

**RANCANG BANGUN APLIKASI PENGENALAN HURUF
HIJAIYAH MENGGUNAKAN METODE NEURAL NETWORK
BACKPROPAGATION**

SKRIPSI



Disusun Oleh :

**IMAM AHMAD ASKALANI SYUKRI
12.18.066**

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2016**

LEMBAR PERSETUJUAN

**RANCANG BANGUN APLIKASI PENGENALAN HURUF HIJAIYAH
MENGUNAKAN METODE NEURAL NETWORK
BACKPROPAGATION**

SKRIPSI

*Disusun dan Diajukan untuk melengkapi dan memenuhi persyaratan guna
mencapai Gelar Sarjana Komputer Strata Satu (S-1)*

Disusun Oleh :

IMAM AHMAD ASKALANI SYUKRI

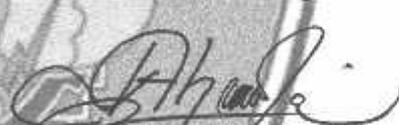
NIM : 12.18.066

Diperiksa dan Disetujui

Dosen Pembimbing I

Dosen Pembimbing II


Yosep Agus Pranoto, ST., MT.
NIP.P. 1031000432


Ahmad Faisal, ST., MT.
NIP.P. 1031000431

Mengetahui

Ketua Program Studi Teknik Informatika S-1


Joseph Dedy Irawan, ST., MT.
NIP. 197404162005011002

**PROGRAM STUDI TEKNIK INFORMATIKA S-1
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2016**

LEMBAR KEASLIAN
PERNYATAAN KEASLIAN SKRIPSI

Saya yang bertanda tangan dibawah ini :

Nama : IMAM AHMAD ASKALANI SYUKRI

NIM : 12.18.066

Program Studi : Teknik Informatika S-1

Fakultas : Fakultas Teknologi Industri

Menyatakan dengan sesungguhnya bahwa Skripsi saya yang berjudul :

**“Rancang Bangun Aplikasi Pengenalan Huruf Hijaiyah Menggunakan
Metode Neural Network Backpropagation”**

Adalah skripsi sendiri bukan duplikasi serta mengutip atau menyadur seluruhnya karya orang lain kecuali dari sumber aslinya.



Malang, 15 Januari 2016

membuat pernyataan



Imam Ahmad Askalani Syukri

Abstrak

Aplikasi pengenalan huruf merupakan salah satu cabang dari implementasi teknologi computer vision. Computer vision merupakan teknologi pengembangan dari pengolahan citra yang nantinya akan menghasilkan sesuatu berupa informasi kepada pengguna. Namun teknologi computer vision harus didukung dengan sebuah metode kecerdasan buatan untuk menunjang performa dan keakuratan dalam pengenalan citranya.

Maka dari itu penulis membuat sebuah aplikasi pengenalan huruf hijaiyah menggunakan metode neural network backpropagation yang merupakan salah satu dari metode kecerdasan buatan. Metode backpropagation merupakan salah satu metode yang paling populer dan banyak digunakan untuk proses pengenalan pola dan proses peramalan yang memiliki tingkat akurasi diatas standar karena menerapkan konsep pembelajaran terlebih dahulu baru kemudian pengujian.

Hasil yang diperoleh dari pembuatan aplikasi ini adalah aplikasi dapat melakukan proses training dan pengujian. Aplikasi ini mampu mengenali data citra huruf hijaiyah yang ditraining dengan tingkat akurasi 100% serta dapat mengenali data citra huruf hijaiyah asing dengan tingkat akurasi 100%. Disamping itu hasil pengujian juga menunjukkan bahwa tingkat akurasi pengenalan citra tidak terpengaruh oleh nilai learning rate yang berubah-ubah.

Kata kunci: *Pengolahan Citra, Computer Vision, Neural Network Backpropagation, Huruf Hijaiyah.*

KATA PENGANTAR

Puji syukur kehadiran Allah Yang Maha Kuasa, karena telah memberikan rahmat dan hidayah-Nya sehingga penyusun dapat menyelesaikan skripsi dengan judul RANCANG BANGUN APLIKASI PENGENALAN HURUF HIJAIYAH MENGGUNAKAN METODE NEURAL NETWORK BACKPROPAGATION sesuai dengan waktu yang ditentukan.

Penyusunan skripsi ini merupakan salah satu persyaratan untuk menyelesaikan program pendidikan Strata Satu (S-1) Teknik Informatika, Fakultas Teknologi Industri di Institut Teknologi Nasional Malang.

Pada penyusunan skripsi ini penyusun mengucapkan terima kasih kepada:

1. Bapak Drs. H. Bakri Umar dan Ibu Hj. Suryani Syahwan yang merupakan kedua orang tua dan pendukung utama dari segi moril maupun materiil.
2. Dr. H. Lalu Mulyadi, ST., MT. selaku Rektor Institut Teknologi Nasional Malang.
3. Ir. Anang Subardi, MT. selaku Dekan Fakultas Teknologi Industri, Institut Teknologi Nasional Malang.
4. Joseph Dedy Irawan, ST., MT. selaku Ketua Program Studi Teknik Informatika S-1, Institut Teknologi Nasional Malang.
5. Sonny Prasetyo, ST., MT. selaku Sekertaris Program Studi Teknik Informatika S-1, Institut Teknologi Nasional Malang.
6. Yosep Agus Pranoto, ST., MT. selaku Dosen Pembimbing I, yang selalu memberikan masukan dan arahan.
7. Ahmad Faisol, ST., MT. selaku Dosen Pembimbing II, yang selalu memberikan masukan dan arahan.
8. Bapak Ali Mahmudi, B. Eng, PhD selaku Kepala Laboratorium Pemrograman dan Rekayasa Perangkat Lunak Teknik Informatika.
9. Asisten Laboratorium Pemrograman dan Rekayasa Perangkat Lunak yang merupakan tempat tinggal kedua dan merupakan tempat bernaung dalam melakukan riset dan bersosialisasi selama masa kuliah dan penyusunan skripsi.

10. Semua Dosen dan Karyawan Program Studi Teknik Informatika S-1 Institut Teknologi Nasional Malang yang telah membantu dalam memberikan masukan selama penyusunan skripsi.
11. Semua teman seperjuangan Mahasiswa Teknik Informatika S-1 yang telah banyak membantu demi terselesaikannya skripsi ini.

Penyusun menyadari bahwa skripsi ini masih jauh dari kata sempurna, oleh karena itu penyusun mengharapkan kritik dan saran dari pembaca sekalian, semoga skripsi ini bisa bermanfaat bagi penulis dan pembaca.

Malang, Januari 2016
Penulis,

(Imam Ahmad Askalani Syukri)

DAFTAR ISI

ABTRAK	iv
KATA PENGANTAR	v
DAFTAR ISI	vii
DAFTAR GAMBAR	ix
DAFTAR TABEL	x
BAB I	1
PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan	3
1.5 Manfaat	3
1.6 Metodologi	3
1.7 Sistematika Penulisan Laporan	4
BAB II	6
LANDASAN TEORI	6
2.1 Computer Vision	6
2.2 Jaringan Syaraf Tiruan (Artificial Neural Network)	6
2.2.1 Backpropagation	9
2.3 Visual Basic .NET	15
2.3 Huruf Hijaiyah	18
BAB III	20
ANALISIS DAN PERANCANGAN	20
3.1 Analisis Sistem	20

3.1.1	Kebutuhan Fungsional.....	20
3.1.2	Kebutuhan Non Fungsional	20
3.1.3	Pengguna Sistem	21
3.1.4	Spesifikasi Kebutuhan Sistem.....	21
3.2	Perancangan Sistem.....	21
3.2.1	Desain Sistem.....	21
3.2.2	Desain Antar Muka Aplikasi	25
BAB IV		27
IMPLEMENTASI DAN PENGUJIAN		27
4.1	Implementasi Sistem	27
4.1.1	Tampilan Aplikasi	27
4.1.2	Tampilan Pelatihan Jaringan Saraf Tiruan	28
4.1.3	Tampilan Input Citra Uji	29
4.1.4	Tampilan Ubah ke Grayscale.....	30
4.1.5	Tampilan Konversi Citra Ke Hitam Putih	31
4.1.6	Tampilan Pengujian Citra	32
4.2	Pengujian	33
4.2.1	Pengujian Keakuratan.....	33
4.2.2	Pengujian Laju Pembelajaran.....	46
BAB V.....		50
PENUTUP.....		50
5.1	Kesimpulan	50
5.2	Saran	50
DAFTAR PUSTAKA		51
LAMPIRAN		52

DAFTAR GAMBAR

Gambar 1.1 30 Jenis Huruf Hijaiyah	1
Gambar 2.1 Proses Pada Computer Vision.....	6
Gambar 2.2 Sel Saraf Biologis.....	7
Gambar 2.3 Algoritma Jaringan Saraf Tiruan Backpropagation.....	10
Gambar 2.4 Arsitektur Jaringan Saraf Tiruan Backpropagation.....	10
Gambar 2.5 Fungsi Sigmoid Biner Dengan Range (0,1).....	15
Gambar 3.1 Diagram Blok Aplikasi	22
Gambar 3.2 Flowchart Aplikasi	23
Gambar 3.3 Flowchart Metode.....	24
Gambar 3.4 Desain Antarmuka Aplikasi	26
Gambar 4.1 Form Utama	27
Gambar 4.2 Tampilan Pelatihan Jaringan Saraf Tiruan 1	28
Gambar 4.3 Tampilan Pelatihan Jaringan Saraf Tiruan 2.....	28
Gambar 4.4 Tampilan Input Citra Uji	29
Gambar 4.5 Tampilan konversi Citra RGB ke Grayscale	30
Gambar 4.6 Tampilan konversi Citra Grayscale ke Hitam Putih.....	31
Gambar 4.7 Tampilan pengujian Citra 1	32
Gambar 4.8 Tampilan pengujian Citra 2	32

DAFTAR TABEL

Tabel 2.1 Keanalogan JST Terhadap Jaringan Saraf Biologis.....	8
Tabel 4.1 Pengujian tingkat keakuratan menggunakan citra yang dilatih	34
Tabel 4.2 Pengujian tingkat keakuratan menggunakan citra baru.....	36
Tabel 4.3 Pengujian tingkat keakuratan menggunakan citra cacat	41
Tabel 4.4 Nilai <i>learning rate</i> (α) = 0.7	46
Tabel 4.5 Nilai <i>learning rate</i> (α) = 0.8	47
Tabel 4.6 Nilai <i>learning rate</i> (α) = 0.9	48

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dewasa ini pengembangan teknologi semakin cepat, segala sesuatu yang dahulunya dilakukan secara manual atau konvensional kini satu per-satu mulai diubah menjadi proses otomatis dengan bantuan teknologi komputer. Hal ini tentu berdasarkan dari keuntungan yang didapat dengan proses otomatis yakni efisiensi, efektivitas, dan kecepatan yang lebih maksimal dibandingkan dengan proses manual. Semua informasi yang dibutuhkan akan lebih mudah diperoleh secara cepat, tepat dan akurat. Dengan sistem yang dijalankan menggunakan teknologi komputer akan memudahkan suatu pekerjaan menjadi lebih baik dan menguntungkan

Al-Qur'an merupakan firman Allah SWT yang sekaligus menjadi kitab suci agama Islam. Al-Qur'an juga digunakan sebagai panduan atau pedoman hidup umat islam diseluruh dunia. Oleh karena itu, setiap umat islam diwajibkan mampu membaca dan memahami isi dari Al-Qur'an. Al-Qur'an tersusun menggunakan Bahasa Arab yang mana hurufnya biasa disebut sebagai huruf hijaiyah. Huruf hijaiyah sendiri terdiri dari 30 huruf, yang terdiri dari:

ا ب ت ث ج ح خ د ذ ر ز س ش
ص ض ط ظ غ غ ف ف ك ل م
ن و ه ل ا ء ي

Gambar 1.1 30 Jenis Huruf Hijaiyah

Mempelajari huruf hijaiyah merupakan langkah awal untuk dapat belajar membaca Al-Qur'an. Di zaman modern saat ini masih banyak umat islam yang belum bisa membaca Al-Qur'an, baik itu anak kecil maupun orang dewasa. Hal ini disebabkan masih banyak muslim yang belum mengenal huruf hijaiyah karena terlambat belajar, baru memeluk agama islam (mualaf) atau sudah malu belajar karena sudah berumur, namun disamping itu mereka juga kesulitan untuk belajar sendiri secara otodidak.

Oleh karena itu, berdasarkan perumusan latar belakang diatas, penulis mencoba memberikan sebuah solusi yakni dengan mengembangkan rekayasa perangkat lunak dengan menggabungkan teknologi *Computer Vision* dan *Neural Network Backpropagation* untuk melakukan pengenalan huruf hijaiyah sebagai sarana memudahkan proses pembelajaran tentang huruf hijaiyah. Aplikasi ini nantinya akan mengolah citra huruf hijaiyah agar menghasilkan output berupa nama huruf hijaiyah beserta cara pelafalan dan pengucapannya. Solusi yang penulis tawarkan ini diharapkan mampu membantu pengguna agar lebih cepat memahami jenis-jenis huruf hijaiyah beserta cara pelafalannya.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan diatas, maka timbul beberapa perumusan masalah, antara lain:

1. Bagaimana cara mengembangkan sebuah aplikasi yang dapat membantu mengenali jenis jenis huruf hijaiyah beserta cara pelafalannya.
2. Bagaimana menerapkan metode *Neural Network Backpropagation* dalam mengenali huruf hijaiyah.
3. Bagaiaman memberikan output yang berupa informasi karakter huruf hijaiyah yang diinputkan ke dalam sistem

1.3 Batasan Masalah

Batasan masalah dalam skripsi ini adalah:

1. Aplikasi ini hanya mengenali 30 jenis huruf hijaiyah dengan font yang sudah ditentukan yakni *Times New Roman* dengan ukuran 17x20pixel.
 2. Aplikasi ini hanya mendeteksi satu persatu huruf hijaiyah (tidak berambung), tidak mengandung harakat, tidak mengandung huruf hiasan (seperti huruf terbalik dan lain lain), memiliki background hitam putih dalam bentuk format .JPG
 3. Hasil output berupa text dan suara pelafalan dari huruf yang dikenali, output suara dikirimkan melalui speaker.
 4. Aplikasi ini disusun menggunakan teknologi *Computer Vision* dengan metode *Neural Network Backpropagation*.
 5. Aplikasi ini dikembangkan menggunakan software *Visual Studio 2013 Ultimate*.
-

1.4 Tujuan

Tujuan yang diharapkan dari pembuatan skripsi ini adalah:

1. Menerapkan teknologi informasi dan komunikasi untuk pengenalan pola atau karakter huruf.
2. Mengembangkan sebuah aplikasi pengolahan citra dan *computer vision* yang mampu mengenali jenis-jenis huruf hijaiyah.
3. Menerapkan konsep *Neural Network Backpropagation* dengan teknologi *Computer Vision* dalam mengenali jenis-jenis huruf hijaiyah.
4. Mampu mengimplementasikan Algoritma *Neural Network Backpropagation* ke dalam bahasa Visual Basic.Net

1.5 Manfaat

Beberapa manfaat yang dapat diperoleh dari proposal ini adalah:

1. Memberikan gambaran teknik pengolahan citra dan *computer vision* untuk pengenalan huruf.
2. Mengembangkan aplikasi *Computer Vision* yang mampu menerapkan metode *Neural Network Backpropagation*.
3. Memberikan informasi berupa text dan suara sebagai output dari aplikasi pengenalan huruf hijaiyah.
4. Memberikan solusi alternatif pembelajaran huruf hijaiyah dengan memanfaatkan teknologi komputer.

1.6 Metodologi

Metodologi yang digunakan dalam penyusunan skripsi ini adalah:

1. Studi Literatur
Penelitian ini dilakukan dengan mempelajari buku-buku baik dari perpustakaan, internet, maupun milik pribadi.
 2. Analisa Kebutuhan
 - a. Hardware
 - 1) Laptop atau Pc Minimum Core i3
 - b. Software
 - 1) Visual Studio 2013 Ultimate
 3. Perancangan Sistem
Meliputi perancangan desain sistem dan desain interface.
-

4. Pembuatan Sistem

Tahap pembuatan sistem ini merupakan implementasi dari desain sistem yang telah direncanakan sebelumnya. Pada tahap ini meliputi tahap pembuatan interface dan fitur yang disediakan di dalam sistem ini.

5. Pengujian dan Analisis

Tahapan pengujian sistem merupakan tahap untuk melakukan pengujian terhadap sistem yang telah dibuat, sekaligus melakukan analisa untuk melakukan perbaikan apabila ada kekurangan dan kesalahan terhadap system yang telah dibuat.

6. Penarikan Kesimpulan

Pada tahap ini akan dibuat kesimpulan tentang sistem yang sudah dibuat berdasarkan perencanaan dan analisa yang sudah dilakukan.

1.7 Sistematika Penulisan Laporan

Sistematika penyusunan skripsi ini dibagi menjadi 5 (lima) bab yaitu pendahuluan, dasar teori, analisa dan perancangan sistem, implementasi dan pengujian, serta penutup.

BAB 1 PENDAHULUAN

Bab ini berisi latar belakang, rumusan masalah, batasan masalah, tujuan, metodologi, dan sistematika penulisan laporan.

BAB 2 DASAR TEORI

Bab ini berisi tentang penjelasan teori-teori sebagai referensi dalam pembuatan aplikasi.

BAB 3 ANALISA DAN PERANCANGAN SISTEM

Bab ini berisi tentang perancangan sistem berupa analisa dan desain sistem. Proses analisa menjelaskan tentang deskripsi sistem, pengguna sistem, dan spesifikasi kebutuhan sistem. Sedangkan pada tahap desain dibagi menjadi desain dasar sistem secara keseluruhan.

BAB 4 IMPLEMENTASI DAN PENGUJIAN

Pada bab ini membahas tentang implementasi dan pengujian sistem yang dibuat secara keseluruhan.

BAB 5 PENUTUP

Berisi tentang kesimpulan dan saran yang didapat setelah melakukan analisa, desain, implementasi dan pengujian terhadap pembuatan sistem.

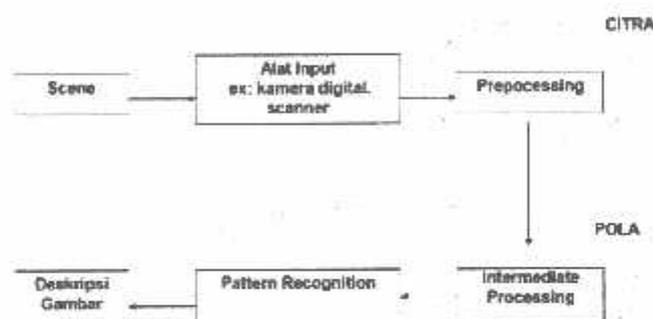
BAB II

LANDASAN TEORI

2.1 Computer Vision

Computer Vision merupakan proses otomatis yang mengintegrasikan sejumlah besar proses untuk persepsi visual, seperti akuisisi citra, pengolahan citra, pengenalan dan membuat keputusan. *Computer Vision* adalah kombinasi antara Pengolahan Citra dan Pengenalan Pola yang hubungan antara ketiganya dapat dilihat pada gambar 2.1. Pengolahan citra merupakan proses awal dari computer vision, sedangkan pengenalan pola merupakan proses menginterpretasikan citra. Ada terdapat 3 proses yang terjadi dalam *computer vision* (Winarno, 2009), yaitu:

- Memperoleh atau mengakuisisi citra digital.
- Operasi pengolahan citra.
- Menganalisis dan menginterpretasi citra dan menggunakan hasil pemrosesan untuk tujuan tertentu, misal memandu robot, mengontrol peralatan, dll.



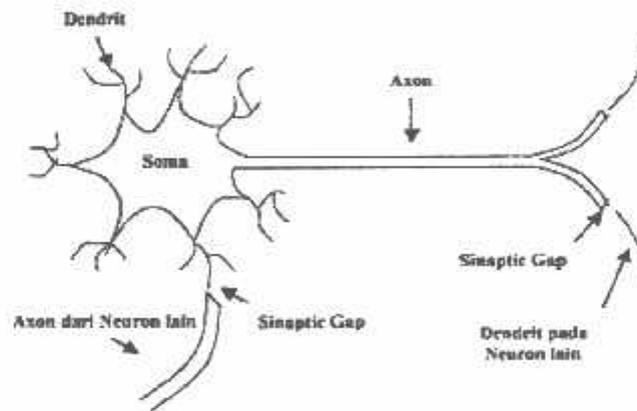
Gambar 2.1 Proses Pada Computer Vision (Sumber : Winarno, Edy. 2009)

2.2 Jaringan Syaraf Tiruan (Artificial Neural Network)

Menurut Puspitaningrum (2006), jaringan saraf tiruan bisa dibayangkan seperti otak buatan dalam cerita fiksi ilmiah. Otak buatan ini dapat berfikir seperti manusia, dan juga sepandai manusia dalam menyimpulkan sesuatu dari potongan-potongan informasi yang diterima. Khayalan manusia tersebut mendorong para peneliti untuk mewujudkannya. komputer diusahakan agar bias berpikir sama seperti cara berpikir manusia. Caranya adalah dengan melakukan peniruan terhadap aktivitas-aktivitas yang terjadi di dalam sebuah jaringan saraf biologis.

Ketika manusia berfikir, aktivitas-aktivitas yang terjadi adalah aktivitas mengingat, memahami, menyimpan, dan memanggil kembali apa yang pernah dipelajari oleh otak. Sesungguhnya apa yang terjadi di dalam otak manusia jauh lebih rumit dari apa yang telah disebutkan di atas. Salah satu contoh pengambilan ide dari jaringan saraf biologis adalah adanya elemen-elemen pemrosesan pada jaringan saraf tiruan yang saling terhubung dan beroperasi secara paralel. Ini meniru jaringan saraf biologis yang tersusun dari sel-sel saraf (neuron). Cara kerja dari elemen-elemen pemrosesan jaringan saraf tiruan juga sama seperti cara neuron meng-encode informasi yang diterimanya (Puspitaningrum, 2006).

Hal yang perlu mendapat perhatian istimewa adalah bahwa jaringan saraf tiruan tidak deprogram untuk menghasilkan keluaran tertentu. Semua keluaran atau kesimpulan yang ditarik oleh jaringan didasarkan pada pengalamannya selama mengikuti proses pembelajaran. Pada proses pembelajaran, ke dalam jaringan saraf tiruan dimasukkan pola-pola input (dan output) lalu jaringan akan diajari untuk memberikan jawaban yang bias diterima (Puspitaningrum, 2006).



Gambar 2.2 Sel Saraf Biologis (sumber: Puspitaningrum, Diah. 2006)

Sebelum membandingkan jaringan saraf biologis dengan jaringan saraf tiruan, sangat baik bila dipahami terlebih dahulu apa yang terjadi di dalam sebuah jaringan saraf biologis. Jaringan saraf biologis merupakan kumpulan dari sel-sel saraf (neuron). Neuron mempunyai tugas mengolah informasi. Komponen-komponen utama dari sebuah neuron dapat dikelompokkan menjadi 3 bagian (Puspitaningrum, 2006), yaitu:

1. *Dendrit*. Dendrit bertugas untuk menerima informasi.

2. *Badan sel (soma)*. Badan sel berfungsi sebagai tempat pengolahan informasi.
3. *Akson (neurit)*. Akson mengirimkan impuls-impuls ke sel saraf lainnya.

(Puspitaningrum, 2006), Jaringan saraf tiruan disusun dengan asumsi yang sama seperti jaringan saraf biologis:

1. Pengolahan informasi terjadi pada elemen-elemen pemrosesan (neuron-neuron).
2. Sinyal antara dua buah neuron diteruskan melalui link-link koneksi.
3. Setiap link koneksi memiliki bobot terasosiasi.
4. Setiap neuron menerapkan sebuah fungsi aktivasi terhadap input jaringan (jumlah sinyal input berbobot). Tujuannya adalah untuk menentukan sinyal output. Fungsi aktivasi yang digunakan biasanya fungsi yang nonlinier.

Adapun cara belajar jaringan saraf tiruan (Puspitaningrum, 2006) sebagai berikut: ke dalam jaringan saraf tiruan diinputkan informasi yang sebelumnya telah diketahui hasil keluarannya. penginputan informasi ini dilakukan leat node-node atau unit-unit input. Bobot-bobot antarkoneksi dalam suatu arsitektur diberi nilai awal dan kemudian jaringan saraf tiruan dijalankan. Bobot-bobot ini bagi jaringan digunakan untuk belajar dan mengingat suatu informasi. Pengaturan bobot dilakukan secara terus-menerus dan dengan menggunakan kriteria tertentu sampai diperoleh keluaran yang diharapkan. Tabel 2.1 memperlihatkan keanalogan antara jaringan saraf tiruan dengan jaringan saraf bilogois.

Tabel 2.1 Keanalogan JST Terhadap Jaringan Saraf Biologis (*sumber: Puspitaningrum, Diah. 2006*)

Jaringan Saraf Tiruan	Jaringan Saraf Biologis
Node atau Unit	Badan sel (soma)
Input	Dendrit
Output	Akson
Bobot	Sinapsis

Hal yang ingin dicapai dengan melatih/mengajari jaringan saraf tiruan adalah untuk mencapai keseimbangan antara kemampuan *memorisasi* dan *generalisasi*. Yang dimaksud dengan kemampuan memorisasi adalah kemampuan jaringan saraf tiruan untuk memanggil kembali secara sempurna sebuah pola yang

telah dipelajari. Kemampuan *generalisasi* adalah kemampuan jaringan saraf tiruan untuk menghasilkan respons yang bisa diterima terhadap pola-pola input yang serupa (namun tidak identik) dengan pola-pola yang sebelumnya telah dipelajari. Hal ini sangat bermanfaat bila pada suatu saat ke dalam jaringan saraf tiruan itu diinputkan informasi baru yang belum pernah dipelajari, maka jaringan saraf tiruan itu masih akan tetap dapat memberikan tanggapan yang baik, memberikan keluaran yang paling mendekati (Puspitaningrum, 2006).

2.2.1 Backpropagation

Metode backpropagation merupakan metode yang sangat baik dalam menangani masalah pengenalan pola-pola kompleks. Metode ini merupakan metode jaringan saraf tiruan yang populer. Beberapa contoh aplikasi yang melibatkan metode ini adalah pengompresian data, pendeteksian virus komputer, pengidentifikasian objek, sintesis suara dari teks, dan lain lain (Puspitaningrum, 2006).

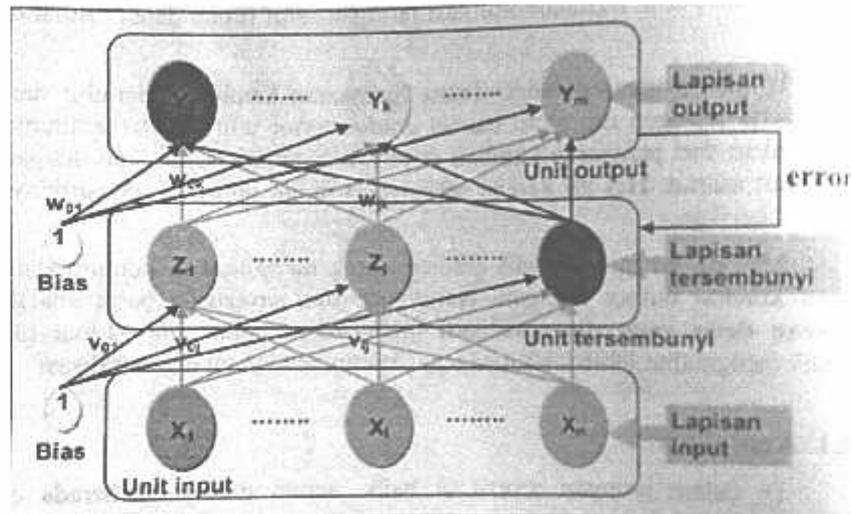
Istilah backpropagation diambil dari cara kerja jaringan ini, yaitu bahwa gradient error unit-unit tersembunyi diturunkan dari penyiaran kembali error-error yang diasosiasikan dengan unit-unit output. Hal ini karena nilai target untuk unit-unit tersembunyi tidak diberikan. Metode ini menurunkan gradient untuk meminimalkan penjumlahan error kuadrat output jaringan. Nama lain dari propagasi balik adalah aturan delta yang digeneralisasi (*generalized delta rule*) (Puspitaningrum, 2006).

Di dalam jaringan backpropagation, setiap unit yang berada di lapisan input terhubung dengan setiap unit yang ada di lapisan tersembunyi. Hal serupa berlaku pula pada lapisan tersembunyi. Setiap unit yang ada di lapisan tersembunyi terhubung dengan setiap unit yang ada di lapisan output. Jaringan saraf tiruan backpropagation terdiri dari banyak lapisan (*multilayer neural networks*) (Puspitaningrum, 2006):

1. Lapisan input (1 buah). Lapisan input terdiri dari neuron-neuron atau unit-unit input, mulai dari unit input 1 sampai unit input n
 2. Lapisan tersembunyi (minimal 1 buah). Lapisan tersembunyi terdiri dari unit-unit tersembunyi mulai dari unit tersembunyi 1 sampai unit tersembunyi p .
-

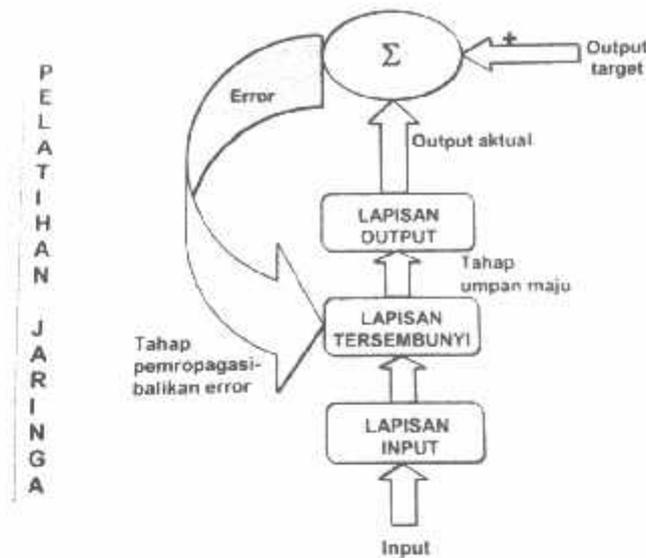
3. Lapisan output (1 buah). Lapisan output terdiri dari unit-unit output mulai dari unit output 1 sampai unit output m .

n , p , m masing-masing adalah bilangan integer sembarang yang menurut arsitektur jaringan saraf tiruan yang dirancang. v_0j dan w_0k masing-masing adalah bias untuk unit tersembunyi ke- j dan untuk unit tersembunyi ke- k . bias v_0j dan w_0k berperilaku seperti bobot dimana output bias ini selalu sama dengan 1. v_{ij} adalah bobot koneksi antara unit ke- i lapisan input dengan unit ke- j lapisan tersembunyi, sedangkan w_{jk} adalah bobot koneksi antara unit ke- i lapisan tersembunyi dengan unit ke- j lapisan output (Puspitaningrum, 2006).



Gambar 2.3 Arsitektur Jaringan Saraf Tiruan Backpropagation (sumber: Puspitaningrum, Diah. 2006)

Alur kerja atau algoritma dari jaringan backpropagation ini dapat dilihat pada gambar dibawah ini:



Gambar 2.4 Algoritma Jaringan Saraf Tiruan Backpropagation (*sumber: Puspitaningrum, Diah, 2006*)

Keterangan berikut mengacu pada gambar 2.4 diatas. Agar dapat digunakan untuk suatu aplikasi, jaringan saraf tiruan perlu 'belajar' terlebih dahulu. Caranya, pada jaringan dimasukkan sekumpulan contoh pelatihan yang disebut set pelatihan. Set pelatihan ini digambarkan dengan sebuah *vector feature* yang disebut vector input yang diasosiasikan dengan sebuah output yang menjadi target pelatihannya. Pelatihan kemudian dilangsungkan dengan maksud membuat jaringan saraf tiruan beradaptasi terhadap karakteristik-karakteristik dari contoh-contoh pada set pelatihan dengan cara melakukan perubahan/pengupdate-an bobot-bobot yang ada dalam jaringan (Puspitaningrum, 2006).

Cara kerja jaringan backpropagation adalah sebagai berikut: mula-mula jaringan diinisialisasi dengan bobot yang diset dengan bilangan acak. Lalu contoh-contoh pelatihan dimasukkan ke dalam jaringan, Contoh pelatihan terdiri dari pasangan vector input dan vector output target. Keluaran dari jaringan berupa sebuah vector output actual. Selanjutnya vector output actual jaringan dibandingkan dengan vector output target untuk mengetahui apakah output jaringan suda sesuai dengan harapan (output actual sudah sama dengan output target) (Puspitaningrum, 2006).

Error yang timbul akibat perbedaan antara output actual dengan output target tersebut kemudian dihitung dan digunakan untuk mengupdate bobot-bobot

relevan dengan jalan mempropagasikan kembali error. Setiap perubahan bobot yang terjadi diharapkan dapat mengurangi besar error. Epoch (siklus setiap pola pelatihan) seperti ini dilakukan pada semua set pelatihan sampai unjuk kerja jaringan mencapai tingkat yang diinginkan atau sampai kondisi berhenti terpenuhi. Yang dimaksud dengan kondisi berhenti di sini misalnya: pelatihan akan dihentikan setelah epoch mencapai 10.000 kali, atau pelatihan akan dihentikan sampai sebuah nilai ambang yang ditetapkan terlampaui. Setelah proses pelatihan selesai, barulah diterapkan algoritma aplikasi. Biasanya sebelum digunakan untuk aplikasi sebenarnya, pengujian unjuk kerja jaringan dilakukan dengan cara memasukkan set pengujian (set tes) ke dalam jaringan. Karena bersifat untuk menguji, set pengujian hanya berupa input saja. Dari respons jaringan dapat dinilai kemampuan memorisasi dan generalisasi jaringan dalam menebak output berdasarkan pada apa yang telah dipelajarinya selama ini (Puspitaningrum, 2006).

Algoritma propagasi balik dapat dibagi ke dalam 2 bagian (Puspitaningrum, 2006):

1. Algoritma pelatihan

Terdiri dari 3 tahap: tahap umpan maju pola pelatihan input, tahap pempropagasibalikkan error, dan tahap pengaturan bobot.

2. Algoritma aplikasi

Yang digunakan hanyalah tahap umpan maju saja.

Berikut adalah algoritma pelatihan untuk jaringan backpropagation dengan sebuah lapisan tersembunyi yang disusun menurut gambar 4.3 dan 4.4 (Puspitaningrum, 2009)

Algoritma pelatihan

0. Inisialisasi bobot

Tentukan angka pembelajaran (α).

Tentukan pula nilai toleransi error atau nilai ambang (bisa menggunakan nilai ambang sebagai kondisi berhenti); atau set maksimal epoch (bila menggunakan banyak epoch sebagai kondisi berhenti).

1. **While** kondisi berhenti tidak terpenuhi **do** langkah ke-2 sampai langkah ke-9.

2. Untuk setiap pasangan pola pelatihan, lakukan langkah ke-3 sampai langkah ke-8.

Tahap umpan maju/feedforward:

3. Setiap unit input x_i (dari unit ke-1 sampai unit ke- n pada lapisan input) mengirimkan sinyal input ke semua unit yang ada di lapisan atasnya (ke lapisan tersembunyi).
4. Setiap unit di lapisan tersembunyi z_j (dari unit ke-1 sampai unit ke- p ; $i=1, \dots, n$; $j=1, \dots, p$) sinyal output lapisan tersembunyinya dihitung menerapkan fungsi aktivasi terhadap penjumlahan sinyal-sinyal input berbobot x_i :

$$z_{net_j} = v_{j0} + \sum_{i=1}^n x_i v_{ji} \quad \dots\dots\dots (2.1)$$

Kemudian dikirim ke semua unit di lapisan atasnya.

5. Setiap unit di lapisan output y_k (dari unit ke-1 sampai unit ke- m ; $i=1, \dots, n$; $k=1, \dots, m$) dihitung sinyal outputnya dengan menerapkan fungsi aktivasi terhadap penjumlahan sinyal-sinyal input berbobot z_j bagi lapisan ini :

$$y_{net_k} = w_{k0} + \sum_{j=1}^p z_j w_{kj} \quad \dots\dots\dots (2.2)$$

Tahap Pemropagasikan Error:

6. Setiap unit output y_k (dari unit ke-1 sampai unit ke- m ; $j=1, \dots, p$; $k=1, \dots, m$) menerima pola target t_k lalu informasi kesalahan lapisan output (δ_k) dihitung. δ_k dikirim ke lapisan di bawahnya dan digunakan untuk menghitung besar koreksi bobot dan bias (Δw_{jk} dan Δw_{0k}) antara lapisan tersembunyi dengan lapisan output:

$$\delta_k = (t_k - y_k) f'(y_{net_k}) = (t_k - y_k) y_k (1 - y_k) \quad \dots\dots\dots (2.3)$$

$$\Delta w_{jk} = \alpha \delta_k z_j \quad \dots\dots\dots (2.4)$$

$$\Delta w_{0k} = \alpha \delta_k \quad \dots\dots\dots (2.5)$$

7. Pada setiap unit di lapisan tersembunyi (dari unit ke-1 sampai unit ke- p ; $i=1, \dots, n$; $j=1, \dots, p$; $k=1, \dots, m$) dilakukan perhitungan informasi kesalahan lapisan tersembunyi (δ_j). δ_j kemudian digunakan untuk menghitung besar

koreksi bobot dan bias (Δv_{ij} dan Δv_{0j}) antara lapisan input dan lapisan tersembunyi.

$$\delta_i = \delta_{net_i} f'(z_{net_i}) = \delta_{net_i} z_i (1 - z_i) \dots\dots\dots (2.6)$$

$$\Delta v_{ij} = \alpha \delta_j x_i \dots\dots\dots (2.7)$$

$$\Delta v_{0j} = \alpha \delta_j \dots\dots\dots (2.8)$$

Tahap Peng-update-an Bobot Dan Bias

8. Pada setiap unit output y_k (dari unit ke-1 sampai unit ke-m) dilakukan peng-update-an bias dan bobot ($j=0, \dots, p; k=1, \dots, m$) sehingga bias dan bobot yang baru menjadi:

$$w_{jk}(\text{baru}) = w_{jk}(\text{lama}) + \Delta w_{jk} \dots\dots\dots (2.9)$$

Dari unit ke-1 sampai unit ke-p di lapisan tersembunyi juga dilakukan peng-update-an pada bias dan bobotnya ($i=0, \dots, n; j=1, \dots, p$):

$$v_{ij}(\text{baru}) = v_{ij}(\text{lama}) + \Delta v_{ij} \dots\dots\dots (2.10)$$

9. Tes kondisi berhenti.

Algoritma Aplikasi:

0. Inisialisasi bobot. Bobot ini diambil dari bobot-bobot terakhir yang diperoleh dari algoritma pelatihan.
1. Untuk setiap vector input, lakukan langkah ke-2 sampai ke-4.
2. Setiap unit input x_i (dari unit ke-1 sampai unit ke-n pada lapisan input; $i=1, \dots, n$) menerima sinyal input pengujian x_i dan menyiarkan sinyal x_i ke semua unit pada lapisan di atasnya (unit-unit tersembunyi):
3. Setiap unit di lapisan tersembunyi (dari unit ke-1 sampai unit ke-p pada lapisan input; $j=1, \dots, p$) menghitung sinyal outputnya dengan menerapkan fungsi aktivasi terhadap penjumlahan sinyal-sinyal input x_i . Sinyal output dari lapisan tersembunyi kemudian dikirim ke semua unit pada lapisan di atasnya:

$$z_{net_j} = v_{j0} + \sum_{i=1}^n v_{ij} x_i \dots\dots\dots (2.11)$$

4. Setiap unit output y_k (dari unit ke-1 sampai unit ke-m; $j=1, \dots, p; k=1, \dots, m$) menghitung sinyal outputnya dengan menerapkan fungsi aktivasi terhadap

penjumlahan sinyal-sinyal input bagi lapisan ini, yaitu sinyal-sinyal input z_j dari lapisan tersembunyi:

$$net_k = w_{k0} + \sum_{j=1}^p x_j w_{kj} \dots\dots\dots (2.12)$$

Fungsi aktivasi

Puspitaningrum (2006) menyatakan ada beberapa pilihan fungsi aktivasi yang digunakan di dalam metode propagasi balik, seperti fungsi sigmoid biner, sigmoid bipolar, dan tangen hiperbolik. Karakteristik yang harus dimiliki fungsi aktivasi tersebut adalah kontinu, diferensiabel, dan tidak menurun secara monoton. Fungsi aktivasi diharapkan jenuh (mendekati nilai-nilai maksimum dan minimum secara asimtot).

Fungsi Aktivas Biner

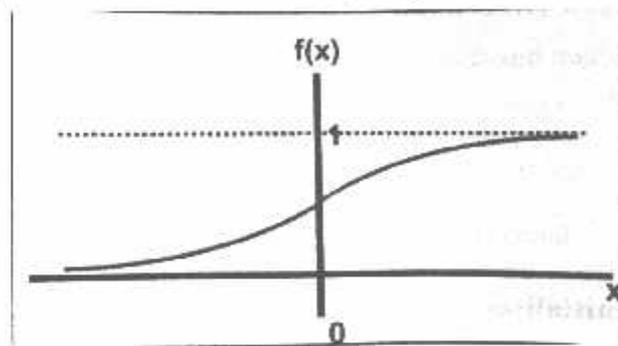
Fungsi ini merupakan fungsi yang umum digunakan (Puspitaningrum, 2006). Range-nya adalah (0,1) dan didefinisikan sebagai:

$$f(x) = 1 / (1 + e^{-x}) \dots\dots\dots (2.13)$$

dengan turunan:

$$f'(x) = f(x) (1 - f(x)) \dots\dots\dots (2.14)$$

fungsi sigmoid biner ini diilustrasikan pada gambar 2.5 dibawah ini:



Gambar 2.5 Fungsi Sigmoid Biner Dengan Range (0,1) (sumber: Puspitaningrum, Diah. 2006)

2.3 Visual Basic .NET

Dahulu ada sebuah Bahasa pemrograman yang diberi nama *Basic* (Beginner's All -Purpose Symbolic Instruction Code). Sesuai dengan namanya, Basic ditujukan sebagai Bahasa yang paling sederhana bagi mereka yang tidak

terlalu familiar dengan dunia pemrograman. Pada tahun 1991 *Microsoft* mengeluarkan Visual Basic, pengembangan dari Basic yang berubah dari sisi pembuatan antarmukanya. Visual basic sampai sekarang masih menjadi salah satu Bahasa pemrograman terpopuler di dunia (Hidayatullah, 2012).

Pada akhir tahun 1999, teknologi .NET diumumkan. Microsoft memosisikan teknologi tersebut sebagai *platform* untuk membangun XML *Web Services*. XML *Web services* memungkinkan aplikasi tipe apa pun dapat berjalan pada sistem komputer dengan tipe manapun dan dapat mengambil data yang tersimpan pada server dengan tipe apapun melalui internet (Hidayatullah, 2012).

Visual basic .NET adalah Visual Basic yang direkayasa kembali untuk digunakan pada platform .NET sehingga aplikasi yang dibuat menggunakan Visual Basic .NET dapat berjalan pada sistem computer apa pun, dan dapat mengambil data dari server dengan tipe apapun asalkan terinstal .NET Framework (Hidayatullah, 2012).

Berikut ini perkembangan Visual Basic .NET (Hidayatullah, 2012):

- a. Visual Basic .NET 2002 (VB 7.0)
- b. Visual Basic .NET 2003 (VB 7.1)
- c. Visual Basic 2005 (VB 8.0)
- d. Visual Basic 2008 (VB 9.0)
- e. Visual Basic 2010 (VB 10.0)
- f. Visual Basic 2012 (VB 11.0)
- g. Visual Basic 2013

Pada umumnya Visual Basic .NET terpaket dalam Visual Studio .NET. pada distribusinya, terdapat berbagai versi visual studio .NET yaitu versi professional, premium dan yang paling lengkap adalah versi ultimate (Hidayatullah, 2012).

Beberapa kelebihan VB .NET antara lain (Hidayatullah, 2012):

1. Sederhana dan mudah dipahami
Seperti pada VB, Bahasa yang digunakan pada VB .NET sangat sederhana sehingga lebih mudah dipahami bagi mereka yang masih awam terhadap dunia pemrograman.
 2. Mendukung GUI
VB .NET bisa membuat software dengan antarmuka grafis yang lebih *User Friendly*
-

3. Menyederhanakan *deployment*

VB .NET mengatasi masalah *deployment* dari aplikasi berbasis windows yaitu DLL Hell dan registrasi COM (*Component Object Model*). Selain itu tersedia wizard yang memudahkan dalam pembuatan *file setup*.

4. Menyederhanakan pengembangan perangkat lunak

Ketika terjadi kesalahan penulisan kode dari sisi sintaks (Bahasa), maka VB .NET langsung menuliskan kesalahannya pada bagian *Message Windows* sehingga *programmer* dapat memperbaiki kode dengan lebih cepat. Editor menu bersifat WYSIWYG (*What You See Is What You Get*). Adanya berbagai wizard yang memandu programmer dalam membuat software. Tersedianya *Crystal Report* (CR) untuk membuat laporan (pada visual studio 2010, *Crystal Report* gratis namun harus diinstal secara terpisah). Adanya *code snippets* yaitu fitur untuk menyisipkan kode-kode koleksi kita pada program yang sedang kita buat. Di atas adalah hal-hal yang membuat pengembangan perangkat lunak menjadi lebih mudah.

5. Mendukung penuh OOP

Memiliki fitur Bahasa pemrograman berorientasi objek seperti *inheritent* (pewarisan), *encapsulation* (pembungkusan), dan *pholymorphism* (banyak bentuk).

6. Mempermudah pengembangan aplikasi berbasis Web

Disediakan desainer form Web, selain itu disediakan layanan Web XML sehingga memungkinkan suatu aplikasi “berkomunikasi” dengan aplikasi lainnya dari berbagai *platform* menggunakan protocol internet terbuka.

7. Migrasi ke VB .NET dapat dilakukan dengan mudah

Jika anda sudah mengembangkan aplikasi di VB, maka konversi ke VB .NET dapat anda jalankan dengan mudah.

8. Banyak digunakan oleh *programmer-programmer* di seluruh dunia. Salah satu keuntungannya adalah jika kita memiliki masalah/pertanyaan, maka kita bisa tanyakan kepada *programmer-programmer* lain di seluruh dunia melalui forum-forum di Internet.

Pada intinya, Visual Basic .NET adalah suatu dari kumpulan *tools* pemrograman yang terdapat pada paket Visual Studio .NET. Pada Visual Studio

.NET terdapat beberapa tools pemrograman lain seperti: Visual C++ .NET, Visual C# .NET, dan Visual J# .NET. Lingkungan pengembangan dari VB .NET disebut dengan .NET Framework. Framework ini menangani bagaimana .NET programming membangun tipe intrinsic, class, dan antarmuka. Pada Desember 2013, NET Framework terbaru adalah versi 4.5.1. komponen paling penting dari framework ini adalah Common Language Runtime (CLR) (Hidayatullah, 2012).

2.3 Huruf Hijaiyah

Huruf Hijaiyah merupakan huruf penyusun kata dalam Al Qur an. Seperti halnya di Indonesia yang memiliki huruf alfabet dalam menyusun sebuah kata menjadi kalimat, huruf hijaiyah juga memiliki peran yang sama. Makna huruf Hijaiyah, dapat diketahui jelas dalam sebuah riwayat. Hakim al-Jarjani meriwayatkan dari sanad Amirul Mukminin Ali bin Abi Thalib. Seorang laki-laki Yahudi mendatangi Rasulullah saw seraya bertanya, "*Apa makna huruf hijaiyah?*" Rasulullah berkata kepada Imam Ali, "*Jawablah pertanyaannya wahai Ali!*" Kemudian Rasulullah saw berdoa, "*Ya Allah, jadikanlah dia berhasil dan batulah dia.*" Amirul Mukminin Ali berkata, "*Setiap huruf hijaiyah adalah nama-nama Allah*" beliau melanjutkan:

Alif: Ismullah (nama Allah), yang tiada Tuhan selain-Nya. Dia maha hidup, Maha mandiri dan Mahakuasa.

Ba` : al-Baqi (Yang Mahakekal), setelah musnahnya makhluk.

Ta` : al-Tawwab (Yang Maha Penerima taubat) dari hamba-hambanya.

Tsa` : al-Tsabit (Yang Menetapkan) keimanan hamba-hamba-Nya.

Jim: Jalla Tsanauhu (Yang Mahatinggi pujian-Nya), kesucian-Nya dan nama-nama-Nya yang tiada terbatas.

Ha: al-Haq, al-Hayyu, wa al-Halim (Yang Mahabener, Mahahidup, dan Mahabijak).

Kha: al-Khabir (Yang Mahatahu) dan maha melihat. Sesungguhnya Allah Mahatahu apa yang kalian kerjakan.

Dal: Dayyanu yaumi al-din (Yang Mahakuasa di hari pembalasan).

Dzal: Dzu al-Jalal wa al-ikram (Pemilik Keagungan dan Kemuliaan)

Ra: al-Rauf (Mahasayang). Dialah pengasih dan Mahapenyayang.

Zai: Zainul Ma`budin (Kebanggaan para hamba).

Sin: al-Sami` al-Bashir (Mahadengar dan Mahalihat).

Syin: Syakur (Maha Penerima ungkapan terima kasih dari hamba-hamba-Nya).

Shad: al-shadiq (Mahajujur) dalam menepati janji dan memberikan ancaman. Sesungguhnya Allah tidak mengingkari janji-Nya.

Dhad: al-Dhar wa al-Nafi (Yang Menangkal bahaya dan Mendatangkan manfaat).

Tha': al-Thahir wa al-Muthahhir (Yang Mahasuci dan Menyucikan).

Dha': al-Dhahir (Yang tampak dan Menampakkan tanda-tanda kebesaran-Nya).

`Ain: al-`Alim (Yang Mahatahu) atas hamba-hamba-Nya dan segala sesuatu.

Ghain: Ghiyats al-Mustaghitsin (Penolong bagi orang-orang yang meminta pertolongan) dan Pemberi perlindungan di setiap masa dan tempat.

Kaf: al-Kafi (Yang Memberikan Kecukupan) bagi setiap makhluk di setiap tempat dan waktu, tiada suatu yang serupa dengan-Nya dan tidak ada yang mampu menandingi-Nya.

Lam: Lathif (Maha Lembut) terhadap hamba-hamba-Nya dengan kelembutan khusus dan kelembutan yang tersembunyi.

Mim: Malik al-dunya wa al-akhirah (Pemilik dunia dan akhirat).

Nun: Nur (Cahaya) langit, cahaya bumi, dan cahaya hati orang-orang yang beriman.

Waw: al-Wahid (Yang Mahaesa) dan tempat bergantung.

Haa': al-Hadi (Maha Pemberi petunjuk) bagi makhluk-Nya. Dialah yang telah menciptakan segala sesuatu dan memberikan petunjuk.

Lam alif: lam tasydid dalam lafadz "Allah" untuk menekankan kekuasaan Allah, yang tidak ada sekutu bagi-Nya.

Ya': Yadullah basithun Lil khalqi (Tangan Allah terbuka bagi makhluk). Maksudnya, kekuasaan dan kekuatan-Nya meliputi semua tempat dan keberadaan di setiap waktu dan ruang (Zainuddin, 2014)

BAB III

ANALISIS DAN PERANCANGAN

3.1 Analisis Sistem

Analisis sistem berguna untuk mengetahui kebutuhan perangkat lunak dan kebutuhan aplikasi yang akan dibangun. Pada tahap ini dilakukan pencarian dan pengumpulan data serta pengetahuan yang diperlukan oleh aplikasi, sehingga pada akhirnya analisis didapatkan hasil berupa sistem yang strukturnya dapat didefinisikan dengan baik dan jelas.

3.1.1 Kebutuhan Fungsional

Aplikasi pendeteksi huruf hijaiyah ini merupakan aplikasi yang dapat mengenali 30 macam huruf hijaiyah menggunakan konsep computer vision dengan menggunakan metode *neural network backpropagation*. Cara kerja aplikasi ini adalah ata citra yang diterima oleh aplikasi akan diolah terlebih dahulu untuk mendapatkan nilai citra yang lebih kongkret sebagai masukan atau input untuk metode *neural network*. Masukan yang diperoleh nantinya diolah oleh metode *neural network* untuk mengenali macam-macam huruf hijaiyah, jika ada huruf hijaiyah yang terdeteksi, maka aplikasi akan menghasilkan output berupa suara pelafalan dari huruf hijaiyah yang terdeteksi. Pada penelitian ini objek yang akan dideteksi berupa huruf hijaiyah cetakan yang sudah dicetak masing-masing pada kertas per huruf. Jadi, pengguna hanya perlu mengarahkan kertas yang sudah berisi cetakan satu huruf hijaiyah ke depan kamera untuk dideteksi dan dikenali. Aplikasi ini diharapkan dapat membantu untuk mengenali 30 macam huruf hijaiyah beserta pelafalannya, khususnya bagi anak usia dini yang akan mempelajari al-qur'an.

3.1.2 Kebutuhan Non Fungsional

Kebutuhan non fungsional merupakan kebutuhan yang merujuk kepada karakteristik sistem yang harus dimiliki. Berikut beberapa kebutuhan non fungsional yang harus dipenuhi:

1. Aplikasi dapat berjalan pada komputer yang memiliki Sistem Operasi Windows 7.

2. Aplikasi mampu mengambil gambar melalui camera dan browse file dari computer atau laptop.
3. Aplikasi mampu mengenali huruf hijaiyah dengan cakupan waktu 3 detik.
4. Aplikasi berjalan secara *offline* dan tidak membutuhkan koneksi internet.

3.1.3 Pengguna Sistem

Pengguna aplikasi pengenalan huruf hijaiyah ini ditujukan kepada pengguna semua umur, baik anak kecil, remaja, dewasa maupun orang tua. Pengguna sistem diharapkan sebelumnya sudah familiar dengan penggunaan komputer.

3.1.4 Spesifikasi Kebutuhan Sistem

Spesifikasi kebutuhan sistem berisi kebutuhan perangkat baik perangkat keras maupun perangkat lunak.

1. Perangkat keras (Hardware)

Perangkat keras yang dibutuhkan adalah sebuah komputer atau laptop dengan webcam VGA dan speaker. Spesifikasi minimal komputer yang digunakan yakni mampu menjalankan semua kebutuhan perangkat lunak yang dibutuhkan.

2. Perangkat lunak (software)

- a. Sistem operasi Windows 7 professional 64-bit
- b. Visual Studio 2013 Ultimate
- c. NetFramework 4.5

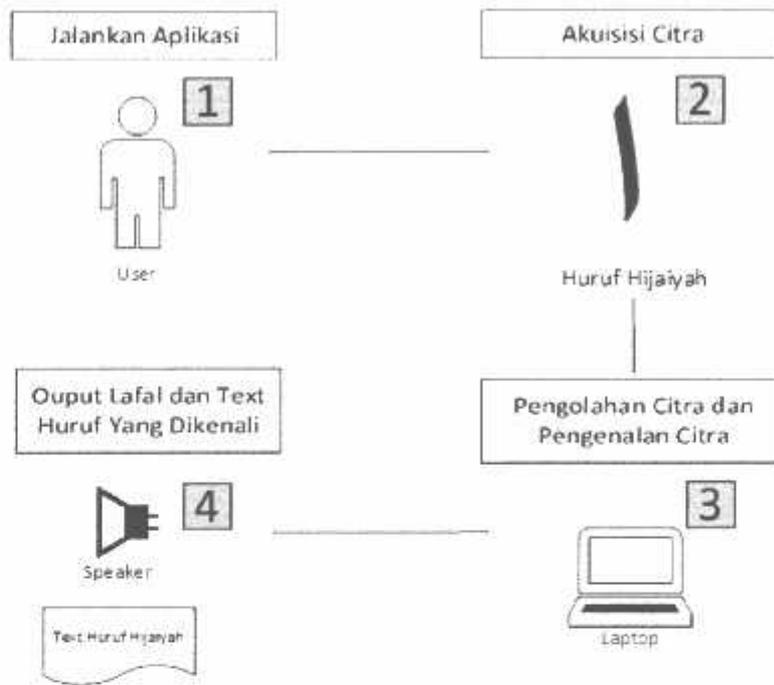
3.2 Perancangan Sistem

Setelah melakukan analisis sistem, berikut adalah perancangan proses. Proses kerja aplikasi pendeteksi huruf hijaiyah ini adalah user mengarahkan huruf hijaiyah yang sudah dicetak pada kertas di depan camera kemudian camera akan mengakuisisi citra yang diterima dan kemudian aplikasi akan mengolah citra tersebut untuk dijadikan input untuk *neural network*, hasil pengenalan citra berupa suara dan text dari huruf yang terdeteksi.

3.2.1 Desain Sistem

Diagram blok berfungsi untuk memudahkan pemahaman dalam alur kerja sistem yang akan dibuat serta memudahkan dalam pemahaman komponen-

komponen yang saling terkait. Adapun diagram blok dari aplikasi pendeteksi huruf hijaiyah ini adalah:



Gambar 3.1 Diagram Blok Aplikasi

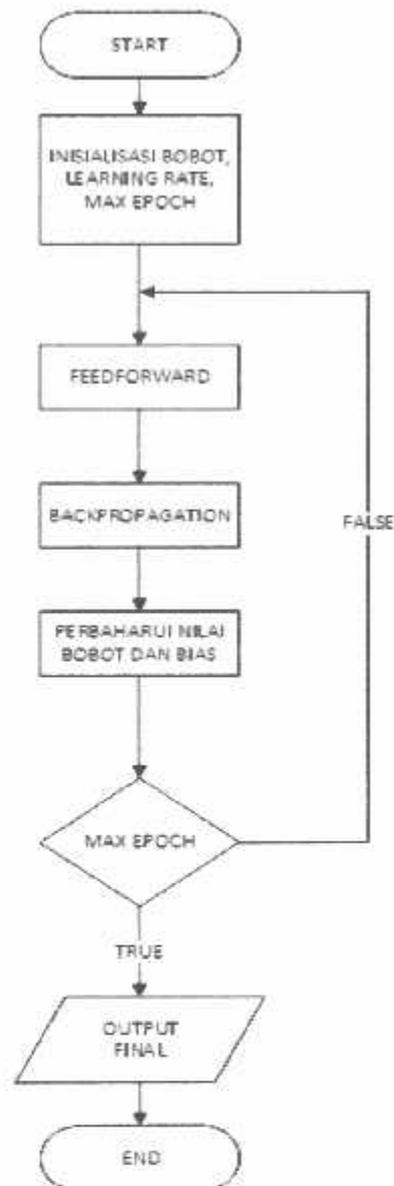
Seperti yang ditunjukkan oleh gambar 3.1, langkah pertama yang dilakukan adalah user menjalankan aplikasi terlebih dahulu, kemudian user melakukan proses akuisisi citra dengan cara menginput citra dari penyimpanan komputer. Setelah itu citra yang sudah diakuisisi diolah terlebih dahulu kemudian baru dilakukan pengenalan citra. Jika huruf sudah dikenali, output dari aplikasi berupa suara pelafalan huruf dan text huruf yang dikenali. Suara yang dihasilkan disampaikan melalui speaker. Berikut adalah flowchart sistem secara umum:



Gambar 3.2 Flowchart Aplikasi

Berdasarkan gambar 3.2 di atas, dapat dilihat bahwa proses jalannya aplikasi dimulai dari langkah pelatihan JST, karena di dalam program sudah terdapat 90 sampel data citra yang akan dilatih. Setelah proses pelatihan JST selesai, maka langkah selanjutnya adalah mengakuisisi citra huruf hijaiyah yang akan dikenali. Citra huruf hijaiyah yang sudah diakuisisi kemudian dilakukan proses pengolahan citra, pada proses ini dilakukan tiga sub-proses yakni konversi citra RGB ke Grayscale, Grayscale ke Hitam Putih, ubah ukuran citra menjadi 17x20pixel, kemudian citra tersebut dikonversi menjadi biner yang bernilai 0 dan 1. Setelah dihasilkan citra biner, proses selanjutnya adalah proses pengujian citra, hasil dari pengujian pengenalan citra yakni berupa suara pelafalan dan text dari huruf yang terdeteksi.

Proses berjalannya metode *Backpropagation* dapat dilihat pada gambar dibawah ini:



Gambar 3.3 Flowchart Metode Backpropagation

Dari gambar 3.3 diatas, dapat dilihat bahwa ilustrasi proses berjalannya pelatihan dengan metode *Backpropagation* adalah sebagai berikut:

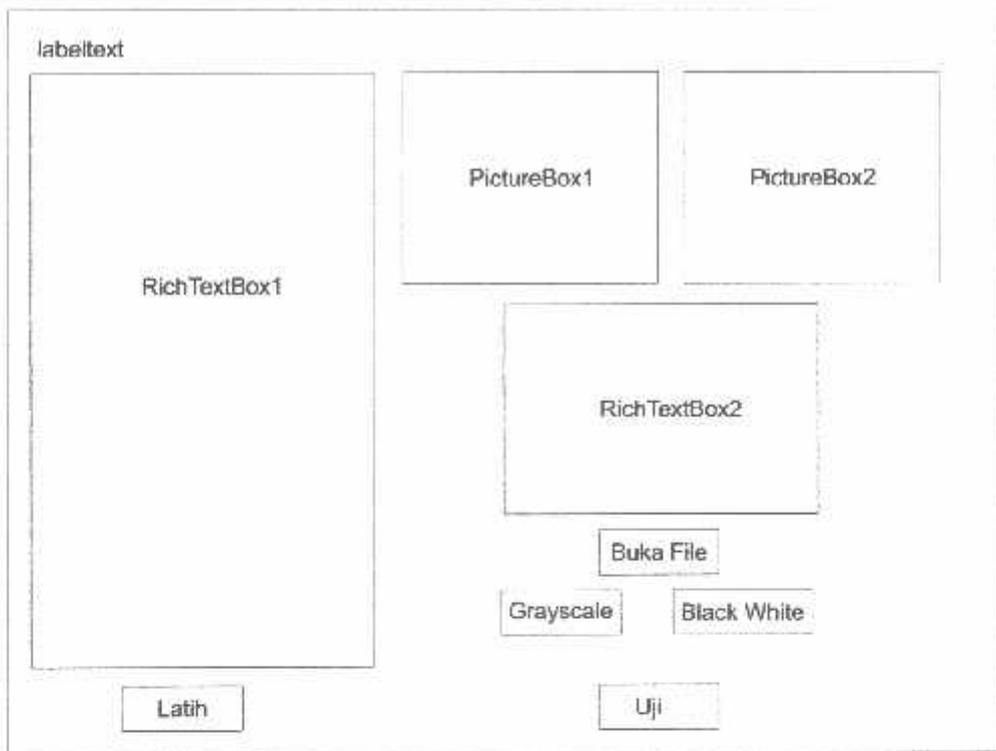
1. Inisialisasi nilai bobot dan bias (*random*), learning rate dan max epoch.
2. Feedforward:
 - a. Tiap unit input akan mengirimkan sinyal ke hidden layer. Tiap unit hidden layer menghitung outputnya menggunakan fungsi aktivasi

terhadap penjumlahan sinyal-sinyal input berbobot. Kemudian mengirim ke unit di atasnya.

- b. Tiap unit hidden akan mengirimkan sinyal ke output layer. Tiap unit output layer menghitung outputnya menggunakan fungsi aktivasi terhadap penjumlahan sinyal-sinyal hidden berbobot.
3. **Backpropagation:**
 - a. Setiap unit output layer menerima pola target dan kemudian menghitung informasi nilai error. Nilai error dikirim ke lapisan dibawahnya (hidden unit) dan digunakan untuk menghitung besar koreksi bobot dan bias antara lapisan tersembunyi dengan lapisan output
 - b. Setiap unit hidden layer dilakukan perhitungan informasi error lapisan tersembunyi. Nilai error kemudian digunakan untuk menghitung besar koreksi bobot dan bias antara lapisan input dan lapisan tersembunyi
 4. **Perbaharui bobot dan bias:**
 - a. Setiap unit output dilakukan perubahan bobot dan bias
 - b. Setiap unit hidden dilakukan perubahan bobot dan bias
 5. Tes kondisi berhenti, jika sudah memenuhi max epoch maka proses pelatihan berhenti. Jika tidak, lakukan kembali proses feedforward dan backpropagation serta peruban bobot dan bias.

3.2.2 Desain Antar Muka Aplikasi

Aplikasi pendeteksi huruf hijaiyah disini didesain sangat sederhana, yakni hanya menggunakan satu form utama tanpa ada tambahan form yang lain. Form utama pada aplikasi ini berisi dua buah RichTextBox, dua buah PictureBox, 5 buah tombol, dan satu buah labelText. Berikut adalah tampilan dari desain antarmuka aplikasi:



Gambar 3.4 Desain Antarmuka Aplikasi

Dari gambar 3.3 diatas, berikut adalah penjelasannya: *LabelText* berfungsi untuk menampilkan hasil atau stats dari proses training. *RichTextbox1* berfungsi untuk menampilkan output dari hasil training, sedangkan *RichTexBox2* yang satunya berfungsi menampilkan nilai biner dari citra huruf hijaiyah yang diuji. *PictureBox1* berfungsi menampung citra yang diambil dari penyimpanan komputer sekaligus menampilkan citra hasil konversi grayscale, sedangkan *PictureBox2* berfungsi menampilkan citra hasil konversi dari grayscale ke Hitam Putih (*black white*). Tombol *Buka File* berfungsi untuk mengambil file citra dari komputer, tombol *Grayscale* berfungsi mengkonversi citra RGB menjadi Grayscale, tombol *Black White* berfungsi mengkonversi citra dari Grayscale ke Hitam Putih, dan tombol *Uji* berfungsi untuk menjalankan proses pengujian sistem.

BAB IV

IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi Sistem

Implementasi sistem adalah proses menerapkan rancangan sistem yang telah dibuat agar bisa dijalankan pada kenyataannya. Implementasi sistem ini dapat mengetahui sejauh mana keberhasilan dari rancangan yang telah dibuat. Interface dari aplikasi pengenalan huruf hijaiyah ini terdiri satu form utama saja. Didalam form utama terdapat beberapa atribut, diantaranya adalah *label*, *Richtextbox*, *picture box* dan *button*. Aplikasi ini menggunakan 1 layer input, 1 layer hidden dan 1 layer output. Dalam layer input terdapat 340 *neuron*, 30 *neuron* pada layer hidden dan 30 *neuron* pada layer output. Nilai *learning rate* yang digunakan adalah 0.9, *maximum epoch* sebanyak 10.000 dan fungsi aktivasi yang digunakan adalah *Sigmoid Biner*.

4.1.1 Tampilan Aplikasi

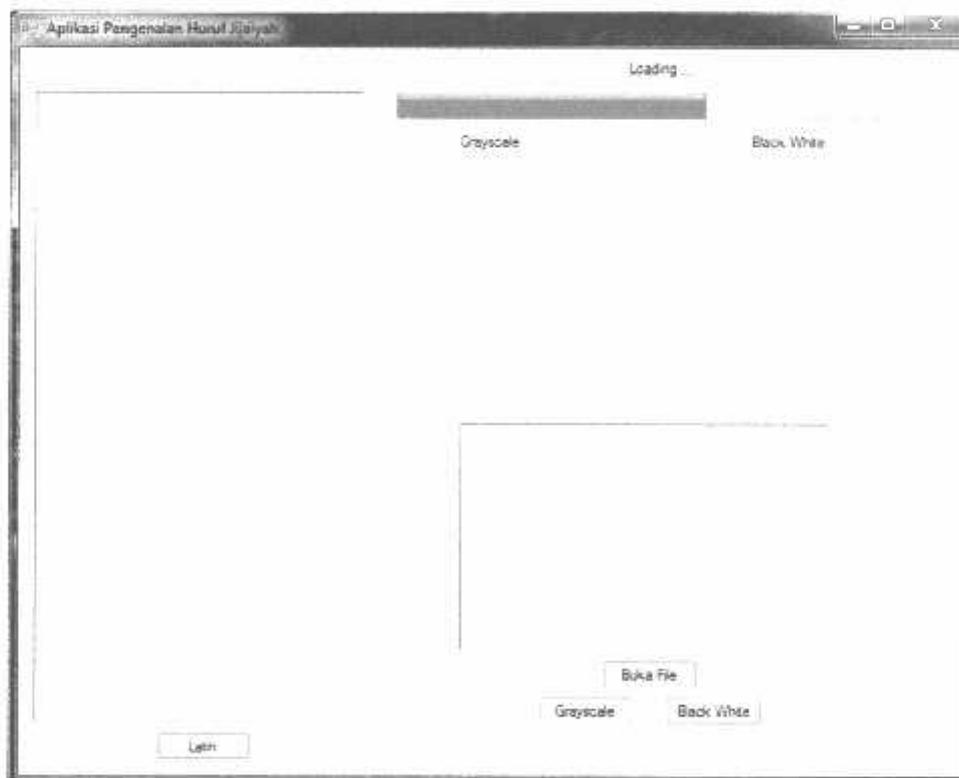


Gambar 4.1 Form Utama

Pada form utama diatas terdapat dua buah *RichTextBox* yang berfungsi untuk menyimpan data hasil training dan data biner dari citra yang akan diuji. Pada form juga terdapat dua buah *picture box* berukuran 170x200 pixel untuk

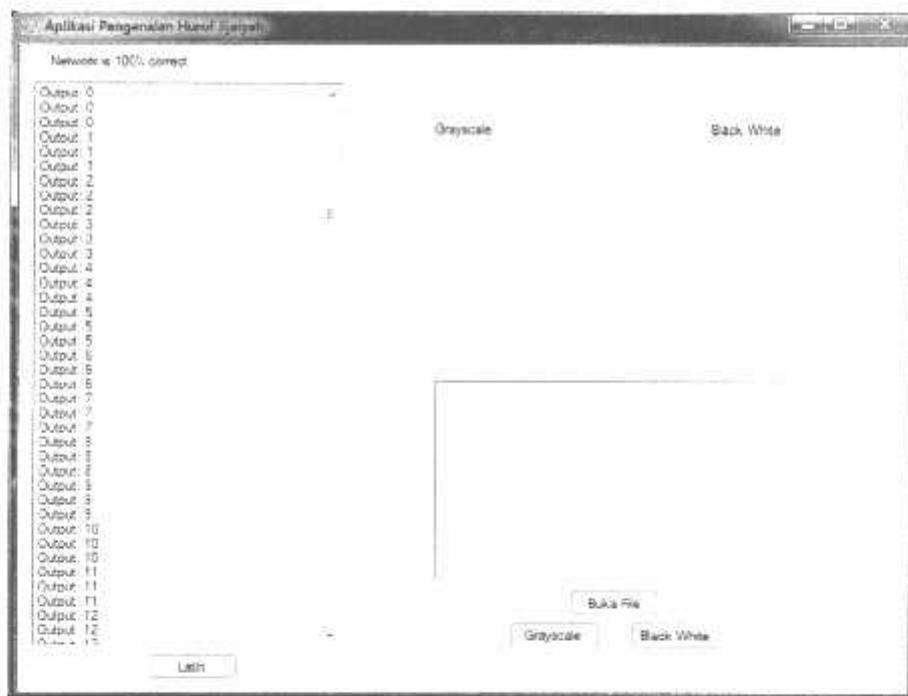
menampung citra gambar yang dipilih untuk diuji. Selain itu juga terdapat beberapa tombol, yakni tombol *Latih* berfungsi untuk melatih jaringan saraf tiruan, tombol *Uji* untuk melakukan pengujian data citra yang baru, tombol *Buka File* yang berfungsi untuk mengambil file citra dari penyimpanan komputer, tombol *Grayscale* berfungsi mengkonversi citra yang ada di *picturebox1* menjadi citra grayscale dan tombol *Black White* berfungsi untuk mengkonversi citra yang ada di *picturebox1* menjadi citra hitam putih sekaligus mengubahnya menjadi citra biner. Citra biner yang didapatkan ditampug pada *RichTextBox2*, sedangkan citra hitam putihnya diletakkan pada *picturebox2*.

4.1.2 Tampilan Pelatihan Jaringan Saraf Tiruan



Gambar 4.2 Tampilan Pelatihan Jaringan Saraf Tiruan 1

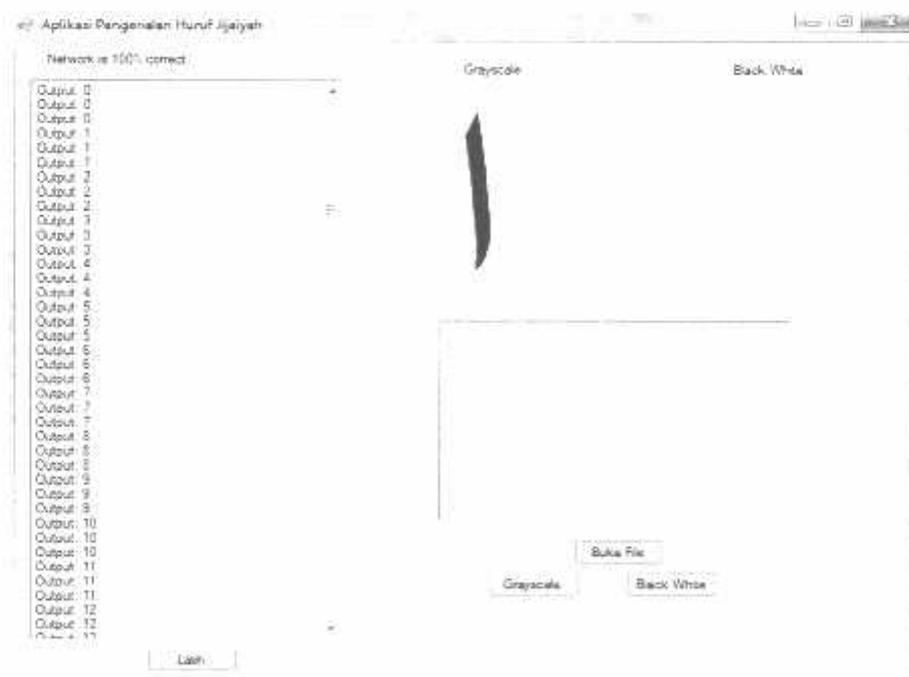
Pada gambar 4.2 terlihat bahwa ketika menekan tombol latih, aplikasi akan menjalankan proses pelatihan jaringan saraf tiruan. Proses akan terus berjalan hingga progress bar berhenti atau menghilang. Ketika proses sudah selesai, maka akan muncul *message box*, dan hasil dari pelatihan jaringan saraf tiruan akan ditampilkan pada *ReachTextBox1*, seperti terlihat pada gambar 4.3 berikut:



Gambar 4.3 Tampilan Pelatihan Jaringan Saraf Tiruan 2

Pada gambar 4.3 dapat dilihat bahwa setelah melakukan proses pelatihan, hasil dari pelatihan jaringan saraf tiruan akan ditampilkan pada *ReachTextBox1*, dan persentasi dari keberhasilan pelatihan akan ditampilkan pada *labeltext1* di bagian pojok kiri atas form utama.

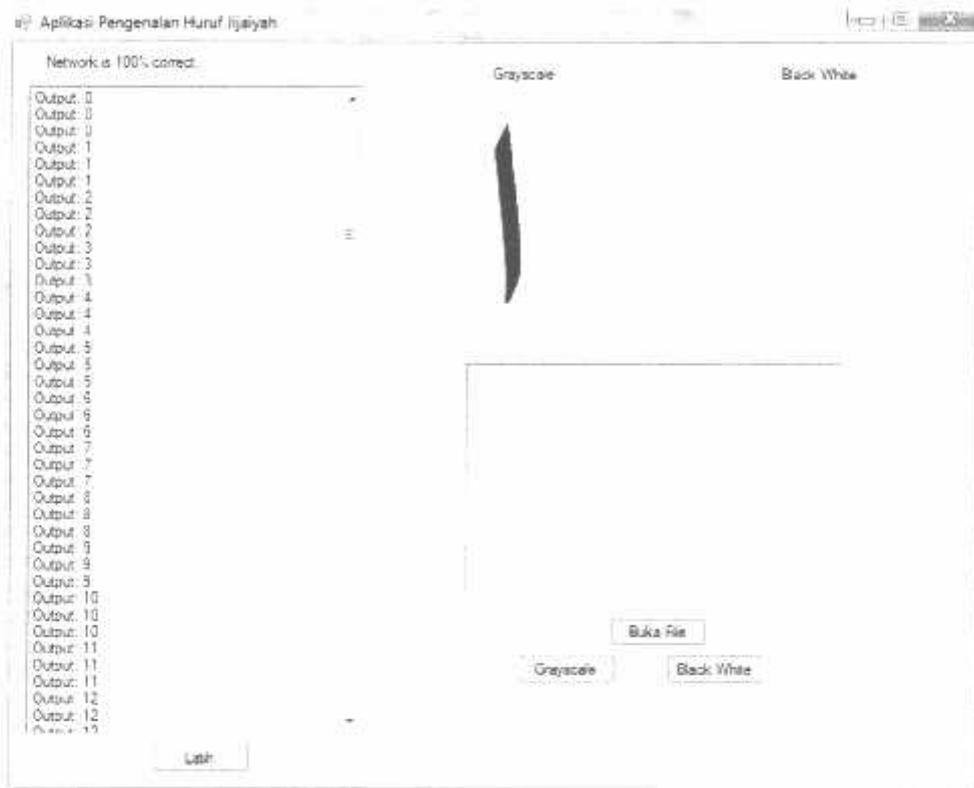
4.1.3 Tampilan Input Citra Uji



Gambar 4.4 Tampilan Input Citra Uji

Pada gambar 4.4 dapat dilihat ketika menekan tombol *Open File*, kita dapat memilih citra huruf hijaiyah yang akan diuji. Citra yang dipilih tadi diletakkan pada *picturebox1* yang terletak disebelah kanan atas form utama. Ukuran citra yang dimasukkan harus berukuran 170x200 pixel.

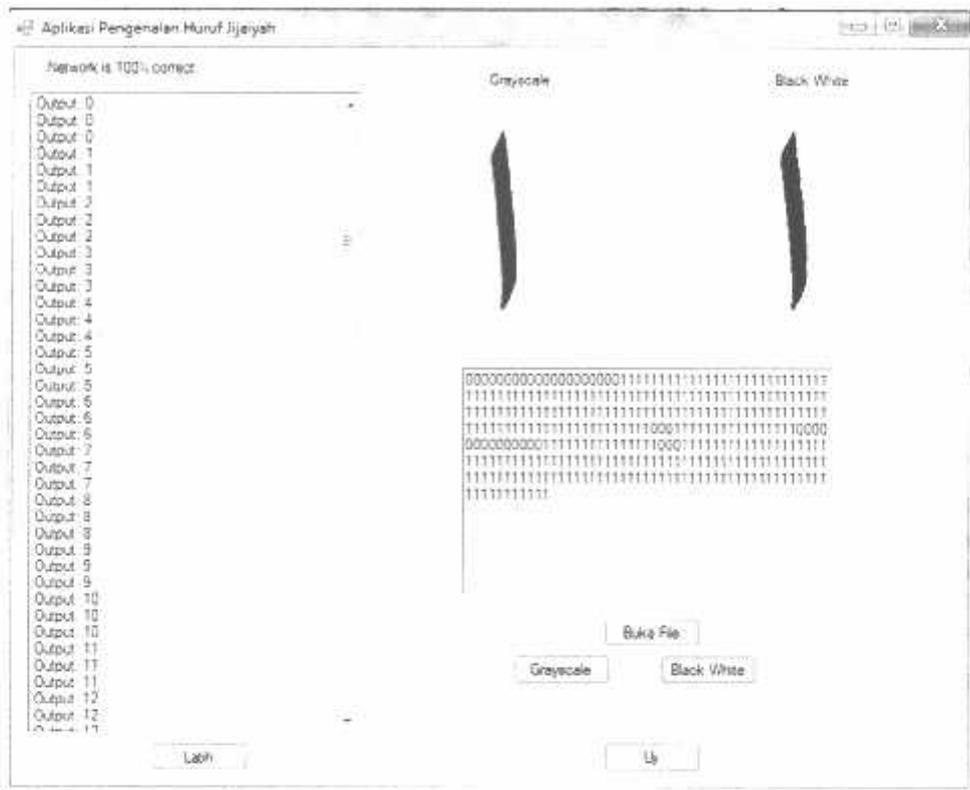
4.1.4 Tampilan Ubah ke Grayscale



Gambar 4.5 Tampilan konversi citra RGB ke Grayscale

Pada gambar 4.5 terdapat tampilan konversi citra hasil inputan dari RGB menjadi citra grayscale. Karena citra asli sudah berwarna hitam putih oleh karena itu tampilan grayscale tidak terlihat. Hasil konversi citra diletakkan di *Picturebox1*.

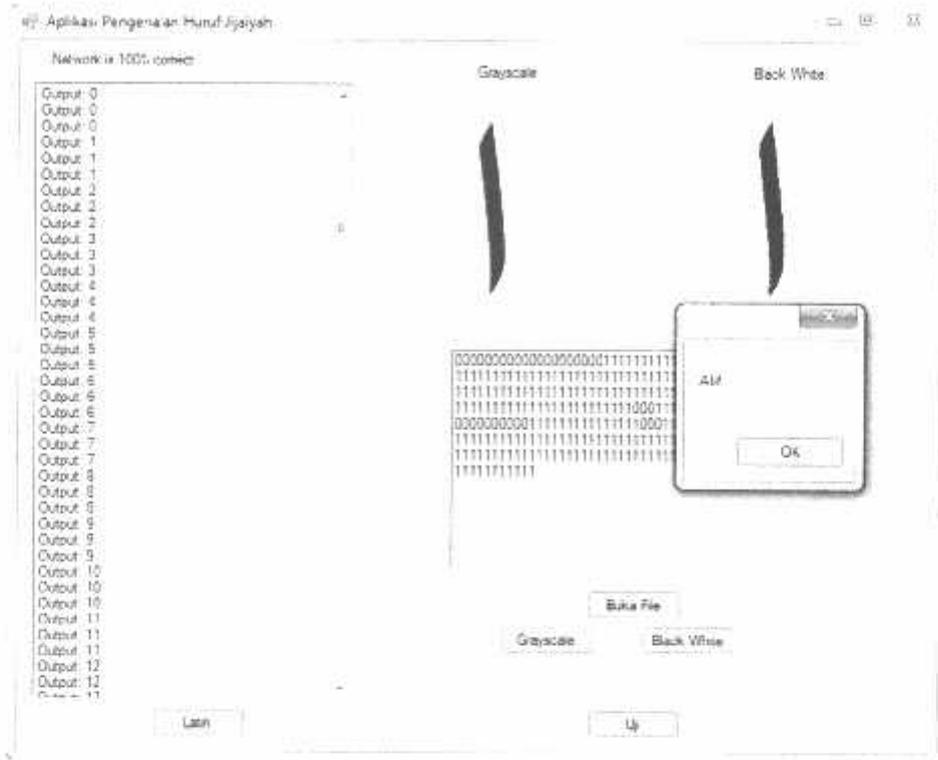
4.1.5 Tampilan Konversi Citra Ke Hitam Putih



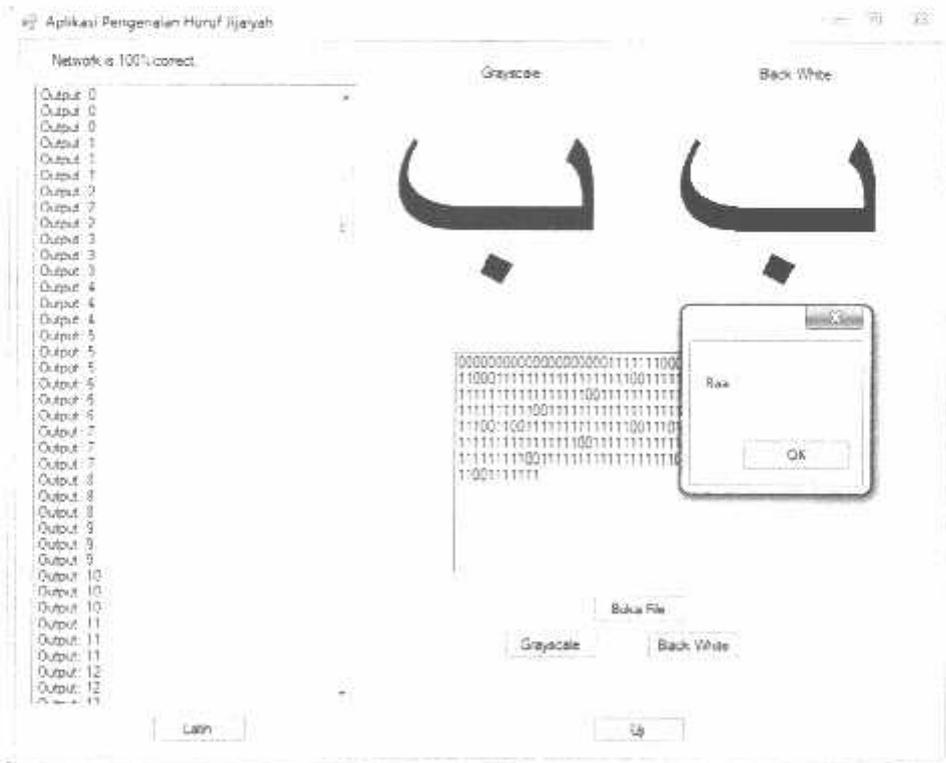
Gambar 4.6 Tampilan Konversi Citra Grayscale ke Hitam Putih

Pada gambar 4.6 dapat dilihat bahwa ketika melakukan konversi citra menjadi hitam putih, citra pada *picturebox1* akan dikonversi menjadi citra hitam putih yang bernilai 0 dan 255 kemudian diletakkan pada *picturebox2*. Setelah itu, citra yang sudah diubah menjadi hitam putih tadi diubah ukurannya menjadi 17x20 pixel. Setelah diubah ukurannya, citra yang bernilai 0 dan 255 diubah lagi menjadi citra bernilai biner yakni 0 dan 1. Data biner yang didapatkan kemudian ditampung ke dalam *ReachTextBox2* (sebelah kanan) dan data biner inilah yang digunakan sebagai data inputan untuk melakukan pengujian pengenalan huruf hijaiyah dengan total 340 bit biner data.

4.1.6 Tampilan Pengujian Citra



Gambar 4.7 Tampilan Pengujian Citra 1



Gambar 4.8 Tampilan Pengujian Citra 2

Untuk melakukan pengujian pengenalan citra, dimasukkan citra yang akan diuji. Pada aplikasi ini terdapat 30 jenis huruf yang akan dikenali, yakni huruf alif, ba, ta, tsa sampai huruf ya. Masing masing huruf memiliki 3 buah sampel data untuk dimasukkan sebagai inputan training jaringan. Sehingga dihasilkan 90 buah data training. Untuk pengujian disediakan dua buah citra pada masing-masing huruf sehingga terdapat 60 buah citra asing yang akan dikenali.

Pada gambar 4.6 dapat dilihat bahwa inputan citra yang akan dikenali merupakan huruf alif. Setelah dilakukan pengolahan citra dari RGB ke grayscale, kemudian dari grayscale menjadi black and white dan kemudian di konversi ke biner maka citra huruf alif sudah siap dikenali oleh aplikasi. Untuk melakukan pengujian yakni dengan menekan tombol Uji, setelah itu akan muncul hasilnya berupa suara pelafalan huruf dan message box yang berisi nama atau tulisan huruf hijaiyah yang dikenali. Untuk pengujian pertama output yang dihasilkan adalah huruf alif, dan untuk pengujian kedua output yang dihasilkan adalah huruf ba seperti terlihat pada gambar 4.7 dan gambar 4.8.

4.2 Pengujian

Pengujian dilakukan untuk mengetahui tingkat keakuratan aplikasi dan pengaruh laju pembelajaran dalam mengenali tanda tangan. Dalam pengujian ini di ambil 150 sampel huruf hijaiyah dari 30 huruf, 90 sampel data huruf hijaiyah untuk dilatih dan 60 data huruf hijaiyah untuk diujikan.

4.2.1 Pengujian Keakuratan

Pengujian keakuratan aplikasi pengenalan huruf hijaiyah dengan menggunakan metode *Neural Network Backpropagation* ini dilakukan untuk mengetahui hasil sistem yang telah dibangun menggunakan metode *Backpropagation*. Pada tahap ini, proses pengujian akan dilakukan dengan memasukkan data citra tanda tangan yang terbagi menjadi 2 yaitu citra yang dilatih dan citra yang diuji. Data-data tersebut diambil dari font hijaiyah Times New Roman. Dari 30 data huruf hijaiyah akan dibagi menjadi 2 bagian yaitu 90 sampel huruf hijaiyah untuk training dan 60 buah huruf hijaiyah untuk diuji. Pada tabel 4.1 adalah tingkat keakuratan citra yang dilatih, table 4.2 adalah tingkat keakuratan citra asing yang diuji, table 4.3 adalah tingkat keakuratan citra yang diberi cacat.

Tabel 4.1 Pengujian tingkat keakuratan menggunakan citra yang dilatih

Nama Huruf	Huruf Hijaiyah	Perhitungan	Tingkat Keakuratan
Alif	ا ا ا	$\frac{\sum \text{Benar}}{\sum \text{Data}} \times 100\% = \frac{3}{3} \times 100\%$	100%
Ba	ب ب ب	$\frac{\sum \text{Benar}}{\sum \text{Data}} \times 100\% = \frac{3}{3} \times 100\%$	100%
Ta	ت ت ت	$\frac{\sum \text{Benar}}{\sum \text{Data}} \times 100\% = \frac{3}{3} \times 100\%$	100%
Tsa	ث ث ث	$\frac{\sum \text{Benar}}{\sum \text{Data}} \times 100\% = \frac{3}{3} \times 100\%$	100%
Jim	ج ج ج	$\frac{\sum \text{Benar}}{\sum \text{Data}} \times 100\% = \frac{3}{3} \times 100\%$	100%
Ha'	ح ح ح	$\frac{\sum \text{Benar}}{\sum \text{Data}} \times 100\% = \frac{3}{3} \times 100\%$	100%
Kha	خ خ خ	$\frac{\sum \text{Benar}}{\sum \text{Data}} \times 100\% = \frac{3}{3} \times 100\%$	100%
Dal	د د د	$\frac{\sum \text{Benar}}{\sum \text{Data}} \times 100\% = \frac{3}{3} \times 100\%$	100%
Dzal	ذ ذ ذ	$\frac{\sum \text{Benar}}{\sum \text{Data}} \times 100\% = \frac{3}{3} \times 100\%$	100%
Ra	ر ر ر	$\frac{\sum \text{Benar}}{\sum \text{Data}} \times 100\% = \frac{3}{3} \times 100\%$	100%
Zae	ز ز ز	$\frac{\sum \text{Benar}}{\sum \text{Data}} \times 100\% = \frac{3}{3} \times 100\%$	100%
Sin	س س س	$\frac{\sum \text{Benar}}{\sum \text{Data}} \times 100\% = \frac{3}{3} \times 100\%$	100%
Syuin	ش ش ش	$\frac{\sum \text{Benar}}{\sum \text{Data}} \times 100\% = \frac{3}{3} \times 100\%$	100%
Shad	ص ص ص	$\frac{\sum \text{Benar}}{\sum \text{Data}} \times 100\% = \frac{3}{3} \times 100\%$	100%
Dhad	ض ض ض	$\frac{\sum \text{Benar}}{\sum \text{Data}} \times 100\% = \frac{3}{3} \times 100\%$	100%
Tha	ط ط ط	$\frac{\sum \text{Benar}}{\sum \text{Data}} \times 100\% = \frac{3}{3} \times 100\%$	100%

Zho	ظ ظ ظ	$\frac{\sum \text{Benar}}{\sum \text{Data}} \times 100\% = \frac{3}{3} \times 100\%$	100%
'ain	ع ع ع	$\frac{\sum \text{Benar}}{\sum \text{Data}} \times 100\% = \frac{3}{3} \times 100\%$	100%
Ghain	غ غ غ	$\frac{\sum \text{Benar}}{\sum \text{Data}} \times 100\% = \frac{3}{3} \times 100\%$	100%
Fa	ف ف ف	$\frac{\sum \text{Benar}}{\sum \text{Data}} \times 100\% = \frac{3}{3} \times 100\%$	100%
Qaf	ق ق ق	$\frac{\sum \text{Benar}}{\sum \text{Data}} \times 100\% = \frac{3}{3} \times 100\%$	100%
Kaf	ك ك ك	$\frac{\sum \text{Benar}}{\sum \text{Data}} \times 100\% = \frac{3}{3} \times 100\%$	100%
Lam	ل ل ل	$\frac{\sum \text{Benar}}{\sum \text{Data}} \times 100\% = \frac{3}{3} \times 100\%$	100%
Mim	م م م	$\frac{\sum \text{Benar}}{\sum \text{Data}} \times 100\% = \frac{3}{3} \times 100\%$	100%
Nun	ن ن ن	$\frac{\sum \text{Benar}}{\sum \text{Data}} \times 100\% = \frac{3}{3} \times 100\%$	100%
Waw	و و و	$\frac{\sum \text{Benar}}{\sum \text{Data}} \times 100\% = \frac{3}{3} \times 100\%$	100%
Ha	ه ه ه	$\frac{\sum \text{Benar}}{\sum \text{Data}} \times 100\% = \frac{3}{3} \times 100\%$	100%
Lamalif	لا لا لا	$\frac{\sum \text{Benar}}{\sum \text{Data}} \times 100\% = \frac{3}{3} \times 100\%$	100%
Hamzah	ء ء ء	$\frac{\sum \text{Benar}}{\sum \text{Data}} \times 100\% = \frac{3}{3} \times 100\%$	100%
Ya	ي ي ي	$\frac{\sum \text{Benar}}{\sum \text{Data}} \times 100\% = \frac{3}{3} \times 100\%$	100%

Berdasarkan table 4.1, terdapat tiga buah data citra yang sebelumnya digunakan sebagai inputan pelatihan JST. Perbedaan dari ketiga citra tersebut terletak pada tingkat kemiringannya, yakni 0° , 2° dan 4° . Tabel 4.1 menunjukkan bahwa tingkat keakuratan pengujian JST menggunakan 3 buah data citra yang dilatih sebesar 100%.

Tabel 4.2 Pengujian tingkat keakuratan menggunakan citra baru

Nama Huruf	Huruf Hijaiyah	Perhitungan	Tingkat Keakuratan
Alif		$\frac{\sum \text{Benar}}{\sum \text{Data}} \times 100\% = \frac{1}{1} \times 100\%$	100%
Ba		$\frac{\sum \text{Benar}}{\sum \text{Data}} \times 100\% = \frac{1}{1} \times 100\%$	100%
Ta		$\frac{\sum \text{Benar}}{\sum \text{Data}} \times 100\% = \frac{1}{1} \times 100\%$	100%
Tsa		$\frac{\sum \text{Benar}}{\sum \text{Data}} \times 100\% = \frac{1}{1} \times 100\%$	100%
Jim		$\frac{\sum \text{Benar}}{\sum \text{Data}} \times 100\% = \frac{1}{1} \times 100\%$	100%
Ha'		$\frac{\sum \text{Benar}}{\sum \text{Data}} \times 100\% = \frac{1}{1} \times 100\%$	100%

Kha	خ	$\frac{\sum \text{Benar}}{\sum \text{Data}} \times 100\% = \frac{1}{1} \times 100\%$	100%
Dal	د	$\frac{\sum \text{Benar}}{\sum \text{Data}} \times 100\% = \frac{1}{1} \times 100\%$	100%
Dzal	ذ	$\frac{\sum \text{Benar}}{\sum \text{Data}} \times 100\% = \frac{1}{1} \times 100\%$	100%
Ra	ر	$\frac{\sum \text{Benar}}{\sum \text{Data}} \times 100\% = \frac{1}{1} \times 100\%$	100%
Zae	ز	$\frac{\sum \text{Benar}}{\sum \text{Data}} \times 100\% = \frac{1}{1} \times 100\%$	100%
Sin	س	$\frac{\sum \text{Benar}}{\sum \text{Data}} \times 100\% = \frac{1}{1} \times 100\%$	100%

Syuin		$\frac{\sum \text{Benar}}{\sum \text{Data}} \times 100\% = \frac{1}{1} \times 100\%$	100%
Shad		$\frac{\sum \text{Benar}}{\sum \text{Data}} \times 100\% = \frac{1}{1} \times 100\%$	100%
Dhad		$\frac{\sum \text{Benar}}{\sum \text{Data}} \times 100\% = \frac{1}{1} \times 100\%$	100%
Tha		$\frac{\sum \text{Benar}}{\sum \text{Data}} \times 100\% = \frac{1}{1} \times 100\%$	100%
Zho		$\frac{\sum \text{Benar}}{\sum \text{Data}} \times 100\% = \frac{1}{1} \times 100\%$	100%
'ain		$\frac{\sum \text{Benar}}{\sum \text{Data}} \times 100\% = \frac{1}{1} \times 100\%$	100%

Ghain	غ	$\frac{\sum \text{Benar}}{\sum \text{Data}} \times 100\% = \frac{1}{1} \times 100\%$	100%
Fa	ف	$\frac{\sum \text{Benar}}{\sum \text{Data}} \times 100\% = \frac{1}{1} \times 100\%$	100%
Qaf	ق	$\frac{\sum \text{Benar}}{\sum \text{Data}} \times 100\% = \frac{1}{1} \times 100\%$	100%
Kaf	ك	$\frac{\sum \text{Benar}}{\sum \text{Data}} \times 100\% = \frac{1}{1} \times 100\%$	100%
Lam	ل	$\frac{\sum \text{Benar}}{\sum \text{Data}} \times 100\% = \frac{1}{1} \times 100\%$	100%
Mim	م	$\frac{\sum \text{Benar}}{\sum \text{Data}} \times 100\% = \frac{1}{1} \times 100\%$	100%

Nun		$\frac{\sum \text{Benar}}{\sum \text{Data}} \times 100\% = \frac{1}{1} \times 100\%$	100%
Waw		$\frac{\sum \text{Benar}}{\sum \text{Data}} \times 100\% = \frac{1}{1} \times 100\%$	100%
Ha		$\frac{\sum \text{Benar}}{\sum \text{Data}} \times 100\% = \frac{1}{1} \times 100\%$	100%
Lamalif		$\frac{\sum \text{Benar}}{\sum \text{Data}} \times 100\% = \frac{1}{1} \times 100\%$	100%
Hamzah		$\frac{\sum \text{Benar}}{\sum \text{Data}} \times 100\% = \frac{1}{1} \times 100\%$	100%
Ya		$\frac{\sum \text{Benar}}{\sum \text{Data}} \times 100\% = \frac{1}{1} \times 100\%$	100%

Berdasarkan tabel 4.2, citra huruf hijaiyah yang diujikan memiliki kemiringan 358° yang berarti tidak sama dengan 3 citra huruf hijaiyah yang

dilatihkan. Table 4.2 menunjukkan bahwa tingkat keakuratan JST untuk mengenali huruf hijaiyah dengan data baru mencapai 100%.

Table 4.3 Pengujian tingkat keakuratan menggunakan citra cacat

Nama Huruf	Huruf hijaiyah	Perhitungan	Tingkat Keakuratan
Alif		$\frac{\sum \text{Benar}}{\sum \text{Data}} \times 100\% = \frac{1}{1} \times 100\%$	100%
Ba		$\frac{\sum \text{Benar}}{\sum \text{Data}} \times 100\% = \frac{1}{1} \times 100\%$	100%
Ta		$\frac{\sum \text{Benar}}{\sum \text{Data}} \times 100\% = \frac{1}{1} \times 100\%$	100%
Tsa		$\frac{\sum \text{Benar}}{\sum \text{Data}} \times 100\% = \frac{1}{1} \times 100\%$	100%
Jim		$\frac{\sum \text{Benar}}{\sum \text{Data}} \times 100\% = \frac{1}{1} \times 100\%$	100%

Ha'		$\frac{\sum \text{Benar}}{\sum \text{Data}} \times 100\% = \frac{1}{1} \times 100\%$	100%
Kha		$\frac{\sum \text{Benar}}{\sum \text{Data}} \times 100\% = \frac{1}{1} \times 100\%$	100%
Dal		$\frac{\sum \text{Benar}}{\sum \text{Data}} \times 100\% = \frac{1}{1} \times 100\%$	100%
Dzal		$\frac{\sum \text{Benar}}{\sum \text{Data}} \times 100\% = \frac{1}{1} \times 100\%$	100%
Ra		$\frac{\sum \text{Benar}}{\sum \text{Data}} \times 100\% = \frac{1}{1} \times 100\%$	100%
Zae		$\frac{\sum \text{Benar}}{\sum \text{Data}} \times 100\% = \frac{1}{1} \times 100\%$	100%

Sin		$\frac{\sum \text{Benar}}{\sum \text{Data}} \times 100\% = \frac{1}{1} \times 100\%$	100%
Syuin		$\frac{\sum \text{Benar}}{\sum \text{Data}} \times 100\% = \frac{1}{1} \times 100\%$	100%
Shad		$\frac{\sum \text{Benar}}{\sum \text{Data}} \times 100\% = \frac{1}{1} \times 100\%$	100%
Dhad		$\frac{\sum \text{Benar}}{\sum \text{Data}} \times 100\% = \frac{1}{1} \times 100\%$	100%
Tha		$\frac{\sum \text{Benar}}{\sum \text{Data}} \times 100\% = \frac{1}{1} \times 100\%$	100%
Zho		$\frac{\sum \text{Benar}}{\sum \text{Data}} \times 100\% = \frac{1}{1} \times 100\%$	100%

'ain		$\frac{\sum \text{Benar}}{\sum \text{Data}} \times 100\% = \frac{1}{1} \times 100\%$	100%
Ghain		$\frac{\sum \text{Benar}}{\sum \text{Data}} \times 100\% = \frac{1}{1} \times 100\%$	100%
Fa		$\frac{\sum \text{Benar}}{\sum \text{Data}} \times 100\% = \frac{1}{1} \times 100\%$	100%
Qaf		$\frac{\sum \text{Benar}}{\sum \text{Data}} \times 100\% = \frac{1}{1} \times 100\%$	100%
Kaf		$\frac{\sum \text{Benar}}{\sum \text{Data}} \times 100\% = \frac{1}{1} \times 100\%$	100%
Lam		$\frac{\sum \text{Benar}}{\sum \text{Data}} \times 100\% = \frac{1}{1} \times 100\%$	100%

Mim		$\frac{\sum \text{Benar}}{\sum \text{Data}} \times 100\% = \frac{1}{1} \times 100\%$	100%
Nun		$\frac{\sum \text{Benar}}{\sum \text{Data}} \times 100\% = \frac{1}{1} \times 100\%$	100%
Waw		$\frac{\sum \text{Benar}}{\sum \text{Data}} \times 100\% = \frac{1}{1} \times 100\%$	100%
Ha		$\frac{\sum \text{Benar}}{\sum \text{Data}} \times 100\% = \frac{1}{1} \times 100\%$	100%
Lamalif		$\frac{\sum \text{Benar}}{\sum \text{Data}} \times 100\% = \frac{1}{1} \times 100\%$	100%
Hamzah		$\frac{\sum \text{Benar}}{\sum \text{Data}} \times 100\% = \frac{1}{1} \times 100\%$	100%

Ya		$\frac{\sum \text{Benar}}{\sum \text{Data}} \times 100\% = \frac{1}{1} \times 100\%$	100%
----	---	--	------

Berdasarkan tabel 4.3, citra huruf hijaiyah yang diujikan diberikan cacat di beberapa sisinya. Table 4.3 menunjukkan bahwa tingkat keakuratan JST untuk mengenali huruf hijaiyah dengan data yang diberi cacat mencapai 100%.

4.2.2 Pengujian Laju Pembelajaran

Pada pengujian laju pembelajaran ini, penulis melakukan pengujian untuk mencari tingkat keakuratan pengenalan huruf dengan nilai laju pembelajaran yang berbeda-beda. Pengujian ini menggunakan jumlah neuron *hidden* sebanyak 30, jumlah neuron *input* sebanyak 340, jumlah neuron *output* sebanyak 30, maksimum *epoch* 10.000, dengan nilai *learning rate* (α) atau laju pembelajaran yang di ubah-ubah. Pengujian ini akan menguji tiap huruf hijaiyah sebanyak 1 kali masukan. Hasil pengujian dapat dilihat pada table 4.5, 4.6 dan 4.7.

Tabel 4.4 Nilai *learning rate* (α) = 0.7

Huruf Hijaiyah	Dikenali	Tidak Dikenali	Presentase (%)
Alif	1	0	100%
Ba	1	0	100%
Ta	1	0	100%
Tsa	1	0	100%
Jim	1	0	100%
Ha'	1	0	100%
Kha	1	0	100%
Dal	1	0	100%
Dzal	1	0	100%
Ra	1	0	100%
Zae	1	0	100%
Sin	1	0	100%

Syuin	1	0	100%
Shad	1	0	100%
Dhad	1	0	100%
Tha	1	0	100%
Zha	1	0	100%
'ain	1	0	100%
Ghain	1	0	100%
Fa	1	0	100%
Qaf	1	0	100%
Kaf	1	0	100%
Lam	1	0	100%
Mim	1	0	100%
Nun	1	0	100%
Waw	1	0	100%
Ha	1	0	100%
Lamalif	1	0	100%
Hamzah	1	0	100%
Ya	1	0	100%

Tabel 4.5 Nilai *learning rate* (α) = 0.8

Huruf Hijaiyah	Dikenali	Tidak Dikenali	Presentase (%)
Alif	1	0	100%
Ba	1	0	100%
Ta	1	0	100%
Tsa	1	0	100%
Jim	1	0	100%
Ha'	1	0	100%
Kha	1	0	100%
Dal	1	0	100%
Dzal	1	0	100%

Ra	1	0	100%
Zae	1	0	100%
Sin	1	0	100%
Syuin	1	0	100%
Shad	1	0	100%
Dhad	1	0	100%
Tha	1	0	100%
Zha	1	0	100%
'ain	1	0	100%
Ghain	1	0	100%
Fa	1	0	100%
Qaf	1	0	100%
Kaf	1	0	100%
Lam	1	0	100%
Mim	1	0	100%
Nun	1	0	100%
Waw	1	0	100%
Ha	1	0	100%
Lamalif	1	0	100%
Hamzah	1	0	100%
Ya	1	0	100%

Tabel 4.6 Nilai *learning rate* (α) = 0,9

Huruf Hijaiyah	Dikenali	Tidak Dikenali	Presentase (%)
Alif	1	0	100%
Ba	1	0	100%
Ta	1	0	100%
Tsa	1	0	100%
Jim	1	0	100%
Ha'	1	0	100%

Kha	1	0	100%
Dal	1	0	100%
Dzal	1	0	100%
Ra	1	0	100%
Zae	1	0	100%
Sin	1	0	100%
Syuin	1	0	100%
Shad	1	0	100%
Dhad	1	0	100%
Tha	1	0	100%
Zha	1	0	100%
'ain	1	0	100%
Ghain	1	0	100%
Fa	1	0	100%
Qaf	1	0	100%
Kaf	1	0	100%
Lam	1	0	100%
Mim	1	0	100%
Nun	1	0	100%
Waw	1	0	100%
Ha	1	0	100%
Lamalif	1	0	100%
Hamzah	1	0	100%
Ya	1	0	100%

Berdasarkan tabel 4.4, tabel 4.5 dan tabel 4.6 terlihat bahwa perubahan nilai *learning rate*/laju pembelajaran tidak mempengaruhi tingkat akurasi hasil pengujian. Namun pada dasarnya semakin tinggi nilai *learning rate* maka semakin cepat proses training yang terjadi pada sistem.

BAB V

PENUTUP

5.1 Kesimpulan

Kesimpulan dari penelitian ini adalah sebagai berikut:

1. Aplikasi ini hanya mampu mengenali Huruf Hijaiyah dengan data citra berukuran 17x20 pixel dengan format JPG.
2. Aplikasi ini berhasil melakukan proses pelatihan jaringan saraf tiruan dengan menggunakan 90 sampel citra huruf hijaiyah dan mampu melakukan proses pengenalan menggunakan 60 sampel citra huruf hijaiyah dengan tingkat akurasi 100%.
3. Aplikasi ini berhasil mengenali 60 sampel data citra huruf hijaiyah dengan tingkat akurasi 100% meskipun dibcrikan perubahan nilai *learning rate* yang berbeda-beda.

5.2 Saran

Saran untuk pengembangan lebih lanjut adalah:

1. Menambahkan lebih banyak data training pada masing-masing huruf hijaiyah agar proses pengenalan semakin baik.
2. Menambahkan data pengujian 30 jenis huruf hijaiyah menggunakan citra hasil pemindaian dari tulisan tangan dengan ukuran 170x200 pixel.
3. Menambahkan data pengujian melalui masukan kamera webcam.
4. Data sampel dan training memiliki warna yang bermacam-macam (tidak hitam putih).

DAFTAR PUSTAKA

- David & Sandy Kosasi. 2013. *Penerapan Algoritma Jaringan Syaraf Tiruan Backpropagation Untuk Pengenalan Pola Tanda Tangan*. Jurnal. Pontianak.
- Falisyah. 2007. *Computer Vision Dan Pengolahan Citra*. Yogyakarta. Andi Offset.
- Hermawan, Arief. 2007. *Jaringan Saraf Tiruan teori dan aplikasi*. Yogyakarta: Andi Offset.
- Hidayatno, Ahmad dkk. 2008. *Identifikasi Tanda-Tangan Menggunakan Jaringan Saraf Tiruan Perambatan-balik (Backpropagation)*. Jurnal. UNDIP, Semarang.
- Hidayatullah, Priyanto. 2014. *Visual Basic. Net Membuat Aplikasi Database dan program Kreatif*. Bandung: Informatika.
- Puspitaningrum, Diah. 2006. *Pengantar Jaringan Saraf Tiruan*. Yogyakarta: Andi Offset.
- Setiawan, Sari Indah Anatta. 2011. *Penerapan Jaringan Saraf Tiruan Metode Backpropagation Menggunakan VB6*. Ultimatics Vol.III, No.2. SofTech, Tangerang.
- Widodo, Pudjo Prabowo & Rahmadya Trias Handayanto. 2012. *Penerapan Soft Computing dengan Matlab*. Bandung: Rekayasa Sains.
- Winarno, Edy. 2009. *Pengolahan Citra*. Jurnal. UNISBANK, Semarang.
- Zainuddin. 2014. *Aplikasi Pengenalan Huruf Hijaiyah Untuk Smartphone Android*. Jurnal. Surabaya.

LAMPIRAN



INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
PROGRAM STUDI TEKNIK INFORMATIKA S-1
Jl. Raya Karanglo Km. 2 Malang

BERITA ACARA UJIAN KOMPRESI
FAKULTAS TEKNOLOGI INDUSTRI

Nama : Imam Ahmad Askalani Syukri
NIM : 12.18.066
Program Studi : Teknik Informatika S-1
Judul Skripsi : Rancang Bangun Aplikasi Pengenalan Huruf Hijaiyah
Menggunakan Metode Neural Network Backpropagation

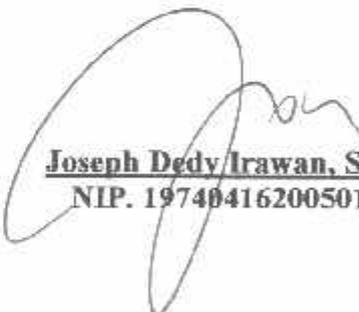
Dipertahankan dihadapan Majelis Penguji Skripsi Jenjang Strata Satu (S1)

pada :

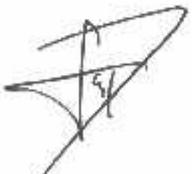
Hari : Jum'at
Tanggal : 15 Januari 2016
Nilai : 87,58 (A)

Panitia Ujian Skripsi

Ketua Majelis Penguji


Joseph Dedy Irawan, ST. MT.
NIP. 197404162005011002

Dosen Penguji I


Sonny Prasetyo, ST. MT.
NIP. P. 1031000433

Dosen Penguji II


Karina Auliasari, ST. M.Eng.
NIP. 1031000426



FORMULIR PERBAIKAN SKRIPSI

Nama : Imam Ahmad Askalani Syukri
Nim : 12.18.066
Program Studi : Teknik Informatika S-1
Judul Skripsi : Rancang Bangun Aplikasi Pengenalan Huruf Hijaiyah Menggunakan Metode Neural Network Backpropagation

Tanggal	Penguji	Uraian	Paraf
15 Januari 2016	I	- Tambahkan progress bar pada aplikasi - Perbaiki kesimpulan	
15 Januari 2016	II	- Landasan teori harus diberi sitasi - Kesimpulan diperbaiki sesuai catatan - Saran diperbaiki sesuai catatan - Daftar pustaka tambahkan referensi	

Anggota Penguji

Dosen Penguji I

Sonny Prasetio, ST. MT.
NIP. P. 1031000433

Dosen Penguji II

Karina Auliasari, ST. M.Eng.
NIP. 1031000426

Mengetahui,

Dosen Pembimbing I

Yosep Agus Pranoto, ST. MT.
NIP. P. 1031000432

Dosen Pembimbing II

Ahmad Fuisok ST. MT.
NIP. P. 1031000431



PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG

FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

PT. BNI (PERSERO) MALANG
BANK NIAGA MALANG

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

Malang, 23 Oktober 2015

Nomor : ITN-593/LINF/TA/2015
Lampiran : --
Perihal : Bimbingan Skripsi

Kepada : Yth. Bpk/Ibu Yosep Agus Pranoto, ST, MT
Dosen Pembina Program Studi Teknik Informatika S-1
Institut Teknologi Nasional
Malang

Dengan Hormat,

Sesuai dengan permohonan dan persetujuan dalam proposal skripsi untuk mahasiswa :

Nama : IMAM AHMAD ASKALANI
Nim : 1218066
Prodi : Teknik Informatika S-1
Fakultas : Teknologi Industri

Maka dengan ini pembimbingan kami serahkan sepenuhnya kepada Saudara/i selama waktu 6 (enam) bulan, terhitung mulai tanggal :

23 Oktober 2015 S/D 23 Maret 2016

Sebagai satu syarat untuk menempuh Ujian Akhir Sarjana Teknik, Program Studi Teknik Informatika S-1.

Demikian agar maklum dan atas perhatian serta bantuannya kami sampaikan terima kasih.

Mengetahui
Program Studi Teknik Informatika S-1

Ketua,


Joseph Dedy Irawan, ST., MT.

NIP : 197404162005021002

Form S-4a



PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG

**FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK**

PT. BNI (PERSERO) MALANG
BANK NIAGA MALANG

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

Malang, 23 Oktober 2015

Nomor : ITN-593/LINF/TA/2015

Lampiran : ---

Perihal : Bimbingan Skripsi

Kepada : Yth. Bpk/Ibu Ahmad Faisol, ST, MT
Dosen Pembina Program Studi Teknik Informatika S-1
Institut Teknologi Nasional
Malang

Dengan Hormat,

Sesuai dengan permohonan dan persetujuan dalam proposal skripsi untuk mahasiswa :

Nama : IMAM AHMAD ASKALANI
Nim : 1218066
Prodi : Teknik Informatika S-1
Fakultas : Teknologi Industri

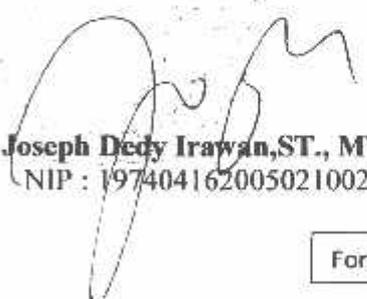
Maka dengan ini pembimbingan kami serahkan sepenuhnya kepada Saudara/i selama waktu 6 (enam) bulan, terhitung mulai tanggal :

23 Oktober 2015 S/D 23 Maret 2016

Sebagai satu syarat untuk menempuh Ujian Akhir Sarjana Teknik, Program Studi Teknik Informatika S-1.

Demikian agar maklum dan atas perhatian serta bantuannya kami sampaikan terima kasih.

Mengetahui
Program Studi Teknik Informatika S-1
Ketua,


Joseph Dedy Irawan, ST., MT.
NIP : 197404162005021002

Form S-4a



BERITA FORMULIR BIMBINGAN SKRIPSI

Nama : Imam Ahmad Askalani Syukri
NIM : 1218066
Masa Bimbingan : 23 Oktober 2015 s/d 23 Maret 2016
Judul Skripsi : Rancang Bangun Aplikasi Pengenalan Huruf Hijaiyah
Menggunakan Metode Neural Network
Backpropagation

No	Tanggal	Uraian	Paraf Pembimbing
1	18/11/2015	- Bimbingan Laporan Bab I, Bab II dan Bab III - Bimbingan Program	
2	19/11/2015	- Revisi Laporan Bab II dan Bab III - Program : tambah ke segmentasi dan JST	
3	20/11/2015	Acc Laporan Bab I - Bab III & Program Fix	
4	02/12/2015	Konsultasi Algoritma Program	
5	04/12/2015	Konsultasi Program	
6	08/12/2015	Konsultasi Program & koreksi Judul	
7	11/12/2015	Bimbingan Laporan Bab IV, Bab V dan Program	
8	08/01/2016	Konsultasi Program & Laporan Kompre	
9	09/01/2016	Laporan Bab I - BabV dan Program Fix	

Malang, 23 -1- 2016
Dosen Pembimbing I

Yosep Agus Pranoto, ST., MT.
NIP. P. 1031000432



FORMULIR BIMBINGAN SKRIPSI

Nama : Imam Ahmad Askalani Syukri
NIM : 1218066
Masa Bimbingan : 23 Oktober 2015 s/d 23 Maret 2016
Judul Skripsi : Rancang Bangun Aplikasi Pengenalan Huruf Hijaiyah
Menggunakan Metode Neural Network
Backpropagation

No	Tanggal	Uraian	Paraf Pembimbing
1	17/11/2015	Konsultasi algoritma program	
2	20/11/2015	Konsultasi laporan Bab I, Bab II dan Bab III	
3	23/11/2015	Acc Laporan Bab I, Bab II dan Bab III	
4	01/12/2016	Konsultasi Program	
5	07/12/2016	Konsultasi Revisi Program dan Laporan Bab IV dan V	
6	08/12/2016	Konsultasi Laporan Bab I – Bab V	
7	11/12/2016	Revisi Bab IV	
8	12/12/2016	Acc Laporan dan Program	
9	09/01/016	Laporan Bab I – Bab V dan Program Fix	

Malang, 2016
Dosen Pembimbing II

Ahmad Faisol, ST., MT.
NIP. P. 1031000431

Lampiran listing program:

1. Script Neural Network

```
Private Sub NeuralNetwork()  
    Dim i, epoch As Integer  
    Dim sample As Integer = 0  
    Dim TempString As String = ""  
    Dim TempString2 As String = ""  
    RichTextBox1.Text = ""  
  
    bobotRandom ()  
    'Train the network.  
    For epoch = 0 To TRAINING_REPS - 1  
        sample += 1  
        If sample = MAX_SAMPLES Then  
            sample = 0  
        End If  
  
        For i = 0 To INPUT_NEURONS - 1  
            inputs(i) = trainInputs(sample, i)  
        Next i  
  
        For i = 0 To OUTPUT_NEURONS - 1  
            target(i) = trainOutput(sample, i)  
        Next i  
  
        feedForward()  
  
        backPropagate()  
  
    Next epoch  
    getTrainingStats()  
End Sub
```

2. Script Pemberian Nilai Bobot (Random)

```
Dim inp As Integer  
Dim hid As Integer  
Dim out As Integer  
Dim sum As Double  
Dim tempString2 As String = ""  
'Calculate input to hidden layer.  
For hid = 0 To HIDDEN_NEURONS - 1  
    sum = 0.0  
    For inp = 0 To INPUT_NEURONS - 1  
        sum += inputsuji(inp) * wih(inp, hid)  
    Next  
  
    'Add in bias.  
    sum += wih(INPUT_NEURONS, hid)  
    hidden(hid) = sigmoid(sum)  
Next  
'Calculate the hidden to output layer.  
For out = 0 To OUTPUT_NEURONS - 1  
    sum = 0.0  
    For hid = 0 To HIDDEN_NEURONS - 1
```

```

        sum += hidden(hid) * who(hid, out)
    Next
    'Add in bias.
    sum += who(HIDDEN_NEURONS, out)
    actualuj1(out) = sigmoid(sum)
    ' tempString2 += CStr(maximum(actualuj1))
Next
End Sub

```

3. Script Feedforward

```

Private Sub feedForward()
    Dim inp As Integer
    Dim hid As Integer
    Dim out As Integer
    Dim sum As Double
    'Calculate input to hidden layer.
    For hid = 0 To HIDDEN_NEURONS - 1
        sum = 0.0
        For inp = 0 To INPUT_NEURONS - 1
            sum += inputs(inp) * wih(inp, hid)
        Next
        'Add in bias.
        sum += wih(INPUT_NEURONS, hid)
        hidden(hid) = sigmoid(sum)
    Next
    'Calculate the hidden to output layer.
    For out = 0 To OUTPUT_NEURONS - 1
        sum = 0.0
        For hid = 0 To HIDDEN_NEURONS - 1
            sum += hidden(hid) * who(hid, out)
        Next
        'Add in bias.
        sum += who(HIDDEN_NEURONS, out)
        actual(out) = sigmoid(sum)
    Next
End Sub

```

4. Script Backpropagation

```

Private Sub feedForward()
    Dim inp As Integer
    Dim hid As Integer
    Dim out As Integer
    Dim sum As Double
    'Calculate input to hidden layer.
    For hid = 0 To HIDDEN_NEURONS - 1
        sum = 0.0
        For inp = 0 To INPUT_NEURONS - 1
            sum += inputs(inp) * wih(inp, hid)
        Next
        'Add in bias.
        sum += wih(INPUT_NEURONS, hid)
        hidden(hid) = sigmoid(sum)
    Next

```

```

'Calculate the hidden to output layer.
  For out = 0 To OUTPUT_NEURONS - 1
    sum = 0.0
    For hid = 0 To HIDDEN_NEURONS - 1
      sum += hidden(hid) * who(hid, out)
    Next
    'Add in bias.
    sum += who(HIDDEN_NEURONS, out)
    actual(out) = sigmoid(sum)
  Next
End Sub

```

5. Script Hitung Stat Training

```

Private Sub getTrainingStats() 'nampilkan statistik training
  Dim i, j As Integer
  Dim sum As Double = 0.0
  Dim TempString As String = ""

  For i = 0 To MAX_SAMPLES - 1
    For j = 0 To INPUT_NEURONS - 1
      inputs(j) = trainInputs(i, j) 'masukkan sampel ke
dalam neuron input (0-339)
    Next j

    For j = 0 To OUTPUT_NEURONS - 1
      target(j) = trainOutput(i, j) 'masukkan target ke
dalam neuron target (0-29)
    Next j

    feedForward() 'lakukan langkah maju

    If maximum(actual) = maximum(target) Then 'cek apakah
output sama dengan target
      sum += 1
    End If
  Next i
  Dim persentase As Double = sum / MAX_SAMPLES * 100
  TempString += "Network is " + CStr(persentase) + _
"% correct." + Chr(13) + Chr(10)
'hitung persentasi keberhasilan
percent = persentase
Label1.Text = TempString
End Sub

```

6. Script Test Network

```

Private Sub testNetworkTraining()
  'This function simply tests the training vectors
against network.
  Dim TempString As String = ""
  Dim i, j As Integer
  For i = 0 To MAX_SAMPLES - 1
    For j = 0 To INPUT_NEURONS - 1
      inputs(j) = trainInputs(i, j)
    Next j
    feedForward()
    For j = 0 To INPUT_NEURONS - 1

```

```

        TempString += CStr(inputs(j)) + vbTab
    Next j
    TempString += " Output: " + CStr(maximum(actual)) +
vbCrLf
    Next i
    RichTextBox1.Text += TempString + vbCrLf
End Sub

```

7. Script Pengujian

```

Dim inp As Integer
Dim hid As Integer
Dim out As Integer
Dim sum As Double
Dim tempString2 As String = ""
'Calculate input to hidden layer.
For hid = 0 To HIDDEN_NEURONS - 1
    sum = 0.0
    For inp = 0 To INPUT_NEURONS - 1
        sum += inputsuji(inp) * wih(inp, hid)
    Next
    'Add in bias.
    sum += wih(INPUT_NEURONS, hid)
    hidden(hid) = sigmoid(sum)
Next
'Calculate the hidden to output layer.
For out = 0 To OUTPUT_NEURONS - 1
    sum = 0.0
    For hid = 0 To HIDDEN_NEURONS - 1
        sum += hidden(hid) * who(hid, out)
    Next
    'Add in bias.
    sum += who(HIDDEN_NEURONS, out)
    actualuji(out) = sigmoid(sum)
    tempString2 += CStr(maximum(actualuji))
Next
End Sub

```

8. Script Fungsi Aktivasi

```

Private Function sigmoid(ByVal val As Double) As Double
    Return (1.0 / (1.0 + Math.Exp(-val)))
End Function

```

9. Script Turunan Fungsi Aktifasi

```

Private Function sigmoidDerivative(ByVal val As Double) As Double
    Return (val * (1.0 - val))
End Function

```