -*.
- Isnanto, R. R. (2017). *Herb Leaves Recognition Using Combinations of Hu's Moment Variants - Backpropagation Neural Network and 2-D Gabor Filter - Learning Vector Quantization (LVQ)*
- Iwan, Suhardi (2007). Evaluasi Pengaruh Fungsi Aktivasi Dan Parameter Kemiringan Terhadap Unjuk Kerja Pengenalan Jaringan Saraf Tiruan. *Jurnal Ilmiah Semesta Teknika* Vol. 10 No.1, 2007:53-68
- Julpan, Nababan, E. B. & Zarlis, M. 2016. Analisis Fungsi Aktivasi Sigmoid Biner Dan Sigmoid Bipolar Dalam Algoritma *Backpropagation* Pada Prediksi Kemampuan Siswa. *Jurnal Teknovasi* Volume 02, Nomor 1, 2015, 103 – 116 ISSN : 2355-701X
- Kozok, U. (2009). Sejarah Perkembangan Tulisan Batak. *Kepustakaan Populer Gramedia., -*.

- Quantisation for Pattern Recognition of Hijaiyah Letters. 6th International Conference on Information and Communication Techno*, -.
- Nugroho, W. &. (2017). *Transliterasi Citra Aksara Hiragana Mempergunakan Jaringan Backpropagation*. Yogyakarta, -.
- Prasetaningtyas, Y. (2016). *Klasifikasi Pola Tanda Tangan Menggunakan Backpropagation*. Yogyakarta: Skripsi Fakultas Sains dan Teknologi. Universitas Sanata Dharma..
- Putri, A. R. (2017). *Pengenalan Pola Aksara Batak Toba Menggunakan Metode Backpropagation*. Yogyakarta: Skripsi Program Studi Teknik Informatika Jurusan Teknik Informatika Fakultas Sains Dan Teknologi Universitas Sanata Dharma .
- Putro B. C. S., M. I. (2018). *Optimized Backpropagation Artificial Neural Network Algorithm for Smart Agriculture Applications*,. *International Conference on Science and Technology (ICST)* (pp. -). Yogyakarta, Indonesia.: -.
- Riansyah, R. R. (2017). 2017. *Sistem Pengenalan Aksara Sunda Menggunakan Metode Modified Direction Feature Dan Learning Vector Quantization*. *Jurnal Teknik Informatika dan Sistem Informasi*, Volume 3 Nomor 1 .
- Rina Septriana & Vitalis Ayu. (2015) *Penerapan Metode Backpropagation Pada Pengenalan Objek Menggunakan Multiple Hidden Layer Teknik Informatika Dan Komputer Semnaskit ISSN 2477-5649*.
- Sri Redjeki. (2015) *Analisis Fungsi Aktivasi Sigmoid Algoritma Backpropagation Pada Prediksi Data*
- Sutoyo, T. M. (2010). *Teori Pengolahan Citra Digital*. Yogyakarta: Penerbit ANDI .
- Syafria, F. B. (2016). *A Comparison of Backpropagation and LVQ: a Case Study Of Lung Sound Recognition Propagation Classifier for Gearbox Fault Diagnosis. Applied Artificial Intelligence An International Journal*,
- Winardi, S. *Penerapan Metode Jaringan Syaraf Tiruan Backpropagation Dalam Pengenalan Pola Aksara Hanacaraka. Jurnal Teknologi Informasi. Vol . IX Nomor 27 Nopember 2015.*, -.

LAMPIRAN

- Data yang Digunakan

Aksara Batak Toba										
A										
BA										
DA										
GA										
HA										
I										
JA										
LA										
MA										
NA										
NGA										
NYA										
PA										
RA										
SA										
TA										
U										
WA										
YA										

- **Sourcode Home**

```

1.function varargout = Menu(varargin)
2. % MENU MATLAB code for Menu.fig
3. %   MENU, by itself, creates a new MENU or raises the existing
4. %   singleton*.
5. %
6. %   H = MENU returns the handle to a new MENU or the handle to
7. %   the existing singleton*.
8. %
9. %   MENU('CALLBACK',hObject,eventData,handles,...) calls the local
10.%   function named CALLBACK in MENU.M with the given input
    arguments.
11.%
12.%   MENU('Property','Value',...) creates a new MENU or raises the
13.%   existing singleton*. Starting from the left, property value pairs are
14.%   applied to the GUI before Menu_OpeningFcn gets called. An
15.%   unrecognized property name or invalid value makes property application
16.%   stop. All inputs are passed to Menu_OpeningFcn via varargin.
17.%
18.%   *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only
    one
19.%   instance to run (singleton)".
20.%
21.% See also: GUIDE, GUIDATA, GUIHANDLES
22.% Edit the above text to modify the response to help Menu
23.% Last Modified by GUIDE v2.5 13-Oct-2020 10:19:14
24.% Begin initialization code - DO NOT EDIT
25. gui_Singleton = 1;
26. gui_State = struct('gui_Name',       mfilename, ...
27.                   'gui_Singleton',  gui_Singleton, ...
28.                   'gui_OpeningFcn', @Menu_OpeningFcn, ...
29.                   'gui_OutputFcn',  @Menu_OutputFcn, ...
30.                   'gui_LayoutFcn',  [], ...
31.                   'gui_Callback',   []);
32. if nargin && ischar(varargin{1})
33.     gui_State.gui_Callback = str2func(varargin{1});
34. end
35. if nargout
36.     [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
37. else
38.     gui_mainfcn(gui_State, varargin{:});
39. end
40.% End initialization code - DO NOT EDIT
41.% --- Executes just before Menu is made visible.
42.function Menu_OpeningFcn(hObject, eventdata, handles, varargin)
43.% This function has no output args, see OutputFcn.

```

```

44. % hObject handle to figure
45. % eventdata reserved - to be defined in a future version of MATLAB
46. % handles structure with handles and user data (see GUIDATA)
47. % varargin command line arguments to Menu (see VARARGIN)
48. % Choose default command line output for Menu
49. handles.output = hObject;
50. % Update handles structure
51. guidata(hObject, handles);
52. % UIWAIT makes Menu wait for user response (see UIRESUME)
53. % uiwait(handles.figure1);
54. % --- Outputs from this function are returned to the command line.
55. function varargout = Menu_OutputFcn(hObject, eventdata, handles)
56. % varargout cell array for returning output args (see VARARGOUT);
57. % hObject handle to figure
58. % eventdata reserved - to be defined in a future version of MATLAB
59. % handles structure with handles and user data (see GUIDATA)
60. % Get default command line output from handles structure
61. varargout{1} = handles.output;
62. axes(handles.axes1);
63. imshow('Logo-UMA.png');
64. % -----
65. function mnuPelatihan_Callback(hObject, eventdata, handles)
66. % hObject handle to mnuPelatihan (see GCBO)
67. % eventdata reserved - to be defined in a future version of MATLAB
68. % handles structure with handles and user data (see GUIDATA)
69. Pelatihan;
70. % clc; clear; close all;
71. % image_folder = 'Citra Latih';
72. % filenames = dir(fullfile(image_folder, '*.bmp'));
73. % total_images = numel(filenames);
74. % area = zeros(1,total_images);
75. % perimeter = zeros(1,total_images);
76. % metric = zeros(1,total_images);
77. % eccentricity = zeros(1,total_images);
78. %
79. % for n = 1:total_images
80. % full_name= fullfile(image_folder, filenames(n).name);
81. % I = imread(full_name);
82. % J = I(:,:,1);
83. % K = I; %im2bw(J,.6);
84. % L = imcomplement(K);
85. % str = strel('disk',5);
86. % M = imclose(L,str);
87. % N = imfill(M,'holes');
88. % %O = bwareaopen(N,5000);
89. % O = bwareaopen(N,50);
90. % stats = regionprops(O,'Area','Perimeter','Eccentricity');
91. % area(n) = stats.Area;
92. % perimeter(n) = stats.Perimeter;

```



```

93. % metric(n) = 4*pi*area(n)/(perimeter(n)^2);
94. % eccentricity(n) = stats.Eccentricity;
95. % end
96. %
97. % input = [metric;eccentricity];
98. % target = zeros(1,25);
99. % target(:,1:1) = 1;
100. % target(:,2:2) = 2;
101. % target(:,3:3) = 3;
102. % target(:,4:4) = 4;
103. % target(:,5:5) = 5;
104. %
105. %
106. % netBP = newff(input,target,[10 5],{'logsig','logsig'},'trainlm');
107. % netBP.trainParam.epochs = 1000;
108. % netBP.trainParam.goal = 1e-6;
109. % netBP = train(netBP,input,target);
110. % output = round(sim(netBP,input));
111. % save netBP.mat netBP;
112. % [m,n] = find(output==target);
113. % akurasi = (sum(m)/total_images)*100
114. % msgbox('Pelatihan selesai.....');
115. % -----
116. function mnuPengenalan_Callback(hObject, eventdata, handles)
117. % hObject handle to mnuPengenalan (see GCBO)
118. % eventdata reserved - to be defined in a future version of MATLAB
119. % handles structure with handles and user data (see GUIDATA)
120. RecogAksaraBtk;
121. % -----
122. function mnuKeluar_Callback(hObject, eventdata, handles)
123. % hObject handle to mnuKeluar (see GCBO)
124. % eventdata reserved - to be defined in a future version of MATLAB
125. % handles structure with handles and user data (see GUIDATA)
126. close;

```

- **Sourcode Pelatihan**

1. `clc; clear; close all;`
2. `image_folder = 'Citra Latih';`
3. `filenames = dir(fullfile(image_folder, '*.png'));`
4. `total_images = numel(filenames);`
5. `area = zeros(1,total_images);`
6. `perimeter = zeros(1,total_images);`
7. `metric = zeros(1,total_images);`
8. `eccentricity = zeros(1,total_images);`
9. `for n = 1:total_images`
10. `full_name= fullfile(image_folder, filenames(n).name);`
11. `I = imread(full_name);`
12. `J = I(:,:,1);`

```

13. K = im2bw(J,.6);
14. %K = I; %(J,.6);
15. L = imcomplement(K);
16. str = strel('disk',6);
17. M = imclose(L,str);
18. N = imfill(M,'holes');
19. O = bwareaopen(N,50);
20. Stats=regionprops(O,'Area','Perimeter','Eccentricity');regionprops(O,
21. area(n) = stats.Area;
22. perimeter(n) = stats.Perimeter;
23. metric(n) = 4*pi*area(n)/(perimeter(n)^2);
24. eccentricity(n) = stats.Eccentricity;
25. end
26. input = [metric;eccentricity];
27. target = zeros(1,total_images);
28. target(:,1:10) = 1;
29. target(:,11:20) = 2;
30. target(:,21:30) = 3;
31. target(:,31:40) = 4;
32. target(:,41:50) = 5;
33. target(:,51:60) = 6;
34. target(:,61:70) = 7;
35. target(:,71:80) = 8;
36. target(:,81:90) = 9;
37. target(:,91:100) = 10;
38. target(:,101:110) = 11;
39. target(:,111:120) = 12;
40. target(:,121:130) = 13;
41. target(:,131:140) = 14;
42. target(:,141:150) = 15;
43. target(:,151:160) = 16;
44. target(:,161:170) = 17;
45. target(:,171:180) = 18;
46. target(:,181:total_images) = 19;
47. netBP = newff(input,target,[total_images 15],{'tansig','logsig'},'trainlm');
48. %netBP = newff (minmax (total_images), [10, 15, 1], {'tansig','purelin',
tansig'},'traingd');
49. %P = (Berisi nilai input data training tahun 2010-2012)
50. %T = (Berisi nilai target dari inputan data training tahun 2013)
51. % net = newff (minmax (P) , [10, 15, 1], {'tansig', 'purelin', tansig'} ,
'traingd');
52. %net. LW { 1, 1 };
53. %net. b { 1 };
54. %net. LW { 2, 1 };
55. %net. b { 2 };
56. %net. LW { 3, 2 };
57. %net. trainParam. epochs = 150000;
58. %net. trainParam. goal = 0.001;
59. %net. trainParam. Lr = 0.01;

```

```

60. %net.trainParam.show = 1000;
61. %net.b { 3 };
62. %net = train ( net, P, T)
63. netBP.trainParam.epochs = 10000;
64. netBP.trainParam.goal = 0.001;
65. netBP.trainParam.Lr = 0.01;
66. netBP.trainParam.show = 1000;
67. netBP = train(netBP,input,target);
68. output = round(sim(netBP,input));
69. save netBP.mat netBP;
70. [m,n] = find(output==target);
71. akurasi = sum(m)/total_images*100;
72. msgbox('Pelatihan selesai....');
73. Menu;

```

- **Sourcode Pengenalan**

```

1. function varargout = PengenalanAksaraBtk(varargin)
2. gui_Singleton = 1;
3. gui_State = struct('gui_Name',      mfilename, ...
4.   'gui_Singleton',  gui_Singleton, ...
5.   'gui_OpeningFcn', @PengenalanAksaraBtk_OpeningFcn, ...
6.   'gui_OutputFcn',  @PengenalanAksaraBtk_OutputFcn, ...
7.   'gui_LayoutFcn',  [] , ...
8.   'gui_Callback',   []);
9. if nargin && ischar(varargin{1})
10.     gui_State.gui_Callback = str2func(varargin{1});
11. end
12. if nargout
13.     [varargout{1:nargout}] = gui_mainfcn(gui_State,
varargin{:});
14. else
15.     gui_mainfcn(gui_State, varargin{:});
16. end
17. % End initialization code - DO NOT EDIT
18. % --- Executes just before PengenalanAksaraBtk is made
visible.
19. function PengenalanAksaraBtk_OpeningFcn(hObject,
eventdata, handles, varargin)
20. % This function has no output args, see OutputFcn.
21. % hObject    handle to figure
22. % eventdata  reserved - to be defined in a future version
of MATLAB
23. % handles    structure with handles and user data (see
GUIDATA)
24. % varargin   command line arguments to
PengenalanAksaraBtk (see VARARGIN)
25. % Choose default command line output for
PengenalanAksaraBtk
26. handles.output = hObject;

```

```

27.     guidata(hObject, handles);
28.     movegui(hObject, 'center');
29.     % --- Outputs from this function are returned to the
        command line.
30.     function varargout =
        PengenalanAksaraBtk_OutputFcn(hObject, eventdata, handles)
31.     % varargout cell array for returning output args (see
        VARARGOUT);
32.     % hObject    handle to figure
33.     % eventdata reserved - to be defined in a future version
        of MATLAB
34.     % handles    structure with handles and user data (see
        GUIDATA)
35.     % Get default command line output from handles structure
36.     varargout{1} = handles.output;
37.     % --- Executes on button press in btnFile.
38.     function btnFile_Callback(hObject, eventdata, handles)
39.     % hObject    handle to btnFile (see GCBO)
40.     % eventdata reserved - to be defined in a future version
        of MATLAB
41.     % handles    structure with handles and user data (see
        GUIDATA)
42.     [nama_file,nama_path] = uigetfile({'*.*'});
43.
44.     if ~isequal(nama_file,0)
45.         I = imread(fullfile(nama_path,nama_file));
46.         axes(handles.axes1)
47.         imshow(I)
48.         handles.I = I;
49.         guidata(hObject,handles)
50.         lebar=size(I,2);
51.         tinggi=size(I,1);
52.         Dimensi=lebar*tinggi;
53.         set
        (handles.txtNamaFile, 'string', fullfile(nama_path,nama_file));
54.         set (handles.txtSize, 'string', num2str(Dimensi));
55.         else
56.             return
57.         end
58.     % --- Executes on button press in btnProses.
59.     function btnProses_Callback(hObject, eventdata, handles)
60.     % hObject    handle to btnProses (see GCBO)
61.     % eventdata reserved - to be defined in a future version
        of MATLAB
62.     % handles    structure with handles and user data (see
        GUIDATA)
63.     I = handles.I;
64.     CitraGray=I;%rgb2gray(I);
65.     axes(handles.axes4);
66.     imshow(CitraGray);
67.     J = I(:, :, 1);
68.     K = im2bw(J, .6);
69.     L = imcomplement(K);

```

```

70.     str = strel('disk',5);
71.     M = imclose(L,str);
72.     N = imfill(M,'holes');
73.     %O = bwareaopen(N,5000);
74.     O = bwareaopen(N,50);
75.     stats = regionprops(O, 'Area', 'Perimeter', 'Eccentricity');
76.     area = stats.Area;
77.     perimeter = stats.Perimeter;
78.     metric = 4*pi*area/(perimeter^2);
79.     eccentricity = stats.Eccentricity;
80.     input = [metric;eccentricity];
81.     load netBP
82.     output = round(sim(netBP,input));
83.     axes(handles.axes2)
84.     imshow(O)
85.     %handles = guidata(hObject);
86.     %switch get(handles.cboAktifasi, 'Value')
87.     %case 1
88.     %     aktifasi='Bipolar';
89.     %otherwise
90.     %     aktifasi='Biner';
91.     %end
92.     imshow(O)
93.     if output == 1
94.         kelas = 'A';
95.     elseif output == 2
96.         kelas = 'HA';
97.     elseif output == 3
98.         kelas = 'MA';
99.     elseif output == 4
100.        kelas = 'NA';
101.    elseif output == 5
102.        kelas = 'RA';
103.    elseif output == 6
104.        kelas = 'TA';
105.    elseif output == 6
106.        kelas = 'SA';
107.    elseif output == 6
108.        kelas = 'PA';
109.    elseif output == 6
110.        kelas = 'LA';
111.    elseif output == 6
112.        kelas = 'GA';
113.    elseif output == 6
114.        kelas = 'JA';
115.    elseif output == 6
116.        kelas = 'DA';
117.    elseif output == 6
118.        kelas = 'NGA';
119.    elseif output == 6
120.        kelas = 'BA';
121.    elseif output == 6


```

```

122.         kelas = 'WA';
123.     elseif output == 6
124.         kelas = 'YA';
125.     elseif output == 6
126.         kelas = 'NYA';
127.     elseif output == 6
128.         kelas = 'I';
129.     elseif output == 6
130.         kelas = 'U';
131.     end
132.     set(handles.txtHasil,'String',kelas)
133.     % --- Executes on button press in btnClear.
134.     function btnClear_Callback(hObject, eventdata, handles)
135.     % hObject    handle to btnClear (see GCBO)
136.     % eventdata  reserved - to be defined in a future version
of MATLAB
137.     % handles    structure with handles and user data (see
GUIDATA)
138.     set (handles.txtNamaFile,'string','');
139.     set (handles.txtHasil,'string','');
140.     set (handles.txtSize,'string','');
141.     axes(handles.axes1);
142.     cla('reset');
143.     axes(handles.axes2);
144.     cla('reset');
145.     axes(handles.axes3);
146.     cla('reset');
147.     axes(handles.axes4);
148.     cla('reset');
149.     clc;
150.     clc;
151.     clc;
152.     clc;
153.     clear all;
154.     % --- Executes during object creation, after setting all
properties.
155.     function figure1_CreateFcn(hObject, eventdata, handles)
156.     % hObject    handle to figure1 (see GCBO)
157.     % eventdata  reserved - to be defined in a future version
of MATLAB
158.     % handles    empty - handles not created until after all
CreateFcns called
159.     % --- Executes on mouse press over figure background,
over a disabled or
160.     % --- inactive control, or over an axes background.
161.     function figure1_WindowButtonUpFcn(hObject, eventdata,
handles)
162.     % hObject    handle to figure1 (see GCBO)
163.     % eventdata  reserved - to be defined in a future version
of MATLAB
164.     % handles    structure with handles and user data (see
GUIDATA)
165.     % --- Executes on selection change in cboAktifasi.

```

```
166.     function cboAktifasi_Callback(hObject, eventdata,
        handles)
167.     % hObject    handle to cboAktifasi (see GCBO)
168.     % eventdata  reserved - to be defined in a future version
        of MATLAB
169.     % handles    structure with handles and user data (see
        GUIDATA)
170.     % Hints: contents = cellstr(get(hObject,'String'))
        returns cboAktifasi contents as cell array
171.     %           contents{get(hObject,'Value')} returns selected
        item from cboAktifasi
172.     % --- Executes during object creation, after setting all
        properties.
173.     function cboAktifasi_CreateFcn(hObject, eventdata,
        handles)
174.     % hObject    handle to cboAktifasi (see GCBO)
175.     % eventdata  reserved - to be defined in a future version
        of MATLAB
176.     % handles    empty - handles not created until after all
        CreateFcns called
177.     % Hint: popupmenu controls usually have a white
        background on Windows.
178.     %           See ISPC and COMPUTER.
179.     if ispc && isequal(get(hObject,'BackgroundColor'),
        get(0,'defaultUicontrolBackgroundColor'))
180.         set(hObject,'BackgroundColor','white');
181.     end
```



UNIVERSITAS MEDAN AREA
FAKULTAS TEKNIK

Kampus I : Jalan Kolan Nomor 1 Medan Estate/Jalan PBSI Nomor 1 ☎ (061) 7366878, 7360168, 7364348, 7366781, Fax. (061) 7366998 Medan 20223
Kampus II : Jalan Setiabudi Nomor 79 / Jalan Sei Serayu Nomor 70 A, ☎ (061) 8225602, Fax. (061) 8226331 Medan 20122
Website: www.teknik.uma.ac.id E-mail: univ_medanana@uma.ac.id

Nomor : 165/FT.6/01.10/VI/2022 22 Juni 2022
Lamp : -
Hal : **Perubahan Judul Tugas Akhir & Perpanjangan SK Pembimbing Tugas Akhir**

Yth, Pembimbing Tugas Akhir
Nurul Khairina, S.Kom, M.Kom
Zulfikar Sembiring, S.Kom., M.Kom
di
Tempat

Dengan hormat,
Sehubungan dengan adanya perubahan judul tugas akhir dan telah berakhirnya waktu masa berlaku SK pembimbing nomor 191/FT.6/01.10/X/2021 pada tanggal 28 Oktober 2021 maka perlu diterbitkan kembali SK Pembimbing Skripsi baru atas nama mahasiswa tersebut :

Nama : Esrayanti Simanjuntak
N P M : 178160055
Jurusan : Informatika

Maka dengan hormat kami mengharapkan kesediaan saudara :



1. **Nurul Khairina, S.Kom, M.Kom** (Sebagai Pembimbing I)
2. **Zulfikar Sembiring, S.Kom., M.Kom** (Sebagai Pembimbing II)

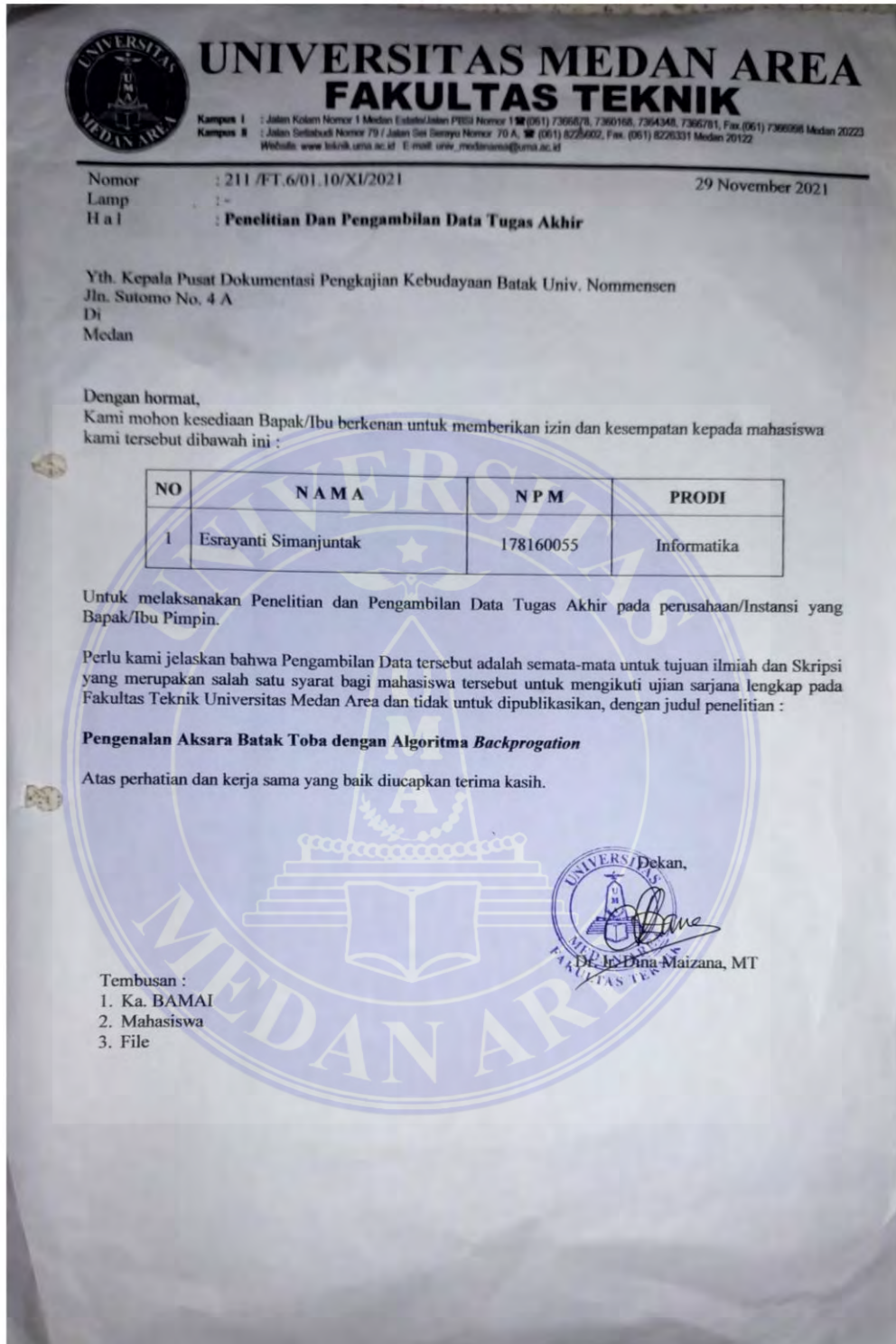
Adapun Tugas Akhir Skripsi berjudul :

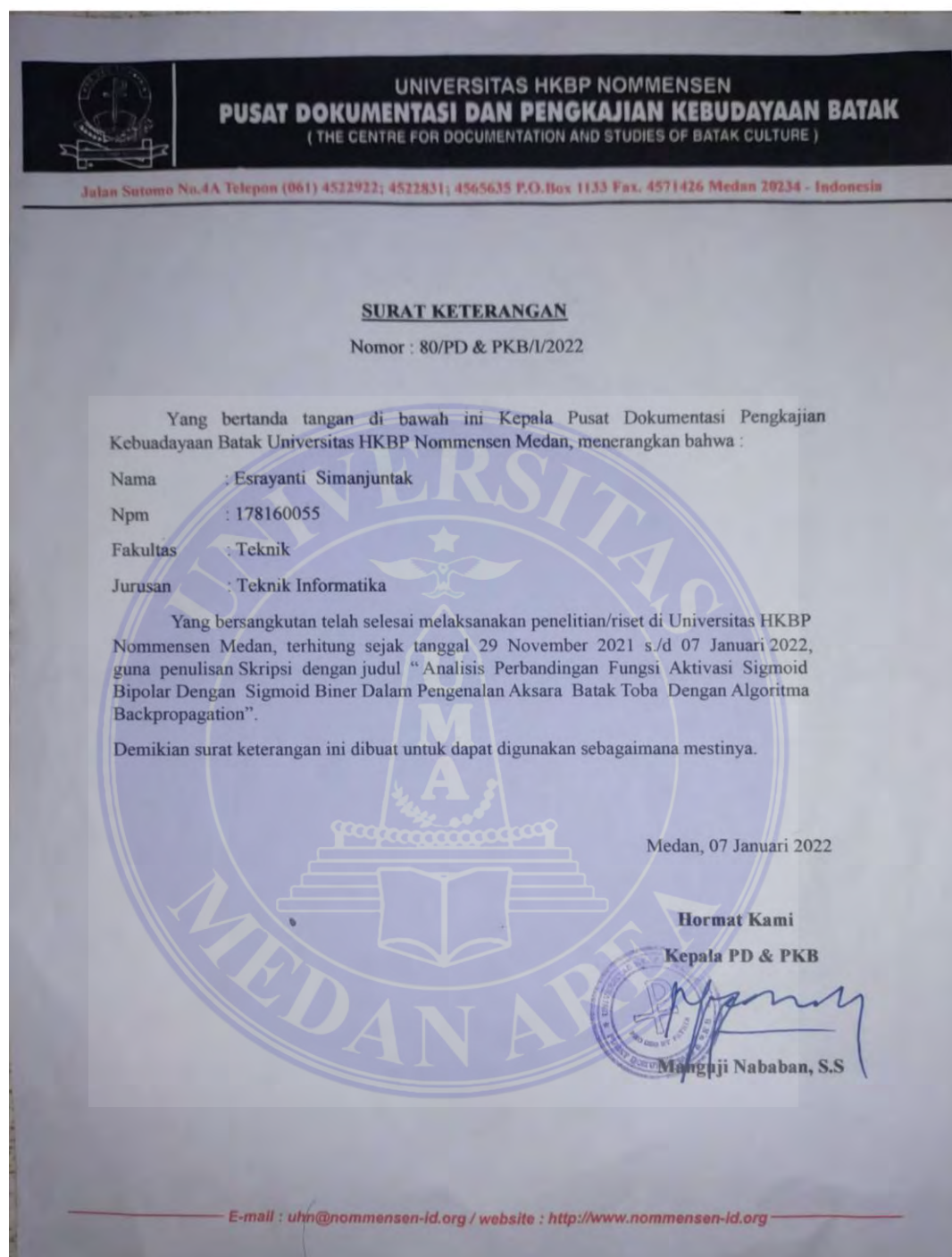
“Analisis Perbandingan Fungsi Aktivasi Sigmoid Biner dengan Sigmoid Bipolar dalam Pengenalan Aksara Batak Toba dengan Algoritma *Backpropagation*”.

SK Pembimbing ini berlaku selama enam bulan terhitung sejak SK ini diterbitkan. Jika proses pembimbing melebihi batas waktu yang telah ditetapkan, SK ini dapat ditinjau ulang.

Demikian kami sampaikan, atas kesediaan saudara diucapkan terima kasih.

Dekan,


Nurul Khairina, S. Kom, M. Kom





The image shows a Turnitin Similarity Report for a document named 'Esraaa 1.docx'. The report includes the following details:

- PAPER NAME:** Esraaa 1.docx
- AUTHOR:** 178160055 Esrayanti Simanjuntak
- WORD COUNT:** 14561 Words
- CHARACTER COUNT:** 78904 Characters
- PAGE COUNT:** 90 Pages
- FILE SIZE:** 2.0MB
- SUBMISSION DATE:** Sep 8, 2022 5:53 PM GMT+7
- REPORT DATE:** Sep 8, 2022 5:58 PM GMT+7

The report indicates a **20% Overall Similarity**. The combined total of all matches, including overlapping sources, for each database is as follows:

- 19% Internet database
- 3% Publications database
- Crossref database
- Crossref Posted Content database
- 9% Submitted Works database

Excluded from Similarity Report:

- Small Matches (Less than 10 words)

A large watermark of Universitas Medan Area is visible in the background of the report.

