

**DETEKSI HURUF TULISAN TANGAN MENGGUNAKAN JARINGAN
SARAF TIRUAN BACKPROPAGATION**

SKRIPSI



**Disusun Oleh :
DWI FERNANDUS TURNIP
06.12.608**

**PROGRAM STUDI TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK KOMPUTER DAN INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2013**

LEMBAR PERSETUJUAN

DETEKSI HURUF TULISAN TANGAN MENGGUNAKAN JARINGAN SARAF TIRUAN BACKPROPAGATION

SKRIPSI

*Disusun dan Diajukan sebagai Salah Satu Syarat Untuk Memperoleh
Gelar Sarjana Teknik Komputer dan Informatika Strata Satu (S-1)*

Disusun Oleh :

DWI FERNANDUS TURNIP

NIM : 06.12.608

Diperiksa dan Disetujui

Mengetahui

Ketua Prodi Teknik Elektro S-1



M. Ibrahim Ashari, ST, MT

NIP. P. 1030100358

Dosen Pembimbing I

Irmalia Suryani Farandisa, ST, MT

NIP.P.1030000365

Dosen Pembimbing II

Yuli Wahyuni, ST, MT

NIP.P.1031200456

**PROGRAM STUDI TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK KOMPUTER DAN INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2013**

ABSTRAK

DETEKSI HURUF TULISAN TANGAN MENGGUNAKAN JARINGAN SARAF TIRUAN *BACKPROPAGATION*

DWI FERNANDUS TURNIP
(06.12.608)

Jurusan Teknik Elektro S-1, Konsentrasi Teknik Komputer dan Informatika
Fakultas Teknologi Industri, Institut Teknologi Nasional Malang
Jl. Raya Karanglo Km 2 Malang
Email : nice_live0202@yahoo.com

Dosen Pembimbing I : Irmalia Suryani Farandisa, ST, MT
Dosen Pembimbing II : Yuli Wahyuni, ST, MT

Ada beberapa tipe jaringan saraf, namun hampir semuanya memiliki komponen yang sama. Jaringan saraf tiruan terdiri dari *neuron* yang saling berhubungan dengan neuron-neuron lainnya, dimana neuron-neuron tersebut mentransformasikan informasi yang diterima melalui sambungan keluarnya menuju neuron-neuron lain. Hubungan ini dikenal dengan sebutan bobot. Algoritma pelatihan jaringan saraf perambatan galat mundur terdiri atas dua langkah, yaitu perambatan maju dan perambatan mundur. Perambatan maju dimulai dengan memberikan pola masukan ke lapisan masukan. Pola masukan ini merupakan nilai aktivasi unit-unit masukan. Dengan melakukan perambatan maju di hitung nilai aktivasi pada unit-unit dilapisan berikutnya. Pada setiap lapisan, tiap unit pengolah melakukan penjumlahan berbobot dan menerapkan fungsi sigmoid untuk menghitung keluarannya. Setelah didapat nilai keluarannya maka akan dihitung kembali nilai yang baru dengan perambatan mundur. Penggunaan *Backpropagation Network* terdiri dari 2 tahap, yaitu tahap belajar atau pelatihan, dimana pada tahap ini pada *backpropagation* diberikan sejumlah data pelatihan dan target dan tahap pengujian atau penggunaan, pengujian dan penggunaan dilakukan setelah *backpropagation* selesai belajar.

Kata Kunci: Tulisan tangan, ekstraksi ciri, jaringan saraf tiruan, *backpropagation*.

Penulis telah berusaha semaksimal mungkin dan menyadari sepenuhnya akan keterbatasan pengetahuan dalam menyelesaikan laporan ini. Untuk itu penyusun mengharapkan saran dan kritik yang membangun dari pembaca demi kesempurnaan laporan ini.

Harapan penulis semoga laporan skripsi ini memberikan manfaat bagi perkembangan ilmu pengetahuan dan pembaca.

Malang, Maret 2013

Penulis

DAFTAR ISI

LEMBAR PERSETUJUAN	i
ABSTRAKSI	ii
KATA PENGANTAR	iii
DAFTAR ISI	v
DAFTAR GAMBAR	viii
DAFTAR TABEL	xi
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Batasan Masalah	2
1.4. Tujuan Penelitian	2
1.5. Metode Penelitian	3
1.6. Sistematika Penulisan	3
BAB II TINJAUAN PUTAKA	5
2.1. Jaringan Saraf Tiruan	5
2.2. Sejarah Jaringan Saraf Tiruan	7
2.3. Arsitektur	7
2.4. Perambatan Galat Mundur	10
2.5. Microsoft Access	18
2.6. Sejarah Microsoft Access	19
2.7. Visual Basic 6.0	20
2.8. Interface Antar Muka Visual Basic 6.0.....	21

2.9. Konsep Dasar Pemrograman Dalam Visual Basic 6.0.....	22
2.10. Membuar Project Baru	23
2.11. Thresholding	27
2.12. Scan	27
BAB III ANALISA DAN PERANCANGAN SISTEM	31
3.1. Analisa Sistem	31
3.2. Deskripsi Umum Sistem	31
3.3. Fitur Apikasi	31
3.3.1. Training	31
3.3.2. Test	32
3.3.3. Exit	32
3.4. Analisa Kebutuhan Sistem	32
3.5. Teknik Pengumpulan Data	32
3.6. Prapengelolaan	34
3.6.1. Ekstraksi Ciri	34
3.7. Perancangan Sistem	34
3.8. Desain Sistem Aplikasi	34
3.9. Gambaran Umum Perangkat Lunak	35
3.10. Penjelasan Desain Sistem	36
3.10.1. Penjelasan Desain Sistem	36
3.11. Penetapan Masukan Input	37
3.12. Proses Jaringan Saraf Tiruan Backpropagation	38
3.12.1. Arsitektur	38
3.12.2. Proses Pelatihan	39

DAFTAR GAMBAR

Gambar 2.1. Struktur Neuron Saraf	6
Gambar 2.2. Jaringan Saraf Dengan Lapisan Tunggal	8
Gambar 2.3. Jaringan Saraf Dengan Banyak Lapisan	9
Gambar 2.4. Jaringan Saraf Dengan kompetitif.....	9
Gambar 2.5. Tiga Lapisan Jaringan Perambatan Galat Mundur.....	11
Gambar 2.6. Langkah Perambatan Maju	12
Gambar 2.7. Fungsi Sigmoid Besrta Turunannya	13
Gambar 2.8. Tiga Lapisan Perambatan Galat Mundur	14
Gambar 2.9. Interface Antar Muka Visual Basic 6.0	21
Gambar 2.10. Komponen Standar Dalam Toolbox	22
Gambar 2.11. Layar Pemilihan Jenis Project.....	23
Gambar 2.12. Jendela Form.....	25
Gambar 2.13. Jendela Kode.....	25
Gambar 2.14. Cara Mengambil Label Dari <i>Toolbox</i>	26
Gambar 2.15. Layout Pada <i>Form</i>	27
Gambar 2.16. Canon Lide 25 Scanner.....	28
Gambar 2.17. Tampilan Aplikasi Canon <i>Scan Toolbox</i>	29
Gambar 2.18. Tampilan Aplikasi Scan -1	29
Gambar 2.18. Tampilan Proses <i>Scan</i>	30
Gambar 3.1. Contoh <i>Sample</i>	33
Gambar 3.2. Dcsain Sistem	35
Gambar 3.3. Diagram Alir Sistem Sistem Secara Umum	36
Gambar 3.4. Contoh Sample Huruf	37

Gambar 3.5. Hasil Proses <i>Resize</i>	37
Gambar 3.6. Arsitektur Jaringan Saraf Tiruan <i>Backpropagation</i>	38
Gambar 3.7. Diagram Alir Proses Pelatihan Jaringan Saraf Tiruan <i>Backpropagation</i>	40
Gambar 3.8. Flowchart Aplikasi	41
Gambar 3.9. Diagram Alir Proses <i>Feedforward</i>	44
Gambar 3.10. Diagram Alir Proses <i>Backpropagation</i>	46
Gambar 3.11. Diagram Alir Perubahan Bobot	48
Gambar 3.12. Diagram Alir Proses Pengujian Jaringan Saraf Tiruan.....	50
Gambar 3.13. Sample Huruf a	51
Gambar 3.14. Sample Baris dan Kolom	51
Gambar 3.15. Inisialisasi Nilai Bobot	52
Gambar 3.16. Form Menu Untuk Deteksi Huruf Tulisan Tangan	57
Gambar 3.17. Tombol Image Processing	58
Gambar 3.18. Tombol Training	58
Gambar 3.19. Form Hidden Layer	58
Gambar 3.20. Form Learning Constanta	58
Gambar 3.21. Form Max Iteration.....	59
Gambar 3.22. Form Learnin Iteration.....	59
Gambar 3.23. Form RMS Error.....	59
Gambar 3.24. Menu Form Image Processing	59
Gambar 3.25. Tombol <i>Open</i>	60
Gambar 3.26. <i>Form Common Dialog</i>	60

DAFTAR TABEL

Tabel 2.1. Daftar sejumlah pernyataan Microsoft Access.....	18
Tabel 3.2. Tabel Kode Huruf.....	63

BAB I

PENDAHULUAN

1.1 Latar Belakang

Penulisan huruf menggunakan tulisan tangan pada media kertas tidak sejalan dimana lingkungan sedang gencar-gencarnya untuk dilestarikan. Oleh karena itu perlu adanya media yang dapat membantu mengurangi penggunaan kertas. Media tersebut dapat berupa komputer atau PC dimana user tidak lagi menulis melainkan mengetik. Namun permasalahan yang ditemukan selanjutnya adalah bahwa tidak semua user dapat mengetik dengan baik dan lancar ke dalam PC menggunakan *keyboard*. Sehingga diperlukan suatu teknologi dimana user dapat mengetik selayaknya menulis tulisan tangan menggunakan kertas namun media yang digunakan bukan lagi kertas, melainkan suatu teknologi yang terintegrasi. Teknologi yang menjadi titik berat pada skripsi ini bukan terletak pada hardware melainkan pada software yang mampu mendeteksi huruf tulisan tangan menjadi teks komputer.

Di dalam pendeteksian penulisan huruf menggunakan tangan menjadi teks komputer diperlukan 2 metode, yang pertama adalah metode yang mampu mengenali tulisan tangan dalam bentuk gambar menjadi suatu matriks pengkodean. Sedangkan metode yang kedua adalah metode yang mampu mengenali dan mengklasifikasi berbagai macam matriks pengkodean menjadi suatu kelompok matriks yang memiliki suatu kemiripan pola tertentu.

Dengan melihat metode yang pertama maka metode yang digunakan adalah metode pencitraan dengan menggunakan metode tipografi yaitu sebuah metode yang digunakan untuk mempelajari mengenai seluk beluk huruf (*font*) dari berbagai macam pola bentuknya. Dan metode yang kedua adalah metode *backpropagation* yang bertujuan untuk memproses dari berbagai macam huruf tulisan tersebut menjadi desain publikasi atau huruf cetak.

Gambaran diatas menjadi suatu pertimbangan bagi penulis untuk membuat judul "*Deteksi Huruf Tulisan Tangan menggunakan Jaringan Saraf Tiruan Backpropagation*" sebagai upaya untuk mengembangkan dan menemukan cara yang lebih efektif dan cepat didalam mengetik melalui sistem komputerisasi.

1.2 Rumusan Masalah

Tugas skripsi ini membahas tentang deteksi huruf tulisan tangan dimana penyusunannya didasarkan pada rumusan sebagai berikut :

1. Bagaimana membuat implementasi dimana inputnya merupakan citra yang dikonversikan menjadi teks digital ?
2. Bagaimana cara membuat suatu sistem untuk menganalisa huruf tulisan tangan dengan menggunakan Jaringan Saraf Tiruan ?
3. Bagaimana cara membuat suatu aplikasi deteksi huruf tulisan tangan dengan menggunakan Visual Basic 6.0 ?
4. Bagaimana menganalisis hasil yang diperoleh dari implementasi pengenalan tulisan tangan ?

1.3 Batasan Masalah

Batasan masalah yang ada pada penyusunan tugas akhir ini adalah sebagai berikut :

1. Objek yang menjadi bahan pengujian dan penelitian adalah huruf abjad dan bukan angka.
2. Huruf yang menjadi objek adalah huruf kecil bukan huruf besar atau kapital.
3. Metode *Backpropagation* dengan menggunakan 1 layer tersembunyi dan fungsi yang digunakan adalah fungsi sigmoid.
4. Perancangan sistem dilakukan berdasarkan algoritma yang berlaku di dalam Visual Basic 6.0.
5. Objek gambar berformat *.bmp dan berukuran 10x10 *pixel*.
6. Implementasi yang digunakan untuk membuat aplikasi deteksi huruf tulisan tangan adalah *Microsoft Visual Basic 6.0*.

1.4 Tujuan dan Manfaat Penulisan

Adapun tujuan dari penulisan dalam penyusunan skripsi ini adalah :

1. Agar dapat digunakan oleh beberapa *user* (pengguna) untuk mengolah tulisan tangan dalam bentuk file citra menjadi huruf cetak.
2. Untuk mengimplementasikan Metode *Backpropagation* dengan menggunakan *Visual Basic 6.0*.

3. Untuk mengetahui tingkat efisiensi memori file hasil deteksi citra dengan Metode *Backpropagation*.

1.5 Metode Penelitian

Data merupakan sumber atau bahan mentah yang sangat berharga bagi proses menghasilkan informasi. Oleh sebab itu dalam pengambilan data perlu dilakukan penanganan secara cermat dan hati-hati, sehingga data yang diperoleh dapat bermanfaat dan berkualitas.

Dalam pengumpulan data penulis menggunakan metode sebagai berikut :

1. Contoh / *sample*

Teknik pengumpulan data dengan mengumpulkan beberapa contoh atau *sample* dari beberapa tulisan tangan dari orang atau *personal* yang bersedia dijadikan objek langsung.

2. Obsevasi / *observation*

Teknik pengumpulan data dengan mengadakan pengamatan atau kegiatan yang sistematis terhadap objek yang dituju secara langsung.

1.6 Sistematika Penulisan

Agar lebih jelas dan mudah untuk dimengerti, maka penulis akan memberikan rangkuman dari pembuatan tugas akhir ini untuk memberikan gambaran-gambaran pokok sebagai berikut :

BAB I. PENDAHULUAN

Bab ini menjelaskan secara umum mengenai latar belakang masalah, permasalahan dengan batasan-batasan masalah yang digunakan, tujuan, metode serta pembahasan tugas akhir ini.

BAB II. LANDASAN TEORI

Pada bab ini akan disajikan landasan teori yang akan digunakan sebagai penyelesaian permasalahan pada sistem yang dibuat.

BAB III. PERANCANGAN DAN PEMBUATAN PERANGKAT LUNAK

Bab ini membahas tentang perancangan umum maupun uraian lebih lanjut mengenai perancangan sistem dalam pembuatan

perangkat lunak. Uraian perancangan sistem ini meliputi perancangan data mengenai data input dan output sistem, perancangan proses mengenai bagaimana sistem akan bekerja dengan proses-proses dan implementasi yang akan digunakan dalam pembuatan tugas akhir ini.

BAB IV. PENGUJIAN DAN ANALISA

Pada bab ini akan dibahas Implementasi yang akan berisi hasil *output* dari pengujian aplikasi. kemudian dianalisa kembali apakah telah sesuai dengan tujuan pembuatan yang terdapat pada bab I.

BAB V. PENUTUP

Bab ini berisi kesimpulan dan saran.

BAB II TINJAUAN PUSTAKA

2.1 Jaringan Saraf Tiruan

Jaringan Saraf Tiruan (*artificial neural network*) atau disingkat JST, adalah sistem komputasi dimana arsitektur dan operasi diilhami dari pengetahuan tentang sel saraf biologis di dalam otak, yang merupakan salah satu representasi buatan dari otak manusia yang selalu mencoba menstimulasi proses pembelajaran pada otak manusia tersebut. JST dapat digambarkan sebagai model matematis dan komputasi untuk fungsi aproksimasi *non-linier*, klasifikasi dan cluster dan regresi non-parametrik atau sebuah simulasi dari koreksi model saraf biologi.

Model saraf ditunjukkan dengan kemampuan dalam emulasi, analisis, prediksi, dan asosiasi. Kemampuan yang dimiliki JST dapat digunakan untuk belajar dan menghasilkan aturan atau operasi dari beberapa contoh atau input yang dimasukkan dan membuat prediksi tentang kemungkinan *output* yang akan muncul atau menyimpan karakteristik dari input yang disimpan kepadanya.

Vallu B.Rao dan Hayagriva V. Rao (1993) mendefinisikan jaringan saraf sebagai sebuah kelompok pengolahan elemen dalam suatu kelompok yang khusus membuat perhitungan sendiri dan memberikan hasilnya pada kelompok kedua atau berikutnya. Setiap sub-kelompok menurut gilirannya harus membuat perhitungan sendiri dan memberikan hasilnya untuk subgrup atau kelompok yang belum melakukan perhitungan. Pada akhirnya sebuah kelompok dari satu atau beberapa pengolahan elemen tersebut menghasilkan keluaran (*Output*) dari jaringan.

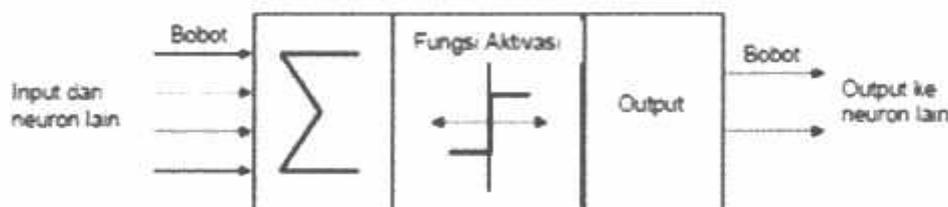
Setiap pengolahan elemen membuat perhitungan berdasarkan pada jumlah masukan (*input*). Sebuah kelompok pengolahan elemen disebut *layer* atau lapisan dalam jaringan. lapisan pertama adalah *input* dan yang terakhir adalah *output*. Lapisan diantara lapisan input dan lapisan output disebut dengan lapisan tersembunyi (*Hidden Layer*).

Jaringan saraf tiruan merupakan suatu bentuk arsitektur yang terdistribusi paralel dengan sejumlah besar node dan hubungan antar-node tersebut. Tiap titik hubungan dari satu node ke node yang lain mempunyai harga yang

diasosiasikan dengan bobot. Setiap node memiliki suatu nilai yang diasosiasikan sebagai nilai aktivasi node.

Salah satu organisasi yang dikenal dan sering digunakan dalam paradigma jaringan saraf tiruan adalah perambatan Galat Mundur (*backpropagation*). Sebelum dikenal adanya jaringan saraf perambatan Galat Mundur pada tahun 1950-1960an, dikenal dua paradigma penting yang nantinya akan menjadi dasar dari jaringan saraf perambatan Galat Mundur, yakni *perceptron* dan *Adaline/Madaline* (*adaptive linier neuron/multilayer adaline*).

Ada beberapa tipe jaringan saraf, namun hampir semuanya memiliki komponen yang sama. Jaringan saraf tiruan terdiri dari neuron yang saling berhubungan dengan neuron-neuron lainnya, dimana neuron-neuron tersebut mentransformasikan informasi yang diterima melalui sambungan keluarnya menuju neuron-neuron lain. Hubungan ini dikenal dengan sebutan bobot. Informasi tersebut disimpan pada suatu nilai tertentu pada bobot tersebut. Struktur neuron pada jaringan saraf tiruan dapat dilihat pada gambar 2.1:



Gambar 2.1 Struktur neuron saraf

Pada struktur diatas dijelaskan bahwa informasi (*input*) akan dikirim ke neuron dengan bobot kedatangan tertentu. Input ini akan diproses oleh suatu fungsi perambatan yang akan menjumlahkan nilai-nilai semua bobot yang datang. Hasil penjumlahan ini akan dibandingkan dengan suatu nilai ambang (*threshold*) tertentu melalui fungsi aktivasi setiap neuron. Apabila input tersebut melewati suatu nilai ambang tertentu, maka neuron tersebut akan aktif, tapi kalau tidak, maka neuron tersebut tidak akan aktif. Apabila neuron tersebut diaktifkan, maka neuron tersebut akan mengirim *output* melalui bobot-bobot outputnya ke semua *neuron* yang berhubungan dengannya. Demikian seterusnya.

2.2 Sejarah Jaringan Saraf Tiruan

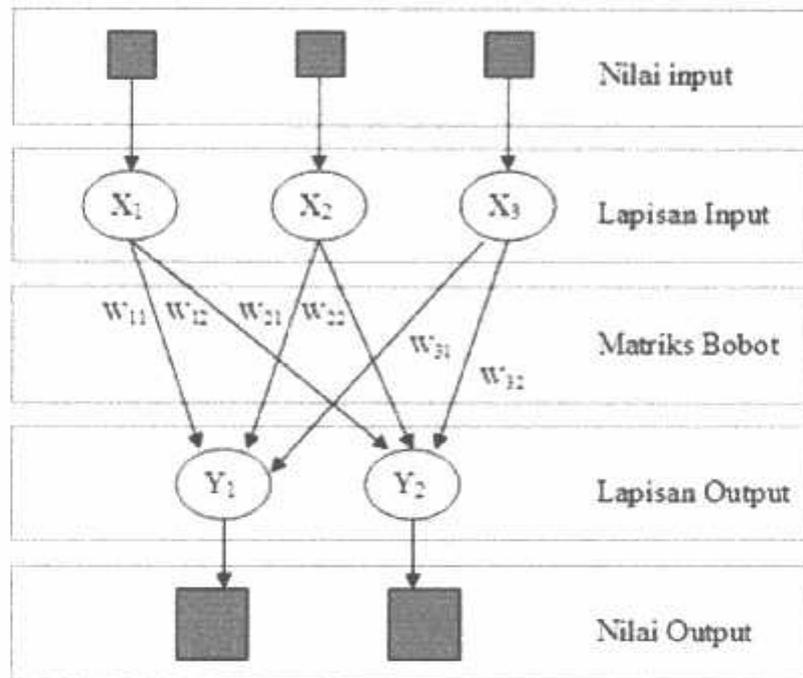
Diperkenalkan secara sederhana pada tahun 1943 oleh McCulloch dan Pitts (Fausett, 1994). Pada saat itu McCulloch dan Pitts melalui beberapa komputasi menggunakan *neuron-neuron* sederhana dapat mengubahnya menjadi sebuah sistem baru (disebut sistem *neural*) yang mempunyai kemampuan komputasi yang lebih baik. Selain itu McCulloch dan Pitts juga mengusulkan pemberian bobot dalam jaringan yang dapat diatur untuk melakukan fungsi logika sederhana. Beliau-beliau ini menggunakan semacam fungsi aktivasi *threshold*. Pada tahun 1958, Rosenblatt beserta Minsky dan Papert mulai mengembangkan model jaringan yang disebut dengan *perceptron*. Dalam model ini mereka mencoba untuk mengoptimalkan hasil iterasinya. Kemudian pada tahun 1960 Widrow dan Hoff mengembangkan model perceptron ini dengan memperkenalkan aturan pelatihan jaringan yang disebut aturan *delta* (sering juga disebut *kuadrat rata-rata terkecil*). Aturan tersebut akan mengubah bobot perceptron apabila keluaran yang dihasilkan tidak lagi sesuai dengan target yang telah ditetapkan. Hal inilah yang menyebabkan komputer dapat “belajar” dengan sendirinya. Kecepatan belajar dapat diatur dengan menggunakan parameter tertentu. Perkembangan selanjutnya dibuat oleh Rumelhart (1986) dengan mencoba mengembangkan sistem layar tunggal (*single layer*) pada perceptron menjadi sistem layar jamak (*multilayers*), yang kemudian disebut dengan sistem *backpropagation*. Setelah itu, muncul beberapa model jaringan saraf tiruan lain yang dikembangkan oleh Kohonen (1972), Hopfield (1982), dan lain-lain.

2.3 Arsitektur

Seperti telah dijelaskan sebelumnya bahwa *neuron-neuron* dikelompokkan dalam lapisan-lapisan. Umumnya, neuron-neuron yang terletak pada lapisan yang sama akan memiliki keadaan yang sama. Faktor terpenting dalam menentukan kelakuan suatu neuron adalah fungsi aktivasi dan pola bobotnya. Pada setiap lapisan yang sama, neuron-neuron akan memiliki fungsi aktivasi yang sama. Jaringan saraf tiruan dengan menggunakan suatu aturan yang bersifat menyeluruh (*general rule*) dimana seluruh model jaringan memiliki konsep dasar yang sama. Arsitektur sebuah jaringan akan menentukan keberhasilan target yang akan dicapai karena tidak semua permasalahan dapat diselesaikan dengan arsitektur yang sama.

1. Jaringan dengan lapisan tunggal (*single layer net*)

Jaringan dengan lapisan tunggal hanya memiliki satu lapisan dengan bobot terhubung. Jaringan ini hanya menerima input kemudian secara langsung akan mengolahnya menjadi output tanpa harus melalui lapisan tersembunyi (*hidden layer*).

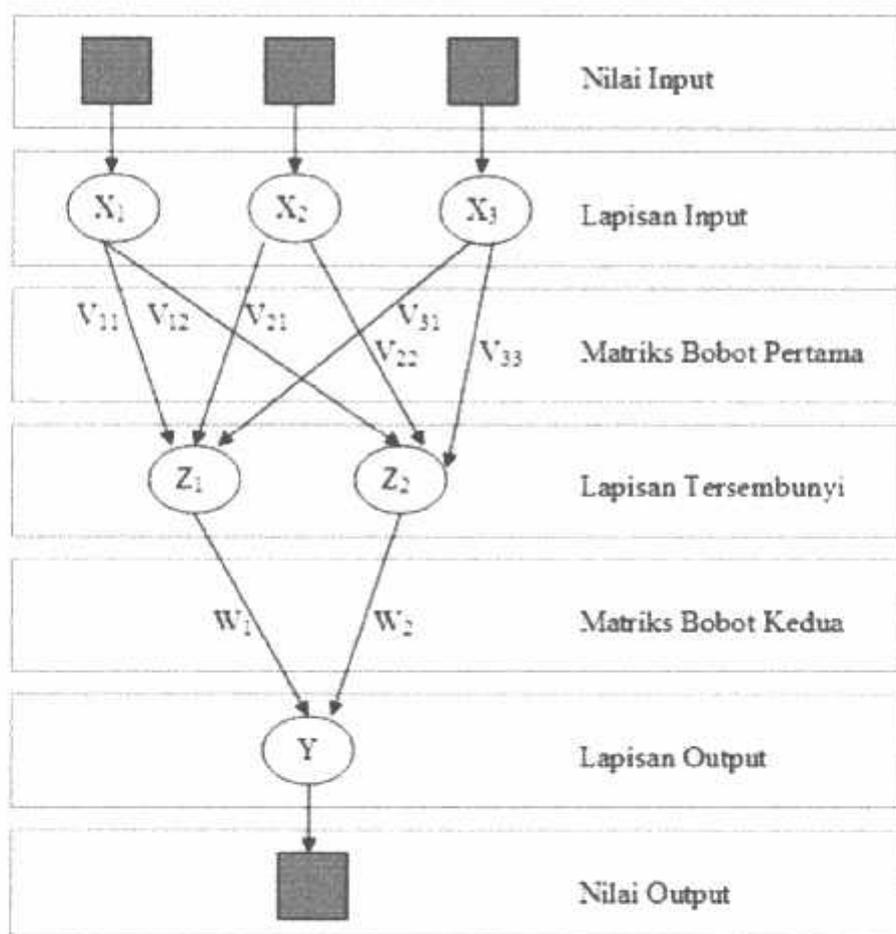


Gambar 2.2 Jaringan saraf dengan lapisan tunggal.

Pada gambar 2.1 diatas, lapisan input memiliki 3 neuron, yaitu X_1 , X_2 dan X_3 . Sedangkan lapisan output memiliki 2 neuron, yaitu Y_1 dan Y_2 . Neuron-neuron pada kedua lapisan saling berhubungan. Seberapa besar hubungan antara 2 neuron ditentukan oleh bobot yang bersesuaian. Semua unit input akan dihubungkan dengan setiap unit output.

2. Jaringan dengan banyak lapisan (*multilayer net*)

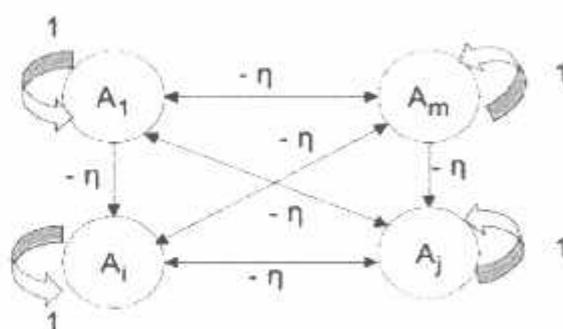
Jaringan dengan banyak lapisan memiliki satu atau lebih lapisan yang terletak diantara lapisan input dan lapisan output seperti terlihat pada Gambar 2.3. Umumnya terdapat lapisan bobot-bobot yang terletak antara dua lapisan yang bersebelahan. Jaringan dengan banyak lapisan ini dapat menyelesaikan permasalahan yang lebih sulit daripada dengan lapisan tunggal, dengan pembelajaran yang lebih rumit.



Gambar 2.3 Jaringan saraf dengan banyak lapisan.

3. Jaringan dengan lapisan kompetitif (*competitif layer net*)

Hubungan antar neuron pada lapisan kompetitif ini tidak diperlihatkan pada diagram arsitektur. Gambar 2.4 menunjukkan salah satu contoh arsitektur jaringan dengan lapisan kompetitif yang memiliki bobot $-\eta$.



Gambar 2.4 Jaringan saraf dengan kompetitif.

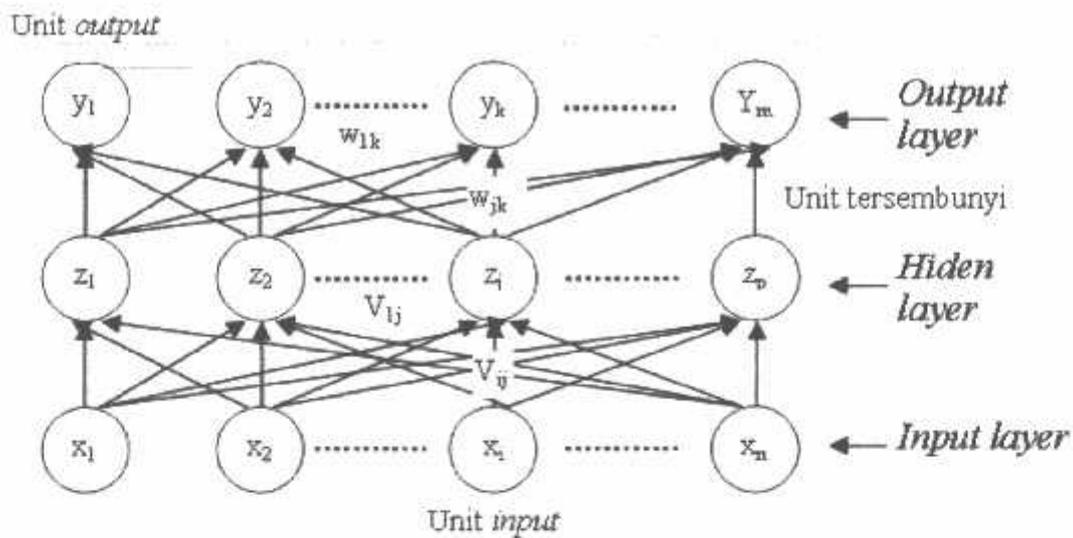
Jaringan saraf tiruan memiliki sejumlah besar kelebihan dibandingkan dengan metode perhitungan lainnya (sistem pakar, statistik, dll) yaitu:

1. Kemampuan mempresentasikan pengetahuan secara fleksibel. JST dapat menciptakan sendiri representasi melalui pengaturan diri sendiri atau kemampuan belajar (*self organizing*).
2. Kemampuan mengakuisisi pengetahuan walaupun dalam kondisi adanya gangguan dan ketidakpastian. Hal ini dikarenakan JST mampu melakukan generalisasi, abstraksi, dan ekstraksi terhadap properti statistik dari data.
3. Kemampuan mentolerir suatu distorsi (*error/fault*). Dimana gangguan kecil pada data dapat dianggap hanya *noise* (guncangan) belaka.
4. Kemampuan memproses pengetahuan secara efisien karena memakai sistem paralel, maka waktu yang diperlukan untuk mengoperasikannya menjadi lebih singkat.
5. Kemampuan untuk memperoleh pengetahuan melalui pembelajaran dari pengalaman.

2.4 Perambatan Galat Mundur (*Backpropagation*)

Jaringan perambatan galat mundur (*backpropagation*) merupakan salah satu algoritma yang sering digunakan dalam menyelesaikan masalah-masalah yang rumit. Hal ini dimungkinkan karena jaringan dengan algoritma ini dilatih dengan menggunakan metode belajar terbimbing. Pada jaringan diberikan sepasang pola yang terdiri atas pola masukan dan pola yang diinginkan. Ketika suatu pola diberikan kepada jaringan, bobot-bobot diubah untuk memperkecil perbedaan pola keluaran dan pola yang diinginkan. Latihan ini berulang-ulang sehingga semu pola yang diketahui jaringan dapat memenuhi pola yang diinginkan.

Algoritma pelatihan jaringan saraf perambatan galat mundur terdiri atas dua langkah, yaitu perambatan maju dan perambatan mundur. Langkah perambatan maju dan perambatan mundur ini dilakukan pada jaringan untuk setiap pola yang diberikan selama jaringan mengalami pelatihan. Jaringan perambatan galat mundur sendiri terdiri atas tiga lapisan atau lebih rumit unit pengolah. Gambar 2.5 menunjukkan jaringan perambatan galat mundur dengan tiga lapisan pengolah, bagian bawah sebagai masukan, bagian tengah disebut sebagai lapisan tersembunyi dan bagian atas disebut lapisan keluaran. Ketiga lapisan ini terhubung secara penuh.



Gambar 2.5 Tiga lapisan jaringan perambatan galat mundur.

Perambatan maju dimulai dengan memberikan pola masukan ke lapisan masukan. Pola masukan ini merupakan nilai aktivasi unit-unit masukan. Dengan melakukan perambatan maju di hitung nilai aktivasi pada unit-unit dilapisan berikutnya. Pada setiap lapisan, tiap unit pengolah melakukan penjumlahan berbobot dan menerapkan fungsi sigmoid untuk menghitung keluarannya.

Untuk menghitung nilai penjumlahan berbobot digunakan rumus :

$$S_j = \sum_{i=0}^n a_i w_{ji}$$

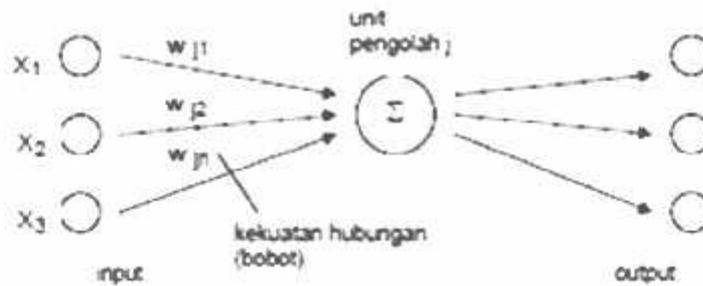
Dengan :

a_i = Masukan yang berasal unit i

w_{ji} = Bobot sambungan dari unit i ke unit j

setelah nilai S_j dihitung. Fungsi sigmoid diterapkan pada S_j untuk membentuk $f(S_j)$. Fungsi sigmoid ini mempunyai persamaan:

$$f(S_j) = \frac{1}{1 + e^{-S_j}}$$



Gambar 2.6 Langkah perambatan maju.

Hasil perhitungan $f(S_j)$ ini merupakan nilai aktivasi j . Nilai ini dikirim ke seluruh keluaran unit j . Setelah perambatan maju selesai dikerjakan maka jaringan siap melakukan perambatan mundur.

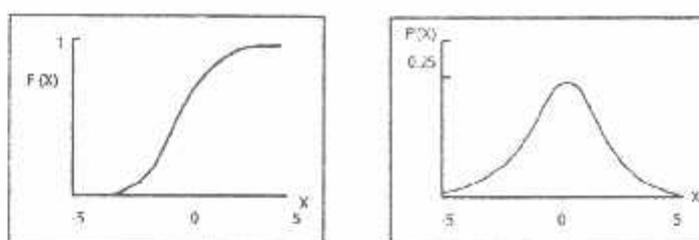
Yang dilakukan pada langkah perambatan mundur adalah menghitung galat dan mengubah bobot-bobot pada semua interkoneksinya. Disini galat dihitung pada semua unit pengolah dan bobotpun diubah pada semua sambungan. Perhitungan dimulai dari lapisan keluaran dan mundur sampai lapisan masukan. Hasil keluaran dari perambatan maju akan dibandingkan dengan hasil keluaran yang diinginkan. Berdasarkan perbedaan ini kemudian dihitung galat untuk tiap-tiap lapisan pada jaringan. Pertama-taman dihitung galat untuk lapisan keluaran (Gambar 2.7a), kemudian bobot-bobot setiap sambungan yang menuju ke lapisan keluaran disesuaikan. Setelah itu dihitung harga galat pada lapisan tersembunyi (Gambar 2.7b) dan dihitung perubahan bobot yang menuju ke lapisan tersembunyi. Demikian proses dilakukan mundur sampai ke lapisan masukan secara iteratif.

Jika j adalah salah satu unit pada lapisan keluaran maka galat lapisan keluaran dapat dihitung dengan rumus :

$$\delta_j = (t_j - a_j) f'(S_j)$$

Dengan :

- T_j = Keluaran yang diinginkan dari unit j
- a_j = Keluaran dari unit j
- $f'(S_j)$ = Turunan dari fungsi sigmoid
- S_j = Hasil penjumlahan berbobot



(a)

(b)

Gambar 2.7 fungsi sigmoid beserta turunannya.

Jika j adalah suatu lapisan tersembunyi, maka galat lapisan tersembunyi dapat dihitung dengan menggunakan rumus :

$$\delta_j = k \left[\sum_k \delta_k w_{kj} \right] f'(s_j)$$

$$\Delta w_{ji} = \eta \delta_j a_i$$

Dengan :

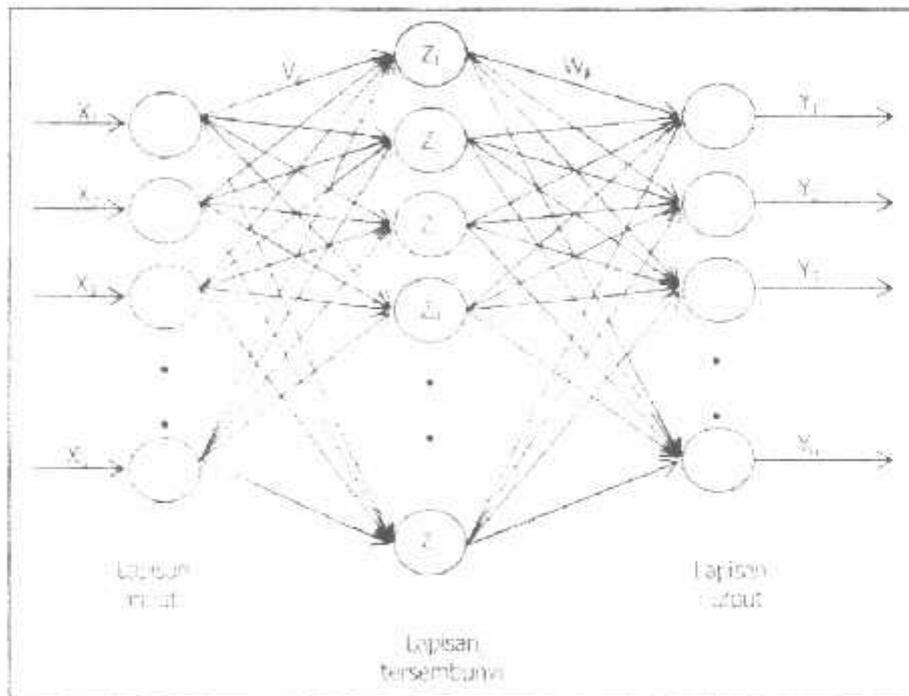
- Δw_{ji} : Perubahan bobot dari unit i ke unit j
- η : Laju belajar/konvergensi
- δ_j : Galat lapisan tersembunyi
- a_i : Masukan yang berasal dari unit i

Variabel η menyatakan suatu konstanta belajar yang berharga antara 0,25-0,75. Nilai ini menunjukkan kecepatan belajar dari jaringan. Nilai yang terlalu tinggi dapat menyebabkan jaringan menjadi tidak stabil sedangkan nilai yang terlalu kecil dapat menjadikan waktu belajar yang lama. Oleh karena itu pemilihan nilai η harus seoptimal mungkin agar didapatkan proses belajar yang cepat.

Jaringan perambatan mundur dilatih dengan metode belajar terbimbing. Pada metode ini jaringan diberi sekumpulan pasangan pola yang terdiri dari pola masukan dan pola yang diinginkan. Pelatihan dilakukan berulang-ulang sehingga dihasilkan jaringan yang memberikan tanggapan yang benar terhadap semua masukannya. Nilai 'benar' disini ditunjukkan dengan nilai RMS/SSE galatnya yang biasanya mempunyai nilai dibawah 0,1. Dengan nilai RMS/SSE di bawah 0,1 maka jaringan sudah boleh dikatakan terlatih.

Untuk mempercepat waktu pelatihan, prosedur perubahan bobot dimodifikasi dengan menggunakan momentum. Dalam perubahan bobot menggunakan momentum nilai bobot ke $t+1$, hasilnya juga ditentukan oleh nilai bobot ke (t) dan ke $(t-1)$, yaitu selisihnya yang dikalikan dengan suatu konstanta momentum (m) yang bernilai antara 0 dan 1.

Kecepatan pelatihan juga dapat ditingkatkan dengan menggunakan laju belajar yang dapat berubah adaptif selama pelatihan. Jika galat yang muncul lebih besar dari galat iterasi sebelumnya maka nilai bobot, prasikap, keluaran, dan galat yang baru disimpan serta laju belajar harus ditingkatkan.



Gambar 2.8 Tiga lapisan perambatan galat mundur

Algoritma selengkapnya pelatihan jaringan *backpropagation* adalah sebagai berikut:

- Langkah 0 : Inialisai bobot-bobot (tetapkan dalam nilai acak kecil).
- Langkah 1 : Bila syarat berhenti adalah salah, kerjakan langkah 2 sampai 9.
- Langkah 2 : Untuk setiap pasangan pelatihan, kerjakan langkah 3- 8

Umpan maju :

Langkah 3 : Tiap unit masukan ($x_i, i=1, \dots, n$) menerima isyarat masukan x_i dan diteruskan ke unit-unit tersembunyi.

Langkah 4 : Tiap unit tersembunyi ($z_j, j=1, \dots, p$) menjumlahkan isyarat masukan terbobot,

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

Dengan menerapkan fungsi aktivasi hitung

$$Z_j = f(z_in_j)$$

dan kirim isyarat ini ke unit-unit keluaran

Langkah 5 : Tiap unit keluaran ($y_k, k=1, \dots, m$) menjumlahkan isyarat masukan terbobot,

$$y_in_k = w_{0k} + \sum_{j=1}^p z_j w_{jk}$$

Dengan menerapkan fungsi aktivasi hitung,

$$y_k = f(y_in_k)$$

Perambatan Balik Galat:

Langkah 6 : Tiap unit keluaran ($y_k, k=1, \dots, m$) menerima pola sasaran berkaitan dengan pola pelatihan masukannya. Hitung galat informasi :

$$\delta_k = (t_k - y_k) f'(y_in_k)$$

Hitung koreksi bobot dan prasikapnya :

$$\Delta w_{jk} = \alpha \delta_k x_j$$

$$\Delta w_{0k} = \alpha \delta_k$$

Langkah 7 : Tiap unit tersembunyi (z_j , $j=1, \dots, p$) menjumlahkan delta masukannya (dari unit-unit di lapisan atasnya).

$$\delta_{in_j} = \sum_{k=1}^m \delta_k w_{jk}$$

Hitung galat informasinya:

$$\delta_j = \delta_{in_j} f'(x_{in_j})$$

Hitung koreksi bobot dan prasikapnya:

$$\Delta v_{ij} = \alpha \delta_j x_i$$

Perbaharui bobot dan prasikap:

Langkah 8 : Tiap unit keluaran (y_k , $k=1, \dots, m$) memperbaharui bobot-bobot dan prasikapnya ($j=0, 1, \dots, p$)

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

Tiap unit tersembunyi (z_j , $j=1, \dots, p$) memperbaharui bobot dan prasikapnya ($j=0, 1, \dots, n$)

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

Langkah 9 : Uji syarat berhenti.

Prosedur pembaharuan bobot-bobot dapat dimodifikasi dengan menggunakan momentum. Dengan menambahkan momentum ke dalam rumus pembaharuan bobot, biasanya konvergensi akan lebih cepat dicapai. Dalam

pembaharuan bobot menggunakan momentum, nilai bobot pada iterasi ke (t+i) ditentukan oleh nilai bobot pada iterasi ke-t dan ke-(t-1).

Rumus pembaharuan bobotnya adalah sebagai berikut:

$$w_{ij}(t+1) = w_{ij}(t) + \alpha \delta_k z_j + \mu [w_{jk}(t) - w_{jk}(t-1)]$$

atau,

$$\Delta w_{ij}(t+1) = \alpha \delta_k z_j + \mu \Delta w_{ij}(t)$$

dan,

$$v_{ij}(t+1) = v_{ij}(t) + \alpha \delta_j x_i + \mu [v_{jk}(t) - v_{jk}(t-1)]$$

atau

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

Dengan:

$X_1 .. X_n$: Masukan

$Y_1 .. Y_n$: Keluaran

$Z_1 .. Z_n$: Nilai lapisan tersembunyi

V_{ij} : Bobot antara lapisan masukan dan lapisan tersembunyi

W_{jk} : Bobot antaral apisant ersembunyid an lapisank eluaran

δ : Galat informasi

α : Konstanta berkelanj utan

μ : Momentum

Kecepatan konvergensi juga dapat ditingkatkan dengan memodifikasi laju belajar menjadi *adaptive* yang berubah selama proses pelatihan. Jika galat yang muncul lebih besar daripada galat sebelumnya maka nilai bobot-bobot, prasikap, keluaran, dan galat yang baru diabaikan, dan nilai laju belajar diturunkan. Jika galat yang muncul lebih kecil daripada galat sebelumnya, maka nilai bobot-bobot, prasikap, keluaran, dan galat yang baru disimpan, dan laju belajar ditingkatkan.

Penggunaan *Backpropagation Network* terdiri dari 2 tahap:

1. Tahap belajar atau pelatihan, di mana pada tahap ini pada *backpropagation* diberikan sejumlah data pelatihan dan target.
2. Tahap pengujian atau penggunaan, pengujian dan penggunaan dilakukan setelah *backpropagation* selesai belajar.

Sedangkan pada tahap belajar atau pelatihan dengan metode *backpropagation* terdiri dari tiga langkah, yaitu:

1. Data dimasukkan ke input jaringan (*feedforward*)
2. Perhitungan dan propagasi balik dari error yang bersangkutan
3. Pembaharuan (*adjustment*) bobot dan bias.

2.5 Microsoft Access

Microsoft Access Adalah sebuah program aplikasi basis dan komputer relasional. Aplikasi ini merupakan anggota dari beberapa aplikasi microsoft office, selain tentunya microsoft word, microsoft excel, dan microsoft PowerPoint. Aplikasi ini menggunakan mesin basis data microsoft Jet Database Engine, dan juga menggunakan tampilan grafis yang intuitif sehingga memudahkan pengguna (*user*).

Pernyataan	Keterangan
TABLE	Untuk menyimpan data
QUERY	Untuk memanipulasi data
FORM	Untuk menampilkan data
REPORT	Untuk membuat laporan

Tabel 2.1 Daftar sejumlah pernyataan Microsoft Access

Microsoft Access dapat menggunakan data yang disimpan didalam format microsoft access, microsoft jet *database engine*, microsoft sql sever, oracle database, atau semua kontainer basis data yang mendukung standart ODBC.

Microsoft access mengizinkan pengembangan yang relatif cepat karena semua tabel basis data, kueri, form, dan report disimpan di dalam berkas basis data miliknya (*.MDB). Untuk membuat Query, Access menggunakan Query Design Grid, sebuah program berbasis grafis yang mengizinkan para penggunanya untuk membuat query tanpa harus mengetahui bahasa

pemrograman SQL. Di dalam Query Design Grid, para pengguna dapat memperlihatkan tabel basis data sumber dari query, dan memilih field-field nama yang hendak dikembalikan oleh proses dengan mengklik dan menyeretnya ke dalam *grid*. *Join* juga dapat dibuat dengan cara mengklik dan menyeret field-field dalam tabel ke dalam field dalam tabel lainnya. Access juga mengizinkan pengguna untuk melihat dan memanipulasi kode SQL jika memang diperlukan.

Bahasa pemrograman yang tersedia di dalam Access adalah Microsoft Visual Basic for Applications (VBA), seperti halnya dalam beberapa aplikasi Microsoft. Dua buah pustaka komponen Component Object Model (COM), untuk mengakses basis data pun disediakan, yaitu Data Access Object (DAO), yang hanya dapat terdapat didalam Access 97, dan ActiveX Data Object (ADO) yang tersedia dalam versi-versi Access terbaru.

2.6 Sejarah Microsoft Access

Microsoft merilis Microsoft Access 1.0 pada bulan November 1992 dan dilanjutkan dengan merilis versi 2.0 pada tahun 1993. Microsoft menentukan spesifikasi minimum untuk menjalankan Microsoft Access 2.0 adalah sebuah komputer dengan sistem operasi Microsoft Windows 3.0, RAM berkapasitas 4 megabyte (6 megabyte lebih disarankan) dan ruangan kosong hard disk yang dibutuhkan 8 megabyte (14 megabyte lebih disarankan). Versi 2.0 dari Microsoft Access ini datang dengan tujuh buah disket floppy 3½ inci berukuran 1.44 megabyte.

Perangkat lunak tersebut bekerja dengan sangat baik pada sebuah basis data dengan banyak record tapi terdapat beberapa kasus di mana data mengalami kerusakan. Sebagai contoh, pada ukuran basis data melebihi 700 megabyte sering mengalami masalah seperti ini (pada saat itu, memang hard disk yang beredar masih berada di bawah 700 megabyte). Buku manual yang dibawanya memperingatkan bahwa beberapa kasus tersebut disebabkan oleh driver perangkat yang kuno atau konfigurasi yang tidak benar.

Nama kode (*codename*) yang digunakan oleh Access pertama kali adalah Cirrus yang dikembangkan sebelum Microsoft mengembangkan Microsoft Visual Basic, sementara mesin pembuat form antarmuka yang digunakannya dinamakan dengan Ruby. *Bill Gates* melihat purwarupa (*prototype*) tersebut dan memutuskan bahwa komponen bahasa pemrograman *BASIC* harus

dikembangkan secara bersama-sama sebagai sebuah aplikasi terpisah tapi dapat diperluas. Proyek ini dinamakan dengan *Thunder*. Kedua proyek tersebut dikembangkan secara terpisah, dan mesin pembuat form yang digunakan oleh keduanya tidak saling cocok satu sama lainnya. Hal tersebut berakhir saat *Microsoft* merilis *Visual Basic for Applications* (VBA).

2.7 Visual Basic 6.0

Bahasa Basic pada dasarnya adalah bahasa yang mudah dimengerti sehingga pemrograman di dalam bahasa Basic dapat dengan mudah dilakukan meskipun oleh orang yang baru belajar membuat program. Hal ini lebih mudah lagi setelah hadirnya *Microsoft Visual Basic*, yang dibangun dari ide untuk membuat bahasa yang sederhana dan mudah dalam pembuatan scriptnya (*simple scripting language*) untuk graphic user interface yang dikembangkan dalam sistem operasi Microsoft Windows.

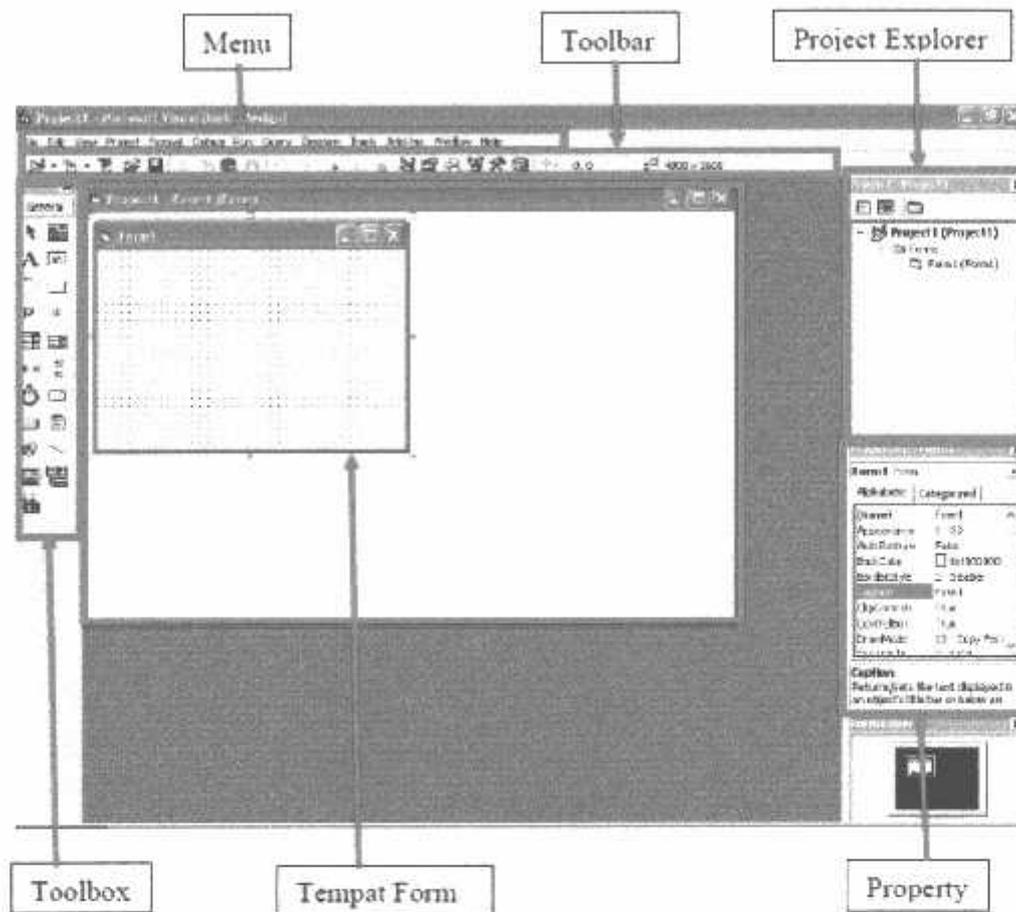
Visual Basic merupakan bahasa pemrograman yang sangat mudah dipelajari, dengan teknik pemrograman visual yang memungkinkan pengguna untuk berkreasi lebih baik dalam menghasilkan suatu program aplikasi. Ini terlihat dari dasar pembuatan dalam visual basic adalah FORM, dimana pengguna dapat mengatur tampilan form kemudian dijalankan dalam script yang sangat mudah.

Ledakan pemakaian Visual Basic ditandai dengan kemampuan Visual Basic untuk dapat berinteraksi dengan aplikasi lain di dalam sistem operasi Windows dengan komponen ActiveX Control. Dengan komponen ini memungkinkan pengguna untuk memanggil dan menggunakan semua model data yang ada di dalam sistem operasi windows. Hal ini juga ditunjang dengan teknik pemrograman di dalam Visual Basic yang mengadopsi dua macam jenis pemrograman yaitu Pemrograman Visual dan *Object Oriented Programming* (OOP).

Visual Basic 6.0 sebetulnya perkembangan dari versi sebelumnya dengan beberapa penambahan komponen yang sedang tren saat ini, seperti kemampuan pemrograman internet dengan DHTML (*Dynamic HyperText Mark Language*), dan beberapa penambahan fitur database dan multimedia yang semakin baik. Sampai saat buku ini ditulis bisa dikatakan bahwa Visual Basic 6.0 masih merupakan pilih pertama di dalam membuat program aplikasi yang ada di pasar perangkat lunak nasional. Hal ini disebabkan oleh kemudahan dalam melakukan proses *development* dari aplikasi yang dibuat.

2.8 Interface Antar Muka Visual Basic 6.0

Interface antar muka Visual Basic 6.0, berisi menu, toolbar, toolbox, form, project explorer dan property seperti terlihat pada gambar 2.9 berikut:

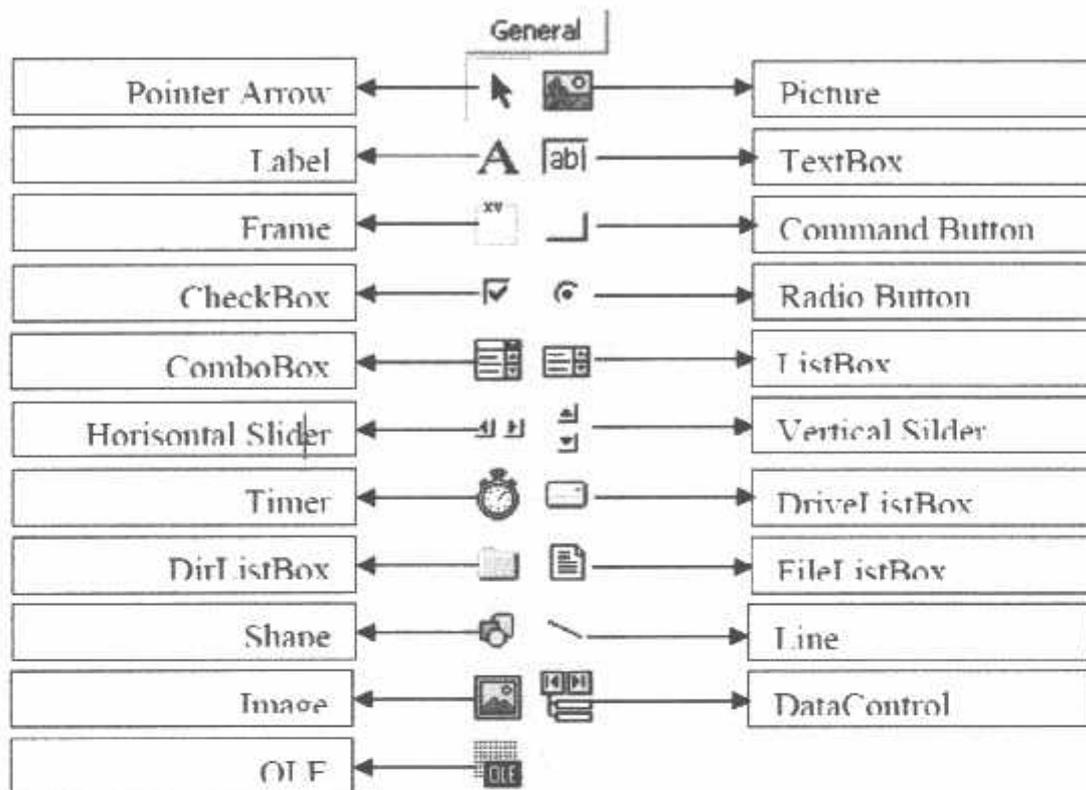


Gambar 2.9 Interface antar muka Visual Basic 6.0.

Pembuatan program aplikasi menggunakan Visual Basic dilakukan dengan membuat tampilan aplikasi pada form, kemudian diberi script program di dalam komponen-komponen yang diperlukan. Form disusun oleh komponen-komponen yang berada di [Toolbox], dan setiap komponen yang dipakai harus diatur propertinya lewat jendela [Property].

Menu pada dasarnya adalah operasional standar di dalam sistem operasi windows, seperti membuat form baru, membuat project baru, membuka project dan menyimpan project. Di samping itu terdapat fasilitas-fasilitas pemakaian visual basic pada menu. Untuk lebih jelasnya *Visual Basic* menyediakan bantuan yang sangat lengkap dan detail dalam MSDN.

Toolbox berisi komponen-komponen yang bisa digunakan oleh suatu project aktif, artinya isi komponen dalam toolbox sangat tergantung pada jenis project yang dibangun. Komponen standar dalam toolbox dapat dilihat pada gambar 2.10 berikut ini.



Gambar 2.10 Komponen standar dalam Toolbox.

2.9 Konsep Dasar Pemrograman Dalam Visual Basic 6.0

Konsep dasar pemrograman Visual Basic 6.0, adalah pembuatan form dengan mengikuti aturan pemrograman **Property, Metode dan Event**. Hal ini berarti:

- (1) **Property**: Setiap komponen di dalam pemrograman Visual Basic dapat diatur propertinya sesuai dengan kebutuhan aplikasi. Property yang tidak boleh dilupakan pada setiap komponen adalah "**Name**", yang berarti nama variabel (komponen) yang akan digunakan dalam scripting. Properti "**Name**" ini hanya bisa diatur melalui jendela Property, sedangkan nilai peroperti yang lain bisa diatur melalui script seperti

```
Command1.Caption="Play"
Text1.Text="Visual Basic"
Label1.Visible=False
```

Timer1.Enable=True

- (2) **Metode:** Bahwa jalannya program dapat diatur sesuai aplikasi dengan menggunakan metode pemrograman yang diatur sebagai aksi dari setiap komponen. Metode inilah tempat untuk mengekspresikan logika pemrograman dari pembuatan suatu program aplikasi.

- (3) **Event:** Setiap komponen dapat beraksi melalui event, seperti event click pada command button yang tertulis dalam layar script Command1_Click, atau event Mouse Down pada picture yang tertulis dengan Picture1_MouseDown. Pengaturan event dalam setiap komponen yang akan menjalankan semua metode yang dibuat.

2.10 Membuat Project Baru

Untuk memulai pembuatan program aplikasi di dalam Visual Basic, yang dilakukan adalah membuat project baru. Project adalah sekumpulan form, modul, fungsi, data dan laporan yang digunakan dalam suatu aplikasi. Membuat project baru dapat dilakukan dengan memilih menu [File] >> [New Project] atau dengan menekan ikon [new project] pada Toolbar yang terletak di pojok kiri atas. Setelah itu akan muncul konfirmasi untuk jenis project dari program aplikasi yang akan dibuat seperti terlihat pada gambar 2.11 berikut :



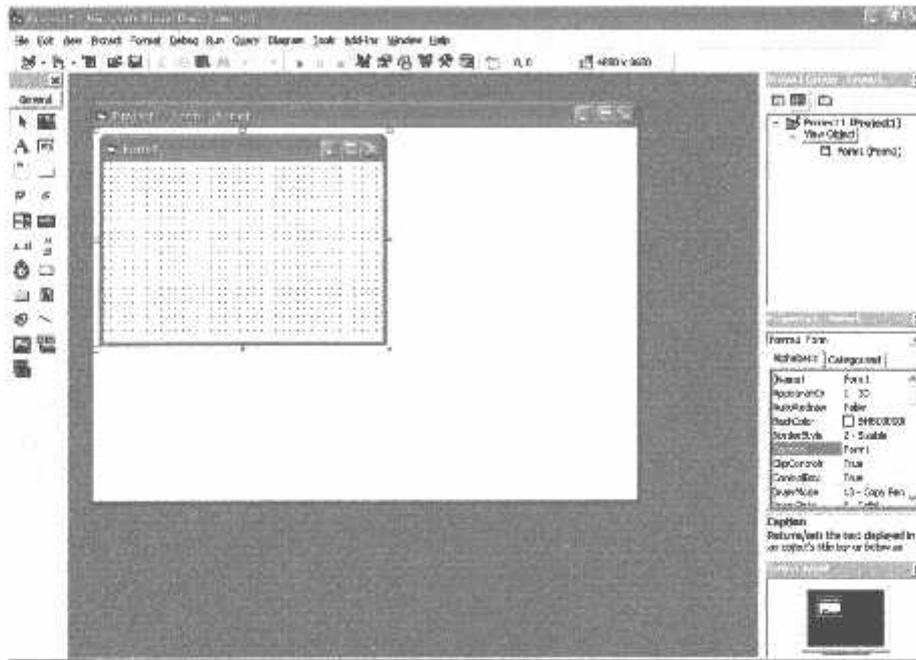
Gambar 2.11 Layar pemilihan jenis project.

Visual Basic 6.0 menyediakan 13 jenis project yang bisa dibuat seperti terlihat pada gambar 2.11 di atas. Ada beberapa project yang biasa digunakan oleh banyak pengguna Visual Basic, antara lain:

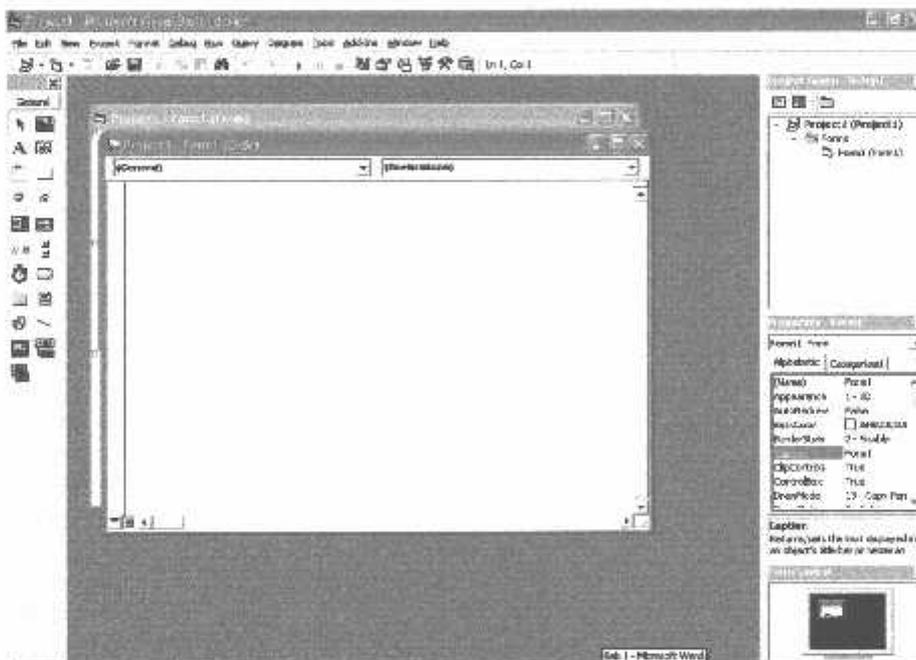
- (1) Standard EXE: Project standar dalam Visual Basic dengan komponen-komponen standar. Jenis project ini sangat sederhana, tetapi memiliki keunggulan bahwa semua komponennya dapat diakui oleh semua unit komputer dan semua user meskipun bukan administrator. Pada buku ini akan digunakan project Standard EXE ini, sebagai konsep pemrograman visualnya.
- (2) ActiveX EXE: Project ini adalah project ActiveX berisi komponen-komponen kemampuan untuk berinteraksi dengan semua aplikasi di sistem operasi windows.
- (3) ActiveX DLL: Project ini menghasilkan sebuah aplikasi library yang selanjutnya dapat digunakan oleh semua aplikasi di sistem operasi windows.
- (4) ActiveX Control: Project ini menghasilkan komponen-komponen baru untuk aplikasi Visual Basic yang lain
- (5) VB Application Wizard: Project ini memandu pengguna untuk membuat aplikasi secara mudah tanpa harus pusing-pusing dengan perintah-perintah pemrograman.
- (6) Addin: Project seperti Standard EXE tetapi dengan berbagai macam komponen tambahan yang memungkinkan kebebasan kreasi dari pengguna.
- (7) Data project: Project ini melengkapi komponennya dengan komponen-komponen database. Sehingga bisa dikatakan project ini memang disediakan untuk keperluan pembuatan aplikasi database.
- (8) DHTML Application: Project ini digunakan untuk membuat aplikasi internet pada sisi client (*client side*) dengan fungsi-fungsi DHTML.
- (9) IIS Application: Project ini menghasilkan aplikasi internet pada sisi server (*server side*) dengan komponen-komponen CGI (*Common Gateway Interface*).

Selanjutnya pilih Standard EXE dan tekan [Ok]. Lalu muncul tampilan dari Standard Exe seperti pada gambar 2.11. Dengan demikian project sudah siap dibuat. Dalam pembuatan project sebelumnya double click pada form yang terbuat maka akan terlihat jendela tersembunyi (*hidden windows*) yang berupa jendela untuk pembuatan program atau jendela kode (*code windows*). Hal ini

Dapat dilakukan dengan cara memilih ikon jendela form atau jendela kode yang ada di [Project Explorer]. Lihat pada gambar 2.12 dan gambar 2.13.



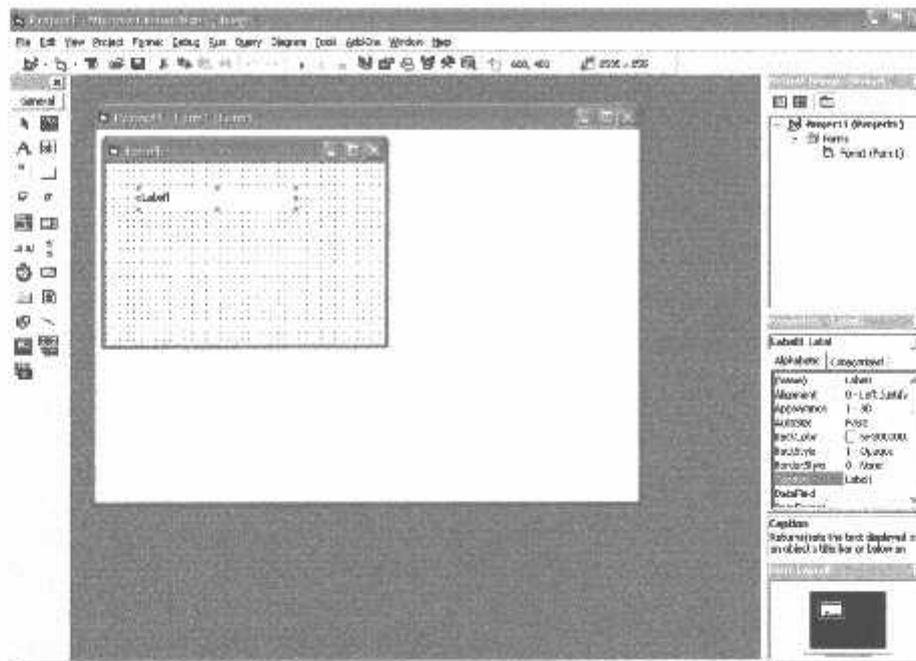
Gambar 2.12 Jendela Form



Gambar 2.13 Jendela Kode

Pada jendela form, pengguna dalam membangun tampilan dari program aplikasi yang akan dibuat dengan mengatur komponen-komponen baik letak, properti dan eventnya. Untuk mengambil suatu komponen dari [Toolbox] dapat dilakukan dengan click komponen tersebut, kemudian klik atau tarik pada posisi

yang benar pada form. Sebagai contoh mengambil label dari Toolbox dapat dilakukan dengan cara seperti gambar 2.14 di bawah ini.

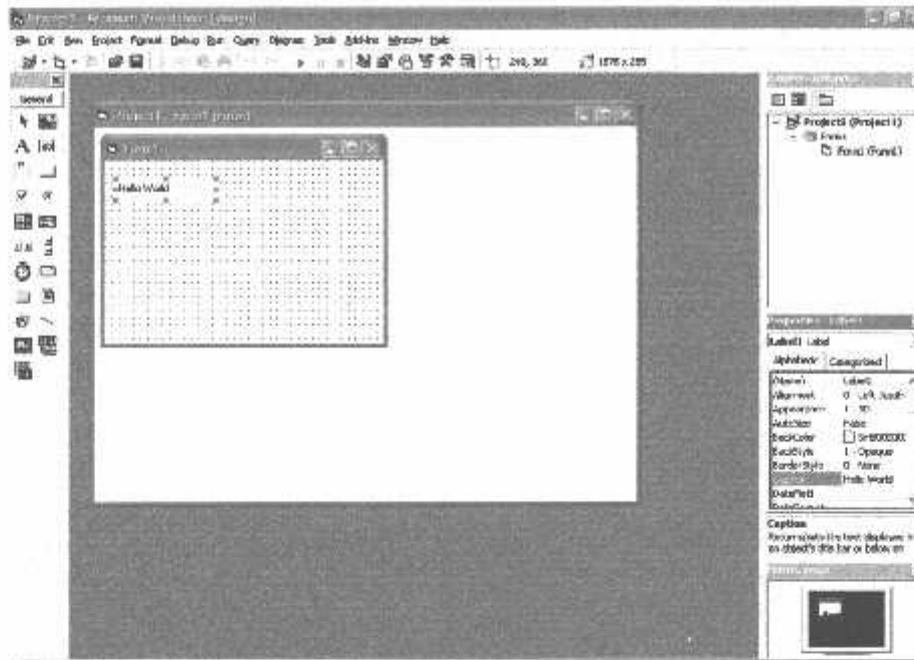


Gambar 2.14 Cara mengambil label dari Toolbox

Langkah-langkah mengambil label dari toolbox untuk dipasangkan dalam form adalah sebagai berikut:

- (1) Click ikon [Label] pada [ToolBox]
- (2) Pindahkan ke posisi dimana label itu akan diletakkan
- (3) Click dan tarik sampai ukurannya benar lalu lepaskan

Langkah berikutnya adalah memberikan teks pada label, misalkan "Hello world", maka pilihlah properti Caption, dan isi dengan Hello world. Hasil tampilan program adalah sebagai berikut:



Gambar 2.15 Layout pada form

Untuk menjalankan program click ikon Run (▶) pada toolbar atau pilih menu [Run] >> [Start], atau dengan tekan tombol [F5].

2.11 Thresholding

Thresholding adalah suatu cara untuk mengubah citra yang memiliki level grayscale atau true colour menjadi citra dengan level warna yang lebih sedikit, dalam kasus ini digunakan level warna bilevel. Citra bilevel adalah suatu citra yang jumlah warnanya terbagi menjadi dua buah warna yaitu 0 (hitam) dan 1 (putih). Dengan menggunakan *thresholding* maka derajat keabuan bisa diubah sesuai dengan keinginan. Proses *thresholding* ini pada dasarnya adalah proses perubahan kuantisasi pada citra.

2.12 Scan

Pada proses *scanning* penulis menggunakan scan dengan kualitas gambar yang baik dan dapat terlihat jelas. Hal ini dilakukan agar gambar yang dihasilkan dari proses scan sample dapat terlihat jelas dan dapat diproses lebih lanjut. Scanner yang digunakan adalah scanner *Canon Lide 25*. Untuk mengetahui spesifikasi dari scanner adalah sebagai berikut:



Gambar 2.16 Canon Lide 25 Scanner

Spesifikasi :

- Scanner type Flatbed.
- Optical resolution 1200 x 2400 dpi.
- Scanner element CIS (Contact Image Sensor).
- Light source 3-colour (RGB) LED.
- Selectable resolution*¹ 25 – 19200 dpi (through software interpolation).
- Scanning bit depth.
- Colour.
- Grayscale.
- 16-bit input, 8-bit output for each colour (RGB).
- 16-bit input, 8-bit output.
- Scanning Speed.
- Colour.
- Grayscale.
- 16 msec/line (600 dpi), 32 msec/line (1200 dpi).
- 5.3 msec/line (600 dpi), 10.6 msec/line (1200 dpi).
- Preview speed *¹ Approx. 16 seconds.
- Interface USB (1.1 equivalent)*².
- Max. document size A4, 216 x 297 mm.
- Scanner buttons (EZ buttons) 3 buttons: Copy, Scan, E-mail.
- Weight Approx. 1.5 kg

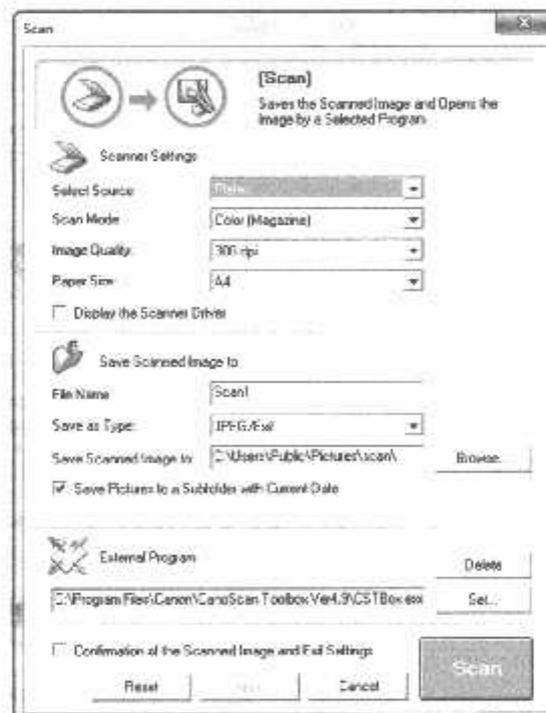
Pada proses scanning yang dilakukan didalam scanner terdapat 2 cara yaitu dengan cara mengklik pada komputer dengan menggunakan aplikasi *CanoScan Toolbox*. Cara kedua adalah dengan menekan tombol *Scan* yang terdapat pada

bagian depan dari *Scanner Canon Lide 25*. Berikut adalah tampilan dari aplikasi *CanoScan Toolbox*.



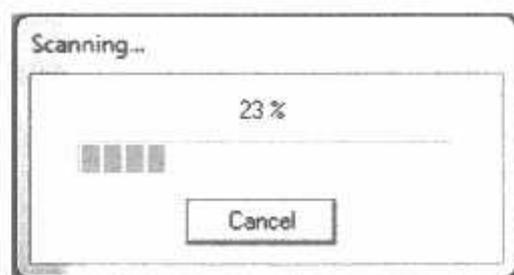
Gambar 2.17 Tampilan Aplikasi CanoScan Toolbox

Setelah Program CanoScan terinstal lakukan proses scanning dengan cara menjalankan aplikasi tersebut. Pada menu terdapat berbagai macam submenu seperti *Copy*, *Print*, *Mail*, *OCR*, *Save*, *PDF*, *Scan-1*, *Scan-2*, *Setting*. Untuk Menu yang digunakan disini ialah menu scan. Klik *scan-1* untuk memulai proses *scanning*.



Gambar 2.18 Tampilan aplikasi Scan-1

Pada menu Scan tahap selanjutnya adalah dengan melakukan proses scanning. Klik tombol scan untuk menjalankan program. Lihat Gambar 2.18.



Gambar 3.19 Tampilan Proses *Scanning*

BAB III

ANALISA DAN PERANCANGAN SISTEM

3.1 Analisa Sistem

Pemahaman konsep dasar dari pembuatan aplikasi dengan metode *backpropagation* sangat diperlukan. Maka dari itu diperlukan referensi guna sebagai penyokong pembuatan aplikasi yang akan dirancang dibangun dan diimplementasikan. Data yang digunakan dalam pengujian ini merupakan data yang bervariasi yang telah disimpan ke dalam *database*. *Database* tersebut berisi informasi tentang data atau ciri dari masing-masing huruf. *Sample* atau contoh dari tulisan tangan berjumlah 50 *sample* dan untuk pemahaman mengenai kesulitan sangat dibutuhkan guna efisiensi di dalam pembuatan aplikasi. Karena dari aspek tersebut aplikasi dapat dibuat semaksimal mungkin.

3.2 Deskripsi Umum Sistem

Aplikasi yang dikembangkan dalam skripsi ini adalah Aplikasi Deteksi Huruf Tulisan Tangan yang dapat mengidentifikasi tulisan tangan seseorang. Dalam mendeteksi tulisan tangan dibutuhkan masukan yaitu data atau ciri dari masing-masing huruf. Pengguna (*user*) hanya perlu memasukan file gambar hasil tulisan yang telah di scan ke dalam aplikasi. Sistem yang akan melakukan presentase kemiripan tiap huruf terhadap huruf yang ada di dalam *database* dengan metode *backpropagation* dan mencari nilai *error* yang didapat. Dengan adanya aplikasi ini diharapkan *user* dapat mengetahui dan mengimplementasikan metode dari *backpropagation* ke dalam dunia kerja.

3.3 Fitur Aplikasi

Pada fitur tampilan terdapat beberapa form yaitu :

3.3.1 Training

Pada menu Training terdapat submenu yang terdiri dari *Image Processing*, *Weight Initialize* dan *training*. Fungsi dari *Image Processing* adalah proses dimana tiap gambar huruf dikenali yaitu dari abjad a sampai z. Setelah *Image Processing* selesai dijalankan langkah selanjutnya adalah melakukan perintah *Weight Initialize*. Proses ini bertujuan untuk mengenali pembobot yang akan

dimasukkan. Setelah nilai telah dimasukkan, maka dapat dilakukan proses *training*.

3.3.2 Test

Pada menu test terdiri dari beberapa submenu, diantaranya adalah Image Processing, Detect dan Exit. Pada saat submenu Image Processing diklik maka akan muncul pola biner dari tiap-tiap nilai masukkan. Setelah itu dilakukan lagi proses *detect*. Proses ini bertujuan untuk mengetahui hasil (*output*) dari proses keseluruhan program. Bila pola biner cocok dengan *database* maka akan muncul tampilan hasil huruf yang diinginkan.

3.3.3 Exit

Setelah semua proses selesai maka klik tombol *Exit* untuk keluar dari program.

3.4 Analisa Kebutuhan Sistem

Untuk perangkat yang akan digunakan merancang aplikasi ini adalah sebagai berikut :

1. Laptop Intel Core i3 M380.
2. RAM DDR3 2 *Gigabyte* (GB).
3. Hardisk 360 *Gigabyte*.
4. VGA card *onboard*.
5. Scanner Canon Lide 25.

Selain perangkat yang digunakan sebagai pembuatan aplikasi, membutuhkan spesifikasi perangkat-perangkat lunak sebagai berikut :

1. Operating System Windows 7.
2. Microsoft Visual Basic 6.0.
3. Microsoft Access 2007.

3.5 Teknik Pengumpulan Data

Dalam teknik pengumpulan data penulis menggunakan beberapa *sample* tulisan dari beberapa orang. Data yang sudah terkumpul tersebut lalu di scan dan diolah ke dalam *database*. Pada hasil crop tiap-tiap huruf berukuran 10x10 pixel lalu disimpan kedalam objek gambar yang berkekestensi *.bmp. Dalam gambar 3.1

menunjukkan contoh dari beberapa *sample* tulisan tangan. Alur pengolahan data yang dilakukan dalam aplikasi antara lain :

a	b
c	d
e	f
g	h
i	j
k	l
m	n
o	p
q	r
s	t
u	v
w	x
y	z

Gambar 3.1 Contoh Sample

- Langkah 1 : Inisialisasi bobot dari data *sample* yang sudah ada.
- Langkah 2 : Setelah melakukan inisialisasi bobot masukkan nilai pembobot dan lakukan proses *test*.
- Langkah 3 : Selesai memasukkan nilai pembobot maka akan muncul pola yang nantinya akan di *test* untuk mendapatkan nilai outputnya.
- Langkah 4 : Bila hasil output telah didapatkan dan sesuai dengan data yang ada didalam database maka deteksi huruf telah berhasil. Akan tetapi jika di dalam proses *test* tersebut belum ditemukan hasil outputnya maka proses deteksi belum bisa dikatakan berhasil dan harus di uji lagi.

3.6 Prapengolahan

Yang dilakukan pada tahap ini adalah menyederhanakan citra sehingga siap untuk dianalisis. Pada proses prapengolahan ini, citra objek warna diubah menjadi keabuan kemudian diambangkan. Citra hasil proses pengambangan adalah citra biner dengan latar-belakang berwarna hitam dan bagian latar depan yang merupakan citra yang akan diproses berwarna putih. Setelah itu dicari batas atas, batas bawah, batas kanan, dan batas kiri dari citra dan mengubah ukuran citra menjadi berukuran 10x10 piksel. Citra inilah yang siap untuk dianalisis pada proses ekstraksi ciri.

3.6.1 Ekstraksi Ciri

Citra hasil prapengolahan, dibagi menjadi 10 baris dan 10 kolom. Pada setiap kotak dilakukan penghitungan piksel, bila ditemukan jumlah piksel 1 berwarna hitam, maka kotak tersebut diberi nilai nol, bila tidak maka kotak tersebut diberi nilai satu. Hasilnya disimpan dalam matriks 10x10.

3.7 Perancangan Sistem

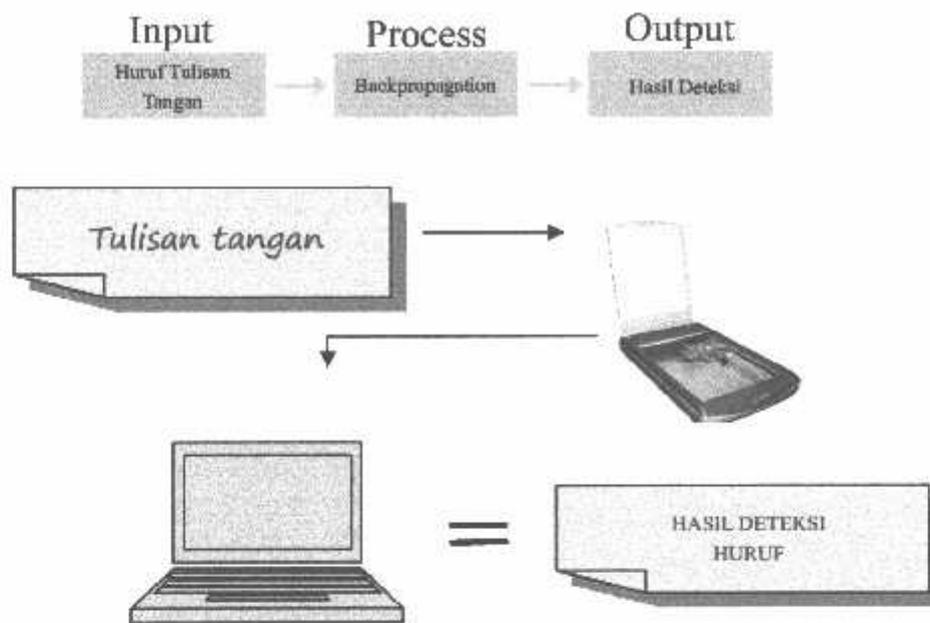
Rancangan dan langkah-langkah yang dilakukan dalam memberikan deteksi huruf tulisan tangan menggunakan Jaringan saraf Tiruan *Backpropagation*.

Penelitian dilakukan dengan langkah-langkah sebagai berikut :

1. Mempelajari metode yang digunakan, yaitu *Backpropagation* dan huruf tulisan tangan.
2. Menganalisa dan merancang sistem dengan menggunakan hasil pembelajaran pada tahap sebelumnya.
3. Membuat sistem berdasarkan analisis dan perancangan yang dilakukan.
4. Uji coba sistem.
5. Evaluasi hasil uji coba.

3.8 Desain Sistem Aplikasi

Pada bagian desain Sistem Aplikasi ini akan diuraikan mulai dari suatu sistem yang utuh ke dalam bagian-bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan yang ada. Bagian ini terdiri atas gambaran alur sistem dalam bentuk *flowchart* dan penjelasan dari setiap tahapan dalam gambaran alur sistem.



Gambar 3.2 Gambar Desain Sistem

3.9 Gambaran Umum Perangkat Lunak

Sistem yang dikembangkan adalah implementasi dari algoritma *backpropagation* dalam bentuk aplikasi untuk mendeteksi huruf tulisan tangan. Sistem akan mendeteksi huruf tulisan tangan yang didapat dari beberapa sample tulisan tangan dan akan memberikan keluaran berupa hasil dari proses deteksi tersebut.

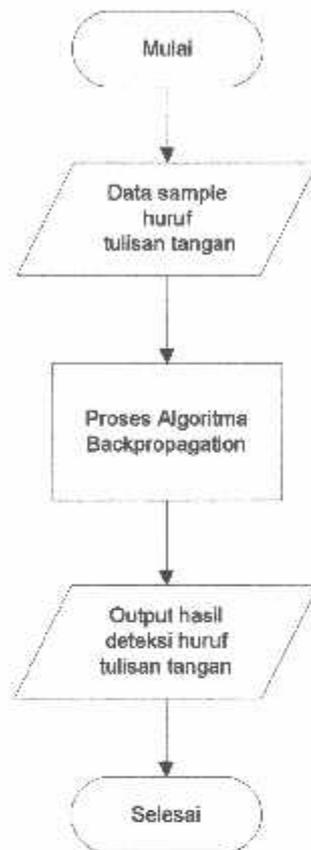
Dari masukan tersebut akan dikelompokkan menjadi 26 buah huruf. Pada tahap pembelajaran menggunakan metode *feedforward*, lalu akan dihitung besarnya nilai error. Jika kesalahan yang ditargetkan belum didapat, maka dilakukan *backpropagation* dan dilatih hingga mendapatkan *error* yang minimum. Bila nilai *error* minimum, maka bobot akan disimpan dalam data penyimpanan. Tempat penyimpanan menggunakan *microsoft access*.

Pada tahap pengujian sistem, dengan menggunakan bobot-bobot akhir yang telah diperoleh selama proses pembelajaran, maka dihasilkan pengelompokan terhadap data pengujian.

Langkah-langkah untuk memprediksi huruf tulisan tangan menggunakan jaringan saraf tiruan *backpropagation* diilustrasikan dalam gambar 3.3 dengan tahapan sebagai berikut :

1. Data masukkan (*input*) gambar *scan* huruf tulisan tangan.

2. Proses algoritma *backpropagation* terdiri dari proses pelatihan dan pengujian data.
3. Keluaran (*Output*), merupakan hasil dari proses deteksi.

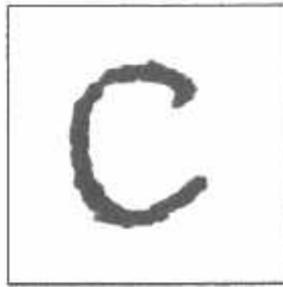


Gambar 3.3 Diagram Alir sistem secara umum

3.10 Penejelasan Desain Sistem

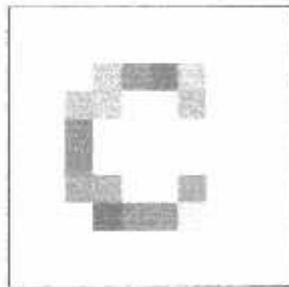
3.10.1 Input Tulisan Tangan

Pengenalan citra pada gambar sangat penting didalam penelitian ini. Dari hasil *sample* yang didapat penulis mengambil beberapa *sample* yang nantinya akan diproses untuk mendapatkan pola yang terbaik. Pada contoh gambar diatas dijelaskan bahwa proses inputan yang dilakukan adalah berupa *sample* tulisan tangan. Sebagai contoh pada gambar 3.4 merupakan *sample* huruf yang telah di *scan*.



Gambar 3.4 Contoh sample huruf

Pada huruf "C" telah dilakukan proses *cropping* dengan ukuran panjang 6 cm dan lebar 6 cm dengan piksel berukuran 118x118 piksel. Setelah itu proses selanjutnya adalah melakukan proses *resize* pada piksel. Untuk proses *resize* piksel yang berukuran 118x118 akan dirubah menjadi 10x10 piksel dengan 10 baris dan 10 kolom. Berikut adalah gambar hasil dari proses *resize* :



Gambar 3.5 Hasil Proses *Resize*

Setelah dilakukan proses *resize*, terlihat jelas perbedaan gambar dari huruf pertama dan kedua (gambar 3.4) dan (gambar 3.5). proses penyimpanan gambar tersebut berekstensi *.*bmp* dan berukuran 10x10 piksel.

3.11 Penetapan Masukan (*Input*)

Data masukan yang digunakan didalam aplikasi terdiri dari dua jenis. Masukan data pertama adalah data *sample* huruf tulisan tangan dan masukan data yang kedua adalah jaringan saraf tiruan untuk pelatihan jaringan. Data *sample* huruf tulisan tangan yang digunakan akan dijadikan data *biner*. Hasil pengolahan data *sample* digunakan sebagai masukan data pelatihan jaringan saraf tiruan.

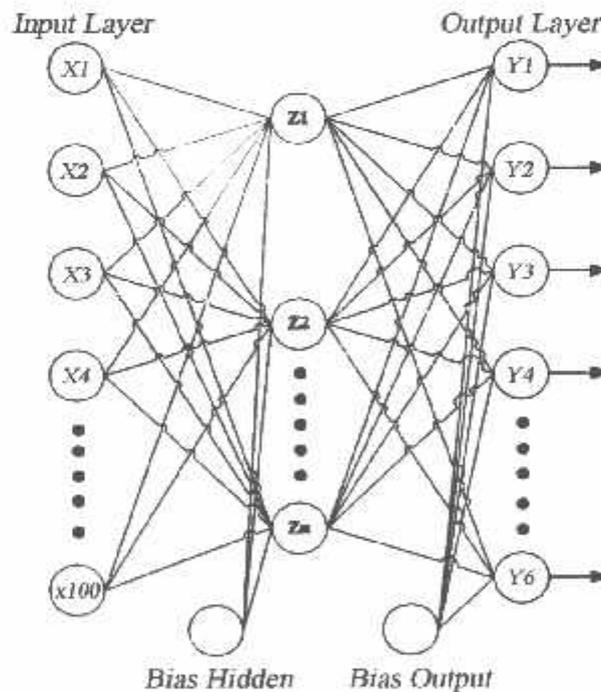
3.12 Proses Jaringan Saraf Tiruan *Backpropagation*

Proses dibagi menjadi tiga, yaitu proses membuat arsitektur jaringan, proses pelatihan dan proses pengujian.

3.12.1 Arsitektur

Arsitektur dari backpropagation terdiri dari lapisan masukan (*input layer*), lapisan tersembunyi (*hidden layer*), dan lapisan keluaran (*output layer*). Arsitektur jaringan saraf yang digunakan terdiri dari neuron. Suatu algoritma belajar dari jaringan saraf tiruan tergantung dari arsitektur atau struktur dari jaringan saraf tersebut. Dengan arsitektur tersebut menggambarkan jumlah neuron dan hubungan bobot yang dipakai. Secara umum arsitektur jaringan saraf dapat dibedakan menjadi tiga kelas yang berbeda, yaitu :

1. Jaringan umpan-maju lapisan tunggal (*single layer feedforward network/Perceptron*).
2. Jaringan umpan-maju lapisan jamak (*multilayer perceptron*).
3. Jaringan saraf tiruan *recurrent*.



Gambar 3.6 Arsitektur jaringan saraf tiruan *Backpropagation*

Proses belajar dari jaringan saraf perambatan-balik (*Backpropagation*) ini secara garis besar adalah setelah menerima masukan pada lapisan masukan,

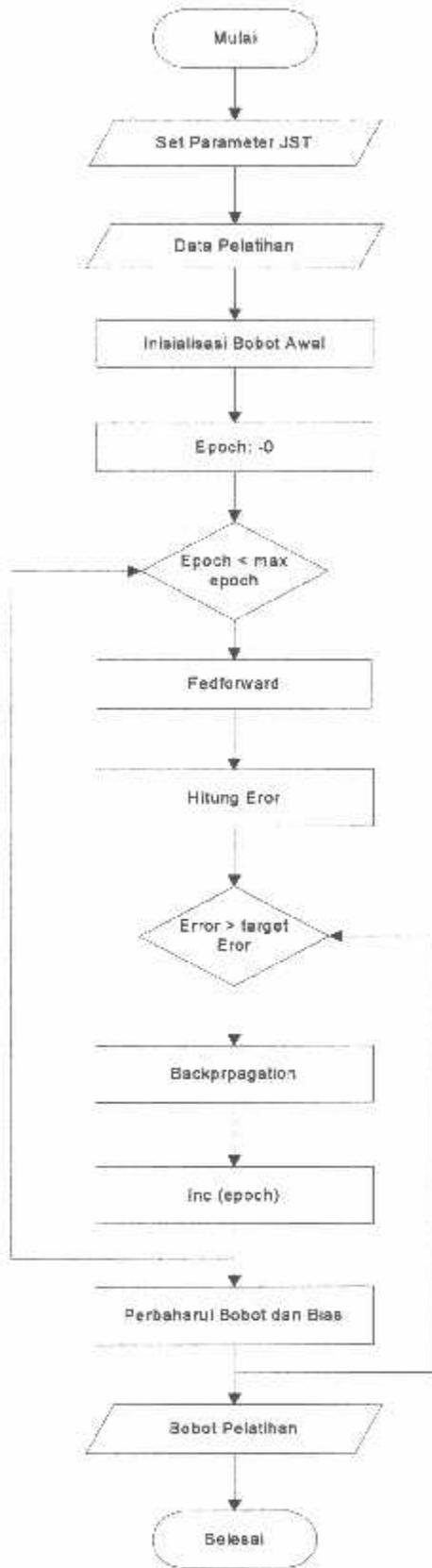
maka masukan itu akan dirambatkan melewati setiap lapisan di atasnya hingga suatu keluaran dihasilkan oleh jaringan. Keluaran yang dihasilkan oleh jaringan akan dibandingkan dengan acuan keluaran, sehingga suatu galat akan dibangkitkan. Selanjutnya, jaringan akan melewatkan turunan dari galat tersebut ke lapisan tersembunyi dengan menggunakan sambungan berbobot yang masih belum diubah nilainya. Kemudian setiap neuron pada lapisan tersembunyi akan menghitung jumlah bobot dari galat yang dirambatbalikkan sebelumnya. Galat perambatan-balik inilah yang memberi nama jaringan ini sebagai jaringan perambatan-balik (*backpropagation*). Setelah masing-masing neuron dari lapisan tersembunyi dan lapisan keluaran menemukan besarnya galat, maka neuron-neuron ini akan mengubah nilai bobotnya untuk mengurangi besarnya galat. Proses ini berlangsung terus-menerus hingga galat yang dihasilkan oleh jaringan tersebut mendekati nol. Jaringan saraf jenis inilah yang digunakan dalam tugas akhir ini.

3.12.2 Proses Pelatihan

Algoritma pelatihan *backpropagation* terdiri dari tiga langkah, yaitu: langkah maju (*feedforward*), propagasi balik (*backpropagation*), dan perubahan bobot.

Langkah-langkah terdapat pada gambar 3.7 yaitu:

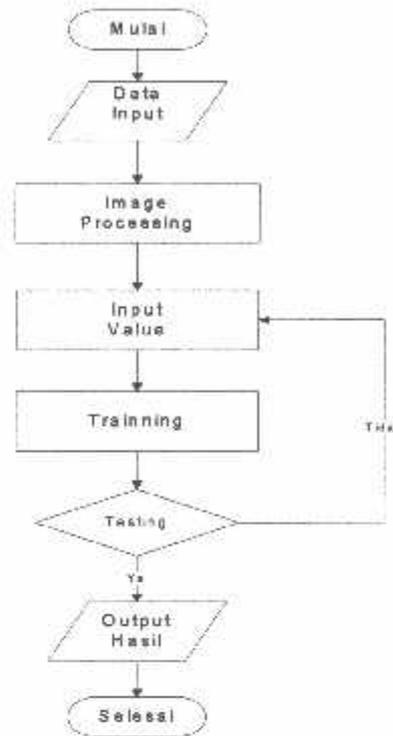
1. Mulai.
2. Tentukan kesalahan yang ditargetkan, epoch maksimum, dan neuron hidden.
3. Masukan data yang dilatih.
4. Inisialisasi bobot awal, yang berkisar antara -0,05 sampai 0,05.
5. Tentukan parameter epoch sama dengan 0 (nol).
6. Lakukan *feedforward* setelah jaringan terisi bobot.
7. Hitung kesalahan antara pola keluaran dan pola target.
8. Apakah kesalahan keluaran lebih besar dari kesalahan yang ditargetkan?
Jika ya lakukan langkah 9, jika tidak, maka lakukan langkah 10.
9. Lakukan *backpropagation*.
10. Proses berhenti, bobot dan bias akhir pelatihan disimpan dan siap untuk digunakan dalam proses pengujian.
11. Selesai.



Gambar 3.7 Diagram alir proses pelatihan jaringan saraf tiruan *backpropagation*

3.13 Flowchart Aplikasi

Flowchart adalah penggambaran dari langkah-langkah dan urutan prosedur dari suatu program. *Flowchart* dapat berfungsi untuk memecahkan masalah kedalam segmen-segmen yang lebih kecil dan menolong dalam menganalisis alternatif-alternatif lain dalam pengoperasian suatu program. Dalam gambar 3.8 berikut ini menunjukkan *flowchart* dari keseluruhan sistem.



Gambar 3.8 *Flowchart* Aplikasi

3.13.1 Penjelasan *Flowchart*

Algoritma dari *flowchart* pada aplikasi Deteksi Huruf ini dapat dijabarkan sebagai berikut :

1. **Mulai** untuk menjalankan program/aplikasi.
2. **Data Input** merupakan data dari beberapa *sample* yang telah dimasukkan ke dalam *database*.
3. **Image Processing** merupakan proses pengenalan gambar *scan* huruf dari tiap-tiap huruf yang didapat dari beberapa *sample* yang sudah ada didalam *database* yaitu huruf abjad dari a sampai huruf z.
4. **Input Value** merupakan pemberian nilai pembobot yang telah disesuaikan dengan nilai konstanta yang berlaku pada metode *backpropagation* yaitu bernilai antara 0,25 sampai 0,75.

5. **Training** bertujuan untuk mendapatkan nilai error yang ada. Nilai ‘benar’ disini ditunjukkan dengan nilai RMS/SSE galatnya yang biasanya mempunyai nilai di bawah 0,1. Dengan nilai RMS/SSE di bawah 0,1 maka jaringan sudah boleh dikatakan terlatih.
6. **Testing** yang berisikan perintah uji kelayakan dari proses *training*. Jika hasil yang didapat sudah memenuhi syarat yaitu nilai RMS/SSE dibawah 0,1 maka akan tampak huruf abjad tersebut. Jika nilai RMS/SSE diatas 0,1 maka jaringan masih belum bisa dikatakan terlatih dan harus dilakukan proses *input value* yaitu proses pemberian nilai bobot.
7. **Output Hasil** merupakan hasil akhir dari proses deteksi huruf.
8. **Selesai** untuk mengakhiri program/aplikasi.

3.14 Perancangan Jaringan Saraf Tiruan

Jaringan Saraf Buatan yang dibuat dalam program aplikasi di sini dirancang dengan struktur sebagai berikut :

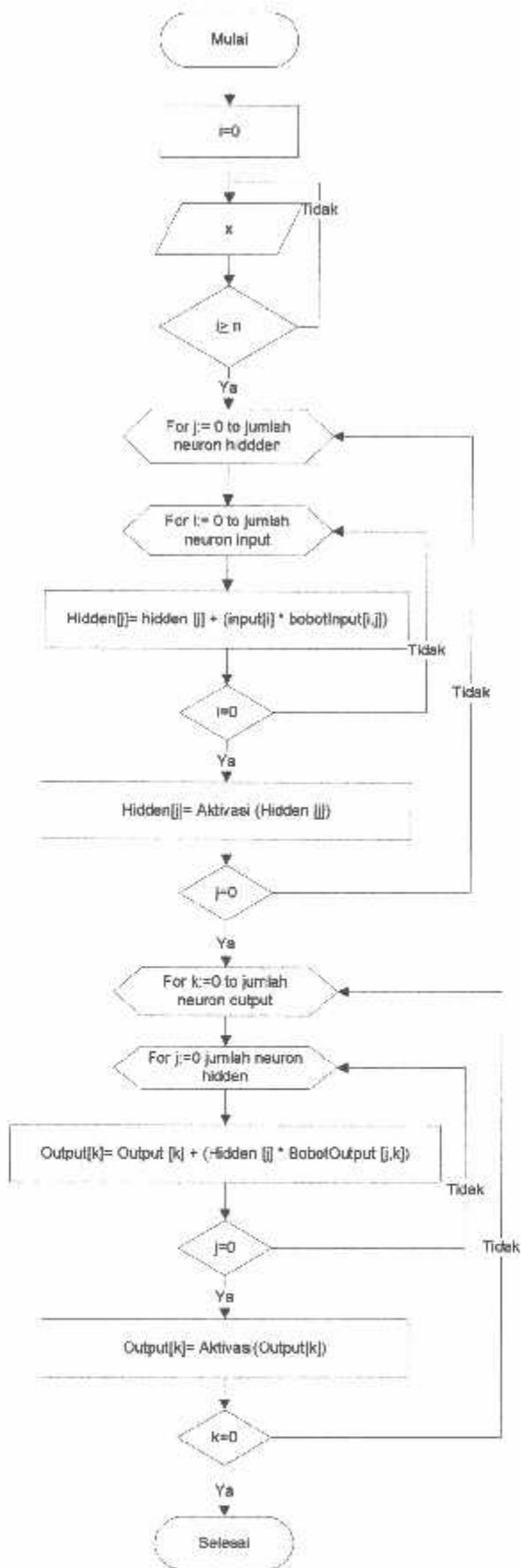
- a. Terdiri atas tiga lapisan *neuron*, yaitu lapisan input (*input layer*), satu lapisan tersembunyi (*hidden layer*), dan lapisan output (*output layer*).
- b. Lapisan input terdiri atas lima *neuron*. Kelima neuron ini menjadi penghubung antara jaringan dengan lingkungannya dan tidak melakukan perubahan apapun terhadap data atau sinyal yang diperolehnya. Setiap *neuron* pada lapisan input ini diberi nomor urut 1 hingga 5 dan seterusnya.
- c. Lapisan tersembunyi terdiri atas sepuluh buah *neuron*. Setiap *neuron* diberi nomor urut 6 hingga 15. Setiap *neuron* pada lapisan ini menerima sinyal aktivasi berupa akumulasi perkalian antara hasil keluaran *neuron-neuron* pada lapisan sebelumnya (lapisan *input*) dengan bobot, yang dikoreksi dengan nilai bias.
Keluaran dari setiap *neuron* pada lapisan ini merupakan fungsi dari sinyal aktivasi, dimana pada aplikasi ini digunakan fungsi *sigmoid*.
- d. Lapisan *output* terdiri atas dua buah *neuron*, yang berkarakteristik sama dengan *neuron-neuron* pada lapisan tersembunyi, kecuali pada sumber sinyal masukannya. *Neuron* pada lapisan ini mendapat masukan berupa sinyal dari *neuron-neuron* pada lapisan sebelumnya (lapisan tersembunyi).

- e. Nilai awal bobot dan bias ditentukan melalui proses randomisasi dalam interval bilangan yang batas-batasnya bisa diset sebelumnya.

3.15 *Feedforward*

Proses yang dilakukan dalam tahap ini adalah menjumlahkan perkalian antara masukan dengan bobot yang ada dan menghitung nilai aktivasinya untuk kemudian hasilnya digunakan untuk masukkan lapisan yang ada di atasnya. Langkah-langkahnya terdapat pada gambar 3.9 yaitu :

1. Mulai
2. Kalikan seluruh data masukan dengan bobot acak pada masing-masing bobot koneksi yang terhubung dengan masukan *neuron*. Lalu jumlahkan seluruh bobot yang menuju unit tersembunyi yang sama.
3. Aktivasikan hasil penjumlahan pada masing-masing *neuron* dilapisan tersembunyi, sehingga keluaran hasil dari deteksi huruf.
4. Kalikan seluruh data hasil aktivasi masing-masing *neuron* pada lapisan tersembunyi dengan bobot masing-masing bobot koneksinya. Lalu jumlahkan seluruh vektor bobot yang menuju keluaran.
5. Aktivasi hasil penjumlahan pada tiap *neuron* lapisan keluaran, sehingga keluaran hasil dari deteksi huruf.
6. Selesai.



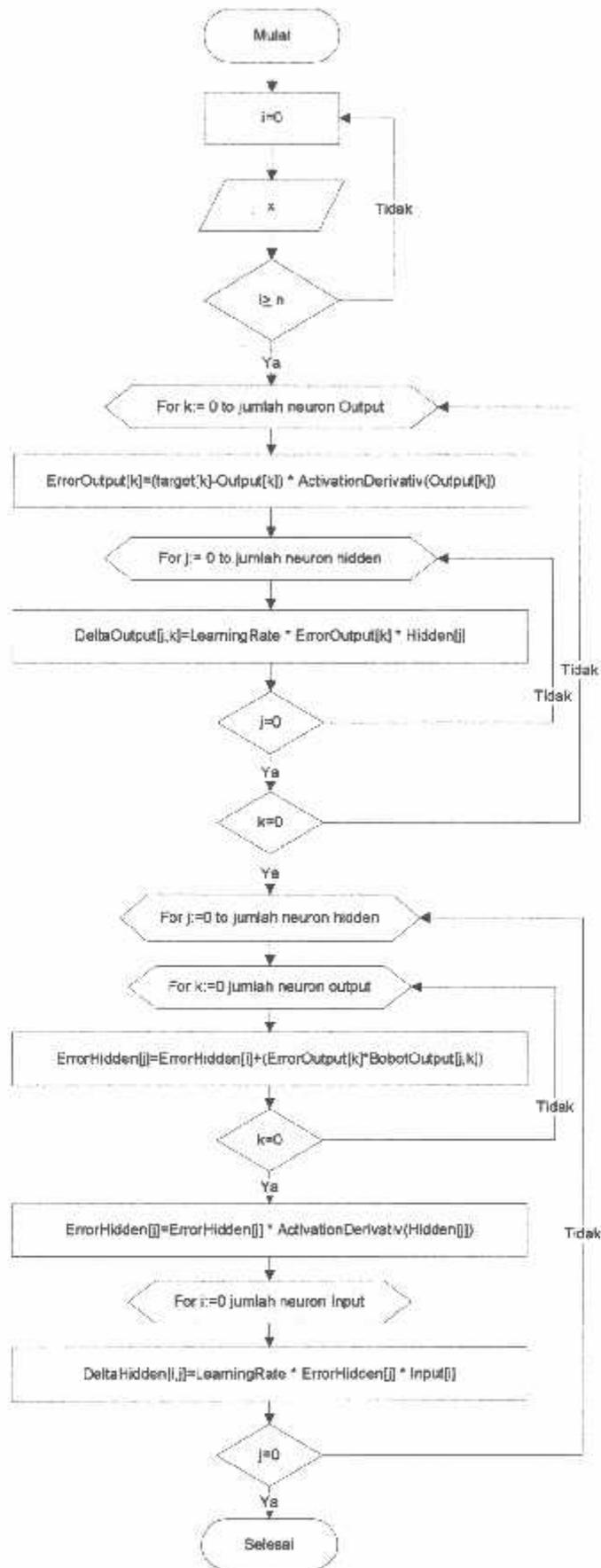
Gambar 3.9 Diagram Alir Proses *Feedforward*

3.16 *Backpropagation*

Proses yang dilakukan dalam tahap ini adalah perhitungan informasi kesalahan pada tiap *neuron* masing-masing lapisan, dimulai dari kesalahan pada lapisan keluaran hingga lapisan tersembunyi. Informasi kesalahan berguna untuk menghitung faktor peubah bobot yang digunakan untuk memperbaiki bobot lama.

Langkah-langkah *backpropagation* terdapat pada gambar 3.10, yaitu :

1. Mulai.
2. Hitung selisih antara target pelatihan dengan keluaran. Selisih disebut *error*. Kalikan selisih dengan keluaran yang telah diaktivasi dengan fungsi turunan aktivasi. Hasil perkalian merupakan faktor kesalahan pada lapisan keluaran dan akan digunakan untuk menghitung faktor kesalahan pada lapisan tersembunyi serta untuk menghitung faktor peubah bobot.
3. Hitung besar faktor peubah bobot terbaru dengan mengalikan *learning rate* dengan lapisan tersembunyi (*hidden layer*) dan *error* pada lapisan keluaran.
4. Masing-masing faktor kesalahan dikeluarkan dikalikan dengan bobot lama yang terkoneksi dengan *neuron* lapisan keluaran. Lalu hasil perkalian pada semua koneksi pada *neuron* dijumlahkan. Faktor kesalahan ini akan digunakan untuk menghitung peubah bobot dari lapisan masukan ke lapisan tersembunyi.
5. Mengalikan *learning rate* dengan lapisan masukan dan *error* (pada lapisan tersembunyi) untuk mendapatkan nilai faktor peubah bobot baru pada tiap vektor.
6. Selesai.

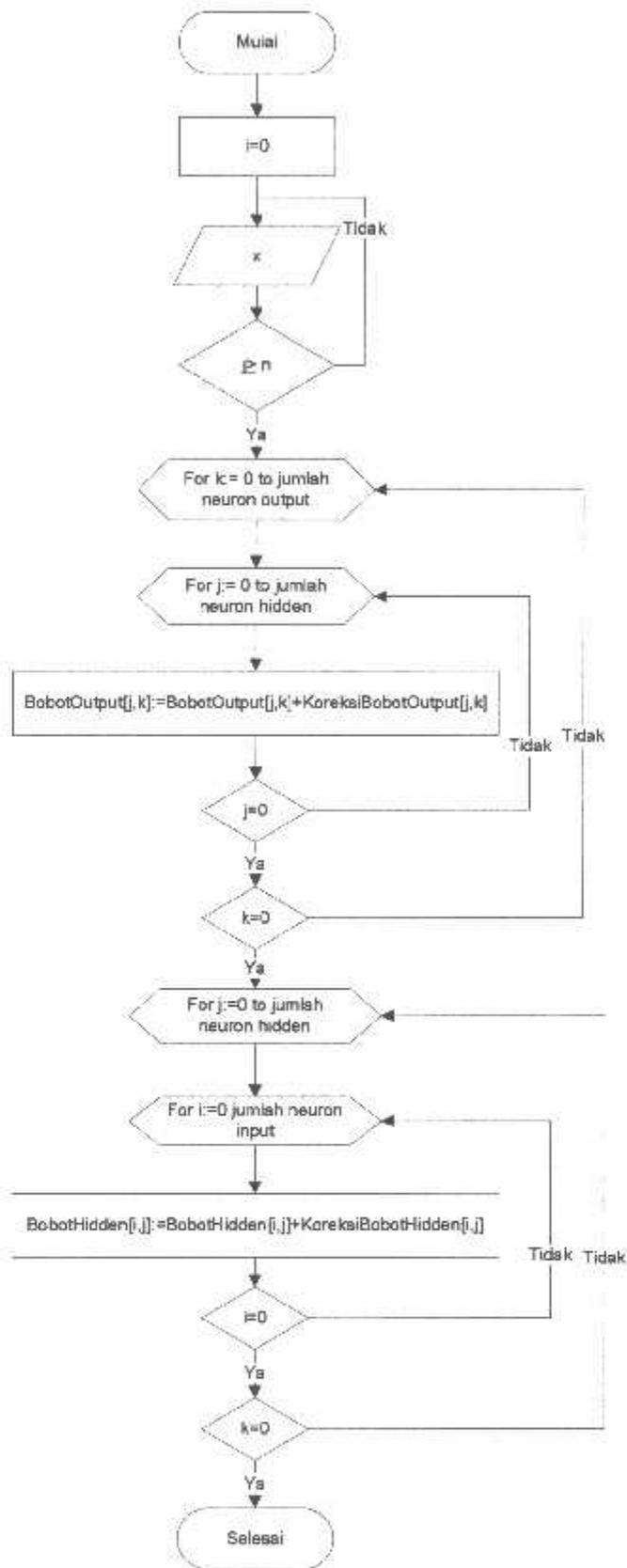


Gambar 3.10 Diagram Alir Proses *Backpropagation*

3.17 Perubahan Bobot

Hasil dari proses *backpropagation* adalah nilai faktor peubah bobot yang digunakan untuk melakukan perubahan bobot baru. Langkah-langkah dalam update bobot terdapat pada gambar 3.11, yaitu sebagai berikut :

1. Mulai.
2. Perbaiki nilai bobot untuk tiap koneksi menuju lapisan keluaran dengan cara menjumlahkan nilai bobot lama dengan peubah bobot yang telah dihitung pada proses *backpropagation*.
3. Perbaiki nilai bobot ke lapisan tersembunyi dengan menjumlahkan nilai bobot lama dengan peubah bobot yang telah dihitung pada proses *backpropagation*.
4. Selesai.

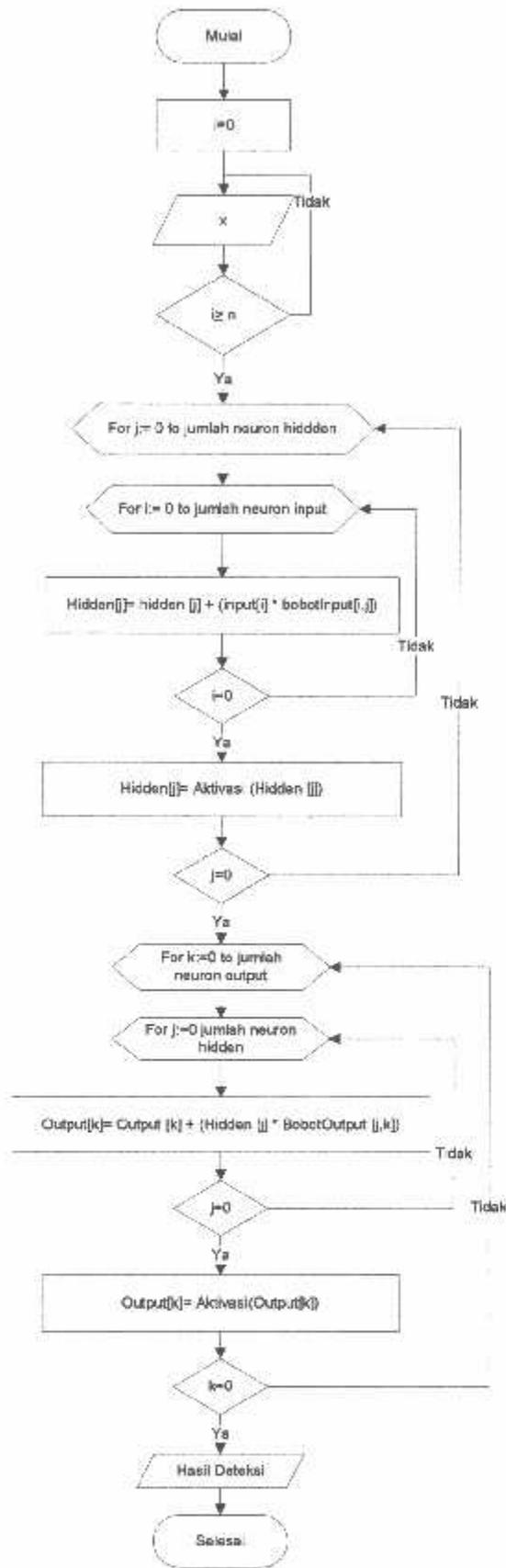


Gambar 3.11 Diagram Alir perubahan Bobot

3.18 Proses Pengujian

Tahap pengujian Jaringan Saraf Tiruan *backpropagation* dilakukan dengan tahap *feedforward* dari algoritma pelatihan. Dalam proses pengujian, hasil keluaran jaringan tidak diproses ke *backpropagation*. Langkah-langkah pengujian terdapat pada gambar 3.12, yaitu :

1. Mulai.
2. Inisialisasi bobot-bobot yang diperoleh dari proses pelatihan jaringan.
3. Masukkan data pengujian.
4. Kalikan seluruh data masukan pada *neuron* masukan dengan bobot acak pada masing-masing bobot yang terhubung dengan *neuron* masukan. Lalu jumlahkan seluruh vektor bobot menuju *neuron* tersembunyi.
5. Aktivasikan hasil penjumlahan pada masing-masing *neuron* di lapisan tersembunyi, sehingga keluaran hasil dari deteksi huruf.
6. Kalikan seluruh data hasil aktivasi masing-masing *neuron* lapisan tersembunyi dengan bobot masing-masing koneksi bobot keluaran. Lalu jumlahkan seluruh vektor bobot yang menuju *neuron* keluaran.
7. Aktivasikan hasil penjumlahan pada masing-masing *neuron* dilapisan keluaran sehingga keluaran akan bernilai antara 0 (nol) dan 1 (satu).
8. Keluaran berupa hasil deteksi huruf.
9. Selesai.



Gambar 3.12 Diagram Alir Proses Pengujian Jaringan Saraf Tiruan

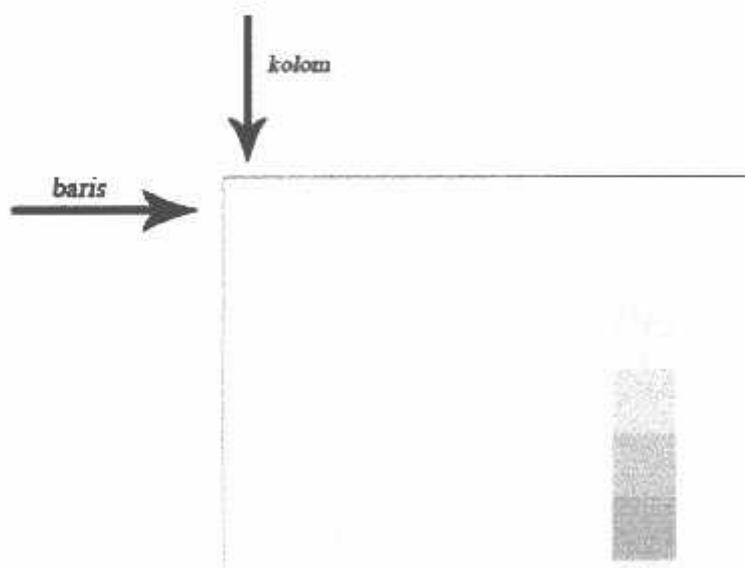
3.19 Contoh perhitungan manual

Pembelajaran Menggunakan *Backpropagation*

Data masukan berupa huruf tulisan tangan dengan inputan 100 dan *output* 6. Sebagai contoh gambar dibawah merupakan inputan dari huruf a :



Gambar 3.13 *sample* huruf a



Gambar 3.14 *sample* baris dan kolom

Dari gambar diatas akan di ambil *pixel* berukuran 10 dengan jumlah matriks 10x10, dengan begitu total baris dan kolom berjumlah 100. Hasil ini yang akan menjadi inputan. Sebagai contoh perhitungan *manual* kita buat suatu algoritma dengan inputan sebagai berikut. Lihat tabel 3.1.

Unit	Input	Output
X1	0	1
X2	1	0
X3	1	1

Tabel 3.1 Tabel data

Diketahui:

Pola Input : 0 1 1

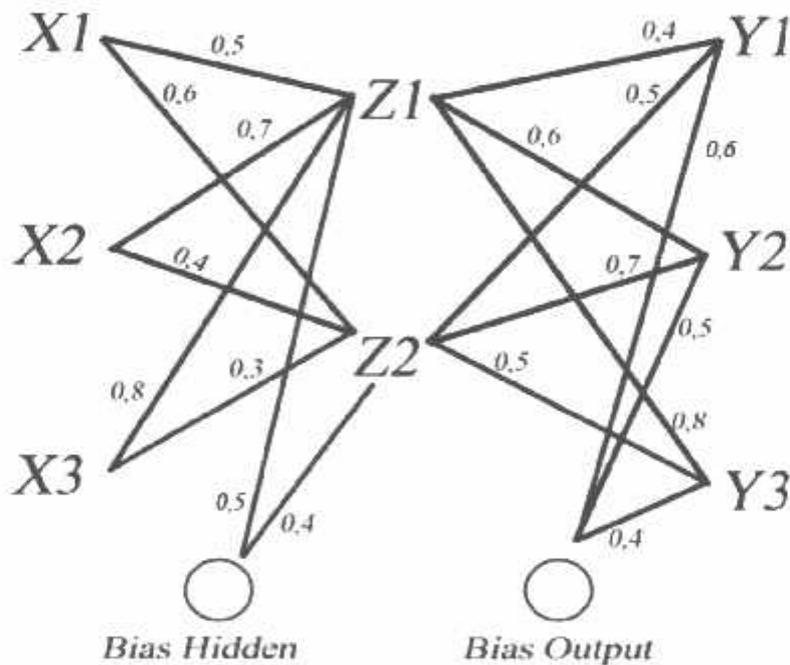
Output yang diinginkan : 1 0 1

Konstanta belajar : 0,4

Fungsi Aktivasi : $f(x) = \frac{1}{1 + e^{-x}}$

Umpan Maju

Setelah melakukan inisialisasi bobot tiap unit masukan menerima isyarat masukan X1 dan diteruskan ke unit tersembunyi (*hidden layer*). Lihat gambar 3.15.



Gambar 3.15 Inisialisasi nilai bobot

Tiap unit tersembunyi menjumlahkan isyarat masukan terbobot. Rumus yang digunakan $z_{in_j} = v_{0j} + \sum_{i=1}^n x_i v_{ij}$ maka perhitungan yang didapat adalah :

$$\begin{aligned} Z_{in_1} &= 0,5 + (1*0,8) + (1*0,7) + (0*0,5) \\ &= 0,5 + 0,8 + 0,7 + 0 \\ &= 2 \end{aligned}$$

$$Z_1 = \frac{1}{1+e^{-z}} = 0,05$$

$$\begin{aligned} Z_{in_2} &= 0,4 + (1*0,3) + (1*0,4) + (0*0,5) \\ &= 0,4 + 0,3 + 0,4 + 0 \\ &= 1,1 \end{aligned}$$

$$Z_2 = \frac{1}{1+e^{-z}} = 0,39$$

Setelah mendapatkan nilai Z_1 dan Z_2 maka langkah selanjutnya adalah menjumlahkan tiap unit keluaran isyarat masukan terbobot pada Y. Rumus yang digunakan $y_{in_k} = w_{0k} + \sum_{j=1}^p z_j w_{jk}$ maka perhitungan yang didapat adalah :

$$\begin{aligned} Y_{in_1} &= 0,6 + (0,05*0,4) + (0,39*0,5) \\ &= 0,6 + 0,02 + 0,195 \\ &= 0,815 \end{aligned}$$

$$\begin{aligned} Y_{in_2} &= 0,5 + (0,05*0,6) + (0,39*0,7) \\ &= 0,5 + 0,03 + 0,273 \\ &= 0,803 \end{aligned}$$

$$\begin{aligned} Y_{in_3} &= 0,4 + (0,05*0,8) + (0,39*0,5) \\ &= 0,4 + 0,04 + 0,195 \\ &= 0,635 \end{aligned}$$

$$Y_1 = \frac{1}{1+e^{-0,815}} = 0,64$$

$$Y_2 = \frac{1}{1+e^{-0,803}} = 0,47$$

$$Y_3 = \frac{1}{1+e^{-0,635}} = 0,50$$

Perambatan Galat Mundur

Tiap unit keluaran menerima pola sasaran berkaitan dengan pola pelatihan masukannya. Rumus yang digunakan adalah $\delta_k = (t_k - y_k) f'(y_{in_k})$.

Maka perhitungan yang didapat adalah :

$$\begin{aligned}\delta^1 &= (1-0,64) * (0,64) * (1-0,64) \\ &= 0,36 * 0,64 * 0,36 \\ &= 0,08\end{aligned}$$

$$\begin{aligned}\delta^2 &= (0-0,47) * (0,47) * (1-0,47) \\ &= -0,47 * 0,47 * 0,53 \\ &= 0,11\end{aligned}$$

$$\begin{aligned}\delta^3 &= (1-0,5) * (0,50) * (1-0,50) \\ &= 0,5 * 0,5 * 0,5 \\ &= 0,125\end{aligned}$$

Hitung koreksi bobot dan prasikapnya :

$$\begin{array}{lll} \Delta W_1 &= 0,5 * 0,8 * 0,05 &= 0,002 \\ \Delta W_2 &= 0,5 * 0,8 * 0,39 &= 0,0156 \\ \Delta W_0 &= 0,5 * 0,8 &= 0,04 \end{array} \left. \vphantom{\begin{array}{l} \Delta W_1 \\ \Delta W_2 \\ \Delta W_0 \end{array}} \right\} \delta^1$$

$$\begin{array}{lll} \Delta W_1 &= 0,5 * 0,11 * 0,05 &= 0,002 \\ \Delta W_2 &= 0,5 * 0,11 * 0,39 &= 0,02 \\ \Delta W_0 &= 0,5 * 0,11 &= 0,05 \end{array} \left. \vphantom{\begin{array}{l} \Delta W_1 \\ \Delta W_2 \\ \Delta W_0 \end{array}} \right\} \delta^2$$

$$\begin{array}{lll} \Delta W_1 &= 0,5 * 0,125 * 0,05 &= 0,003 \\ \Delta W_2 &= 0,5 * 0,125 * 0,39 &= 0,02 \\ \Delta W_0 &= 0,5 * 0,125 &= 0,06 \end{array} \left. \vphantom{\begin{array}{l} \Delta W_1 \\ \Delta W_2 \\ \Delta W_0 \end{array}} \right\} \delta^3$$

Tiap unit tersembunyi menjumlahkan delta masukannya (dari unit unit dilapisan atasnya). Dengan menggunakan rumus $\delta_{in_j} = \sum_{k=1}^m \delta w_{jk}$ maka perhitungan yang didapat adalah :

$$\delta_{in1} = \frac{(0,08 * 0,4)}{0,032} + \frac{(0,11*0,6)}{0,066} + \frac{(0,125*0,8)}{0,1} = 0,198$$

$$\delta_{in2} = \frac{(0,08 * 0,5)}{0,004} + \frac{(0,11*0,7)}{0,077} + \frac{(0,125*0,5)}{0,0625} = 0,1435$$

Hitung galat informasinya :

$$\delta_1 = 0,198 * 0,05 * (1-0,05) = 0,009$$

$$\delta_2 = 0,1435 * 0,39 * (1-0,39) = 0,034$$

$$\begin{array}{l} \Delta V_1 = 0,5 * 0,09 * 0 = 0 \\ \Delta V_2 = 0,5 * 0,034 * 0 = 0 \end{array} \left. \vphantom{\begin{array}{l} \Delta V_1 \\ \Delta V_2 \end{array}} \right\} x_1$$

$$\begin{array}{l} \Delta V_1 = 0,5 * 0,09 * 1 = 0,0045 \\ \Delta V_2 = 0,5 * 0,034 * 1 = 0,017 \end{array} \left. \vphantom{\begin{array}{l} \Delta V_1 \\ \Delta V_2 \end{array}} \right\} x_2$$

$$\begin{array}{l} \Delta V_1 = 0,5 * 0,09 * 1 = 0,0045 \\ \Delta V_2 = 0,5 * 0,034 * 1 = -0,017 \end{array} \left. \vphantom{\begin{array}{l} \Delta V_1 \\ \Delta V_2 \end{array}} \right\} x_3$$

Perbaharui bobot dan prasikapnya :

Tiap unit keluaran memperbaharui bobot-bobot dan prasikapnya.

$$\begin{array}{l} Wbaru_1 = 0,002 + 0,4 = 0,402 \\ Wbaru_2 = 0,0156 + 0,5 = 0,5156 \end{array} \left. \vphantom{\begin{array}{l} Wbaru_1 \\ Wbaru_2 \end{array}} \right\} y_1$$

$$\begin{array}{l} Wbaru_1 = 0,002 + 0,6 = 0,602 \\ Wbaru_2 = 0,002 + 0,7 = 0,72 \end{array} \left. \vphantom{\begin{array}{l} Wbaru_1 \\ Wbaru_2 \end{array}} \right\} y_2$$

$$\begin{array}{l} Wbaru_1 = 0,003 + 0,8 = 0,803 \\ Wbaru_2 = 0,002 + 0,5 = 0,502 \end{array} \left. \vphantom{\begin{array}{l} Wbaru_1 \\ Wbaru_2 \end{array}} \right\} y_3$$

$$\begin{array}{l} Vbaru_1 = 0 + 0,5 = 0,5 \\ Vbaru_2 = 0 + 0,6 = 0,6 \end{array} \left. \vphantom{\begin{array}{l} Vbaru_1 \\ Vbaru_2 \end{array}} \right\} x_1$$

$$\begin{array}{l} V_{\text{baru}_1} = 0,0045 + 0,7 \\ V_{\text{baru}_2} = 0,017 + 0,4 \end{array} \quad \begin{array}{l} = 0,7045 \\ = 0,417 \end{array} \left. \vphantom{\begin{array}{l} V_{\text{baru}_1} \\ V_{\text{baru}_2} \end{array}} \right\} x_2$$

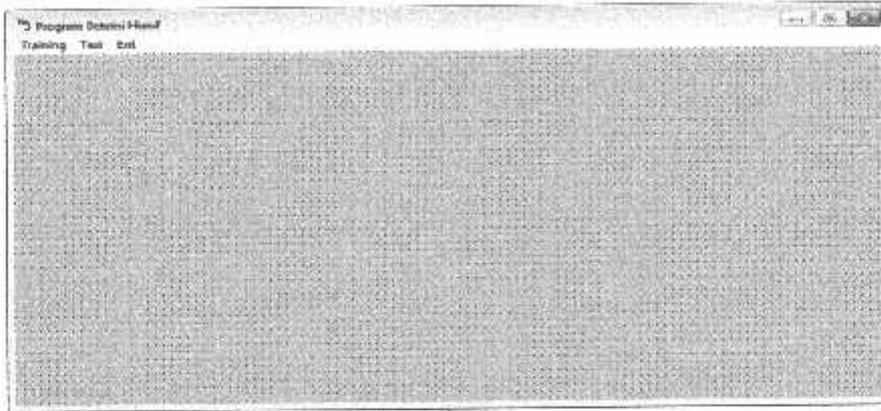
$$\begin{array}{l} V_{\text{baru}_1} = 0,045 + 0,8 \\ V_{\text{baru}_2} = 0,017 + 0,3 \end{array} \quad \begin{array}{l} = 0,8045 \\ = 0,317 \end{array} \left. \vphantom{\begin{array}{l} V_{\text{baru}_1} \\ V_{\text{baru}_2} \end{array}} \right\} x_3$$

Uji syarat berhenti.

3.20 Perancangan *Form* Utama

Perancangan *form* digunakan untuk menggabungkan perangkat lunak dengan pengguna, agar dapat dioperasikan. Berikut adalah gambar rancangan *menu* untuk mendeteksi huruf tulisan tangan.

- Tampilan menu utama (pada *software Microsoft Visual Basic 6.0*).



Gambar 3.16 *Form menu* untuk deteksi huruf tulisan tangan.

a. *Form Image Processing*.

Pada *Form Image Processing* terdiri dari beberapa komponen yang akan dijadikan *menu Image Processing*. Untuk proses pengolahan gambar terdiri 2 tombol yaitu *Image Processing* dan *Training*.

- Tombol *Image Processing* untuk memproses data inputan gambar yang telah di *scan*. Pada aplikasi ini, fitur yang digunakan adalah fitur *CommandButton*.



Gambar 3.17 Tombol *Image Processing*

- Tombol *Training* untuk proses pelatihan deteksi huruf. Pada aplikasi ini, fitur yang digunakan adalah fitur *CommandButton*.



Gambar 3.18 Tombol *Training*

- *Form Hidden Layer* berfungsi untuk memasukkan nilai *layer* tersembunyi (*hidden layer*). Pada aplikasi ini, fitur yang digunakan adalah fitur *TextBox*.



Gambar 3.19 *Form Hidden Layer*

- *Form Learning Constanta* untuk menginputkan nilai konstanta pembelajaran. Pada aplikasi ini, fitur yang digunakan adalah fitur *TextBox*.



Gambar 3.20 *Form Learning Constanta*

- *Form Max Iteration.* Pada form ini nilai iterasi dimasukan dan selanjutnya proses *training* dijalankan. Pada aplikasi ini, fitur yang digunakan adalah fitur *TextBox*.

Max Iteration

Gambar 3.21 *Form Max Iteration*

- *Form Iteration.* Pada *form* ini nilai iterasi yang telah dimasukan akan membaca banyaknya iterasi yang di *training*. Pada aplikasi ini, fitur yang digunakan adalah fitur *TextBox*.

Iteration

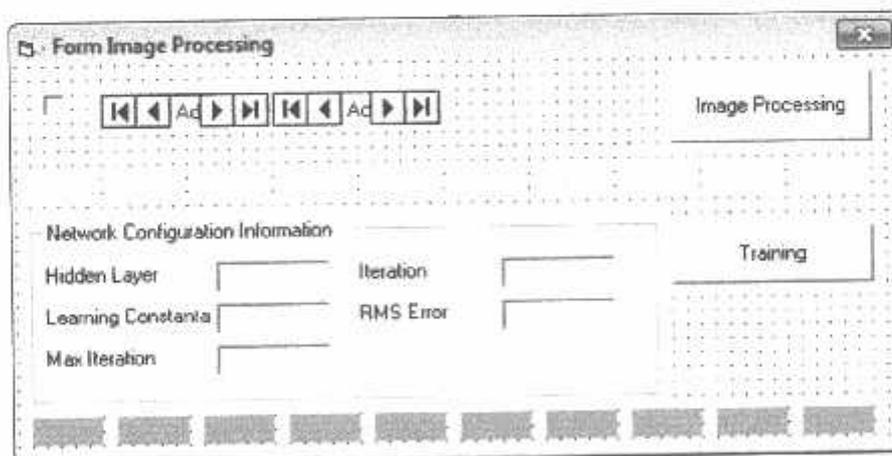
Gambar 3.22 *Form Iteration*

- *Form RMS Error.* Pada *Form RMS Error* akan muncul nilai *error* yang di dapat dari proses *training*. Pada aplikasi ini fitur yang digunakan adalah fitur *TextBox*.

RMS Error

Gambar 3.23 *Form RMS Error*

- Setelah semua *form* dimasukan, maka aplikasi untuk proses *training* sudah bisa dijalankan. Lihat pada gambar 3.24



Gambar 3.24 *Menu Form Image Processing*

b. Form Testing

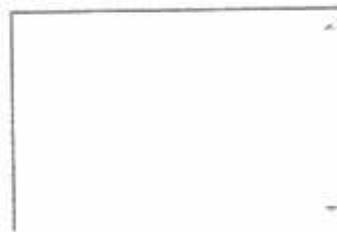
Form Testing terdiri dari 4 tombol yaitu *Open*, *Image Processing*, *Detect* dan *Exit*.

- Setelah melakukan proses *Training*, langkah selanjutnya adalah proses *test*. Pada *form Test* terdapat tombol *Open* yang berfungsi untuk mengambil gambar yang ada pada folder.



Gambar 3.25 Tombol *Open*

- Pada *Form* dibawah berfungsi untuk mengetahui letak dan alamat dari gambar yang telah diambil. Setelah melakukan proses *Open File* maka akan muncul alamat letak dari gambar tersebut.



Gambar 3.26 *Form Common Dialog*

- Pada tombol *Image Processing* disini berfungsi untuk menampilkan pola yang telah di training sebelumnya. Pada aplikasi ini, fitur yang digunakan adalah fitur *CommandButton*.



Gambar 3.27 Tombol *Image Processing*

- *Form Pattern* berfungsi untuk menampilkan nilai biner yang didapat dari proses training. pola yang muncul nantinya akan di *detect*. Pada aplikasi ini, fitur yang digunakan adalah fitur *TextBox*.

.....
Patern

Gambar 3.28 *Form Pattern*

- *Form Output Pattern* berfungsi untuk menampilkan pola keluaran yang dihasilkan. Output yang didapat berjumlah 6 digit. Pada aplikasi ini, fitur yang digunakan adalah fitur *TextBox*.



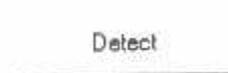
Gambar 3.29 *Form Output Pattern*

- *Form Letter* berfungsi untuk menampilkan huruf dari hasil deteksi melalui proses training. Pada aplikasi ini, fitur yang digunakan adalah fitur *TextBox*.



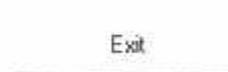
Gambar 3.30 *Form Letter*

- Pada saat tombol *Detect* diklik maka akan muncul pola keluaran yang diinginkan. Pada aplikasi ini, fitur yang digunakan adalah fitur *CommandButton*.



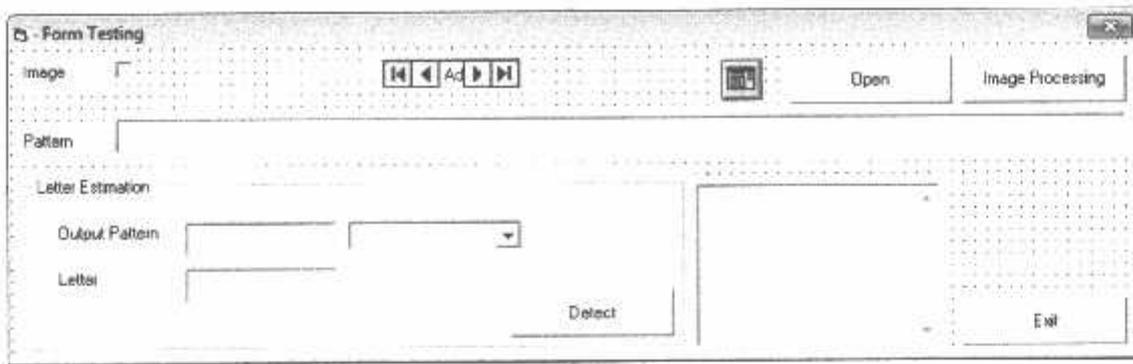
Gambar 3.31 Tombol *Detect*

- Tombol *Exit* berfungsi untuk mengakhiri program deteksi.



Gambar 3.32 Tombol *Exit*

- Setelah semua *form* dimasukan, maka aplikasi untuk proses testing sudah bisa dijalankan. Lihat pada gambar 3.33.



Gambar 3.33 Menu Form Testing

3.21 Tabel Kode Huruf

Huruf	Kode Biner					
a	0	0	0	0	0	0
b	0	0	1	0	0	0
c	0	0	0	1	0	0
d	0	0	0	0	1	0
e	0	0	0	0	0	1
f	0	0	0	0	1	0
g	0	0	0	1	0	0
h	0	0	1	0	0	0
i	0	0	1	1	0	0
j	0	0	1	1	1	1
k	0	1	0	0	0	0
l	0	1	1	0	0	0
m	0	1	0	1	0	0
n	0	1	0	0	1	0
o	0	1	0	0	0	1
p	0	1	0	0	1	0
q	0	1	0	1	0	0
r	0	1	1	0	0	0
s	0	1	1	1	0	0
t	0	1	1	1	1	1
u	1	1	0	0	0	0
v	1	1	1	0	0	0
w	1	1	0	1	0	0
x	1	1	0	0	1	0
y	1	1	0	0	0	1
z	1	1	0	0	1	0

Tabel 3.2 Tabel kode huruf

BAB IV IMPLEMENTASI DAN PENGUJIAN APLIKASI

4.1 Implementasi Aplikasi

Tahap implementasi merupakan lanjutan dari tahap analisa sistem dan perancangan sistem yang sudah dibahas pada bab sebelumnya. Dalam aplikasi deteksi huruf tulisan tangan ini, program yang digunakan adalah *Visual Basic 6.0*. Sebagai media penyimpanan data, aplikasi ini menggunakan *microsoft access 2007*, yang didalamnya terdapat beberapa tabel yang saling berhubungan. Aplikasi yang sudah jadi akan dijalankan pada perangkat komputer.

4.2 Tampilan Form Menu Utama

Pada *menu* utama, terlihat tampilan dari deteksi huruf tulisan tangan. Aplikasi ini yang nantinya akan digunakan untuk proses *training* dan *testing*. Gambar 4.1 merupakan tampilan dari menu utama.



Gambar 4.1 Tampilan *form menu* utama

4.3 Pengujian Sistem

Pada *menu training* terdapat *submenu form Image Processing*. Pada form ini penginputan nilai dilakukan dan akan di proses lebih lanjut. Klik Menu Training maka akan muncul *submenu form Image Processing*. Lihat pada gambar 4.2.



Gambar 4.2 Tampilan Menu *Image Processing*

- Untuk percobaan pertama kita masukan nilai sebagai berikut:

Hidden Layer : 30
Learnining Constanta : 0,60
Max iteration : 100
Output yang diinginkan : 011111

Setelah semua nilai dimasukan, klik tombol *testing*. Maka akan muncul peringatan "*Weight Initialize Just Complite for 30 Hidden Layer*" lalu klik OK. Proses ini bertujuan untuk mendapatakn nilai *error* yang didapat pada saat iterasi telah mencapai nilai 100 (seratus). Output yang di inginkan disini adalah bernilai 011111. Lihat pada gambar 4.3.



Gambar 4.3 Tampilan hasil inputan nilai

- Proses *Testing* sedang berlangsung. Tujuan dari proses *testing* ini ialah mencari nilai *error* yang di dapat pada iterasi ke-100. Untuk proses testing dapat dilihat pada gambar 4.4.



Gambar 4.4 Tampilan Proses *Training* sedang berlangsung

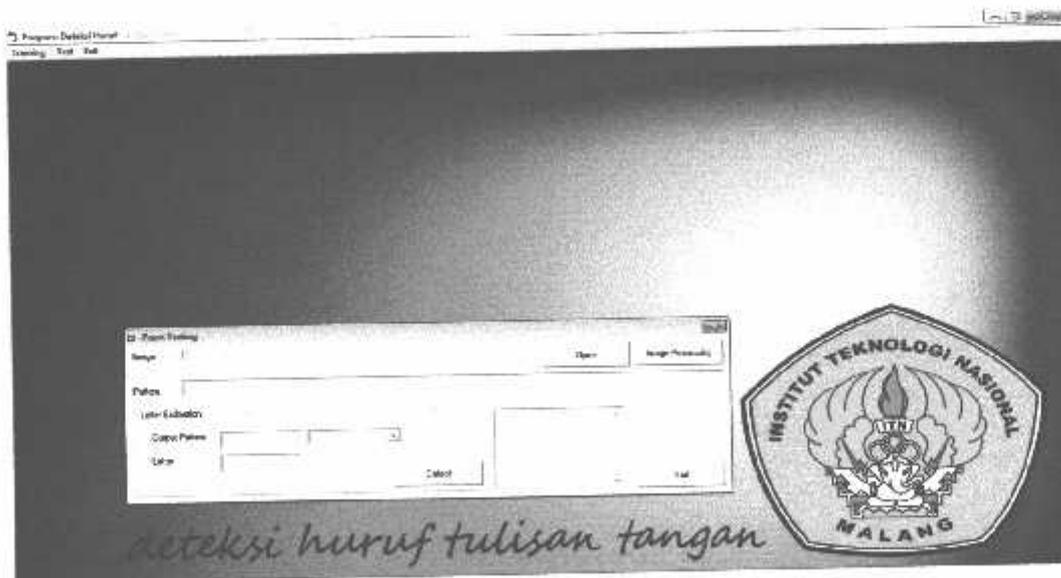
- Setelah Proses *Training* selesai maka akan muncul peringatan "*Data Testing completed*", Klik OK untuk melihat hasil dari proses *training*. Lihat Gambar 4.5.



Gambar 4.5 Tampilan *Training* selesai

Dari gambar diatas dapat dilihat bahwa nilai iterasi telah berhenti pada angka 100, sedangkan untuk nilai *RMS Error* yang didapat ialah bernilai 0,01601. Setelah proses *training* selesai klik ok untuk proses selanjutnya.

- Klik pada *menu form Test* untuk melakukan proses *Testing*. Lihat pada gambar 4.6.



Gambar 4.6 Tampilan *Menu Form Testing*

- Hasil dateksi Pengkodean Matriks 10x10 Pada huruf m adalah sebagai berikut:

1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	0	1	1	1	0	1	1	1
1	1	0	0	1	1	0	0	1	1
1	1	0	0	0	0	1	0	1	1
1	1	0	1	0	0	1	0	1	1
1	1	1	0	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1

Tabel 4.3 Tabel kode huruf m

Terlihat pada tabel diatas menunjukkan huruf m yang diwakili oleh angka 0 yang berarti memiliki warna hitam. Sedangkan angka 1 menunjukkan daerah berwarna putih yang bernilai 1. Apabila daerah yang bernilai 0 tersebut diarsir maka akan terlihat jelas huruf tersebut. Lihat pada tabel 4.4.

1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	0	1	1	1	0	1	1	1
1	1	0	0	1	1	0	0	1	1
1	1	0	0	0	0	1	0	1	1
1	1	0	1	0	0	1	0	1	1
1	1	1	0	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1

Tabel 4.4 Huruf m yang telah diarsir

4.4 Analisa Program

Perlu diketahui, dari hasil percobaan diatas (gambar 4.7), nilai pola yang di dapat dari hasil *test* yaitu pola biner 1 dan 0 dengan jumlah digit yang banyak. Nilai pola inilah yang diambil dari hasil *scan* pada huruf tulisan tangan. Dari *sample* perhitungan 10x10 piksel maka di dapat nilai pola sebanyak 100 digit. Setelah diketahui nilainya, pola inilah yang akan menjadi inputan dan dengan menggunakan metode *backpropagation*. Nilai pola ini selanjutnya akan mengalami perubahan nilai kembali dengan proses perhitungan yang berulang-ulang sesuai dengan nilai iterasi yang dimasukan. Bila hasil output tidak memunculkan hasil deteksi pada *form letter* maka hasil deteksi belum bisa dikatakan berhasil dan harus melakukan proses *training* dan *test*.

Dari uraian mengenai jaringan saraf tiruan diatas nampak bahwa pada dasarnya ada beberapa hal yang mendasari kerja jaringan saraf tiruan, diantaranya mengenai penyimpanan informasi daya ingat, akson dan dendrit yang bercabang-bercabang sedemikian banyaknya, dan proses pengolahan informasi yang terdapat dalam jaringan saraf tiruan (eksitasi, inhibisi, penjumlahan, dan nilai ambang). Keempat hal itu pulalah yang menjadi dasar untuk menciptakan jaringan saraf tiruan.

4.5 Data Hasil Pengujian

Dari beberapa pengujian yang dilakukan didapat hasil dalam tabel berikut :

Hidden Layer	Learning Constanta	Max Iteration	RMS Error	Output Pattern	Letter
20	0,50	100	0,00001	001110	-
20	0,50	300	0,00001	001110	-
20	0,50	800	0	001110	-
20	0,50	1000	0	001110	-

Tabel 4.1 Hasil Pengujian dengan layer 20

Hidden Layer	Learning Constanta	Max Iteration	RMS Error	Output Pattern	Letter
30	0,60	100	0,01961	111111	-
30	0,60	300	0,01961	011111	t
30	0,60	800	0,01601	011111	t
30	0,60	1000	0,00001	010011	-

Tabel 4.2 Hasil Pengujian dengan layer 30

Hidden Layer	Learning Constanta	Max Iteration	RMS Error	Output Pattern	Letter
40	0,45	100	0,01961	111111	-
40	0,45	300	0,01961	111111	-
40	0,45	400	0,01961	111111	-
40	0,45	500	0,01961	111111	-

Tabel 4.3 Hasil Pengujian dengan layer 40

BAB V PENUTUP

5.1 Kesimpulan

Dari implementasi dan uji coba perangkat lunak dapat diperoleh kesimpulan sebagai berikut:

1. Besarnya ukuran penulisan huruf mempengaruhi hasil pendeteksian teks dikarenakan pengecilan citra normalisasi dalam mengambil nilai kode satu pada masukan ke input pattern pada saat proses testing.
2. Pada saat proses *training*, perputaran *epoch* akan berhenti sendiri sesuai dengan nilai bobot yang dimasukan.
3. Pada Proses *training* nilai iterasi yang besar memungkinkan untuk mendapatkan hasil yang maksimal pada saat pendeteksian.
4. Arsitektur dan parameter jaringan yang memberikan hasil untuk kerja paling optimal adalah 20 neuron pada lapisan tersembunyi, laju pembelajaran 0,65 dan iterasi 1000 kali.
5. Untuk lebih mengetahui kemampuan deteksi, semakin banyak data yang dilatihkan maka akan semakin baik namun akan berdampak dengan melambatnya proses pelatihan.

5.2 Saran

Berdasarkan uji coba, ada beberapa saran yang perlu dipertimbangkan dalam pengembangan tugas akhir ini yaitu:

1. Perlu digunakan arsitektur dan parameter jaringan yang memberikan hasil pengenalan yang lebih optimal pada penelitian dan pengembangan berikutnya.
2. Untuk mendapatkan hasil yang baik diharapkan pada inialisasi parameter dan arsitektur jaringan *backpropagation* saat proses *training* diatur dengan baik dan memperbanyak data referensi yang akan digunakan.

DAFTAR PUSTAKA

1. Alexander, I dan Morton, H., (1990), *An Introduction Neural Computing*, Chapman and Hall, London.
2. Demuth, H. Dan Beale, D., 1995, *Neural Network Toolbox for Use with Matlab*, the Math Work, Inc., Massachussets.
3. Fausett, L. (1994). *Fundamentals of Neural Network*. Prentice-Hall International, Inc., Florida Institut of Technology.
4. Hanselman, D. dan B. Littlefield, *MATLAB Bahasa Komputasi Teknis*, (terj. Jozep Edyanto), ANDI, Yogyakarta, 2000.
5. Hartadi, D., *Simulasi Penghitungan Sel Darah Merah*, Tugas Akhir, Jurusan Teknik Elektro, Fakultas Teknik, Universitas Diponegoro, 2003.
6. Haykin, S., *Neural Networks, a Comprehensive Foundation*, Mcmillan College Publishing Company, New York.
7. Ivana, *Pengenalan Ucapan Vokal Bahasa Indonesia Dengan Jaringan Syaraf Tiruan Menggunakan Linear Predictive Coding*, Tugas Akhir, Jurusan Teknik Elektro, Fakultas Teknik, Universitas Diponegoro, 2003.
8. Jain, A.K., *Fundamentals of Digital Image Processing*, Prentice Hall of India Private Limited, New Delhi, 1995.
9. Kusumoputro, B., E. Philipus dan M.W. Rahmat, *Pengenalan Huruf Tulisan Tangan Menggunakan Logika Fuzzy dan Jaringan Syaraf Tiruan*, Fakultas Ilmu Komputer, Universitas Indonesia.
10. Mudjihardjo, P., T.S. Widodo dan A. Susanto, *Penerapan Jaringan Perambatan-Balik untuk Deteksi Kesalahan pada Pengenalan Kode Pos Tulisan Tangan*, Seminar Nasional Teknik Elektro, G-20, 2003.
11. Murni, A., *Pengantar Pengolahan Citra*, PT. Elekmedia Komputindo, Kelompok Gramedia, Jakarta, 1992.
12. Suyatno Addy, 2003, *Prediksi Prestasi Siswa SMU Berdasarkan Nilai Ebtanas Murni menggunakan Jaringan Saraf Tiruan*, Skripsi FTI UTY, Yogyakarta.
13. Turban, E., 1995, *Decision Support and Expert System*, Printice Hall, London.
14. Wilson, R.L., dan Sharda R., 1994, *Bankruptcy Prediction Using Neural Network*, Decision Support System.

LAMPIRAN



PT. BNI (PERSERO) MALANG
BANK NIAGA MALANG

PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

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) 417638 Fax. (0341) 417634 Malang

**BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI**

NAMA : DWI FERNANDUS TURNIP
NIM : 06.12.608
JURUSAN : Teknik Elektro S-1
KONSENTRASI : Teknik Informatika & Komputer
MASA BIMBINGAN: 12 April 2012 s/d 23 September 2012
JUDUL : DETEKSI HURUF TULISAN TANGAN MENGGUNAKAN
JARINGAN SARAF TIRUAN BACKPROPAGATION

Dipertahankan dihadapan Majelis Penguji Skripsi Jenjang Strata Satu (S-1) pada :

Hari : Sabtu
Tanggal : 04 Agustus 2012
Dengan Nilai : 85,7 (A) *r*

PANITIA UJIAN SKRIPSI

Ketua Majelis Penguji,


Ir. Yusuf Ismail Nakhoda, MT
NIP. Y. 1018800189

Sekretaris Majelis Penguji,


Dr. Eng. Ardyanto S, ST, MT
NIP. Y. 1030800417

ANGGOTA PENGUJI

Dosen Penguji I


Ir. Eko Nurcahyo
NIP. Y. 1028700172

Dosen Penguji II


Ahmad Faisal, ST
NIP. P. 1031000431



FORMULIR PERBAIKAN SKRIPSI

Dalam pelaksanaan ujian skripsi jenjang Strata Satu (S-1) Jurusan Teknik Elektro Konsentrasi Teknik Komputer dan Informatika, maka perlu adanya perbaikan skripsi untuk mahasiswa :

Nama : DWI FERNANDUS TURNIP
NIM : 06.12.608
Jurusan : Teknik Elektro S-1
Konsentrasi : Teknik Informatika & Komputer
Masa Bimbingan : 12 April 2012 s/d 23 September 2012
Judul Skripsi : **DETEKSI HURUF TULISAN TANGAN MENGGUNAKAN JARINGAN SARAF TIRUAN BACKPROPAGATION**

Tanggal	Uraian	Paraf
Penguji II 04 Agustus 2012	- Iterasi harus berhenti jika error 0 mendekati target error yang diinginkan - Deteksi tiap kesalahan input atau error dari user	

Disetujui :

Penguji I

Ir. Eko Nurcahvo
NIP. 1028700172

Penguji II

Ahmad Faisol, ST
NIP. P. 1031000431

Mengetahui :

Dosen Pembimbing I

Irmalia Suryani Farandisa, ST, MT
NIP. P. 1030000365

Dosen Pembimbing II

Yuli Wahyuni, ST, MT
NIP. P. 1031200456



FORMULIR BIMBINGAN SKRIPSI

Nama : Dwi Fernandus Turnip
Nim : 06.12.608
Masa Bimbingan : 12 April 2012 s/d 23 September 2012
Judul Skripsi : DETEKSI HURUF TULISAN TANGAN MENGGUNAKAN JARINGAN SARAF TIRUAN BACKPROPAGATION

NO.	TANGGAL	URAIAN	PARAF PEMBIMBING
1.	11/07 2012	Revisi Bab 3	
2.	17/01 2012	Revisi Bab 4	
3.	20/07 2012	Revisi program	
4.	24/07 2012	Revisi Flowchart	
5.	02/08 2012	Demo program	
6.		Ace seminar huruf	
7.		Ace Bab 4, 5	
8.		Ace bagian backprop	
9.		Ace silang	
10.			

Malang,

Dosen Pembimbing I

Irmalia Suryani Faradisa, ST., MT,
NIP. P. 1030000365

Form S-4B



FORMULIR BIMBINGAN SKRIPSI

Nama : **Dwi Fernandus Turnip**
Nim : **06.12.608**
Masa Bimbingan : **12 April 2012 s/d 23 September 2012**
Judul Skripsi : **DETEKSI HURUF TULISAN TANGAN MENGGUNAKAN JARINGAN SARAF TIRUAN BACKPROPAGATION**

NO.	TANGGAL	URAIAN	PARAF PEMBIMBING
1.	5/06 2012	Bimbingan Bab 1 & 2	
2.	7/06 2012	Revisi Bab 2	
3.	14/06 2012	ACC Bab 1 & 2	
4.	21/06 2012	Bimbingan Bab 3	
5.	23/06 2012	Demo program	
6.	7/07 2012	Revisi Bab 3	
7.	17/07 2012	Revisi Bab 4	
8.	02/08 2012	ACC Bab 1, 2, 3, 4, 5	
9.			
10.			

Malang,

Dosen Pembimbing II

Yuti Wahyuni, ST, MT
NIP.Y. 1031200456

Form S-4B



PERMOHONAN PERSETUJUAN SKRIPSI

Yang betanda tangan dibawah ini :

Nama : Dwi-Fernandus Tornip
 N I M : 0612608
 Semester : 12
 Fakultas : Teknologi Industri
 Jurusan : Teknik Elektro S-1
 Konsentrasi : **TEKNIK ELEKTRONIKA**
TEKNIK ENERGI LISTRIK
TEKNIK KOMPUTER DAN INFORMATIKA
TEKNIK KOMPUTER
TEKNIK TELEKOMUNIKASI
 Alamat : Jl. Raya Mojorejo No. 8 Batu

Dengan ini kami mengajukan permohonan untuk mendapatkan persetujuan untuk membuat **SKRIPSI Tingkat Sarjana**. Untuk melengkapi permohonan tersebut, bersama kami lampirkan persyaratan-persyaratan yang harus dipenuhi. Adapun persyaratan-persyaratan pengambilan **SKRIPSI** adalah sebagai berikut :

1. Telah melaksanakan semua praktikum sesuai dengan konsentrasinya (.....)
2. Telah lulus dan menyerahkan Laporan Praktek Kerja (.....)
3. Telah lulus seluruh mata kuliah keahlian (MKB) sesuai konsentrasinya (.....)
4. Telah menempuh mata kuliah ≥ 134 sks dengan IPK ≥ 2 dan tidak ada nilai E (.....)
5. Telah mengikuti secara aktif kegiatan seminar skripsi yang diadakan Jurusan (.....)
6. Memenuhi persyaratan administrasi (.....)

Demikian permohonan ini untuk mendapatkan penyelesaian lebih lanjut dan atas perhatiannya kami ucapkan terima kasih.

Telah diteliti kebenaran data tersebut diatas
 Recording Teknik Elektro

Jung
 (.....)

Malang, 29 Maret2012

Pemohon

Dwi-Fernandus Tornip
 (.....)

SURAT PERNYATAAN ORISINALITAS

Yang bertanda tangan di bawah ini :

Nama : Dwi Fernandus Turnip
NIM : 06.12.608
Program Studi : Teknik Elektro S1
Konsentrasi : Teknik Komputer & Informatika S1

Dengan ini menyatakan bahwa Skripsi yang saya buat adalah hasil karya sendiri, tidak merupakan plagiasi dari karya orang lain. Dalam Skripsi ini tidak memuat karya orang lain, kecuali dicantumkan sumbernya sesuai dengan ketentuan yang berlaku.

Demikian surat pernyataan ini saya buat, dan apabila di kemudian hari ada pelanggaran atas surat pernyataan ini, saya bersedia menerima sanksinya.

Malang, 19-Maret-2013

Yang membuat Pernyataan,



Nama : Dwi Fernandus Turnip
NIM : 06.12.608



INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO

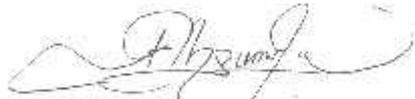
Formulir Perbaikan Ujian Skripsi

Dalam pelaksanaan Ujian Skripsi Janjang Strata 1 Jurusan Teknik Elektro Konsentrasi T. Energi Listrik / T. Elektronika / T. Infokom, maka perlu adanya perbaikan skripsi untuk mahasiswa :

NAMA : Dwi Fernandes T.
NIM : 0612608
Perbaikan meliputi :

1. Aerasi harus bertambah jika error sudah 0 atau mendekati target error yg diinginkan
2. Deteksi tiap kesalahan input / proses dari user

Malang, 4 Agustus


(Ahmad Fauzan, ST)

SAMP
EL
HURUF

Abjad		Tulisan Tangan	
A	B	a	b
C	D	c	d
E	F	e	f
G	H	g	h
I	J	i	j
K	L	k	l
M	N	m	n
O	P	o	p
Q	R	q	r
S	T	s	t
U	V	u	v
W	X	w	x
Y	Z	y	z

Abjad		Tulisan Tangan	
A	B	a	b
C	D	c	d
E	F	e	f
G	H	g	h
I	J	i	j
K	L	k	l
M	N	m	n
O	P	o	p
Q	R	q	r
S	T	s	t
U	V	u	v
W	X	w	x
Y	Z	y	z

Abjad		Tulisan Tangan	
A	B	a	b
C	D	c	d
E	F	e	f
G	H	g	h
I	J	i	j
K	L	k	l
M	N	m	n
O	P	o	p
Q	R	q	r
S	T	s	t
U	V	u	v
W	X	w	x
Y	Z	y	z

Abjad		Tulisan Tangan	
A	B	a	b
C	D	c	d
E	F	e	f
G	H	g	h
I	J	i	j
K	L	k	l
M	N	m	n
O	P	o	p
Q	R	q	r
S	T	s	t
U	V	u	v
W	X	w	x
Y	Z	y	z

Abjad		Tulisan Tangan	
A	B	a	b
C	D	c	d
E	F	e	f
G	H	g	h
I	J	i	j
K	L	k	l
M	N	m	n
O	P	o	p
Q	R	q	r
S	T	s	t
U	V	u	v
W	X	w	x
Y	Z	y	z

Abjad		Tulisan Tangan	
A	B	a	b
C	D	c	d
E	F	e	f
G	H	g	h
I	J	i	j
K	L	k	l
M	N	m	n
O	P	o	p
Q	R	q	r
S	T	s	t
U	V	u	v
W	X	w	x
Y	Z	y	z

Abjad		Tulisan Tangan	
A	B	a	b
C	D	c	d
E	F	e	f
G	H	g	h
I	J	i	j
K	L	k	l
M	N	m	n
O	P	o	p
Q	R	q	r
S	T	s	t
U	V	u	v
W	X	w	x
Y	Z	y	z

Abjad		Tulisan Tangan	
A	B	a	b
C	D	c	d
E	F	e	f
G	H	g	h
I	J	i	j
K	L	k	l
M	N	m	n
O	P	o	p
Q	R	q	r
S	T	s	t
U	V	u	v
W	X	w	x
Y	Z	y	z

Abjad		Tulisan Tangan
A	B	a b
C	D	c d
E	F	e f
G	H	g h
I	J	i j
K	L	k l
M	N	m n
O	P	o p
Q	R	q r
S	T	s t
U	V	u v
W	X	w x
Y	Z	y z

Abjad		Tulisan Tangan
A	B	a b
C	D	c d
E	F	e f
G	H	g h
I	J	i j
K	L	k l
M	N	m n
O	P	o p
Q	R	q r
S	T	s t
U	V	u v
W	X	w x
Y	Z	y z

Abjad		Tulisan Tangan
A	B	a b
C	D	c d
E	F	e f
G	H	g h
I	J	i j
K	L	k l
M	N	m n
O	P	o p
Q	R	q r
S	T	s t
U	V	u v
W	X	w x
Y	Z	y z

Abjad		Tulisan Tangan
A	B	a b
C	D	c d
E	F	e f
G	H	g h
I	J	i j
K	L	k l
M	N	m n
O	P	o p
Q	R	q r
S	T	s t
U	V	u v
W	X	w x
Y	Z	y z

Abjad		Tulisan Tangan
A	B	a b
C	D	c d
E	F	e f
G	H	g h
I	J	i j
K	L	k l
M	N	m n
O	P	o p
Q	R	q r
S	T	s t
U	V	u v
W	X	w x
Y	Z	y z

Abjad		Tulisan Tangan
A	B	a b
C	D	c d
E	F	e f
G	H	g h
I	J	i j
K	L	k l
M	N	m n
O	P	o p
Q	R	q r
S	T	s t
U	V	u v
W	X	w x
Y	Z	y z

Abjad		Tulisan Tangan
A	B	a b
C	D	c d
E	F	e f
G	H	g h
I	J	i j
K	L	k l
M	N	m n
O	P	o p
Q	R	q r
S	T	s t
U	V	u v
W	X	w x
Y	Z	y z

Abjad		Tulisan Tangan
A	B	a b
C	D	c d
E	F	e f
G	H	g h
I	J	i j
K	L	k l
M	N	m n
O	P	o p
Q	R	q r
S	T	s t
U	V	u v
W	X	w x
Y	Z	y z

Abjad		Tulisan Tangan
A	B	a b
C	D	c d
E	F	e f
G	H	g h
I	J	i j
K	L	k l
M	N	m n
O	P	o p
Q	R	q r
S	T	s t
U	V	u v
W	X	w x
Y	Z	y z

Abjad		Tulisan Tangan
A	B	a b
C	D	c d
E	F	e f
G	H	g h
I	J	i j
K	L	k l
M	N	m n
O	P	o p
Q	R	q r
S	T	s t
U	V	u v
W	X	w x
Y	Z	y z

Abjad		Tulisan Tangan
A	B	a b
C	D	c d
E	F	e f
G	H	g h
I	J	i j
K	L	k l
M	N	m n
O	P	o p
Q	R	q r
S	T	s t
U	V	u v
W	X	w x
Y	Z	y z

Abjad		Tulisan Tangan
A	B	a b
C	D	c d
E	F	e f
G	H	g h
I	J	i j
K	L	k l
M	N	m n
O	P	o p
Q	R	q r
S	T	s t
U	V	u v
W	X	w x
Y	Z	y z

Abjad		Tulisan Tangan
A	B	a b
C	D	c d
E	F	e f
G	H	g h
I	J	i j
K	L	k l
M	N	m n
O	P	o p
Q	R	q r
S	T	s t
U	V	u v
W	X	w x
Y	Z	y z

Abjad		Tulisan Tangan
A	B	a b
C	D	c d
E	F	e f
G	H	g h
I	J	i j
K	L	k l
M	N	m n
O	P	o p
Q	R	q r
S	T	s t
U	V	u v
W	X	w x
Y	Z	y z

Abjad		Tulisan Tangan
A	B	a b
C	D	c d
E	F	e f
G	H	g h
I	J	i j
K	L	k l
M	N	m n
O	P	o p
Q	R	q r
S	T	s t
U	V	u v
W	X	w x
Y	Z	y z

Abjad		Tulisan Tangan
A	B	a b
C	D	c d
E	F	e f
G	H	g h
I	J	i j
K	L	k l
M	N	m n
O	P	o p
Q	R	q r
S	T	s t
U	V	u v
W	X	w x
Y	Z	y z

Abjad		Tulisan Tangan	
A	B	a	b
C	D	c	d
E	F	e	f
G	H	g	h
I	J	i	j
K	L	k	l
M	N	m	n
O	P	o	p
Q	R	q	r
S	T	s	t
U	V	u	v
W	X	w	x
Y	Z	y	z

Abjad		Tulisan Tangan	
A	B	a	b
C	D	c	d
E	F	e	f
G	H	g	h
I	J	i	j
K	L	k	l
M	N	m	n
O	P	o	p
Q	R	q	r
S	T	s	t
U	V	u	v
W	X	w	x
Y	Z	y	z

Abjad		Tulisan Tangan	
A	B	a	b
C	D	c	d
E	F	e	f
G	H	g	h
I	J	i	j
K	L	k	l
M	N	m	n
O	P	o	p
Q	R	q	r
S	T	s	t
U	V	u	v
W	X	w	x
Y	Z	y	z

Abjad		Tulisan Tangan	
A	B	a	b
C	D	c	d
E	F	e	f
G	H	g	h
I	J	i	j
K	L	k	l
M	N	m	n
O	P	o	p
Q	R	q	r
S	T	s	t
U	V	u	v
W	X	w	x
Y	Z	y	z

Abjad		Tulisan Tangan	
A	B	a	b
C	D	c	d
E	F	e	f
G	H	g	h
I	J	i	j
K	L	k	l
M	N	m	n
O	P	o	p
Q	R	q	r
S	T	s	t
U	V	u	v
W	X	w	x
Y	Z	y	z

Abjad		Tulisan Tangan	
A	B	a	b
C	D	c	d
E	F	e	f
G	H	g	h
I	J	i	j
K	L	k	l
M	N	m	n
O	P	o	p
Q	R	q	r
S	T	s	t
U	V	u	v
W	X	w	x
Y	Z	y	z

Abjad		Tulisan Tangan	
A	B	a	b
C	D	c	d
E	F	e	f
G	H	g	h
I	J	i	j
K	L	k	l
M	N	m	n
O	P	o	p
Q	R	q	r
S	T	s	t
U	V	u	v
W	X	w	x
Y	Z	y	z

Abjad		Tulisan Tangan	
A	B	a	b
C	D	c	d
E	F	e	f
G	H	g	h
I	J	i	j
K	L	k	l
M	N	m	n
O	P	o	p
Q	R	q	r
S	T	s	t
U	V	u	v
W	X	w	x
Y	Z	y	z

Abjad		Tulisan Tangan
A	B	a b
C	D	c d
E	F	e f
G	H	g h
I	J	i j
K	L	k l
M	N	m n
O	P	o p
Q	R	q r
S	T	s t
U	V	u v
W	X	w x
Y	Z	y z

Abjad		Tulisan Tangan
A	B	a b
C	D	c d
E	F	e f
G	H	g h
I	J	i j
K	L	k l
M	N	m n
O	P	o p
Q	R	q r
S	T	s t
U	V	u v
W	X	w x
Y	Z	y z

Abjad		Tulisan Tangan
A	B	a b
C	D	c d
E	F	e f
G	H	g h
I	J	i j
K	L	k l
M	N	m n
O	P	o p
Q	R	q r
S	T	s t
U	V	u v
W	X	w x
Y	Z	y z

Abjad		Tulisan Tangan
A	B	a b
C	D	c d
E	F	e f
G	H	g h
I	J	i j
K	L	k l
M	N	m n
O	P	o p
Q	R	q r
S	T	s t
U	V	u v
W	X	w x
Y	Z	y z

Abjad		Tulisan Tangan	
A	B	a	b
C	D	c	d
E	F	e	f
G	H	g	h
I	J	i	j
K	L	k	l
M	N	m	n
O	P	o	p
Q	R	q	r
S	T	s	t
U	V	u	v
W	X	w	x
Y	Z	y	z

Abjad		Tulisan Tangan	
A	B	a	b
C	D	c	d
E	F	e	f
G	H	g	h
I	J	i	j
K	L	k	l
M	N	m	n
O	P	o	p
Q	R	q	r
S	T	s	t
U	V	u	v
W	X	w	x
Y	Z	y	z

Abjad		Tulisan Tangan	
A	B	a	b
C	D	c	d
E	F	e	f
G	H	g	h
I	J	i	j
K	L	k	l
M	N	m	n
O	P	o	p
Q	R	q	r
S	T	s	t
U	V	u	v
W	X	w	x
Y	Z	y	z

Abjad		Tulisan Tangan	
A	B	a	b
C	D	c	d
E	F	e	f
G	H	g	h
I	J	i	j
K	L	k	l
M	N	m	n
O	P	o	p
Q	R	q	r
S	T	s	t
U	V	u	v
W	X	w	x
Y	Z	y	z

21/06/19

Abjad		Tulisan Tangan	
A	B	a	b
C	D	c	d
E	F	e	f
G	H	g	h
I	J	i	j
K	L	k	l
M	N	m	n
O	P	o	p
Q	R	q	r
S	T	s	t
U	V	u	v
W	X	w	x
Y	Z	y	z

Abjad		Tulisan Tangan	
A	B	a	b
C	D	c	d
E	F	e	f
G	H	g	h
I	J	i	j
K	L	k	l
M	N	m	n
O	P	o	p
Q	R	q	r
S	T	s	t
U	V	u	v
W	X	w	x
Y	Z	y	z

Abjad		Tulisan Tangan	
A	B	a	b
C	D	c	d
E	F	e	f
G	H	g	h
I	J	i	j
K	L	k	l
M	N	m	n
O	P	o	p
Q	R	q	r
S	T	s	t
U	V	u	v
W	X	w	x
Y	Z	y	z

Abjad		Tulisan Tangan	
A	B	a	b
C	D	c	d
E	F	e	f
G	H	g	h
I	J	i	j
K	L	k	l
M	N	m	n
O	P	o	p
Q	R	q	r
S	T	s	t
U	V	u	v
W	X	w	x
Y	Z	y	z

NOMOR : 1

Nama : Romadiansyah

Abjad		Tulisan Tangan	
A	B	a	b
C	D	c	d
E	F	e	f
G	H	g	h
I	J	i	j
K	L	k	l
M	N	m	n
O	P	o	p
Q	R	q	r
S	T	s	t
U	V	u	v
W	X	w	x
Y	Z	y	z

NOMOR : 2

Nama : Ocin

Abjad		Tulisan Tangan	
A	B	a	b
C	D	c	d
E	F	e	f
G	H	g	h
I	J	i	j
K	L	k	l
M	N	m	n
O	P	o	p
Q	R	q	r
S	T	s	t
U	V	u	v
W	X	w	x
Y	Z	y	z

NOMOR : 3

Nama : Randi Agja

Abjad		Tulisan Tangan	
A	B	a	b
C	D	c	d
E	F	e	f
G	H	g	h
I	J	i	j
K	L	k	l
M	N	m	n
O	P	o	p
Q	R	q	r
S	T	s	t
U	V	u	v
W	X	w	x
Y	Z	y	z

NOMOR : 4

Nama : Melly

Abjad		Tulisan Tangan	
A	B	a	b
C	D	c	d
E	F	e	f
G	H	g	h
I	J	i	j
K	L	k	l
M	N	m	n
O	P	o	p
Q	R	q	r
S	T	s	t
U	V	u	v
W	X	w	x
Y	Z	y	z

Nama : MIA

Abjad		Tulisan Tangan	
A	B	a	b
C	D	c	d
E	F	e	f
G	H	g	h
I	J	i	j
K	L	k	l
M	N	m	n
O	P	o	p
Q	R	q	r
S	T	s	t
U	V	u	v
W	X	w	x
Y	Z	y	z

Nama : RINTO

Abjad		Tulisan Tangan	
A	B	a	b
C	D	c	d
E	F	e	f
G	H	g	h
I	J	i	j
K	L	k	l
M	N	m	n
O	P	o	p
Q	R	q	r
S	T	s	t
U	V	u	v
W	X	w	x
Y	Z	y	z

Nama : ASTRINA

Abjad		Tulisan Tangan	
A	B	a	b
C	D	c	d
E	F	e	f
G	H	g	h
I	J	i	j
K	L	k	l
M	N	m	n
O	P	o	p
Q	R	q	r
S	T	s	t
U	V	u	v
W	X	w	x
Y	Z	y	z

Nama : HENDRA BOCHANTO

Abjad		Tulisan Tangan	
A	B	a	b
C	D	c	d
E	F	e	f
G	H	g	h
I	J	i	j
K	L	k	l
M	N	m	n
O	P	o	p
Q	R	q	r
S	T	s	t
U	V	u	v
W	X	w	x
Y	Z	y	z

Nama : Dea Mariska F.

Abjad	Tulisan Tangan	
	a	b
A	a	b
C	c	d
E	e	f
G	g	h
I	i	j
K	k	l
M	m	n
O	o	p
Q	q	r
S	s	t
U	u	v
W	w	x
Y	y	z

Nama : Evitha

Abjad	Tulisan Tangan	
	A	B
A	A	B
C	C	D
E	E	F
G	G	H
I	I	J
K	K	L
M	M	N
O	O	P
Q	Q	R
S	S	T
U	U	V
W	W	X
Y	Y	Z

Nama : Abel

Abjad	Tulisan Tangan	
	a	b
A	a	b
C	c	d
E	e	f
G	g	h
I	i	j
K	k	l
M	m	n
O	o	p
Q	q	r
S	s	t
U	u	v
W	w	x
Y	y	z

Nama : Retri

Abjad	Tulisan Tangan	
	a	b
A	a	b
C	c	d
E	e	f
G	g	h
I	i	j
K	k	l
M	m	n
O	o	p
Q	q	r
S	s	t
U	u	v
W	w	x
Y	y	z

Nomor: 63

Nama: Risky

Abjad	Tulisan Tangan	
	a	b
A	a	b
C	c	d
E	e	f
G	g	h
I	i	j
K	k	l
M	m	n
O	o	p
Q	q	r
S	s	t
U	u	v
W	w	x
Y	y	z

Nomor: 64

Nama: Iwan

Abjad	Tulisan Tangan	
	a	b
A	a	b
C	c	d
E	e	f
G	g	h
I	i	j
K	k	l
M	m	n
O	o	p
Q	q	r
S	s	t
U	u	v
W	w	x
Y	y	z

Nomor: 65

Nama: Raka

Abjad	Tulisan Tangan	
	a	b
A	a	b
C	c	d
E	e	f
G	g	h
I	i	j
K	k	l
M	m	n
O	o	p
Q	q	r
S	s	t
U	u	v
W	w	x
Y	y	z

Nomor: 66

Nama: Endry

Abjad	Tulisan Tangan	
	a	b
A	a	b
C	c	d
E	e	f
G	g	h
I	i	j
K	k	l
M	m	n
O	o	p
Q	q	r
S	s	t
U	u	v
W	w	x
Y	y	z

Nama : Ikha

Abjad		Tulisan Tangan	
A	B	a	b
C	D	c	d
E	F	e	f
G	H	g	h
I	J	i	j
K	L	k	l
M	N	m	n
O	P	o	p
Q	R	q	r
S	T	s	t
U	V	u	v
W	X	w	x
Y	Z	y	z

Nama : Adhies

Abjad		Tulisan Tangan	
A	B	a	b
C	D	c	d
E	F	e	f
G	H	g	h
I	J	i	j
K	L	k	l
M	N	m	n
O	P	o	p
Q	R	q	r
S	T	s	t
U	V	u	v
W	X	w	x
Y	Z	y	z

Nama : Dewi Rahmadiani

Abjad		Tulisan Tangan	
A	B	a	b
C	D	c	d
E	F	e	f
G	H	g	h
I	J	i	j
K	L	k	l
M	N	m	n
O	P	o	p
Q	R	q	r
S	T	s	t
U	V	u	v
W	X	w	x
Y	Z	y	z

Nama : Aditya Andy Saputra

Abjad		Tulisan Tangan	
A	B	a	b
C	D	c	d
E	F	e	f
G	H	g	h
I	J	i	j
K	L	k	l
M	N	m	n
O	P	o	p
Q	R	q	r
S	T	s	t
U	V	u	v
W	X	w	x
Y	Z	y	z

