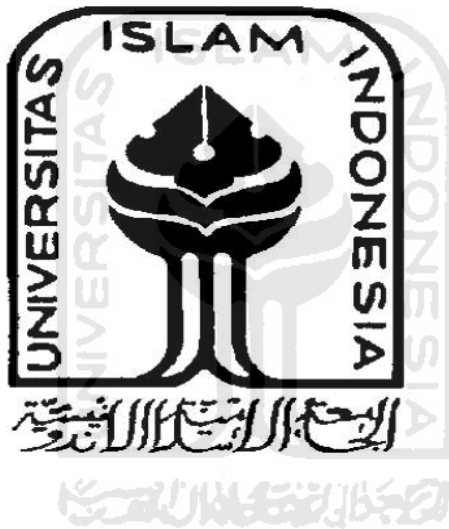


**PENGENALAN UCAPAN MANUSIA
DENGAN METODE JARINGAN SYARAF TIRUAN PROPAGASI
BALIK**

TUGAS AKHIR

**Diajukan Sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana
Jurusan Teknik Informatika**

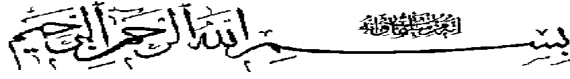


Oleh:

**Nama : Anjar Rahmawati
No.Mahasiswa : 04 523 013**

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
YOGYAKARTA
2011**

PERSEMBAHAN



Alhamdulillah...

*Segala puji bagi Allah SWT, atas rahmat, karunia, dan segala pengetahuan yang
Kau limpahkan hingga karya ini terselesaikan*

*Kupersembahkan setitik pencapaian awal dari masa depanku dengan penuh cinta dan kerendahan
hati kepada:*

-- Bapak Ibuku Tercinta --

*Tiada kata yang mampu terucap untuk membayar semua doa, kasih sayang, perhatian dan
semangat yang selalu kalian berikan kepadaku.*

-- Kakak dan Adikku --

Mas Dani dan Indra

Terima kasih atas kasih sayang, juga dukungan dan motivasi yang sangat berarti

-- My beloved hubby--

Jazakumullah khoiron katsiron, inni uhibuka fillah

Para Guru dan Dosen yang telah membimbingku

Jasamu tak akan kulupakan

Teman dan Sahabatku

dan

Seluruh orang yang menganggap penelitian ini bermanfaat

-- Terima Kasih --

MOTTO

إِنَّ مَعَ الْعُسْرِ يُسْرًا ﴿٦﴾ فَإِذَا فَرَغْتَ فَانصَبْ ﴿٧﴾

“*Sesungguhnya sesudah kesulitan itu ada kemudahan ; Maka apabila kamu telah selesai (dari suatu urusan), kerjakanlah dengan sungguh-sungguh (urusan) yang lain ”.*
(Q.S. Alam Nasyrah ayat 6 dan 7)

لَا يُكَلِّفُ اللَّهُ نَفْسًا إِلَّا وُسْعَهَا لَهَا مَا كَسَبَتْ وَعَلَيْهَا مَا أُكْتَسَبَتْ

“*Allah tidak membebani seseorang, melainkan sesuai dengan kesanggupannya... ”*
(QS. Al Baqarah : 286)

“*Jadilah sabar dan sholat sebagai penolongmu, sesungguhnya Allah beserta orang-orang yang sabar ”.*

(Q.S. Al Baqarah ayat 153)

KATA PENGANTAR



Assalamu'alaikum Warohmatullohi Wabarokatuh

Alhamdulillah, segala puji dan syukur panjatkan kehadirat Allah *subhanahu wata'ala* yang telah melimpahkan rahmat, taufik serta hidayahnya. Sholawat dan salam kepada junjungan kita Nabi Muhammad *shalallahu 'alaihi wasallam* beserta keluarga dan para sahabat, serta orang-orang yang bertaqwa, sehingga penulis dapat menyelesaikan Tugas Akhir ini sebagaimana mestinya.

Tugas akhir ini merupakan salah satu penerapan ilmu yang telah didapatkan selama kuliah. Dengan adanya penelitian ini, penulis berharap Aplikasi Pengenalan Ucapan Manusia dengan Jaringan Syaraf Tiruan Propagasi Balik ini akan dapat membantu dosen dan mahasiswa dalam melakukan kegiatan belajar mengajar.

Penulis menyampaikan ucapan terimakasih dan penghargaan yang setinggi-tingginya atas bantuan, bimbingan dan dukungan dari berbagai pihak yang ikut serta demi kelancaran pelaksanaan Tugas Akhir kepada :

1. Ir. Gumbolo HS M.Sc selaku Dekan Fakultas Teknologi Industri Universitas Islam Indonesia.
2. Bapak Yudi Prayudi, S.Si, M.Kom selaku Ketua Jurusan Teknik Informatika Universitas Islam Indonesia.

3. Bapak Taufiq Hidayat, ST, MCS selaku dosen pembimbing. Terima kasih atas segala bantuan, dukungan, semangat, dan pengetahuannya, serta kemudahan yang telah diberikan.
4. Bapak Zainudin Zukhri, ST, MCS selaku dosen pembimbing. Terima kasih atas segala bantuan, dukungan, semangat, dan pengetahuannya, serta kemudahan yang telah diberikan.
5. Untuk seluruh keluargaku tercinta Bapak, Ibu, kakak, adik, yang selalu mendoakanku dan memberikankan dukungan dan kasih sayang yang tak terhingga. Aku sayang kalian.
6. My hubby, terima kasih ya atas semua kesabarannya dalam menghadapi aku.
7. Teman-teman Informatika '04, terimakasih atas kekompakan dan kebersamaannya selama ini dan semua pihak yang tidak dapat penulis sebutkan satu persatu yang telah membantu sejak pengumpulan data sampai penyusunan Tugas Akhir ini.

Penulis menyadari bahwa Tugas Akhir ini masih jauh dari kesempurnaan, untuk itu penulis sangat mengharapkan kritik serta saran yang bersifat membangun untuk perbaikan di masa mendatang. Semoga Tugas Akhir ini bermanfaat untuk kita semua. Amin.

Wassalum'alaikum Warohmatullohi Wabarokatuh

Yogyakarta, April 2008

penyusun

ABSTRAKSI

Seiring kemajuan zaman, manusia menginginkan kepraktisan dalam teknologi. Membuat teknologi yang user friendly adalah salah satunya. Salah satu bentuk user friendly adalah membuat mesin yang mampu mengenali perintah dalam bentuk suara sehingga pengguna tidak perlu menggunakan keyboard dan mouse.

Jaringan Syaraf Tiruan (JST) adalah model sistem komputasi yang bekerja seperti jaringan syaraf manusia, nama jaringan syaraf tiruan merupakan terjemahan dari Artificial Neural Network, yang dapat diartikan bahwa JST memodelkan sistem komputasi yang dapat menirukan cara kerja jaringan syaraf biologis karena dasar dari jaringan syaraf tiruan adalah jaringan syaraf manusia.

Pengenalan ucapan manusia, khususnya suku kata dapat ditentukan menggunakan pola – pola yang dibuat menggunakan jaringan syaraf tiruan. Jaringan Syaraf Tiruan (JST) sendiri merupakan suatu model matematis yang mengambil sifat-sifat komputasi Jaringan Syaraf Manusia. Digunakan metode propagasi balik karena metode propagasi balik merupakan metode yang paling sederhana, mudah dipahami dari metode-metode yang lain dan dinilai memiliki tingkat akurasi yang baik. Pemrosesan suara menggunakan FFT yaitu proses mengubah sinyal suara yang berada pada ranah waktu menjadi ranah frekuensi.

Hasil penggunaan Jaringan Syaraf Tiruan metode propagasi balik dalam pengenalan pola suara manusia masih menghasilkan nilai kesalahan yang cukup tinggi. Hal ini bisa dikarenakan kesalahan pada jaringan yang dibuat atau karena masukan yang diberikan kepada jaringan . Semakin banyak data yang dilatih tingkat akurasi pada pengujiannya akan semakin baik karena banyaknya pola pelatihan yang tersimpan. Selain nilai bobot random dan nilai parameter juga cukup mempengaruhi akurasi hasil perhitungan.

Kata kunci: *Jaringan Syaraf Tiruan, BackPropagation, FFT*

DAFTAR ISI

HALAMAN JUDUL	i
LEMBAR PENGESAHAN PEMBIMBING.....	ii
LEMBAR PENGESAHAN PENGUJI.....	iii
HALAMAN PERNYATAAN KEASLIAN.....	iv
HALAMAN PERSEMBAHAN	v
HALAMAN MOTTO	vi
KATA PENGANTAR	vii
ABSTRAKSI.....	ix
DAFTAR ISI	x
DAFTAR GAMBAR.....	xiii
DAFTAR TABEL.....	xv
BAB I PENDAHULUAN	
1.1 Latar Belakang	1
1.2 Rumusan Masalah.....	4
1.3 Batasan Masalah.....	4
1.4 Tujuan Penelitiann	5
1.5 Manfaat Penelitian	5
1.6 Metode Penelitian.....	5
1.7 Sistematika Penulisan	6
BAB II LANDASAN TEORI	
2.1 Jaringan Syaraf Tiruan	9
2.1.1 Sejarah Jaringan Syaraf Tiruan	9
2.1.2 Struktur Dasar Pemodelan Jaringan Syaraf Tiruan	11
2.1.3 Jaringan Syaraf Tiruan Propagasi Balik.....	12
2.1.4 Fungsi Aktivasi	13

2.1.5	Arsitektur Propagasi Balik	15
2.1.6	Algoritma Pelatihan Propagasi Balik	16
2.1.7	Prosedur Pengujian Propagasi Balik	20
2.2	Sistem Pengenal Sinyal Tutur	21
2.2.1	FFT (Fast Fourier Transform)	23
2.2.2	Tata Bunyi Bahasa Manusia	24
2.2.3	Karakteristik Sinyal Tutur	25

BAB III ANALISIS KEBUTUHAN & PERANCANGAN PERANGKAT LUNAK

3.1	Tinjauan Sistem	28
3.2	Analisis Kebutuhan Perangkat Lunak	28
3.2.1	Analisis Kebutuhan Fungsional Perangkat Lunak	29
3.2.2	Analisis Kebutuhan Non Fungsional	30
3.2.3	Deskripsi Umum Perangkat Lunak	31
3.3	Perancangan Perangkat Lunak	31
3.3.1	Perancangan Awal	31
3.3.2	Perancangan Rinci	33

BAB IV HASIL DAN PEMBAHASAN

4.1	Batasan Implementasi	38
4.1.1	Aspek Perangkat Lunak	38
4.1.2	Aspek Perangkat Keras	38
4.1.3	Aspek Bahasa Pemrograman	39
4.2	Implementasi Perangkat Lunak	39
4.2.1	Implementasi Form Train Network	39
4.2.2	Implementasi Kode Program	41
4.3	Kinerja Perangkat Lunak	47
4.3.1	Kinerja Aplikasi Pengenalan Ucapan	47

4.3.2 Hasil Pengujian	53
4.3.3 Analisis Hasil Pengujian	56
4.3.4 Kelebihan dan Kekurangan Sistem	59

BAB V SIMPULAN DAN SARAN

5.1 Simpulan	61
5.2 Saran.....	62

DAFTAR PUSTAKA



DAFTAR GAMBAR

Gambar 2.1 Model tiruan sebuah sebuah.....	11
Gambar 2.2 Fungsi aktivasi linear identitas.....	14
Gambar 2.3 Fungsi aktivasi sigmoid biner.....	15
Gambar 2.4 Fungsi aktivasi sigmoid bipolar.....	15
Gambar 2.5 Arsitektur PB jaringan sederhana.....	16
Gambar 2.6 Diagram sinyal analog dan sinyal digital.....	22
Gambar 2.7 Diagram model sistem produksi suara.....	23
Gambar 3.1 Blok diagram pengenalan ucapan.....	28
Gambar 3.2 Use case diagram pelatihan.....	32
Gambar 3.2 Use case diagram pelatihan.....	32
Gambar 3.3 Class diagram jaringan syaraf tiruan propagasi balik.....	34
Gambar 3.4 Sequence diagram pelatihan.....	35
Gambar 3.5 Sequence diagram menua pengujian JST.....	36
Gambar 3.6 Activity diagram pengenalan ucapan.....	37
Gambar 4.1 Menu training network.....	40
Gambar 4.2 Menu utama testing network.....	40
Gambar 4.3 Kesalahan urutan jalannya program.....	48
Gambar 4.4 Grafik MSE dengan learning rate 0.001.....	49
Gambar 4.5 Grafik MSE dengan learning rate 0.5.....	50
Gambar 4.6 Grafik MSE dengan learning rate 0.3.....	50
Gambar 4.7 Grafik MSE dengan learning rate 0.1 dan 1 neuron.....	51
Gambar 4.8 Grafik MSE dengan learning rate 0.1 dan 16 neuron.....	51
Gambar 4.9 Menu pengujian	53
Gambar 4.10 Spektrum frekuensi sample 1 huruf A.....	57
Gambar 4.11 Spektrum frekuensi sample 1 huruf E.....	57
Gambar 4.12 Spektrum frekuensi sample 1 huruf I.....	58
Gambar 4.13 Spektrum frekuensi sample 1 huruf O.....	58
Gambar 4.14 Spektrum frekuensi sample 1 huruf U.....	59

DAFTAR TABEL

Tabel 4.1 Error pada masing-masing nilai.....	52
Tabel 4.2 Nilai error dengan learning rate 0.001.....	54
Tabel 4.3 Nilai error dengan learning rate 0.5.....	54
Tabel 4.4 Nilai error dengan learning rate 0.3.....	55
Tabel 4.5 Nilai error dengan learning rate 0.1 dan 1 neuron.....	55
Tabel 4.6 Nilai error dengan learning rate 0.1 dan 16 neuron.....	56



BAB I PENDAHULUAN

1.1 Latar Belakang Masalah

Perkembangan teknologi mengalami kemajuan yang sangat pesat. Hal ini berdampak pula pada perkembangan komputasi. Komputer tidak hanya digunakan sebagai alat penghitung namun dapat dibuat berfikir layaknya otak manusia. Membuat sistem komputer menjadi “cerdas” seperti manusia bisa dipelajari melalui ilmu komputer yang dikenal dengan kecerdasan komputasional (*computational intelligence*). Kecerdasan komputasional adalah bagian dari kecerdasan buatan (*artificial intelligence*). Dengan kecerdasan buatan sistem komputer mampu memberikan solusi seperti seorang pakar, hal ini dapat diwujudkan melalui sistem pakar. Sistem komputer mampu memberikan keputusan berbasiskan kasus seperti yang dilakukan manusia dengan menerapkan penalaran berbasis kasus (*case based reasoning*).

Tidak hanya itu sistem komputer dapat mengenali pola seperti halnya manusia dengan menggunakan jaringan syaraf tiruan (*neural network*). Algoritma sebuah program pun bisa menemukan solusi terbaik dengan menggunakan teori algoritma genetika (*genethic algorithm*). Selain algoritma genetika, telah berkembang juga metode optimasi yang dikenal dengan algoritma semut (*ant colony*) yang merupakan pengembangan dari metode *trafelling salesman problem*. Algoritma imun (*artificial immune system*) juga dikembangkan sebagai komputasi yang handal meniru kekebalan tubuh manusia. Selain itu juga berkembang teknologi fuzzy yang mengadopsi intuisi dan penalaran manusia.

Berkat adanya *artificial intelligence* manusia mulai berfikir dan mencari ide bagaimana membuat sebuah interaksi dengan komputer seperti interaksi antara manusia dengan manusia. Suara atau bahasa lisan adalah alat komunikasi antar manusia. Komunikasi suara merupakan model komunikasi yang paling efektif. Dalam penerapannya terbukti bahwa komunikasi dengan bahasa lisan akan lebih cepat daripada komunikasi dengan bahasa tulisan. Oleh karena itu dibuatlah suatu interaksi manusia dengan komputer melalui suara. Suara sebagai alat komunikasi manusia dengan komputer telah banyak dikembangkan. Pengenalan suara secara otomatis (*speech recogniton*) merupakan suatu kemampuan manusia yang dicoba diterapkan pada komputer.

Speech recogniton adalah sistem yang berfungsi untuk mengubah bahasa lisan menjadi bahasa tulisan. Masukan sistem berupa ucapan manusia, selanjutnya sistem akan mengidentifikasi kata atau kalimat yang diucapkan dan menghasilkan teks sesuai dengan apa yang diucapkan oleh pembicara. Dalam pengertian lain *speech recogniton* adalah pemberian perintah pada komputer dengan perintah suara. Tujuan utamanya adalah meringankan pekerjaan manusia. Contoh penggunaan *speech recogniton* adalah untuk mengontrol peralatan elektronik dengan mengucapkan perintah melalui peralatan audio (*voice command*). Hal ini memberikan kemudahan karena dapat mengontrol peralatan elektronik tanpa menggunakan tangan. Penerapan lain adalah *voice dialling* yang terdapat pada *handphone*. Selain aplikasi *speech recogniton* dikenal pula aplikasi *text to speech* (TTS). Berbeda dengan *speech recogniton*, TTS membutuhkan masukan berupa

teks atau tulisan. Komputer akan membangkitkan suara yang dapat dimengerti manusia. Aplikasi TTS yang telah dikembangkan adalah *translator* bahasa.

Jaringan syaraf tiruan (JST) merupakan salah satu metode komputasi yang dapat menyelesaikan masalah pengenalan pola meliputi citra, suara, *time series*, *prediction*, dan lain-lain. JST adalah suatu program komputer yang dibuat berdasarkan cara kerja otak manusia. JST ditujukan untuk merancang komputer sehingga dapat difungsikan untuk melakukan proses pelatihan dari suatu contoh. Dalam konteks lain JST adalah suatu rancangan alat penghitung yang ditujukan untuk melakukan proses serupa dengan apa yang dapat dilakukan oleh otak manusia.

Struktur atau arsitektur JST mencoba meniru arsitektur otak manusia. Salah satu struktur yang ingin ditiru adalah bentuk neuron dari otak. JST propagasi balik atau *Back propagation* yang terdiri dari beberapa lapisan banyak dimanfaatkan dalam bidang pengenalan pola. Propagasi balik melatih jaringan untuk mendapatkan keseimbangan antara kemampuan jaringan untuk mengenali pola yang digunakan selama pelatihan serta kemampuan jaringan untuk memberikan respon yang benar terhadap pola masukan yang serupa (tapi tidak sama) dengan pola yang dipakai selama pelatihan.

Untuk penerapan pengenalan ucapan (*speech recognition*) JST propagasi balik digunakan sebagai pengklasifikasi pola suara ke dalam kelas-kelas fonetik yaitu dengan mengenali segmentasi suara. Segmen suara ini dapat berupa suku kata, kata, atau fonem. JST akan menyusun kata dari fonem dan menganalisa hubungan antar kata untuk memastikan pengejaan yang benar pada kata yang

mirip dengan kata yang dilatih. Di sini JST akan melakukan pembelajaran dan menyimpan pengetahuannya. Dengan pengetahuan inilah JST dapat mengambil keputusan.

1.2 Rumusan Masalah

Berdasarkan latar belakang diatas maka dapat diambil suatu rumusan masalah, yaitu sejauh mana JST propagasi balik dapat mengenali sinyal ucapan manusia. Setelah dilakukan proses *training* sampai dicapai bobot yang optimal, selanjutnya dilakukan pengujian terhadap sinyal suara yang sama persis dengan data yang telah dilatih (*training data set*).

1.3 Batasan Masalah

Agar pembahasan dalam masalah ini menjadi lebih terarah dan untuk memahami permasalahan yang akan dibahas perlu batasan-batasan, yaitu :

- a. Pengenalan ucapan ini hanya berlaku pada pengenalan suku kata yang biasa diucapkan dalam percakapan sehari-hari.
- b. Sinyal suara dari *microphone* (sinyal analog) dicuplik menjadi sinyal digital dengan kecepatan 8000 Hz berdasarkan asumsi bahwa sinyal percakapan berada pada daerah frekuensi 300-3400 Hz.
- c. Normalisasi semua *input* yang berupa sinyal digital menggunakan metode menggunakan FFT (*Fast Fourier Transform*) untuk mendapatkan sinyal pada domain frekuensi.
- d. Hasil keluaran FFT merupakan masukan bagi JST propagasi balik.

1.4 Tujuan Penelitian

Tujuan dilakukannya penelitian adalah mengetahui sejauh mana keberhasilan sistem jaringan syaraf tiruan propagasi balik dalam melakukan pengenalan sinyal ucapan, serta mengetahui proses-proses yang berkaitan dengan perancangan penelitian ini.

1.5 Manfaat Penelitian

Dari hasil penelitian, dapat diperoleh manfaat yaitu sebagai berikut :

- a. Memberikan pemahaman mengenai konsep jaringan syaraf tiruan khususnya model propagasi balik.
- b. Memberikan tambahan pengetahuan mengenai ilmu kecerdasan buatan dan pemrosesan sinyal digital.

1.6 Metode Penelitian

Metode yang digunakan dalam penelitian ini adalah :

- a. Metode pengumpulan data

Melakukan proses perekaman suara sebagai data pelatihan. Analisis kebutuhan sistem dilakukan dengan melakukan studi pustaka, yaitu penulis melakukan kegiatan mengumpulkan dan mempelajari teori yang menunjang penulisan dengan bersumber dari buku, karya ilmiah, jurnal dan internet, yang berhubungan dengan jaringan syaraf tiruan dan pemrosesan sinyal digital.

- b. Metode pengembangan perangkat lunak

Pada metode ini dititikberatkan pada analisis kebutuhan perancangan dan implementasi perangkat lunak.

c. Metode pengujian perangkat lunak

Pada tahap ini perangkat lunak yang dihasilkan dari tahap-tahap sebelumnya (pelatihan dan pembelajaran) diuji agar menghasilkan keluaran seperti yang diinginkan.

1.7 Sistematika Penulisan

Berikut ini adalah sistematika penulisan laporan tugas akhir yang dilakukan yaitu:

BAB I PENDAHULUAN

Berisi tentang latar belakang masalah tentang dasar pengenalan ucapan oleh sebuah mesin, rumusan terhadap permasalahan berkaitan dengan metodologi berupa jaringan saraf tiruan, batasan masalah tentang kemampuan sistem menyelesaikan kasus, tujuan penelitian, manfaat penelitian, metode yang digunakan dalam penelitian serta sistematika penulisan laporan tugas akhir.

BAB II LANDASAN TEORI

Berisi penjelasan tentang jaringan syaraf tiruan secara umum, metode propagasi balik dan arsitekturnya serta penjelasan mengenai pengolahan sinyal digital.

BAB III METODOLOGI

Berisi tentang metode analisis kebutuhan perangkat lunak, perancangan perangkat lunak, dan implementasi perangkat lunak. Analisis kebutuhan perangkat lunak meliputi uraian analisis kebutuhan pemakai dan apa saja yang dibutuhkan

oleh program yang akan dibuat, seperti fungsi yang dibutuhkan, aliran data dan interface. Dengan metode analisis, memungkinkan fungsi yang dibutuhkan, masukan sistem dan keluaran sistem dapat didefinisikan dengan mudah. Pada perancangan perangkat lunak diuraikan tentang metode perancangan yang digunakan dan hasil perancangan yang diuraikan dalam bentuk perancangan UML (*unified modelling language*) serta perancangan antar muka (*interface*). Analisis terakhir berisi tentang batasan implementasi perangkat lunak yang memuat dokumentasi pelatihan dan pengujian.

BAB IV HASIL DAN PEMBAHASAN

Berisi tentang pengujian terhadap sistem yang telah dibuat yaitu dalam hal proses menjalankan sistem serta proses analisis sistem. Serta pembahasan mengenai kelebihan dan kekurangan aplikasi pengenalan suku kata manusia dengan metode jaringan saraf tiruan propagasi balik.

BAB V SIMPULAN DAN SARAN

Berisi tentang kesimpulan dari laporan tugas akhir serta saran-saran yang bersifat membangun agar aplikasi pengenalan ucapan suku kata manusia bisa dikembangkan lebih lanjut kedepannya.

BAB II LANDASAN TEORI

2.1 Jaringan Syaraf Tiruan

2.1.1 Sejarah Jaringan Syaraf Tiruan

Jaringan syaraf tiruan (JST) adalah suatu algoritma untuk menyelesaikan persoalan bersifat teori seperti pembelajaran dan optimasi. Dari segi fungsi, JST ditujukan untuk merancang suatu komputer sehingga dapat difungsikan untuk melakukan proses pelatihan dari suatu contoh. Sedangkan dari struktur rancangan, JST adalah suatu rancangan alat penghitung yang ditujukan agar dapat melakukan proses serupa dengan apa yang dilakukan oleh otak manusia. Sebenarnya JST adalah salah satu metode komputasi yang berfungsi seperti halnya metode komputasi yang lain. Istilah lainnya adalah alat pengolah data yang tidak begitu jauh berbeda dengan metode statistic.

Berdasarkan sejarah perkembangannya, JST diakui sebagai salah satu cabang ilmu pengetahuan sejak McCulloch dan Walter Pitts (1943) memperkenalkan teori umum tentang pemrosesan informasi berdasarkan sebuah jaringan yang terdiri dari dua pilihan atau sebuah keputusan masalah yang diistilahkan dengan “syaraf” walaupun sebenarnya secara factor biologi sangat beda jauh dan syaraf tentunya lebih kompleks [MUL95].

Elemen yang paling mendasar dan merupakan satuan unit terkecil dari jaringan syaraf adalah sel syaraf (neuron). Sel-sel syaraf inilah membentuk bagian kesadaran manusia yang meliputi beberapa kemampuan umum. Pada dasarnya sel syaraf biologi menerima masukan dari sumber yang lain dan

mengkombinasikannya dengan beberapa cara, melaksanakan suatu operasi yang *non-linear* untuk mendapatkan hasil dan kemudian mengeluarkan hasil akhir tersebut.

Dalam tubuh manusia terdapat banyak variasi tipe dasar sel syaraf, sehingga proses berpikir manusia menjadi sulit untuk direplikasi secara elektrik. Sekalipun demikian, semua sel syaraf alami mempunyai empat komponen dasar yang sama. Keempat komponen dasar ini diketahui berdasarkan nama biologinya yaitu, dendrit, soma, akson, dan sinapsis. Dendrit merupakan suatu perluasan dari soma (badan sel) yang menyerupai rambut dan bertindak sebagai saluran masukan. Fungsi dendrit adalah sebagai penyampai sinyal dari satu neuron ke neuron lain yang saling terhubung. Dendrit menerima masukan dari sel syaraf lain melalui sinapsis. Kemudian soma memproses nilai masukan dengan menjumlahkan sinyal-sinyal *input* yang masuk menjadi sebuah *output* yang kemudian dikirim ke sel syaraf lainnya melalui akson dan sinapsis.

Sel-sel syaraf ini terhubung satu dengan yang lainnya melalui sinapsis. Sel syaraf dapat menerima rangsangan berupa sinyal elektrokimiawi dari sel-sel syaraf yang lain. Berdasarkan rangsangan tersebut, sel syaraf akan mengirimkan sinyal atau tidak berdasarkan kondisi tertentu. Konsep dasar semacam inilah yang ingin dicoba oleh para ahli dalam menciptakan sel tiruan.

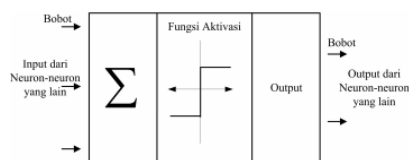
2.1.2 Struktur Dasar Pemodelan Jaringan Syaraf Tiruan

Jaringan syaraf tiruan adalah sebuah prosesor yang terdistribusi dan bekerja paralel serta mempunyai kecenderungan untuk menyimpan pengetahuan

yang didapatkannya dari pengalaman dan membuatnya tetap tersedia untuk digunakan. Hal ini menyerupai kerja otak dalam dua hal yaitu pengetahuan diperoleh oleh jaringan melalui suatu proses belajar dan kekuatan hubungan antar sel syaraf yang dikenal dengan bobot sinapsis yang digunakan untuk menyimpan pengetahuan.

Tiruan neuron biologi dalam struktur JST adalah elemen pemroses yang sering disebut neuron, unit, sel, atau node. Setiap neuron terhubung dengan neuron-neuron lain melalui layer dengan bobot tertentu. Bobot di sini melambangkan informasi yang digunakan oleh jaringan untuk menyelesaikan persoalan. Pada jaringan syaraf biologis, bobot tersebut dapat dianalogikan dengan aksi pada proses kimia yang terjadi pada *synaptic gap*.

Sedangkan, setiap neuron mempunyai kondisi dalam (*internal state*) yang disebut aktivasi. Aktivasi tersebut merupakan fungsi dari input yang diterima. Suatu neuron akan mengirim sinyal ke neuron-neuron lain, tetapi pada suatu saat, hanya ada satu sinyal yang dapat dikeluarkan walaupun sinyal tersebut ditransmisikan pada beberapa neuron lain. Model jaringan syaraf tiruan dapat dilihat pada gambar 2.1



Gambar 2.1 Model tiruan sebuah neuron

Setiap masukan atau *input* (x) akan dikirim ke neuron dengan bobot kedatangan tertentu (w). Input ini akan diproses oleh suatu fungsi perambatan yang akan menjumlahkan nilai-nilai semua bobot yang datang. Hasil penjumlahan

ini kemudian 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 diaktifkan, tapi kalau tidak, maka neuron tersebut tidak akan diaktifkan. Apabila neuron tersebut diaktifkan, maka neuron tersebut akan mengirimkan *output* melalui bobot-bobot keluarannya ke semua neuron yang berhubungan dengannya. Kemudian fungsi aktivasi akan mengaktifkan y_{in} menjadi *output* jaringan. Demikian seterusnya. Berikut adalah persamaannya:

$$y_{in} = w_1x_1 + w_2x_2 + w_3x_3 \quad (2.1)$$

2.1.3 Jaringan Syaraf Tiruan Propagasi Balik

Algoritma pelatihan Propagasi Balik (PB) pertama kali dirumuskan oleh Werbos dan dipopulerkan oleh McClelland. Algoritma ini termasuk metode pelatihan terbimbing (*supervised learning*) dan didesain untuk operasi pada JST *feed forward* lapis jamak (*multi layer*). Ketika JST diberikan pola masukan sebagai pola pelatihan (*feed forward*) maka pola tersebut menuju ke unit-unit pada lapisan tersembunyi untuk diteruskan ke unit-unit pada lapisan *output*. Setiap unit *output* kemudian akan menghitung aktivasinya dan mengirim sinyal ke tiap unit *output*. Kemudian unit-unit lapisan *output* memberikan tanggapan yang disebut sebagai *output* JST. Saat *output* JST tidak sama dengan target yang diharapkan maka keluaran akan disebarkan mundur (*backward*) pada lapis tersembunyi diteruskan ke unit pada lapisan *input*.

Tahap pelatihan ini merupakan langkah untuk melatih suatu JST, yaitu dengan cara melakukan perubahan bobot-bobot yang terhubung dengan neuron-neuron yang ada pada lapisan tersembunyi. Algoritma propagasi balik menggunakan *error output* untuk mengubah nilai bobot-bobotnya dalam arah mundur (*backward*). Untuk mendapatkan error ini, tahap perambatan maju (*forward propagation*) harus dikerjakan terlebih dahulu.

Jaringan propagasi balik yang dilatih secara tepat biasanya akan menghasilkan *output* yang tepat ketika dicoba dengan *input* yang belum pernah dilatihkannya. Suatu *input* baru akan menyebabkan jaringan mengeluarkan *output* sama dengan *output* yang muncul ketika dilatih dengan data *input* yang mirip dengan *input* baru tersebut. Dengan sifat generalisasi ini memungkinkan untuk melatih sebuah jaringan hanya dengan suatu representasi data (*input* dan *output*) saja tanpa perlu mencoba semua pasang vektor-vektor *input* dan *output* yang ada [ARB03].

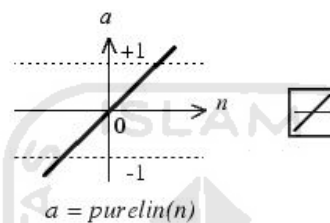
2.1.4 Fungsi Aktivasi

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. Suatu fungsi aktivasi untuk setiap sel sebuah JST Propagasi Balik mempunyai beberapa karakteristik penting sebagai berikut [MUN08]:

1. Harus kontinyu, dapat diturunkan.
2. Tidak naik atau turun secara monoton (*monotonically nondecreasing*).

Untuk mendapatkan efisiensi perhitungan, turunannya mudah dihitung. Pada umumnya nilai turunan fungsi aktivasi dapat diperoleh dari nilai rentang tertentu. Beberapa fungsi aktivasi yang sering digunakan dalam JST, tetapi yang biasa digunakan untuk metode propagasi balik adalah :

1. *Purelin* atau fungsi *linier* (identitas), memiliki nilai *output* yang sama dengan nilai input. Lihat gambar 2.2

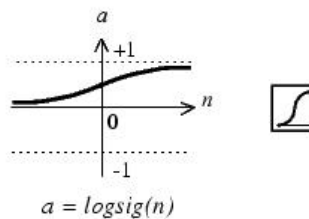


Gambar 2.2 Fungsi aktivasi *linear* identitas

2. *Logsig* (*sigmoid* biner), memiliki nilai *range* antara 0 sampai dengan 1. *Sigmoid* biner adalah fungsi aktivasi yang sering digunakan pada JST Propagasi Balik. Lihat gambar 2.3.

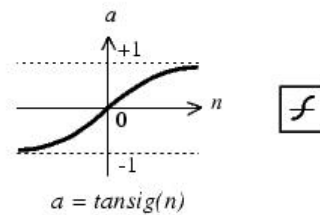
$$f(x) = (1 + e^{-x})^{-1} \quad (2.2)$$

$$\text{Dengan turunan } f'(x) = f(x)(1 - f(x)) \quad (2.3)$$



Gambar 2.3 Fungsi aktivasi *sigmoid* biner

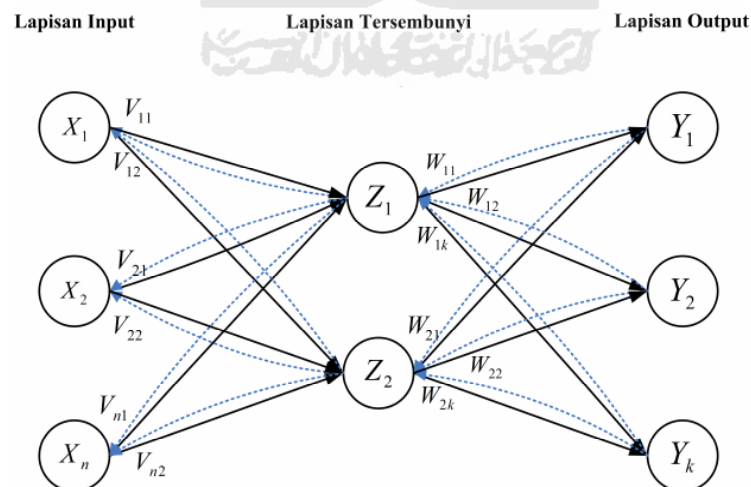
3. *Tansig* (*sigmoid* bipolar), hampir sama dengan fungsi *sigmoid* biner, hanya saja *output* dari fungsi memiliki *range* antara 1 sampai dengan -1. Lihat gambar 2.4.



Gambar 2.4 Fungsi aktivasi *sigmoid* bipolar

2.1.5 Arsitektur Propagasi Balik

JST PB memiliki beberapa unit yang ada dalam satu atau lebih lapisan tersembunyi. Arsitektur PB dapat dilihat pada Gambar 2.5 Dapat kita lihat, sebuah jaringan PB dengan X_1 , X_2 dan X_3 sebagai masukan (*input*), Z_1 dan Z_2 sebagai lapisan tersembunyi, serta Y_1 , Y_2 , dan Y_3 sebagai hasil keluaran (*output*).



Gambar 2.5 Arsitektur PB jaringan sederhana

2.1.6 Algoritma Pelatihan Propagasi Balik

Algoritma pelatihan pada jaringan syaraf tiruan Propagasi Balik adalah sebagai berikut:

Langkah 0 : Inisialisasi nilai bobot (diatur pada nilai acak yang kecil).

Langkah 1 : Selama kondisi berhenti masih tidak terpenuhi, lakukan langkah 2-9.

Langkah 2 : Untuk setiap pasangan pelatihan, lakukan langkah 3-8.

Perambatan maju (Feedforward) :

Langkah 3 : Tiap unit masukan ($X_i, i=1, \dots, n$) menerima sinyal masukan x_i dan menyebarkan sinyal itu ke semua unit pada lapisan di atasnya (lapisan tersembunyi).

Langkah 4 : Setiap unit lapisan tersembunyi ($Z_j, j=1, \dots, p$) dihitung nilai masukan dengan menggunakan nilai bobotnya :

$$z_in_j = v_{0j} + \sum_{i=1}^n x_i v_{ij} \quad (2.4)$$

v_{0j} = bias pada unit tersembunyi j , kemudian dihitung nilai keluaran dengan menggunakan fungsi aktivasi yang dipilih :

$$z_j = f(z_in_j) \quad (2.5)$$

Hasil fungsi tersebut dikirim ke semua unit pada lapisan di atasnya (unit keluaran).

Langkah 5 : Tiap unit keluaran ($Y_k, k=1, \dots, m$) dihitung nilai masukan dengan menggunakan nilai bobotnya :

$$y_in_k = w_{0k} + \sum_{j=1}^p z_j w_{jk} \quad (2.6)$$

w_{0k} = bias pada unit keluaran k , kemudian dihitung nilai keluaran dengan menggunakan fungsi aktivasinya :

$$y_k = f(y_{in_k}) \quad (2.7)$$

Perambatan balik (*Backpropagation*) :

Langkah 6 : Tiap unit keluaran (Y_k , $k=1, \dots, m$) menerima pola target yang berhubungan dengan pola masukan pelatihan, dan kemudian dihitung kesalahan informasinya :

$$\delta_k = (t_k - y_k) f'(y_{in_k}) \quad (2.8)$$

Kemudian dihitung koreksi nilai bobotnya yang kemudian akan digunakan untuk memperbaharui nilai bobot w_{jk} :

$$\Delta w_{jk} = \alpha \delta_k z_j \quad (2.9)$$

Hitung koreksi nilai biasnya yang kemudian akan digunakan untuk memperbaharui nilai w_{0k} :

$$\Delta w_{0k} = \alpha \delta_k \quad (2.10)$$

dan kemudian nilai δ_k dikirim ke unit pada lapisan di bawahnya.

Langkah 7 : Setiap unit lapisan tersembunyi (Z_j , $j=1, \dots, p$) dihitung perubahan masukan yang dari unit-unit pada lapisan di atasnya :

$$\delta_{in_j} = \sum_{k=1}^m \delta_k w_{jk} \quad (2.11)$$

Kemudian nilai tersebut dikalikan dengan nilai turunan dari fungsi aktivasinya untuk menghitung informasi kesalahannya :

$$\delta_j = \delta_{in_j} f'(z_{in_j}) \quad (2.12)$$

Hitung koreksi nilai bobot yang kemudian digunakan untuk memperbaharui nilai U_{ij} :

$$\Delta v_{ij} = \alpha \delta_j x_i \quad (2.13)$$

dan hitung nilai koreksi bias yang kemudian digunakan untuk memperbaharui U_{oj} :

$$\Delta U_{oj} = \alpha \delta_j \quad (2.14)$$

Memperbaharui nilai bobot dan bias :

Langkah 8 : Tiap unit ($Y_k, k=1, \dots, m$) keluaran diperbaharui nilai bias dan bobotnya ($j=0, \dots, p$):

$$w_{jk}(\text{baru}) = w_{jk}(\text{lama}) + \Delta w_{jk} \quad (2.15)$$

dan pada tiap unit lapisan tersembunyi ($Z_j, j = 1, \dots, p$) diperbaharui bias dan bobotnya ($i = 1, \dots, n$):

$$v_{ij}(\text{baru}) = v_{ij}(\text{lama}) + \Delta v_{ij} \quad (2.16)$$

Langkah 9 : Menguji apakah kondisi berhenti sudah terpenuhi. Jika kondisi berhenti telah terpenuhi, pelatihan jaringan dapat dihentikan. Untuk menentukan kondisi berhenti terdapat dua cara yang biasa dipakai, yaitu :

- a. Pertama, dengan membatasi iterasi yang ingin dilakukan.
- b. Kedua, dengan membatasi error. Pada metode Backpropagation, dipakai metode Mean Square Error untuk menghitung rata-rata error antara output yang dikehendaki pada training data dengan

output yang dihasilkan oleh jaringan. Cara memeriksa kondisi berhenti dengan Mean Square Error adalah sebagai berikut :

Mean Square Error :

$$\widehat{\text{MSE}}(\hat{\theta}) = \frac{1}{n} \sum_{j=1}^n (\theta_j - \theta)^2 \quad (2.17)$$

Suatu jangka waktu (*epoch*) adalah satu set putaran vektor-vektor pelatihan. Beberapa *epoch* diperlukan untuk pelatihan sebuah JST PB. Dalam algoritma ini dilakukan perbaikan bobot setelah masing-masing pola pelatihan dimasukan. Setelah pelatihan selesai bobot-bobot yang telah diperbaiki disimpan.

2.1.7 Prosedur Pengujian Propagasi Balik

Setelah pelatihan, sebuah jaringan syaraf Propagasi Balik diaplikasikan hanya pada fase umpan maju (*feedforward*). Prosedur aplikasinya adalah sebagai berikut:

Langkah 0 : Inisialisasi bobot awal (hasil dari pelatihan).

Langkah 1 : Untuk tiap vektor masukan, lakukan langkah 2-4.

Langkah 2: Untuk setiap unit masukan, distribusikan masukan x_i ke setiap unit diatasnya (unit dalam).

Langkah 3 : Untuk $j = 1, \dots, p$; gunakan persamaan berikut :

$$z_in_j = v_{0j} + \sum_{i=1}^n x_i v_{ij} \quad (2.18)$$

$$z_j = f(z_in_j) \quad (2.19)$$

Langkah 4 : Untuk $k = 1, \dots, m$; gunakan persamaan sebagai berikut :

$$y_{-in_k} = w_{0k} + \sum_{j=1}^p z_j w_{jk} \quad (2.20)$$

$$y_k = f(y_{-in_k}) \quad (2.21)$$

2.2 Sistem Pengenal Sinyal Tutar

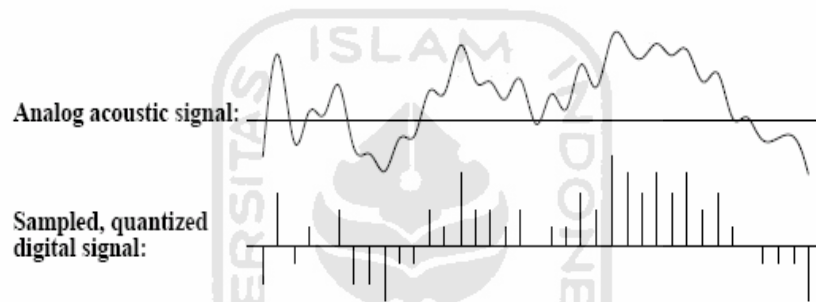
Sinyal dapat didefinisikan sebagai kuantitas fisik yang bervariasi seiring waktu atau variabel bebas lainnya yang menyimpan suatu informasi. Contoh sinyal adalah: suara manusia, kode morse, tegangan listrik di kabel telepon, variasi intensitas cahaya pada sebuah serat optik yang digunakan pada telepon atau jaringan komputer, dan lain-lainnya.

Sinyal dapat diklasifikasikan menjadi beberapa jenis yaitu: sinyal waktu kontinu, sinyal waktu diskrit, sinyal nilai kontinu, sinyal nilai diskrit, sinyal *random*, dan sinyal *nonrandom*. Sinyal waktu kontinu dengan nama lain sinyal analog adalah sinyal yang belum melalui proses apapun. Sedangkan sinyal nilai diskrit atau sinyal digital adalah sinyal analog yang telah melalui proses *sampling*, *quantization*, dan *encoding*. Beberapa proses yang dilakukan dalam pengolahan sinyal digital:

1. *Sampling* adalah proses mengambil nilai-nilai sinyal pada titik-titik diskrit sepanjang variabel waktu dari sinyal waktu kontinu, sehingga didapatkan sinyal waktu diskrit. Jumlah titik-titik yang diambil setiap detik dinamakan sebagai *sampling rate*. Dalam melakukan *sampling*, perlu diperhatikan kriteria *Nyquist* yang menyatakan bahwa sebuah sinyal harus memiliki *sampling rate*

yang lebih besar dari $2f_m$, dengan f_m adalah frekuensi paling tinggi yang muncul disebuah sinyal.

2. *Quantization* adalah proses memetakan nilai-nilai dari sinyal nilai kontinu menjadi nilai-nilai yang diskrit, sehingga didapatkan sinyal nilai diskrit.
3. *Encoding* adalah proses mengubah nilai-nilai sinyal menjadi bilangan biner. Pada gambar 2.6 dapat dilihat perbedaan antara sinyal analog dengan sinyal digital.

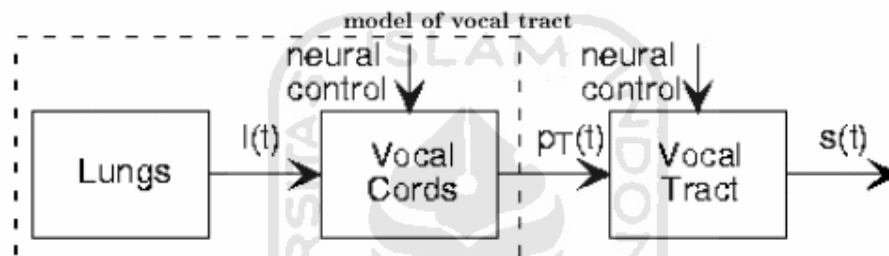


Gambar 2.6 Diagram sinyal analog dan sinyal digital

Sinyal yang berbentuk digital dapat disimpan dalam media penyimpanan di komputer. WAV file (berasal dari kata *wave*) merupakan format umum yang paling sederhana untuk menyimpan data sinyal audio. WAV file terdiri dari 3 potongan informasi yaitu: *RIFF chunk*, *FORMAT chunk*, dan *DATA chunk*. *RIFF chunk* berisi informasi yang menandakan bahwa file berbentuk WAV. *FORMAT chunk* berisi parameter-parameter seperti jumlah channel, *sample rate*, resolusi. *DATA chunk* yang berisi data aktual sinyal digital. File wave terdiri dari dua bagian yaitu *header* dan data, bagian *header* berisi nilai-nilai heksadesimal dan karakter ASCII yang tersimpan dalam variabel-variabel tertentu. Sedangkan pada

bagian data merupakan isi dari file wave yang berupa nilai heksadesimal yang merepresentasikan amplitudo dari file wave tersebut.

Sinyal yang dihasilkan dari suara manusia sewaktu melakukan percakapan disebut sebagai sinyal percakapan. Sinyal percakapan merupakan kombinasi kompleks dari variasi tekanan udara yang melewati pita suara dan *vocal tract*, yaitu mulut, lidah, gigi, bibir, dan langit-langit. Sistem produksi sinyal percakapan dapat dilihat pada gambar 2.7.



Gambar 2.7 Diagram model sistem produksi suara

Sinyal suara manusia yang diterima dengan menggunakan *microphone* masih berupa sinyal analog. Sinyal analog dari *microphone* dicuplik dengan frekuensi 8000 Hz untuk menghindari terjadinya pengaliansan. Kecepatan cuplik tersebut dilakukan dengan didasarkan asumsi bahwa sinyal percakapan (*speech*) berada pada daerah frekuensi 300-3400 Hz sehingga memenuhi kriteria Nyquist yang menyatakan :

$$f_s \geq 2f_h \quad f_h = f_{in\text{ tertinggi}} \quad (2.22)$$

2.2.1 FFT (Fast Fourier Transform)

Fast Fourier Transform merupakan penyederhanaan dari *Discrete Fourier Transform* (DFT). Untuk sinyal waktu diskrit $x(n)$, maka DFT dari sinyal diberikan oleh :

$$X(k) = \sum_{n=0}^{N-1} x(n)e^{-j2\pi nk/N}, k = 0,1,\dots,N-1 \quad (2.23)$$

Faktor eksponensial dalam persamaan tersebut dinamakan *twiddle faktor* yang bersifat periodik dengan periode N dan dilambangkan dengan W_N^{nk} , sehingga DFT dari sinyal waktu diskrit $x(n)$ dapat dituliskan sebagai :

$$X(k) = \sum_{n=0}^{N-1} x(n)W_N^{nk}, k = 0,1,\dots, N-1 \quad (2.24)$$

Dengan *Fast Fourier Transform*, maka jumlah titik data N merupakan bilangan yang dapat difaktorkan sehingga seluruh jumlah titik DFT dapat dipecah ke dalam kelompok-kelompok yang makin lama makin kecil. Pada algoritma FFT *radix 2 Decimation In Frequency* (DIF), data sebanyak N dibagi menjadi dua bagian, yaitu data dengan indeks 0 sampai dengan $N/2-1$ dan data dengan indeks $N/2$ sampai dengan $N-1$. Dengan memanfaatkan sifat FFT yang simetri maka data dengan indeks 0 sampai $N-1$ dapat dibagi dua yaitu hanya dengan menggunakan data 0 sampai $N/2-1$ atau $N/2$ sampai $N-1$. Dengan demikian jumlah perhitungan dapat dikurangi. Sifat simetri tersebut adalah :

$$W_N^{-nk} = -W_N^k \quad (2.25)$$

dimana,

$$W_N = e^{-j2\pi/N} \quad (2.26)$$

2.2.2 Tata Bunyi Bahasa Manusia

Ucapan manusia dihasilkan sebagai rangkaian atau urutan komponen-komponen bunyi pembentuknya. Setiap komponen bunyi yang berbeda dibentuk oleh perbedaan posisi, bentuk, serta ukuran dari alat-alat ucap manusia yang berubah-ubah selama terjadinya proses produksi ucapan. Ucapan manusia sendiri dibentuk oleh suatu sistem produksi ucapan yang dibentuk oleh alat-alat ucap manusia, yaitu *vocal cords (glottis)* yang terdiri dari *pharynx* (koneksi antara esophagus dan mulut) dan mulut.

2.2.3 Karakteristik Sinyal Tutur

Unit bunyi terkecil yang dapat dibedakan manusia disebut fonem. Suatu ucapan kata atau kalimat pada prinsipnya dapat dilihat sebagai urutan fonem. Fonem dibedakan menjadi dua yaitu:

1. Fonem Vokal

Fonem vokal merupakan bunyi yang dihasilkan dengan udara yang keluar dari paru-paru di daerah dasar ucapan yang tidak mengalami hambatan atau rintangan pada waktu bunyi tersebut dilafalkan.

Fonem vokal dibedakan menjadi dua:

- a. Vokal tunggal : / i / e / ə / a / o / u
- b. Vokal diftong : Diftong pada prinsipnya adalah dua fonem vokal berurutan dan diucapkan tanpa jeda. Dalam bahasa Indonesia dijumpai diftong au / ai / oi / atau aw / ay/ dan /oy/.

2. Fonem Konsonan

Bunyi konsonan merupakan bunyi yang dihasilkan dengan mengalami penghambatan atau perintangannya arus udara dari paru-paru pada waktu bunyi itu dilafalkan. Hambatan itu dapat secara total, maupun dapat pula sebagian saja. Fonem konsonan dalam bahasa Indonesia terdiri dari 22 fonem konsonan, yaitu : / b / p / t / d / c / j / k / g / f / s / z / š / x / m / n / ŋ / r / l / w / dan / y /.

Konsonan terbagi menjadi dua :

- a. Konsonan nasal : m, nm, ny, ng (bila keluar melalui rongga hidung)
- b. Konsonan oral : b, c, d, f, g, dst. (kecuali vocal: keluar melalui rongga mulut).

3. Fonem Gabungan 2 konsonan

Gabungan huruf konsonan dalam bahasa Indonesia terdiri dari kh, ng, ny, sy, sw, dan kw.

4. Suku Kata

Suku kata adalah bagian kata yang diucapkan dalam satu hembusan nafas dan umumnya terdiri atas beberapa fonem. Suku kata dalam bahasa Indonesia selalu memiliki vokal yang menjadi puncak suku kata. Puncak itu dapat didahului dan diikuti oleh suatu konsonan atau lebih, meskipun dapat terjadi bahwa suku kata hanya terdiri atas satu vokal, satu vokal dengan satu konsonan, atau satu vokal dengan beberapa konsonan. Suku kata dapat dibedakan menjadi :

- a. Suku kata terbuka, jika suku kata tersebut diakhiri dengan vokal
- b. Suku kata tertutup, jika suku kata tersebut diakhiri dengan konsonan.

Contoh:

V : i-tu

VK :in-tan

KV : ka-um

KVK : pas-ti

KKV : slo-gan

KKVK : kon-trak

KVKK : teks-til

KKKV : stra-ta

KKKVK :struk-tur

KKVKK : kom-pleks

KVKKK : korps

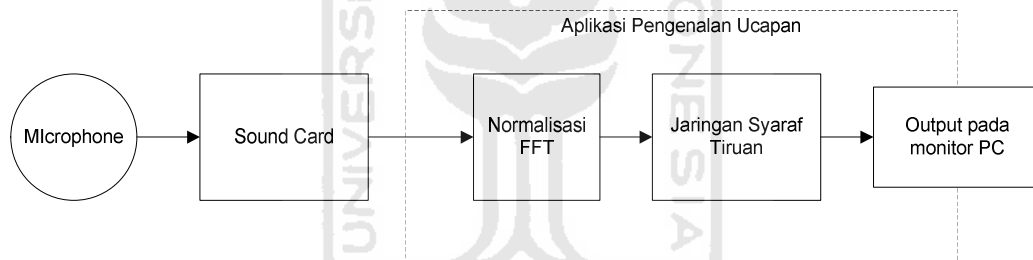
5. Kata

Kata Merupakan kumpulan bunyi ujaran atau satuan bahasa yang memiliki satu pengertian, mengandung arti, atau dalam bahasa tulis, kata dinyatakan sebagai susunan huruf-huruf yang mempunyai arti yang jelas (huruf konsonan dan vokal).

BAB III ANALISIS KEBUTUHAN & PERANCANGAN PERANGKAT LUNAK

3.1 Tinjauan Sistem

Sistem aplikasi ini berbasis PC (Personal Computer). Sistem aplikasi ini menerima masukan berupa sinyal suara analog kemudian dirubah menjadi sinyal digital dengan bantuan *sound card* pada PC. Sinyal digital tersebut yang telah berbentuk PCM dengan format data wav disimpan dalam beberapa file data. Data-data ini kemudian menjadi input bagi aplikasi pengenalan ucapan. Gambar berikut memperlihatkan sistem pengenalan ucapan yang dikembangkan.



Gambar 3.1 Blok diagram pengenalan ucapan

3.2 Analisis Kebutuhan Perangkat Lunak

Analisis kebutuhan perangkat lunak dari aplikasi pengenalan ucapan mencakup analisis terhadap kebutuhan fungsional dan analisis terhadap kebutuhan non fungsional. Analisis kebutuhan yang dilakukan terhadap perangkat lunak akan menghasilkan spesifikasi kebutuhan perangkat lunak tersebut.

Perangkat lunak yang dikembangkan akan digunakan untuk mengenali 5 buah data *output* yang berupa vokal 'a', 'e', 'i', 'o', dan 'u'. Dimana aplikasi akan

dianalisis dengan beberapa parameter berupa *learning rate*, *momentum*, dan jumlah neuron pada *hidden layer*.

3.2.1 Analisis Kebutuhan Fungsional Perangkat Lunak

Analisis kebutuhan fungsional perangkat lunak merupakan analisis terhadap kebutuhan-kebutuhan fungsional yang diperlukan didalam lingkup perangkat lunak yang akan dikembangkan. Kebutuhan fungsi utama perangkat lunak pengenalan ucapan dapat dilihat pada diagram UML (*Unified Modelling Language*). Proses yang akan terjadi pada perangkat lunak pengenalan ucapan adalah:

1. Perekaman yaitu pengubahan sinyal tutur analog ke sinyal tutur digital dengan bantuan *soundcard* pada PC (*personal computer*) melalui piranti *microphone* (pencuplikan). Pencuplikan dilakukan pada kecepatan 8000 Hz, dengan durasi perekaman (t) untuk setiap kata adalah 0.2 detik
2. Sinyal digital hasil cuplikan terlebih dahulu dinormalisasi menggunakan FFT dan merubah menjadi sinyal domain frekuensi.
3. Data hasil keluaran FFT akan menjadi masukan bagi jaringan syaraf tiruan propagasi balik.

3.2.2 Analisis Kebutuhan Non Fungsional

Kebutuhan non-fungsional merupakan kebutuhan perangkat keras dan perangkat lunak yang menyangkut aspek unjuk kerja (*performance*) aplikasi yang harus dipenuhi. Analisis kebutuhan non-fungsional aplikasi pengenalan ucapan meliputi:

1. Kebutuhan Hardware

- Processor Intel Intel Pentium 4, Intel Centrino, Intel Xeon, atau Intel Core Duo (atau yang setara) minimal 1.8 GHz.
- RAM 512 MB (direkomendasikan 2 GB)
- Ruang *hard disc* tersedia 256 MB
- Microphone

2. Kebutuhan Perangkat Lunak

- Sistem operasi Windows XP dengan Service Pack 3, atau Windows Vista Home Premium, Business, Ultimate, atau Enterprise (edisi 32-bit)
- Sun Microsystems Java 2 Standard Edition SDK (J2SDK 1.6)
- AudaCity
- Perancangan UML menggunakan Rational Rose

3. Kebutuhan Bahasa Pemrograman

- NetBeans IDE 6.5

4. Kebutuhan Sistem Operasi

- Windows XP 2000

3.2.3 Deskripsi Umum Perangkat Lunak

Aplikasi pengenalan ucapan akan menerima data *input* berupa sinyal digital dari suara manusia yang berupa ucapan vokal 'a', 'e', 'i', 'o', dan 'u'. Kemudian sinyal ini dinormalisasi dan dirubah menjadi domain frekuensi dengan menggunakan algoritma FFT. Hasil FFT akan menjadi *input* bagi jaringan syaraf tiruan propagasi balik. Output aplikasi pengenalan ucapan berupa teks vokal 'a', 'e', 'i', 'o', dan 'u'.

3.3 Perancangan Perangkat Lunak

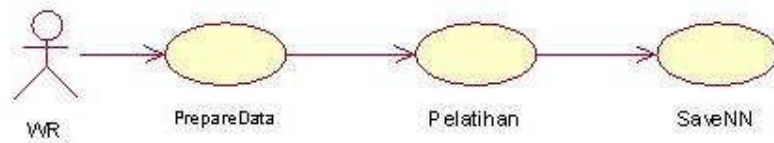
Kebutuhan perancangan adalah kebutuhan perangkat lunak terhadap aspek-aspek yang berhubungan dengan perancangan yang meliputi:

3.3.1 Perancangan Awal

Perancangan awal menggunakan UML. UML merupakan bahasa pemodelan yang dapat digunakan untuk berbagai tujuan, yang menggunakan standar notasi tertentu. UML juga menjadi standar industri yang dibuat di bawah pengawasan *Object Management Group* (OMG). Adapun UML yang dipakai dalam perancangan awal adalah *use case diagram*.

Use case diagram menggambarkan fungsi-fungsi yang berlangsung dilihat dari sisi pengguna. Diagram ini menunjukkan struktur sederhana dari suatu sistem. *Use case diagram* bekerja dengan cara mendeskripsikan tipikal interaksi antar pengguna sebuah sistem dengan sistem itu sendiri melalui sebuah skenario. Skenario adalah urutan langkah-langkah yang menerangkan antar pengguna dan sistem. Setiap skenario mendeskripsikan urutan kejadian, setiap urutan diinisialisasi oleh orang, sistem yang lain, perangkat keras, atau urutan waktu.

Pada aplikasi pengenalan ucapan, *use case* menjelaskan tentang hubungan antara sistem dengan aktor. Hubungan ini dapat berupa *input* aktor ke sistem ataupun *output* ke aktor. Gambar 3.2 menjelaskan aplikasi pengenalan ucapan dalam model *use-case diagram* dari pelatihan jaringan.

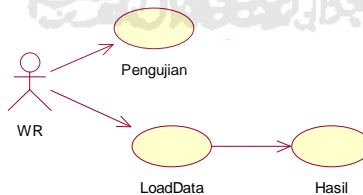


Gambar 3.2 Use case diagram pelatihan

Pada gambar 3.2 menjelaskan proses yang terjadi pada aplikasi pengenalan ucapan. Dalam aplikasi ini hanya terdapat satu *actor* yang terlibat dalam penggunaan aplikasi. Yaitu *actor* WR. Proses yang terjadi adalah :

- a. Prepare data: Mempersiapkan data suara yang akan diproses. File data suara disimpan dalam satu folder. Di dalam proses ini terdapat normalisasi sinyal dan proses FFT.
- b. Pelatihan: Setelah data siap, kemudian data suara dilatih.
- c. Save NN: Setelah dilakukan pelatihan, jaringan syaraf tiruan kemudian disimpan.

Adapun *use case diagram* pengujian dapat dilihat pada gambar 3.3 berikut ini.



Gambar 3.3 Use case diagram pengujian

Pada gambar 3.3 menjelaskan proses yang terjadi pada pengujian jaringan aplikasi pengenalan ucapan. Dalam aplikasi ini hanya terdapat satu *actor* yang terlibat dalam penggunaan aplikasi. Yaitu *actor* WR. Proses yang terjadi adalah:

- a. Load data: Apabila akan melakukan pengujian, actor akan membuka file jaringan yang telah disimpan.
- b. Pengujian: Menguji jaringan yang telah dilatih dan disimpan sebelumnya.
- c. Hasil: Hasil pengenalan dengan jaringan yang digunakan.

3.3.2 Perancangan Rinci

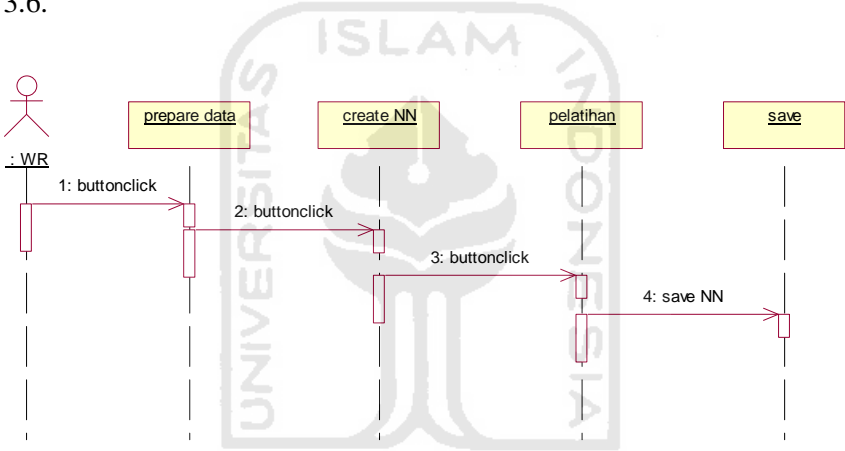
Untuk lebih menjelaskan perancangan aplikasi yang dibangun, digunakan *class diagram*, *sequence diagram* dan *activity diagram*.

A. Class Diagram

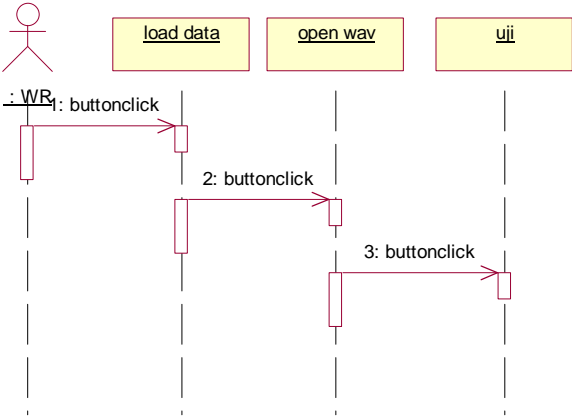
Class Diagram menggambarkan struktur dan deskripsi *class*, *package*, dan *objek* beserta hubungan antar *class* di dalam sistem. *Class diagram* menunjukkan aspek statik sistem terutama untuk mendukung kebutuhan fungsional sistem. Diagram ini menggunakan notasi UML untuk *class diagram*. Hasil perancangan *class diagram* yang digunakan untuk melakukan visualisasi struktur kelas-kelas yang terdapat dalam aplikasi pengenalan ucapan terlihat pada gambar 3.4.

B. Sequence Diagram

Sequence diagram menunjukkan urutan waktu dan proses yang terjadi pada sebuah fungsi sistem sehingga dapat terlihat pertukaran data yang terjadi diantaranya, dan perubahan apa saja yang terjadi secara internal serta output apa yang dihasilkan. Sequence diagram terdiri dari dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). Berikut akan dijelaskan rancangan sequence diagram untuk aplikasi pengenalan ucapan. Lihat gambar 3.5 dan gambar 3.6.



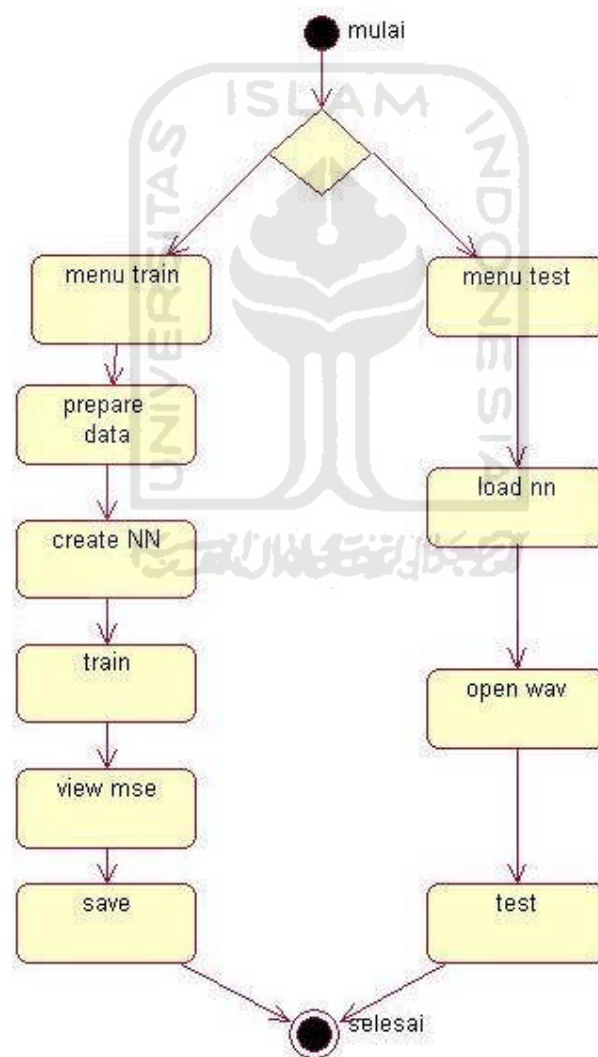
Gambar 3.5 Sequence diagram pelatihan



Gambar 3.6 Sequence diagram menu pengujian JST

C. Activity Diagram

Activity diagram menggambarkan berbagai aliran aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing aliran berawal, *decision* yang mungkin terjadi dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. Lihat gambar 3.7.



Gambar 3.7 Activity diagram pengenalan ucapan

BAB IV HASIL DAN PEMBAHASAN

4.1 Batasan Implementasi

Implementasi merupakan tahap dimana sistem siap dioperasikan pada tahap yang sebenarnya, sehingga akan diketahui apakah sistem yang telah dibuat benar-benar sesuai dengan yang direncanakan. Pada implementasi perangkat lunak ini akan dijelaskan bagaimana program sistem ini bekerja, dengan memberikan tampilan *form-form* yang dibuat. Adapun Batasan implementasi aplikasi ini dapat ditinjau dari 2 aspek, yaitu aspek perangkat lunak dan aspek perangkat keras.

4.1.1 Aspek Perangkat Lunak

Dalam implementasi aplikasi pengenalan ucapan ini digunakan beberapa perangkat lunak pendukung seperti:

- a. Sistem operasi Windows XP Profesional 2002 Service Pack 2.
- b. Perangkat lunak Sun Microsystem Java 2 Standard Edition SDK (J2SDK 1.6).

4.1.2 Aspek Perangkat Keras

Perangkat keras yang dibutuhkan untuk mengimplementasikan aplikasi pengenalan ucapan secara optimum adalah :

- a. Processor Intel® Core (TM) 2 Duo 2 GHz
- b. Har disk 80 GB
- b. RAM 1 GB

4.1.3 Aspek Bahasa Pemrograman

Dipilihnya JAVA sebagai bahasa pemrograman dalam aplikasi ini karena direncanakan dibangun sebuah sistem terpadu yang membutuhkan aplikasi berbasis java. Selain itu bahasa pemrograman java bersifat *multi platform*.

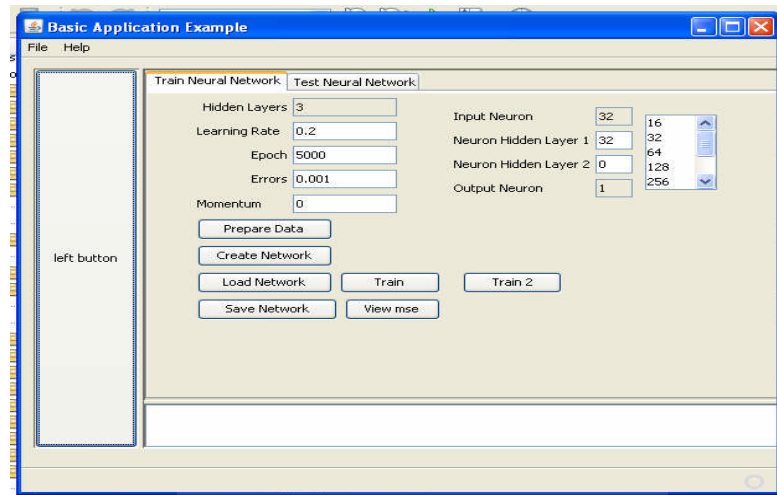
4.2 Implementasi Perangkat Lunak

Pada Bab III telah dijelaskan rancangan aplikasi pengenalan ucapan berupa halaman utama yang terdiri dari menu *testing* dan *training*. Berikut implementasinya.

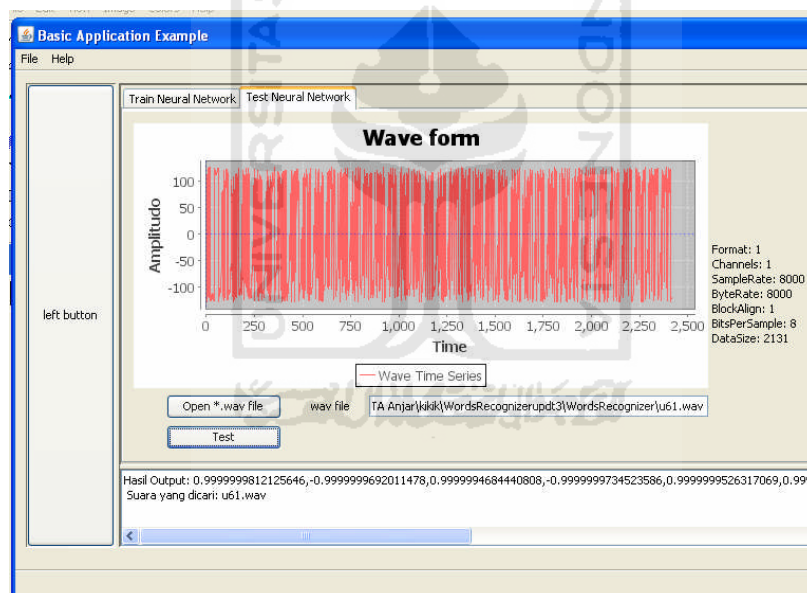
4.2.1 Implementasi Form Train Network

Pada Halaman utama terdapat dua menu yaitu menu train neural network dan menu test neural network. User memilih menu train kemudian memasukkan parameter yang diperlukan untuk proses *training* jaringan. Setelah pelatihan berakhir user menyimpan data pelatihan kemudian melihat *mean square error* dari *training*. Kemudian user memilih menu test untuk menguji jaringan syaraf tiruan. Training network dapat dilihat pada gambar 4.1 dan testing network dapat dilihat pada gambar 4.2.

Pada bagian testing network sebagaimana ditunjukkan gambar 4.2 user juga bisa melihat bentuk gelombang yang dihasilkan dan dibagian bawah bisa dibaca hasil pola output data yang diuji.



Gambar 4.1 Menu training network



Gambar 4.2 Menu utama testing network

4.2.2 Implementasi Kode Program

A. Metode Untuk Mempersiapkan Data Pelatihan Jaringan Syaraf Tiruan

List program dibawah adalah bagaimana aplikasi mengambil file berekstensi .wav pada folder yang sama dengan program. Selanjutnya program menentukan target

sebanyak 5 buah target. Data dalam bentuk file wav akan dibaca satu persatu dan disusun dalam bentuk matriks. Kemudian pada data dilakukan proses FFT.

```
private void waveProcessing(){
    Filter nf = new Filter ("wav");
    // current directory
    File dir = new File (".");
    wavfiles = dir.list(nf);
    M =wavfiles.length; //----- jumlah file wav -----
    ytarget = new double[M][5];

    //String aString = ""+j1Number.getSelectedValue();
    String aString=tfInputNeuron.getText();
    N = Integer.parseInt(aString);
    //-----process wav per file-----
    for (int i = 0; i < M; i++) {
        wavFile = new wavIO(wavfiles[i]);
    //-----process fft-----
        x = new Complex[N];
        if( wavFile.read() == true )
        {
            for (int j = 0; j < N; j++) {
                x[j] = new Complex(j, 0);
                x[j] = new Complex(wavFile.myData[j], 0);
            }
        }
    }

    y = FFT.fft(x);
    System.out.println("file "+i+": "+wavfiles[i]);
}
```

List program selanjutnya adalah proses normalisasi yaitu dengan cara membagi besaran-besarnya dengan nilai terbesar sehingga nanti yang didapatkan adalah nilai maksimal 1 dan minimal -1.

```

//-----Normalisasi besar/scalar-----
yinput = new double[M][N];
double[][] yscal = new double[M][N/2];
for (int j = 0; j < N/2; j++) {
    yscal[i][j] = y[j].abs();
    // System.out.println(""+yinput[i][j]);
}
double[] yi = new double[N/2];
double[] yn = new double[N/2];
for (int j=0; j<N/2; j++)
    yi[j]=yscal[i][j];
yn= getNorm(yi);
for (int k=0; k<N/2; k++){
    yscal[i][k]=yn[k];
    System.out.println(""+yscal[i][k]);
}
//-----normalisasi besar argumen-----
double[][] yarg = new double[M][N/2];
for (int j = 0; j < N/2; j++) {
    yarg[i][j] = Math.atan(y[j].re()/y[j].im());
    // System.out.println(""+yinput[i][j]);
}
double[] yia = new double[N/2];
double[] yna = new double[N/2];
for (int j=0; j<N/2; j++)
    yia[j]=yarg[i][j];
yna= getNorm(yia);
for (int k=0; k<N/2; k++){
    yarg[i][k]=yna[k];
    if (k<N/2-1)
        System.out.print(""+yarg[i][k]+",");
    else
        System.out.println(""+yarg[i][k]);
}
//-----gabungan besar skalar dg argumen-----
for (int j=0; j<N; j++){
    if (j<N/2)
        yinput[i][j]=yscal[i][j];
    else
        yinput[i][j]=yarg[i][j-N/2];
}

```

List program dibawah membentuk target yang disusun dalam matrix.

```
//----- building target matrix -----
double[] identity = getIdentityTarget(5, wavfiles[i]);
for (int j = 0; j < 5; j++)
ytarget[i][j] = identity[j];
}
/*
//-----set target-----
ytarget = new double[M][M];
generateIdentityTarget(M);
*/
for (int i=0; i<M; i++){
for (int j=0; j<5; j++) {
if (j<4)
System.out.print(""+ytarget[i][j]+",");
else
System.out.println(""+ytarget[i][j]);
}
}
}
```

B. Methode Untuk Membuat Jaringan Syaraf Tiruan

Berikut adalah list program untuk membuat arsitektur jaringan dengan menentukan layer dan fungsi aktivasi. Disini ada beberapa fungsi aktivasi yang bisa dipilih yaitu TanH, Sigmoid, Bipolar Sigmoid, dan Linear.

```
private void createNetwork(){
int activation =2;
network = new FeedforwardNetwork();
switch(activation) {
case 1:
network.addLayer(new FeedforwardLayer(new
ActivationTANH(),Integer.valueOf(tfInputNeuron.getText())));
network.addLayer(new FeedforwardLayer(new
ActivationTANH(),Integer.valueOf(tfHiddenNeuron.getText())));
if (Integer.valueOf(tfHHL2.getText())!=0)
network.addLayer(new FeedforwardLayer(new
ActivationTANH(),Integer.valueOf(tfHHL2.getText())));
network.addLayer(new FeedforwardLayer(new
ActivationTANH(),5));
break;
}
```

```

case 2:
network.addLayer(new FeedforwardLayer(new
ActivationSigmoid(),Integer.valueOf(tfInputNeuron.getText()))
);
network.addLayer(new FeedforwardLayer(new
ActivationSigmoid(),Integer.valueOf(tfHiddenNeuron.getText()))
);
if (Integer.valueOf(tfHL2.getText())!=0)
network.addLayer(new FeedforwardLayer(new
ActivationSigmoid(),Integer.valueOf(tfHL2.getText())));
network.addLayer(new FeedforwardLayer(new
ActivationSigmoid(),5));
break;
case 3:
network.addLayer(new FeedforwardLayer(new
ActivationBipolarSigmoid(),Integer.valueOf(tfInputNeuron.getT
ext())));
network.addLayer(new FeedforwardLayer(new
ActivationBipolarSigmoid(),Integer.valueOf(tfHiddenNeuron.get
Text())));
if (Integer.valueOf(tfHL2.getText())!=0)
network.addLayer(new FeedforwardLayer(new
ActivationBipolarSigmoid(),Integer.valueOf(tfHL2.getText()))
);
network.addLayer(new FeedforwardLayer(new
ActivationBipolarSigmoid(),5));
break;
case 4:
network.addLayer(new FeedforwardLayer(new
ActivationLinear(),Integer.valueOf(tfInputNeuron.getText()))
);
network.addLayer(new FeedforwardLayer(new
ActivationLinear(),Integer.valueOf(tfHiddenNeuron.getText()))
);
if (Integer.valueOf(tfHL2.getText())!=0)
network.addLayer(new FeedforwardLayer(new
ActivationLinear(),Integer.valueOf(tfHL2.getText())));
network.addLayer(new FeedforwardLayer(new
ActivationLinear(),5));
break;
}
network.reset();

```

Selanjutnya adalah list program untuk membaca input parameter-parameter seperti Epoch, Learning Rate, Momentum, dan Maximal Error.

```
MaxEpoch = Integer.valueOf(tfEpoch.getText());
LRate = Double.parseDouble(tfLearnRate.getText());
Momentum = Double.parseDouble(tfMomentum.getText());
MaxErr = Double.parseDouble
(tfError.getText());
tfOutputNeuron.setText(String.valueOf(5));
}
```

C. Metode Untuk Melakukan Pelatihan Jaringan Syaraf Tiruan

Berikut ini adalah list program untuk melakuakn proses pelatihan sesuai dengan parameter yang telah dimasukkan. Jaringan dilatih secara maju.

```
public void train() {
//-----set the inputs-----
    inputSynapse.setInputArray(inputArray);
    inputSynapse.setAdvancedColumnSelector("1,2");
//-----set the desired outputs-----
    desiredOutputSynapse.setInputArray(desiredOutputArray);
    desiredOutputSynapse.setAdvancedColumnSelector("1");
//-----get the monitor object to train or feed forward-----
    Monitor monitor = nnet.getMonitor();
//-----set the monitor parameters-----
    monitor.setLearningRate(0.8);
    monitor.setMomentum(0.3);
    monitor.setTrainingPatterns(inputArray.length);
    monitor.setTotCicles(5000);
    monitor.setLearning(true);
    long initms = System.currentTimeMillis();
// Run the network in single-thread, synchronized mode
    nnet.getMonitor().setSingleThreadMode(singleThreadMode);
    // nnet.getMonitor().setSingleThreadMode(false);

    nnet.go(true);
    System.out.println("Total time=
" +(System.currentTimeMillis() - initms)+ " ms");
    //taMessages.setText(taMessages.getText()+"\n"+"Total time=
" +(System.currentTimeMillis() - initms)+ " ms");

}
```

D. Metode Untuk Melakukan Test Pada Jaringan Syaraf Tiruan

Selanjutnya adalah proses pengetesan data dengan jaringan yang telah dihasilkan pada proses pelatihan. Nilai yang paling besar adalah nilai yang dianggap sesuai dengan data yang diuji.

```
private void bTestActionPerformed(java.awt.event.ActionEvent
evt) {
// TODO add your handling code here:
taMessages.setText("");
if (network==null)
JOptionPane.showMessageDialog(null,"Network not created
yet! Load or Create Network
first.", "Error",JOptionPane.ERROR_MESSAGE);
else {
if (wavfiles==null) {
Filter nf = new Filter("wav");
// current directory
File dir = new File(".");
wavfiles = dir.list(nf);
}
System.out.println("Neural Network Results:");
int N =network.getInputLayer().getNeuronCount();
double input[] = new double[N];
for (int i=0;i<N;i++){
input[i]=yinputtest[0][i];
}
double actual[] = network.computeOutputs(input);
int idxmax = getIdxMax(actual);
//String filewav = wavfiles[idxmax];
String filewav = null;
switch(idxmax){
case 0: filewav="huruf a"; break;
case 1: filewav="huruf e"; break;
case 2: filewav="huruf i"; break;
case 3: filewav="huruf o"; break;
case 4: filewav="huruf u"; break;
}
}
```

```

int M = actual.length;
taMessages.setText("Hasil Output: ");
for (int i=0;i<M;i++){
if (i==0)
//System.out.print(""+actual[i]);
taMessages.setText(taMessages.getText()+" "+actual[i]);
else
//System.out.print(", "+actual[i]);
taMessages.setText(taMessages.getText()+", "+actual[i]);
}
taMessages.setText(taMessages.getText()+"\n Suara yang
dicari: "+filewav);
}
}

```

4.3 Kinerja Perangkat Lunak

4.3.1 Kinerja Aplikasi Pengenalan Ucapan

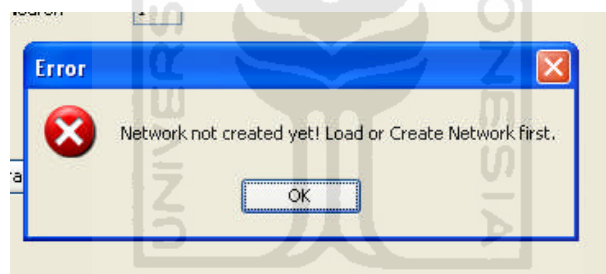
Pada tahap analisis kinerja perangkat lunak dijelaskan tentang pengujian aplikasi pengenalan ucapan. Pengujian dilakukan secara menyeluruh untuk mengetahui kinerja sistem dan diharapkan dapat diketahui kekurangan-kekurangan dari sistem untuk kemudian diperbaiki sehingga kesalahan dari sistem dapat diminimalisasi atau bahkan dihilangkan. Pengujian sistem ini dilakukan untuk mendapatkan hasil perhitungan yang cepat, tepat dan akurat, sesuai dengan perhitungan secara manual.

Beberapa pengujian memiliki dua kemungkinan, yaitu prosedur normal (benar) dan prosedur tidak normal (salah). Pengujian pada prosedur tidak normal digunakan untuk melihat apakah aplikasi mampu menangani kesalahan dengan baik serta mampu mengkomunikasikannya kepada pemakai aplikasi untuk kemudian memberikan saran atau solusi kepada pemakai aplikasi.

A. Pengujian tidak normal

Dilakukan dengan memberikan masukan dengan spesifikasi yang tidak diijinkan sehingga sistem akan memberikan reaksi lain. Reaksi system berupa peringatan (*alert*) atau penanganan kesalahan (*error handling*). Penanganan kesalahan ini dilakukan untuk menangkap error yang terjadi ketika salah dalam proses atau urutan penggunaan program.

Penanganan kesalahan dilakukan untuk menangkap error yang terjadi ketika salah dalam melakukan urutan langkah jalannya program. Contoh penanganan kesalahan pada saat user mengklik button train, sementara belum mengklik button prepare dan create network ditunjukkan pada gambar 4.3.



Gambar 4.3 Kesalahan urutan jalannya program

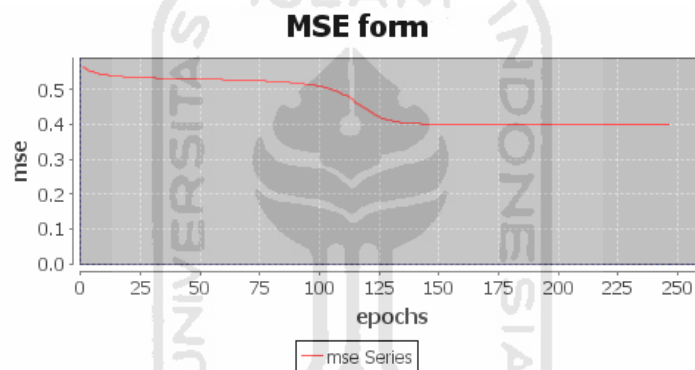
B. Pengujian Normal

Pada tahap pengujian analisis program ini, dilakukan perbandingan antara kebenaran masukan serta kesesuaian program dengan kebutuhan sistem. Terdapat dua tahapan pada pengujian normal, yang pertama adalah tahap pelatihan dan yang kedua adalah tahap pengujian. Data pelatihan adalah sinyal suara dari 10 orang pembicara yang masing-masing mengucapkan suku kata 'a', 'i', 'u', 'e', 'o' sebanyak dua kali. Sehingga didapat 100 buah data pelatihan.

Pada pelatihan jaringan saraf tiruan propagasi balik untuk pengenalan suara manusia dilakukan beberapa pengamatan yaitu pengamatan pada pengaruh fungsi aktivasi yang digunakan, nilai *learning rate*, *epoch* serta *momentum* yang digunakan. Proses pengamatan yang dilakukan adalah sebagai berikut :

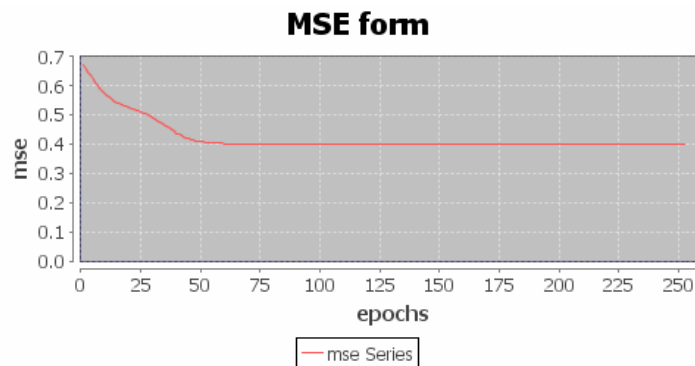
B.1. Pelatihan dengan Perubahan Fungsi Aktivasi Sigmoid Biner, Momentum 0

- a. Pelatihan dengan jumlah input neuron 16, 1 neuron pada hidden layer pertama, dan 1 neuron pada hidden layer kedua, learning rate 0.001. Lihat gambar 4.4



Gambar 4.4 Grafik MSE dengan learning rate 0.001

- b. Pelatihan dengan jumlah input neuron 16, 1 neuron pada hidden layer pertama, dan 1 neuron pada hidden layer kedua, learning rate 0.5. Lihat gambar 4.5

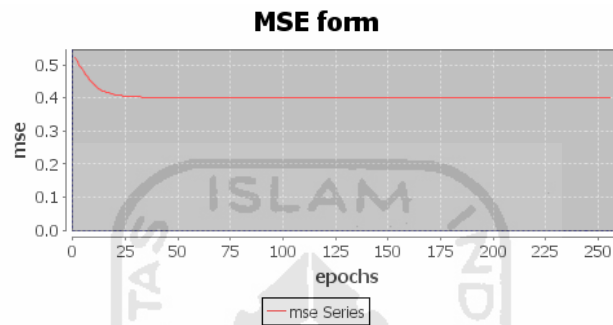


Gambar 4.5 Grafik MSE dengan learning rate 0.5

B.2 Pelatihan dengan Perubahan Fungsi Aktivasi Sigmoid Biner, Momentum

0.5

Pelatihan dengan jumlah input neuron 16, 1 neuron pada hidden layer pertama, dan 1 neuron pada hidden layer kedua, learning rate 0.3. Lihat gambar 4.6

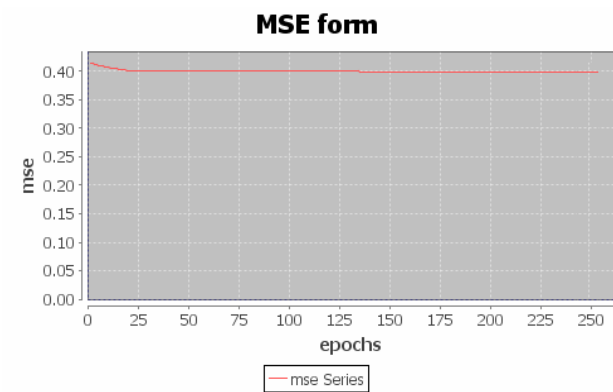


Gambar 4.6 Grafik Grafik MSE dengan learning rate 0.3

B.3 Pelatihan dengan Perubahan Fungsi Aktivasi Sigmoid Biner, Momentum

0.7

Pelatihan dengan jumlah input neuron 16, 1 neuron pada hidden layer pertama, dan 1 neuron pada hidden layer kedua, learning rate 0.1. Lihat gambar 4.7

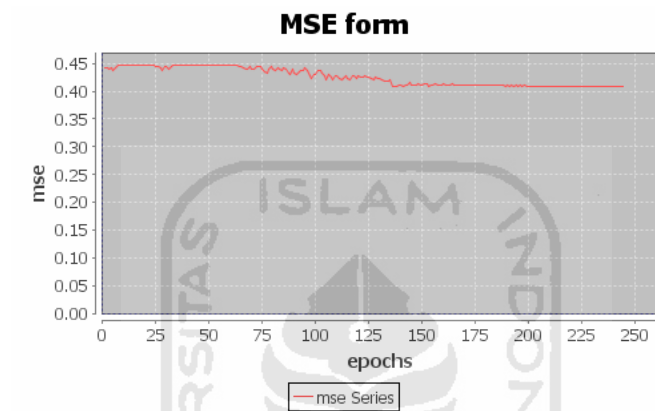


Gambar 4.7 Grafik MSE dengan learning rate 0.1 dan 1 neuron

B.4 Pelatihan dengan Perubahan Fungsi Aktivasi Sigmoid Biner, Momentum

0.9

Pelatihan dengan jumlah input neuron 16, 16 neuron pada hidden layer pertama, dan 16 neuron pada hidden layer kedua, learning rate 0.1. Lihat gambar 4.8.



Gambar 4.8 Grafik MSE dengan learning rate 0.1 dan 16 neuron

Pada tabel 4.1 ditunjukkan nilai error atau kesalahan pada tiap parameter yang dilakukan variasi.

Tabel 4.1 Error pada masing-masing nilai

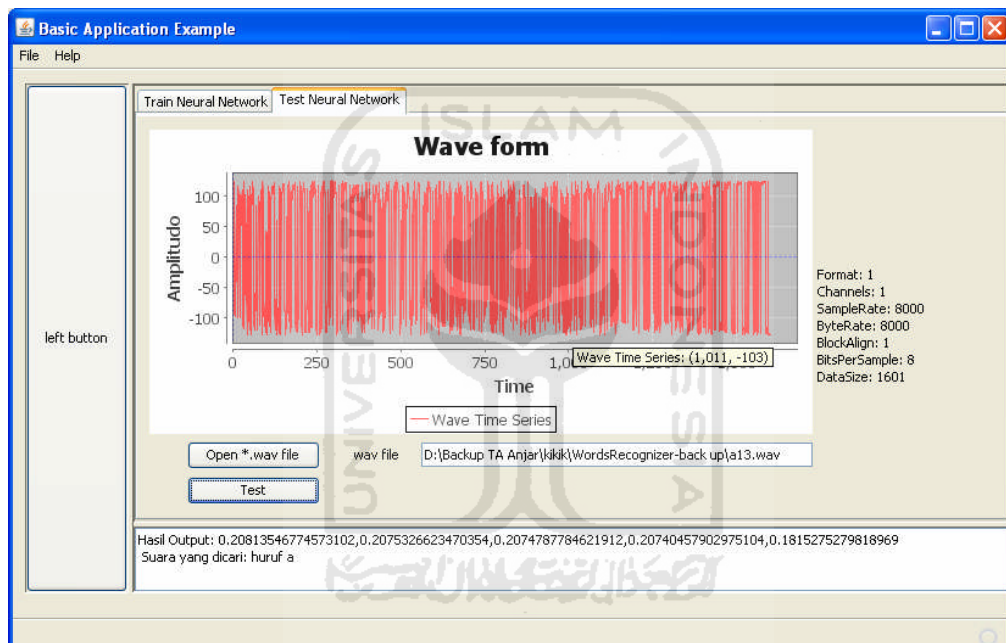
LR	Epoch	Momentum	Input Neuron	HL1	HL2	Mean Square Error
0.001	5000	0	16	1	1	0.40000092737668475
0.01	5000	0	16	1	1	0.39814410052224347
0.1	5000	0	16	1	1	0.39806950751787806
0.3	5000	0	16	1	1	0.3979803770212518
0.5	5000	0	16	1	1	0.39797750483169897
0.9	5000	0	16	1	1	0.4027027111228313
0.01	5000	0.5	16	1	1	0.40000031392275753
0.1	5000	0.5	16	1	1	0.398051967615864
0.3	5000	0.5	16	1	1	0.39798045372752244
0.5	5000	0.5	16	1	1	0.4000001554936066
0.9	5000	0.5	16	1	1	0.43817773213629374
0.01	5000	0.7	16	1	1	0.3999981162918726
0.1	5000	0.7	16	1	1	0.3979796046325251
0.3	5000	0.7	16	1	1	0.39999975161272977
0.5	5000	0.7	16	1	1	0.4380860320636059
0.9	5000	0.7	16	1	1	0.428763451026923
0.1	5000	0.9	16	1	1	0.41923515313020093
0.1	5000	0.9	16	16	16	0.39797612900477913
0.1	5000	0.9	32	32	32	0.44721359549712036

Keterangan Tabel :

1. LR : Learning rate
2. HL1 : Jumlah neuron pada hidden layer 1
3. HL 2 : Jumlah neuron pada hidden layer 2

4.3.2 Hasil Pengujian

Proses pengujian pada sistem ini yaitu pengujian dengan data. Pola kata yang dikenali diwakili oleh masing-masing pola angka dalam bentuk biner untuk setiap kata yang dikenali. Kata yang dikenali dapat dilihat langsung pada GUI berupa tulisan dari kata yang diucapkan. Kata yang akan dikenali yaitu kata seperti kata-kata yang telah dilatihkan. Lihat gambar 4.9



Gambar 4.9 Menu pengujian

Hasil pengujian jaringan saraf tiruan propagasi balik pada sistem pengenalan suara manusia terhadap data pelatihan

- a. Momentum 0, learning rate 0.001, input neuron 16, 1 neuron pada hidden layer pertama, dan 1 neuron pada hidden layer kedua. Lihat tabel 4.2.

Tabel 4.2 Nilai error dengan learning rate 0.001

Kata	Error
A	100%
I	100%
U	60%
E	100%
O	45%

- b. Momentum 0, learning rate 0.5, input neuron 16, 1 neuron pada hidden layer pertama, dan 1 neuron pada hidden layer kedua. Lihat tabel 4.3

Tabel 4.3 Nilai error dengan learning rate 0.5

Kata	Error
A	75%
I	100%
U	5%
E	100%
O	95%

- c. Momentum 0.5, learning rate 0.3, input neuron 16, 1 neuron pada hidden layer pertama, dan 1 neuron pada hidden layer kedua. Lihat tabel 4.4

Tabel 4.4 Nilai error dengan learning rate 0.3

Kata	Error
A	50%
I	100%
U	10%
E	100%
O	100%

- d. Momentum 0.7, learning rate 0.1, input neuron 16, 1 neuron pada hidden layer pertama, dan 1 neuron pada hidden layer kedua. Lihat tabel 4.5

Tabel 4.5 Nilai error dengan learning rate 0.1 dan 1 neuron

Kata	Error
A	100%
I	100%
U	40%
E	45%
O	100%

- e. Momentum 0.9, learning rate 0.1, input neuron 16, 16 neuron pada hidden layer pertama, dan 16 neuron pada hidden layer kedua. Lihat tabel 4.6

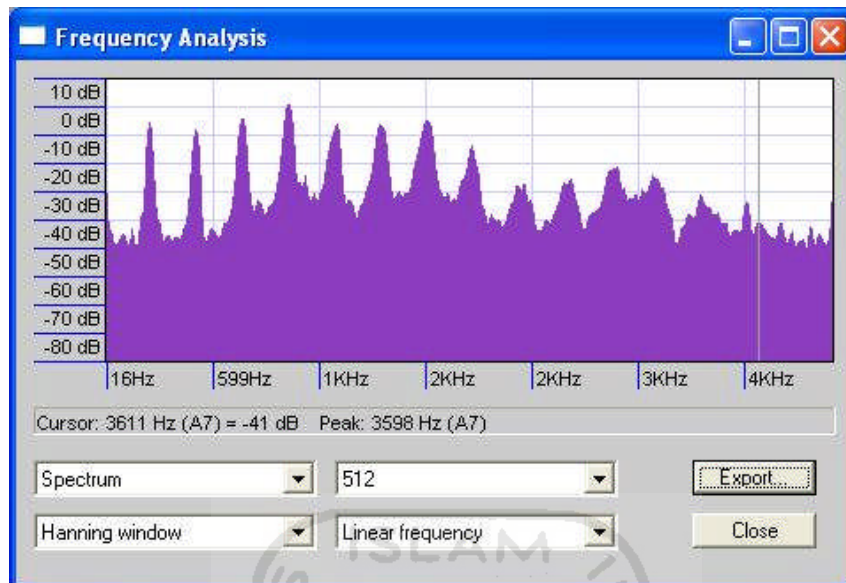
Tabel 4.6 Nilai error dengan learning rate 0.1 dan 16 neuron

Kata	Error
A	100%
I	100%
U	5%
E	100%
O	75%

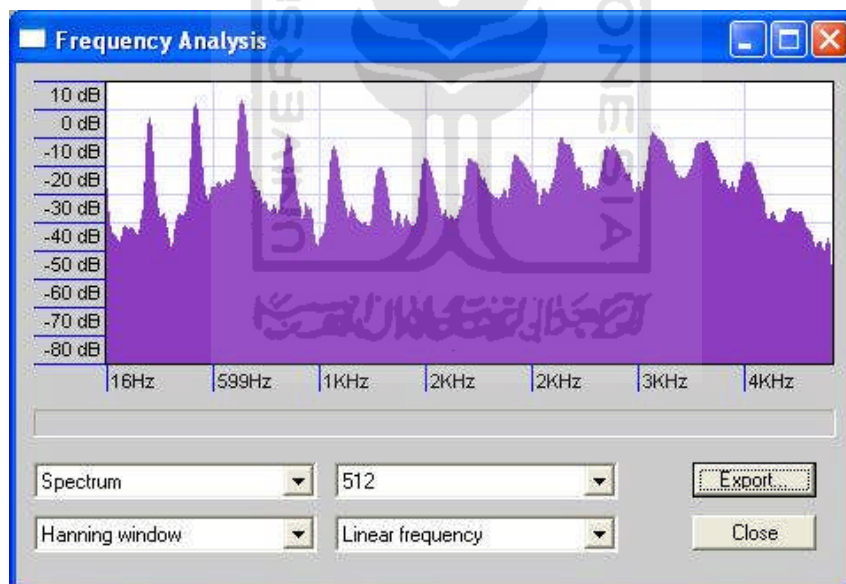
4.3.3 Analisis Hasil Pengujian

Pada hasil pengujian dengan variasi nilai parameter learning rate, momentum, dan jumlah neuron pada setiap hidden layer tidak mendapatkan perubahan yang signifikan. Secara garis besar pelatihan masih memiliki nilai error yang tinggi, yaitu sebesar 60%. Hal ini bisa disebabkan oleh beberapa hal seperti jaringan syaraf tiruan yang belum benar atau sebab yang lainnya.

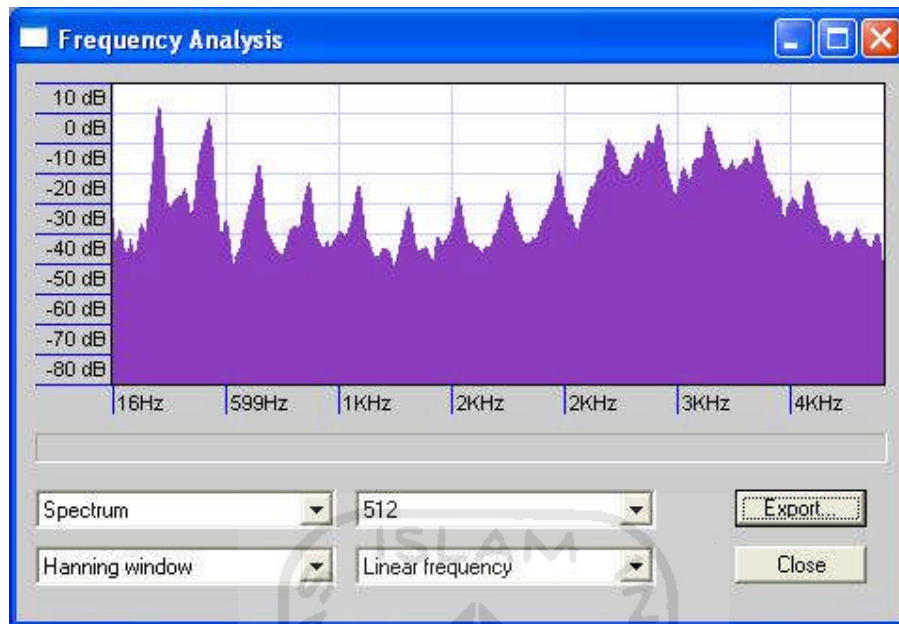
Disini digunakan aplikasi Audacity untuk menganalisa spektrum frekuensi dari salah satu dari sample vokal A, E, I, O, dan U. Lihat Gambar 4.10 sampai dengan 4.14. Terlihat bahwa hasil band pass filter tidak berfungsi dengan baik, hal ini ditunjukkan masih besarnya nilai pada range frekuensi dibawah 300Hz dan diatas 3.400Hz.



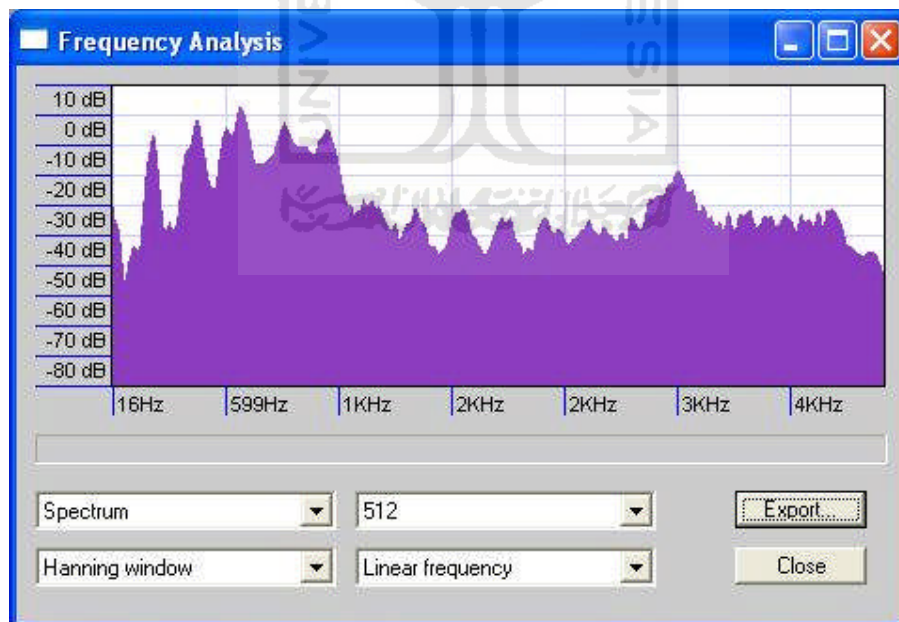
Gambar 4.10 Spektrum frekuensi sample 1 huruf A



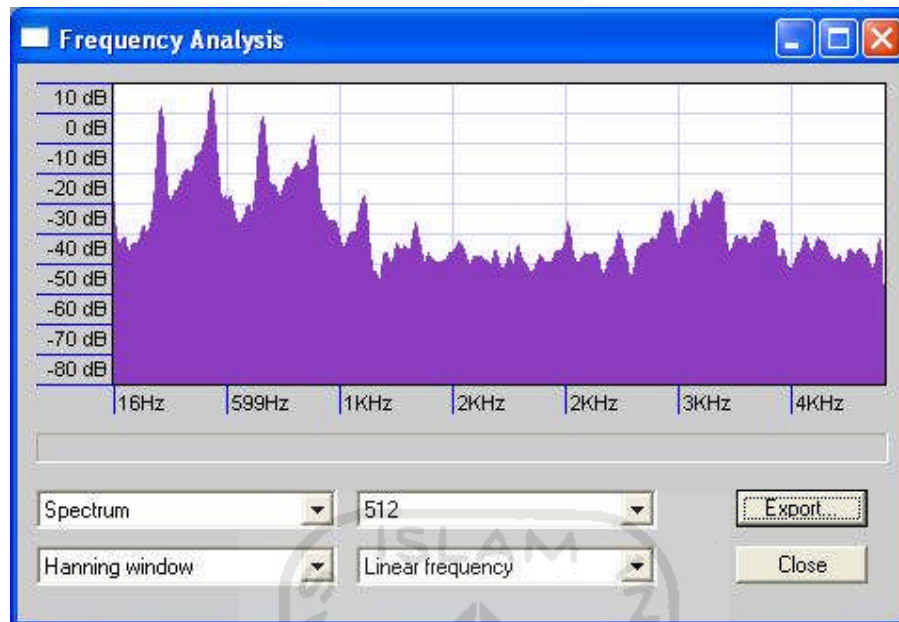
Gambar 4.11 Spektrum frekuensi sample 1 huruf E



Gambar 4.12 Spektrum frekuensi sample 1 huruf I



Gambar 4.13 Spektrum frekuensi sample 1 huruf O



Gambar 4.14 Spektrum frekuensi sample 1 huruf U

4.3.4 Kelebihan dan Kekurangan Sistem

Jaringan syaraf tiruan dengan propagasi balik akan memiliki kinerja yang semakin baik dengan banyaknya nilai sample yang dimasukkan karena berarti nilai pembelajaran yang diberikan semakin tinggi.

Tingginya nilai error/kesalahan sistem ada beberapa kendala yaitu pada sistem perekaman suara yang belum bisa mendapatkan hasil yang maksimal. Suara manusia seharusnya hanya berkisar antara 200 Hz sampai dengan 8000 Hz [GRO09]. Namun pada hasil yang didapat masih tingginya nilai magnitude pada frekuensi lainnya. Hal ini mungkin bisa ditingkatkan dengan cara merekam suara informasi menggunakan microphone jenis analog bukan dengan tipe mic condenser yang digunakan pada saat ini. Langkah lainnya yang bisa ditempuh adalah dengan melakukan pra-proses terhadap data yang akan dijadikan informasi

dengan aplikasi lainnya yang memadai. Contohnya adalah dengan melakukan proses filter ketat (band pass filter) terhadap data yang dimaksud.



BAB V SIMPULAN DAN SARAN

5.1 Simpulan

Dari hasil pengamatan terhadap hasil aplikasi pengenalan ucapan dengan menggunakan jaringan saraf tiruan propagasi balik berbasis Java dapat disimpulkan sebagai berikut :

1. Struktur jaringan saraf tiruan propagasi balik (*backpropagation*) yang terbaik untuk sistem pengenalan suara manusia yaitu dengan menggunakan 2 lapisan tersembunyi dengan jumlah neuron pada masing-masing lapisan adalah 16 neuron pada input layer dan 1 neuron pada hidden layer I dan hidden layer II. Nilai *learning rate* yang digunakan 0,3 dan *momentum* 0,5 serta fungsi aktivasi yang digunakan yaitu *sigmoid binner*.
2. Penambahan jumlah neuron pada input layer maupun hidden layer tidak menghasilkan penurunan jumlah iterasi (*epoch*) yang berarti,
3. *Error* rata-rata hasil pengujian sistem pengenalan suara manusia terhadap data pelatihan yaitu 60%.
4. Tingginya nilai kesalahan namun penulis belum mengetahui secara pasti letak kesalahan. Kesalahan bisa pada arsitektur jaringan syaraf tiruan yang belum benar atau pada data yang diuji.

5.2 Saran

1. Perlunya *preprocessing* sinyal menggunakan metode LPC.

2. Perlu dilakukan pengujian frekuensi cuplik yang lebih dari 8000 Hz untuk mengetahui frekuensi cuplik yang lebih baik.
3. Kondisi lingkungan yang bersih dari derau (noise) saat pengambilan data.
4. Perlu adanya tambahan data pelatihan dengan berbagai jenis karakter suara atau pengucapan dari penutur.
5. Perlu peningkatan mekanisme perekaman suara.



DAFTAR PUSTAKA

- [MUL95] **Müller, Berndt dan Joachim Reinhardt, Michael Thomas Strickland.** *Neural networks: an introduction, Volume 1.* **1995**
Dari <http://books.google.com/books?id=EFUzMYjOXk8C>, diakses 03 Januari 2011
- [ARB03] **Arbib, Michael A.** *The handbook of brain theory and neural networks.* **2003**
Dari <http://books.google.com/books?id=Av6qWhw0-EC>, diakses 03 Januari 2011
- [MUN08] **Munakata, Toshinori.** *Fundamentals of the new artificial intelligence: neural, evolutionary, fuzzy and more.* **2008**
Dari <http://books.google.com/books?id=lei-Zt8UGSQC>, diakses 03 Januari 2011
- [GRO09] **Gross, Tom dan Jan Gulliksen, Paula Kotzé, Lars Oestreicher, Philippe Palanque, Raquel Oliveira Prates, Marco Winckler.** *Human-Computer Interaction - INTERACT 2009: 12th IFIP TC 13 International Conference, Uppsala, Sweden, August 24-28, 2009, Proceedigns, Part 1.* **2009**
Dari <http://books.google.com/books?id=xXXDoORrU9IC>, diakses 03 Januari 2011