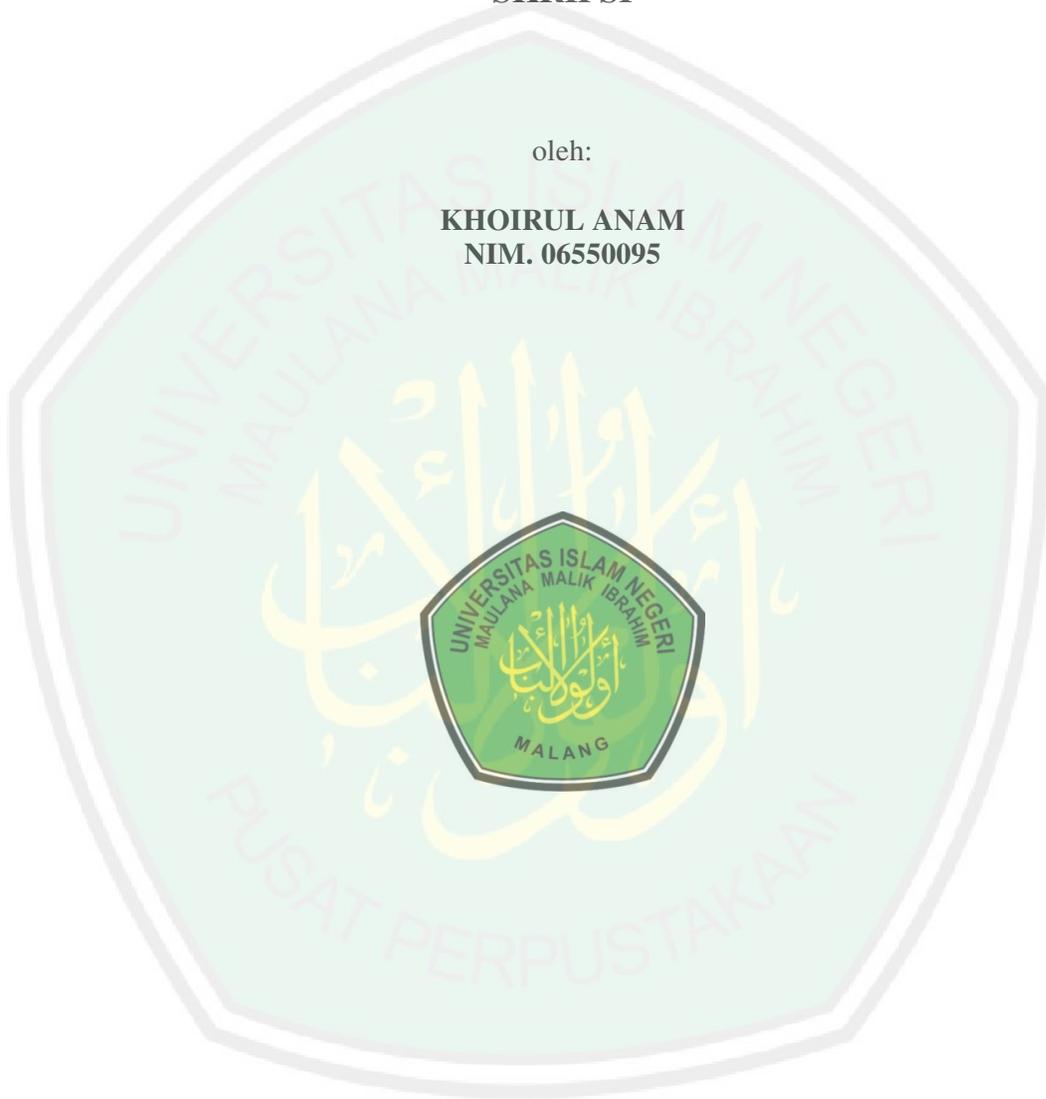


**PENGENALAN SUARA MANUSIA MENGGUNAKAN
METODE *LINIER PREDICTIVE CODING* (LPC)**

SKRIPSI

oleh:

**KHOIRUL ANAM
NIM. 06550095**



**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2013**

**PENGENALAN SUARA MANUSIA MENGGUNAKAN
METODE *LINIER PREDICTIVE CODING* (LPC)**

SKRIPSI

Diajukan Kepada:

**Fakultas Sains dan Teknologi
Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang
Untuk Memenuhi Salah Satu Persyaratan Dalam
Memperoleh Gelar Sarjana Sains (S.Kom)**

Oleh:

**KHOIRUL ANAM
NIM. 06550095**



**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2013**

**PENGENALAN SUARA MANUSIA MENGGUNAKAN
METODE *LINIER PREDICTIVE CODING* (LPC)**

SKRIPSI

Oleh:

**KHOIRUL ANAM
NIM. 06550095**

Telah Disetujui untuk Diuji

Malang, 22 Juli 2013

Dosen Pembimbing I,

Dosen Pembimbing II,

**Totok Chamidy, M.Kom
NIP. 196912222006041001**

**Hani Nurhayati, M.T
NIP. 19780625200812006**

**Mengetahui,
Ketua Jurusan Teknik Informatika**

**Ririen Kusumawati, M.Kom
NIP. 197203092005012002**

LEMBAR PENGESAHAN**PENGENALAN SUARA MANUSIA MENGGUNAKAN
METODE *LINIER PREDICTIVE CODING* (LPC)****SKRIPSI**

Oleh:

**KHOIRUL ANAM
NIM. 06550095**

Telah Dipertahankan di Depan Dewan Penguji Skripsi dan
Dinyatakan Diterima Sebagai Salah Satu Persyaratan
Untuk Memperoleh Gelar Sarjana Sains (S.Kom)

Tanggal 22 Juli 2013

Susunan Dewan Penguji:		Tanda Tangan
1. Penguji Utama:	<u>M. Ainul Yaqin, M.Kom</u> NIP. 197610132006041004	()
2. Ketua:	<u>Zainal Abidin, M.Kom</u> NIP. 197606132005011004	()
3. Sekretaris:	<u>Totok Chamidy, M.Kom</u> NIP. 196912222006041001	()
4. Anggota:	<u>Hani Nurhayati, M.T</u> NIP. 19780625200812006	()

**Mengetahui dan Mengesahkan,
Ketua Jurusan Teknik Informatika**

Ririen Kusumawati, M.Kom
NIP. 197203092005012002

SURAT PERNYATAAN

Yang bertanda tangan di bawah ini :

Nama : KHOIRUL ANAM
NIM : 06550095
Fakultas / Jurusan : Sains Dan Teknologi / Teknik Informatika
Judul Penelitian : Pengenalan Suara Manusia Menggunakan Metode *LINIER PREDICTIVE CODING* (LPC)

Menyatakan dengan sebenar-benarnya bahwa hasil penelitian saya ini tidak terdapat unsur-unsur penjiplakan karya penelitian atau karya ilmiah yang pernah dilakukan atau dibuat oleh orang lain, kecuali yang secara tertulis dikutip dalam naskah ini dan disebutkan dalam sumber kutipan dan daftar pustaka.

Apabila ternyata hasil penelitian ini terbukti terdapat unsur-unsur jiplakan, maka saya bersedia untuk mempertanggung jawabkan, serta diproses sesuai peraturan yang berlaku.

Malang,
Yang Menyatakan,

Khoirul Anam
NIM. 06550095

Motto

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

*"Islam Harus Dipahami Dalam Konteks
Kehidupan Yang Luas Dan Tidak Terbatas
Hanya Menyangkut Persoalan Ritual Dan
Spiritual."*

(Drs. H. Suryadharna Ali, M.Si)

LEMBAR PERSEMBAHAN

Alhamdulillah... puji syukur kehadiran Allah SWT yang selalu memberi kenikmatan serta kekuatan sehingga bisa terselesaikan skripsi ini.

Sholawat beriring salam selalu tercurah kepada manusia pembawa risalah, manusia yang selalu mengajarkan kepada umat manusia betapa indahny iman dan Islam. Nabi agung Muhammad SAW...

Untuk kedua Orang tua, Trimakasih Tuhan telah Kau beri mereka untukku. Mereka yang telah membangkitkan semangat ketika terpuruk, mengajarkan tanpa imbalan, dan menyayangi tanpa akhiran...

Untuk sepasang malaikatku, mereka yang dalam sujudnya berdoa untukku, mereka yang begitu istimewa dalam hidupku, Trimakasih Istri dan buah hati tercinta Jasmine Azka Al-adristi...

Untuk semua keluargaku yang ada di Pandaan, Malang, dan Bali terimakasih atas support dan senyumnya di setiap waktu, di setiap kesedihan dan di setiap kerisauan...

Untuk (Alm) kakak tercinta, biarkanlah namamu selalu mengabadi sehingga aku paham betapa istimewahnya persaudaraan kita, selalu tersenyum meskipun hanya kenangan, dibawah batu nisan itu kau tlah sandarkan kasih sayangmu yang begitu dalam...

Dan trimakasih untuk teman-teman IT UIN Malang edisi "zero six". Teman-teman cyber eXc|CooL, dan lain lain....

-=EVERIDAY IS RACE=-

--THE LAST BUT NOT THE LIST--

Setiap Hari Langkah Kehidupan Begitu Cepat, Bagaikan Pembalap Berebut Dan Melaju Menjadi Yang Nomer Satu, Tetapi Yang Terakhir Bukanlah Yang Terburuk.

KATA PENGANTAR

Assalamu'alaikum Wr. Wb.

Syukur alhamdulillah penulis hanturkan kehadiran Allah SWT yang telah melimpahkan Rahmat dan Hidayah-Nya, sehingga penulis dapat menyelesaikan studi di Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang sekaligus menyelesaikan skripsi dengan judul “Pengenalan Suara Manusia Menggunakan Metode *Linier Predictive Coding* (LPC)”.

Sholawat dan salam semoga selalu tercurah kepada Nabi Muhammad SAW karena bimbingan beliau, Islam telah disampaikan dan membawa umat dari kegelapan menuju zaman yang terang benderang.

Selanjutnya penulis haturkan ucapan terima kasih seiring do'a dan harapan *jazakumullah ahsanal jaza'* kepada semua pihak yang telah membantu terselesaikannya skripsi ini. Dalam kesempatan ini penulis ingin mengucapkan terima kasih yang tak terhingga khususnya kepada:

1. Prof. Dr. H. Mudjia Rahardjo, M.Si. selaku Rektor UIN Maulana Malik Ibrahim Malang yang telah banyak memberikan pengetahuan dan pengalaman yang berharga.
2. Dr. drh. Bayyinatul Muchtaromah, M.Si. selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
3. Bapak Totok Chamidy, M.Kom selaku pembimbing I yang telah meluangkan banyak waktu untuk membimbing dan mengarahkan dalam penyelesaian skripsi.
4. Ibu Hani Nurhayati, M.T. selaku pembimbing II yang telah memberikan pengetahuan tentang perspektif Islam.

5. Seluruh Dosen Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang yang telah mengajar penulis selama empat tahun lamanya, dan memberikan dukungan untuk menyelesaikan penulisan skripsi ini.
6. Kedua orangtua dan seluruh keluarga besar yang selalu mendoakan dan mendukung setiap langkah penulis.
7. Semua pihak yang tidak dapat penulis sebutkan satu persatu, yang telah banyak membantu dalam penyelesaian skripsi ini.

Penulis menyadari bahwa dalam penyusunan skripsi ini masih terdapat kekurangan dan penulis berharap semoga skripsi ini bisa memberikan manfaat kepada para pembaca khususnya bagi penulis secara pribadi. *Amin Ya Rabbal Alamin.*

Wassalamu'alaikum Wr. Wb.

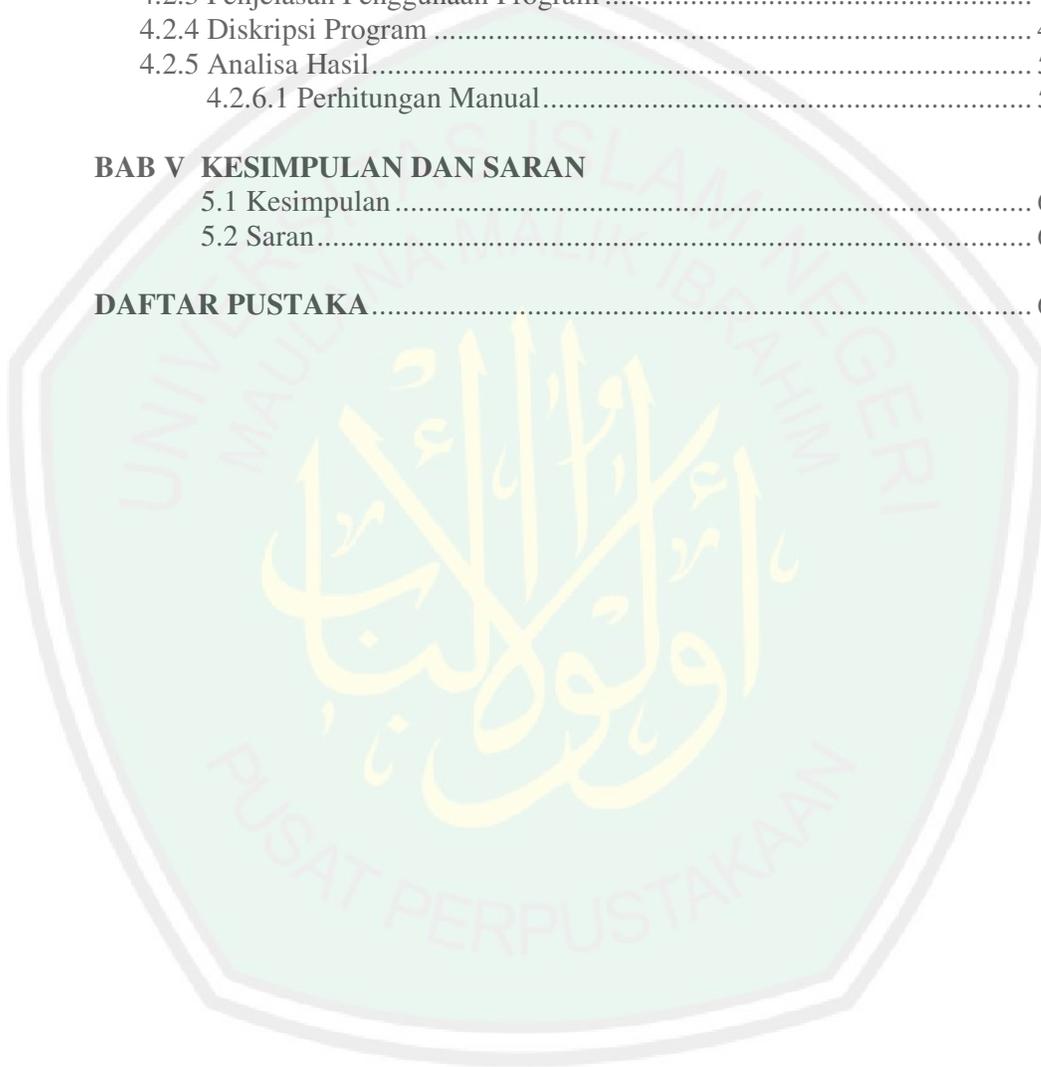
Malang, 22 Juli 2013

Penulis

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGAJUAN	ii
HALAMAN PERSETUJUAN	iii
HALAMAN PENGESAHAN	iv
SURAT PERNYATAAN	v
MOTTO	vii
LEMBAR PERSEMBAHAN	vii
KATA PENGANTAR	viii
DAFTAR ISI	x
DAFTAR TABEL	xii
DAFTAR GAMBAR	xiii
ABSTRAK	xiv
BAB I PENDAHULUAN	
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	5
1.3 Batasan Masalah.....	5
1.4 Tujuan Penelitian.....	5
1.5 Manfaat Penelitian.....	5
1.6 Metode Penelitian.....	6
1.7 Sistematika Penulisan.....	7
BAB II LANDASAN TEORI	
2.1 Pengenalan Suara (<i>Voice recognition</i>).....	8
2.2 Sistem Pengenalan Suara.....	9
2.2.1 Normalisasi.....	10
2.2.2 <i>Linier Predictive Coding</i> (LPC).....	14
2.2.3 <i>Fast Fourier Transform</i> (FFT).....	14
2.3 Pemrosesan Sinyal Digital.....	15
2.4 Skema Utama Dan Algoritma <i>Voice recognition</i>	26
2.5 Java.....	19
2.3.1 Karakteristik Java.....	19
2.6 Pengenalan Suara Pada AL-Qur'an.....	22
BAB III ANALISIS DAN PERANCANGAN SISTEM	
3.1 Analisa Permasalahan Sistem.....	25
3.2 Perancangan Sistem.....	30
3.2.1 Diskripsi Umum Sistem.....	30
3.2.2 Analisa <i>Use Case Diagram</i>	31
3.2.3 Analisa <i>Sequence Diagram</i>	33
3.2.3.1 <i>Sequence Diagram Login</i> Suara.....	33
3.2.4 Diagram <i>Activity</i>	37
3.2.5 Rancangan Antar Muka.....	39

BAB IV HASIL DAN PEMBAHASAN	
4.1 Analisa Dan Implementasi	44
4.2 Intalasi Program.....	44
4.2.1 Kebutuhan Perangkat Keras	44
4.2.2 Kebutuhan Perangkat Keras	44
4.2.3 Penjelasan Penggunaan Program	45
4.2.4 Diskripsi Program	49
4.2.5 Analisa Hasil.....	56
4.2.6.1 Perhitungan Manual.....	59
BAB V KESIMPULAN DAN SARAN	
5.1 Kesimpulan	60
5.2 Saran.....	61
DAFTAR PUSTAKA.....	62



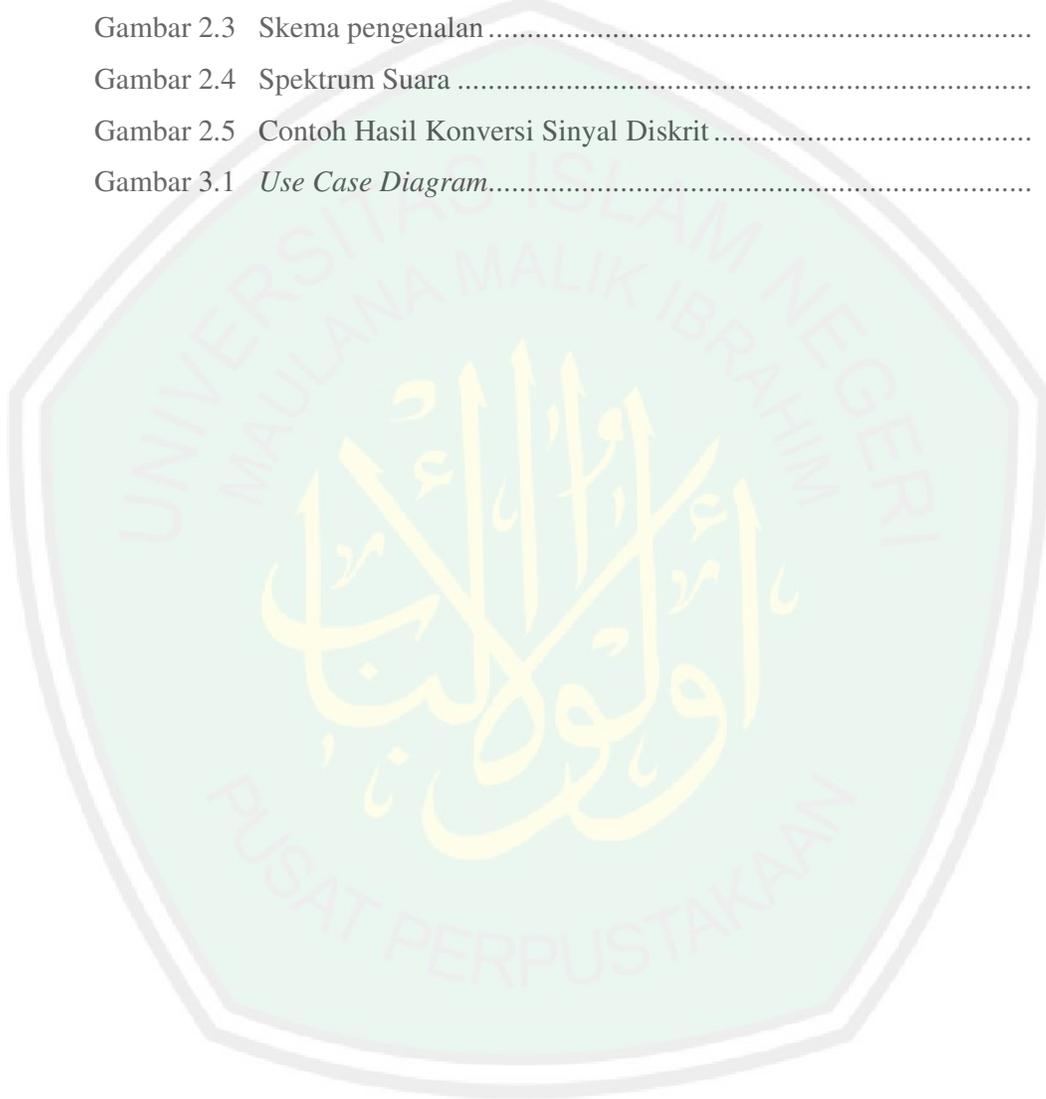
DAFTAR TABEL

Tabel 4.1 Hasil Pengujian untuk <i>Login Suara</i>	58
Tabel 4.2 Hasil Pengujian untuk <i>Voice to command</i>	59



DAFTAR GAMBAR

Gambar 2.1	Blok diagram sistem pengenalan suara	9
Gambar 2.2	Posisi <i>Filter Pre-Emphasis</i> pada sistem pengolah suara	13
Gambar 2.3	Skema pengenalan	16
Gambar 2.4	Spektrum Suara	17
Gambar 2.5	Contoh Hasil Konversi Sinyal Diskrit	18
Gambar 3.1	<i>Use Case Diagram</i>	31



ABSTRAK

Anam, Khoirul.2013. 06550095. **Pengenalan Suara Manusia Menggunakan Metode *Linier Predictive Coding (LPC)***. Skripsi, Jurusan Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang.

Pembimbing : (I) Totok Chamidy, M.Kom (II) Hani Nurhayati, M.T

Kata Kunci : *Linier Predictive Coding, Fast Fourier Transform, Compare*

Pengenalan suara adalah teknologi masa depan yang menggantikan cara interaksi manusia dengan komputer dengan menggabungkan beberapa disiplin ilmu seperti pengenalan sinyal dan pengenalan pola. Dimana interaksi user dengan sistem dapat dilakukan dengan memberikan inputan suara.

Linier Predictive Coding (LPC) adalah salah satu metode parametrik yang digunakan untuk merepresentasikan sinyal. Umumnya LPC digunakan karena menyediakan pemodelan yang baik untuk sinyal suara, LPC dapat dengan mudah dan langsung diterapkan baik secara perangkat lunak maupun perangkat keras karena perhitungan matematis yang dilibatkan realtif lebih singkat dari metode-metode yang dikenal sebelumnya.

Fast Fourier transform (FFT) menjadi penting untuk bermacam – macam aplikasi, dari pengolahan sinyal digital dan memecahkan persamaan *diferensial parsial* menjadi algoritma-algoritma untuk penggandaan bilangan *integer* dalam jumlah yang banyak.

Secara garis besar, cara kerja sistem pengenalan suara ini ialah mula-mula sinyal suara manusia yang diterima dengan menggunakan *microphone* (sinyal *analog*) dicuplik sehingga menjadi sinyal *digital* dengan bantuan *sound card* pada komputer. Sinyal *digital* hasil cuplikan ini terlebih dulu dinormalisasi kemudian diproses awal menggunakan metode LPC sehingga didapat beberapa koefisien LPC yang merupakan *feature* (ciri) dari suara pembicaraan. Kemudian koefisien LPC tersebut diproses dengan *Fast Fourier Transform (FFT)* untuk mendapatkan sinyal pada *domain* frekuensi. Hal ini bertujuan agar perbedaan antar pola kata yang satu dengan yang lain terlihat lebih jelas sehingga ekstraksi parameter sinyal memberikan hasil yang lebih baik. Hasil keluaran FFT ini akan di *compare* sebagai fungsi utama dari sistem untuk proses pengenalan

ABSTRACT

Anam, Khoirul.2013. 06550095. **Human Voice Introduction with Linear Predictive Coding (LPC) Method.** Thesis, Department of Informatics, Faculty of Science and Technology of the State Islamic University of Maulana Malik Ibrahim Malang.

Supervisor: (I) Totok Chamidy, M.Kom (II) Hani Nurhayati, M.T.

Keywords: Linier Predictive Coding, Fast Fourier Transform, Compare

recognition of voice is future technology which substitute the way of human interaction with computer by combine some knowledge discipline like introduction of signal and format. where user interaction with system can be done by giving voice input.

Linear Predictive Coding (LPC) is one of parametric method which is use for signal presentation. Generally, LPC used because provide good modeling for voice signal. LPC can be easily and directly be applied either by software or hardware because mathematical calculations which involved relative shorter than the previously methods known.

Fast Fourier transform (FFT) become important for any kind application, from digital signal process and solving partial differential equations into algorithms for integer multiplication in significant amounts.

In outline, the workings of the voice recognition system is first human voice signal which receive by microphone (analog signal) sampled so that become digital signal helped by sound card in computer. sample of digital signal normalized first then processed with LPC method in order to get some of the LPC coefficients which is feature from speech voice. Then that LPC coefficients processed with Fast Fourier Transform (FFT) in order to get signal in frequency domain. it is instead that the different between each pattern of word looks more clearly so that signal parameter extraction gives better results. The result from this FFT will compare as main function from system for recognize process.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Manusia memiliki akal (otak) dan indera yang merupakan anugerah Tuhan Yang Maha Esa. Akal dan indera berfungsi sebagai alat untuk berpikir dan menerima rangsangan (kepekaan). Indera menerima rangsangan dan diteruskan ke otak (akal) guna mengenali rangsangan yang diterima. Manusia melakukan proses pengenalan dengan cara belajar sejak lahir. Seseorang bayi pertama kali belajar mengenali suara dan wajah ibunya, kemudian belajar mengenali lingkungan sekitarnya. Manusia diwajibkan untuk belajar mengartikan sendiri arti-arti dari kehidupannya, karena manusia merupakan makhluk yang sempurna dibanding makhluk lainnya.

Di dalam kitab suci Al-Qur'an, surat At-Tiin ayat 4, sebagaimana firman Allah SWT, yaitu :

لَقَدْ خَلَقْنَا الْإِنْسَانَ فِي أَحْسَنِ تَقْوِيمٍ ﴿٤﴾

Artinya: *Sesungguhnya Kami telah menciptakan manusia dalam bentuk yang sebaik-baiknya .*

Dari kesempurnaan yang telah Allah anugerahkan, manusia diwajibkan mengetahui semua yang telah diberikan Allah di muka bumi ini. Manusia diwajibkan untuk membaca semua yang ada di dunia. Seperti yang ada dalam kitab suci Al-Qur'an surat Al-Alaq ayat 1-5

أَقْرَأْ بِأَسْمِ رَبِّكَ الَّذِي خَلَقَ ﴿١﴾ خَلَقَ الْإِنْسَانَ مِنْ عَلَقٍ ﴿٢﴾ أَلَمْ يَكُنْ مِنْ عَلَقٍ ﴿٣﴾ أَلَمْ يَكُنْ مِنْ عَلَقٍ ﴿٤﴾ أَلَمْ يَكُنْ مِنْ عَلَقٍ ﴿٥﴾ أَلَمْ يَكُنْ مِنْ عَلَقٍ ﴿٦﴾ أَلَمْ يَكُنْ مِنْ عَلَقٍ ﴿٧﴾ أَلَمْ يَكُنْ مِنْ عَلَقٍ ﴿٨﴾ أَلَمْ يَكُنْ مِنْ عَلَقٍ ﴿٩﴾ أَلَمْ يَكُنْ مِنْ عَلَقٍ ﴿١٠﴾

بِالْقَلَمِ ﴿١١﴾ عَلَّمَ الْإِنْسَانَ مَا لَمْ يَعْلَمْ ﴿١٢﴾ كَلَّا إِنَّ الْإِنْسَانَ لِرَبِّهِ لَكَنَافٍ ﴿١٣﴾

Artinya: *Bacalah dengan (menyebut) nama Tuhanmu yang Menciptakan, Dia telah menciptakan manusia dari segumpal darah. Bacalah, dan Tuhanmulah yang Maha pemurah, Yang mengajar (manusia) dengan perantaran kalam. Dia mengajar kepada manusia apa yang tidak diketahuinya. Ketahuilah! Sesungguhnya manusia benar-benar melampaui batas.*

Dengan turunnya ayat tersebut di atas, manusia diberi kesempurnaan akal dan pikiran untuk membaca dan mengintegrasikan akal dan pikirannya untuk membuka ilmu-ilmu yang sudah Allah titipkan di dalam diri manusia. Dan dengan adanya kecanggihan teknologi akhir-akhir ini, manusia diwajibkan untuk memperluas pengetahuannya. Salah satunya dengan adanya teknologi komputer.

Dengan menggunakan teknologi komputer segalanya menjadi lebih efisien dan praktis. Tetapi, teknologi juga dapat disalahgunakan bahkan bisa digunakan sebagai alat untuk melakukan kejahatan. Muncul beberapa wacana-wacana dalam dunia teknologi, misalnya bagaimana memecahkan masalah keamanan data. Saat ini, pelaku dunia industri lebih sering menggunakan teknologi komputer dibandingkan menggunakan tempat penyimpanan dalam menyimpan suatu informasi atau data penting, akan tetapi keamanan data tersebut yang disimpan dalam komputer kurang memadai. Masalah ini yang

muncul dalam perkembangan teknologi komputer adalah kesulitan *user* atau pengguna komputer untuk menggunakan sejumlah *tools* yang terdapat di suatu aplikasi komputer, keinginan manusia untuk mempermudah penggunaan suatu alat dan (dalam dunia kerja) kurangnya efektifitas produksi kerja *user* dikarenakan harus menggunakan *tools* yang rumit sehingga pengolahan informasi menjadi lebih lama.

Untuk memecahkan beberapa masalah tersebut, pengenalan karakteristik alami manusia dapat digunakan. Suara merupakan salah satu bentuk *biometric* yang dapat digunakan sebagai *person identification*, contoh lainnya adalah wajah, sidik jari, iris mata, atau tanda tangan. Bila dibandingkan dengan *person identification* yang lain, pengenalan suara pembicara (*speech recognition*)¹ tidak membutuhkan biaya yang besar dan tidak membutuhkan peralatan khusus. Dari segi keamanan, sistem ini juga sangat baik, karena karakteristik suara manusia yang berbeda.

Oleh karena itu penulis membuat judul “Pengenalan Suara Menggunakan Metode *Linier Predictive Coding* (LPC)” sebagai judul tugas akhir. Sistem ini bermula dari pencuplikan sinyal analog menjadi sinyal digital. Untuk proses ekstraksi parameter suara digunakan metode *Linier Prediktive Coding* (LPC) untuk mendapatkan koefisien *cepstral*. Dari hasil keluaran LPC ini akan ditransformasikan ke dalam domain frekuensi dengan *fast fourier transform*

¹ Rabiner, Lawrence, and Biing-Hwang Juang, 1993, “Fundamentals of Speech Recognition”. Prentice Hall, New Jersey.

(FFT). Selanjutnya keluaran dari FFT akan di *compare* untuk memadukan suara asli dengan suara yang sudah disimpan di *database*.

Topik dalam skripsi ini, sebelumnya pernah ditulis oleh Rudy Adipranata dengan judul pengenalan suara manusia dengan metode LPC (*Linier Predictive Coding*) dan Jaringan Syaraf Tiruan (*Backpropagasi*) pada tahun 1999 di Universitas Petra Surabaya² dan oleh Indrawati dengan judul yang sama pada tahun 2003 di Sekolah Tinggi Ilmu Komputer Indonesia (STIKI) Malang³. Akan tetapi, skripsi yang sudah ada dengan skripsi penulis mempunyai perbedaan, dengan letak perbedaannya adalah:

a. Bahasa pemrograman yang digunakan

Pada skripsi Rudy Adipranata dan Indrawati menggunakan Visual Basic Versi 6.0 dan Matlab 6.1, sedangkan penulis menggunakan Java.

b. Tampilan layar

Tampilan layar berbeda antara penulis dengan Rudy Adipranata dan Indrawati.

c. Rudy Adipranata dan Indrawati hanya membuat aplikasi pengenalan suara sebagai *log in (speech recognition)*, penulis membuat pengenalan suara (*speech recognition*) dan ditambah *voice to command* sebagai pengenalan suara untuk memanggil perangkat lunak yang sudah ada di dalam komputer.

² faculty.petra.ac.id/resmana/publications.htm

³ perpustakaan.stiki.ac.id

1.2 Rumusan Masalah

Bagaimana membuat aplikasi pengenalan suara manusia sebagai keamanan komputer dengan menggunakan metode LPC (*Linier Predictive Coding*)?

1.3 Batasan Masalah

- a. Suara yang digunakan adalah suara manusia
- b. Suara yang diperkenalkan adalah suara dalam keadaan normal dalam artian orang tersebut tidak dalam keadaan sakit (flu, batuk, dan sakit tenggorokan) dan bisu.
- c. Perekaman suara di dalam ruangan yang kedap suara atau terbebas dari suara bising

1.4 Tujuan

Tujuan penelitian ini adalah untuk membuat aplikasi pengenalan suara manusia menggunakan metode *Linier Predictive Coding* (LPC) sehingga komputer bisa membaca dan mengenali suara (orang) dalam mengucapkan suatu kalimat tertentu.

1.5 Manfaat Penelitian

Adapun penelitian ini diharapkan memberikan manfaat bagi pihak-pihak yang terkait dengan penelitian ini, antara lain:

- **Bagi Peneliti**
Menambah ilmu, dan pengalaman dalam bidang Teknik Informatika, serta sebagai salah satu syarat untuk meraih gelar Sarjana Strata Satu (S-1) di Universitas Islam Negeri (UIN) Malang.
- **Bagi Lembaga/Universitas**
Hasil dari penelitian ini kiranya dapat digunakan sebagai tambahan informasi dalam meningkatkan *output* pendidikan khususnya di perguruan tinggi, yakni Universitas Islam Negeri (UIN) Maliki Malang.
- **Bagi Pengguna**
Menyimpan data di komputer bisa lebih aman dikarenakan hanyalah pemilik komputer yang bisa membuka komputernya.

1.6 Metode Penelitian

Dalam penelitian ini, langkah- langkah yang dilakukan adalah :

1. **Studi Literatur**
Mempelajari teori-teori yang berhubungan dengan pengenalan suara (*speech recognition*) dari berbagai referensi.
2. **Pendefinisian dan analisa masalah**
Mendefinisikan dan menganalisa masalah untuk mencari solusi yang tepat.
3. **Perancangan dan implementasi sistem**
Membuat perancangan perangkat lunak dengan analisa terstruktur dan mengimplementasikan hasil rancangan tersebut yaitu pengenalan suara manusia menggunakan metode LPC (*Linier Predictive Coding*).

4. Uji coba dan analisa hasil implementasi

Cara pengujiannya adalah dengan melakukan *sampling* kepada lima orang untuk menguji perangkat lunak dan menganalisa hasil dari implementasi tersebut apakah sudah sesuai tujuan yang dirumuskan sebelumnya untuk kemudian menarik kesimpulan

1.7 Sistematika Pembahasan

BAB I : Pendahuluan, membahas tentang latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian dan sistematika pembahasan.

BAB II : Kajian teori penunjang, yang berisi dasar-dasar teori tentang *voice recognition* dan *voice to command* serta integrasi dalam perspektif Al-Qur'an

BAB III : Analisis dan perancangan sistem, berisi tentang proses dan langkah-langkah pengenalan suara manusia.

BAB IV : Pembahasan aplikasi.

BAB V : Penutup, berisi kesimpulan dan saran.

BAB II

LANDASAN TEORI

2.1 Pengenalan Suara (*Voice recognition*)

Voice recognition adalah proses identifikasi suara berdasarkan kata yang diucapkan dengan melakukan konversi sebuah sinyal akustik, yang ditangkap oleh audio device (perangkat input suara). *Voice recognition* juga merupakan sistem yang digunakan untuk mengenali perintah kata dari suara manusia dan kemudian diterjemahkan menjadi suatu data yang dimengerti oleh komputer. Pada saat ini, sistem ini digunakan untuk menggantikan peranan input dari *keyboard* dan *mouse*.

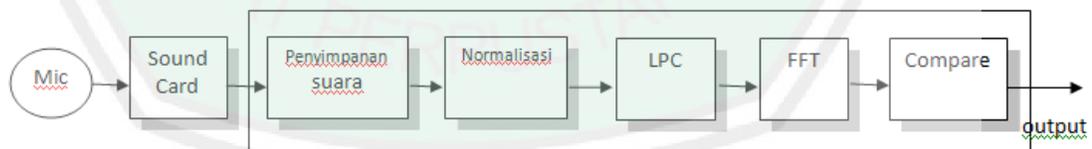
Keuntungan dari sistem ini adalah pada kecepatan dan kemudahan dalam penggunaannya. Kata – kata yang ditangkap dan dikenali bisa jadi sebagai hasil akhir, untuk sebuah aplikasi seperti *command & control*, penginputan data, dan persiapan dokumen. Parameter yang dibandingkan ialah tingkat penekanan suara yang kemudian akan dicocokkan dengan *template database* yang tersedia. Sedangkan sistem pengenalan suara berdasarkan orang yang berbicara dinamakan *speaker recognition*. Pada skripsi ini hanya akan dibahas mengenai *Voice recognition* karena kompleksitas algoritma yang diimplementasikan lebih sederhana daripada *speaker recognition*. Algoritma yang akan diimplementasikan pada bahasan mengenai proses *Voice recognition* ini adalah algoritma FFT (*Fast Fourier Transform*), yaitu algoritma yang cukup efisien dalam pemrosesan sinyal digital (dalam hal ini suara)

dalam bentuk diskrit. Algoritma ini mengimplementasikan algoritma *Divide and Conquer* untuk pemrosesannya. Konsep utama algoritma ini adalah mengubah sinyal suara yang berbasis waktu menjadi berbasis frekuensi dengan membagi masalah menjadi beberapa masalah yang lebih kecil. Kemudian, setiap masalah diselesaikan dengan cara melakukan pencocokan pola digital suara.

Secara umum proses pengenalan ucapan merupakan urutan proses pencuplikan suara atau disebut juga dengan konversi analog ke digital, mengekstraksi ciri suara, kemudian pengenalan. Bagian utama dengan pengenalan ucapan yaitu ekstraksi ciri suara dan proses pengenalan. Untuk proses ekstraksi ciri dilakukan dengan pengolahan sinyal digital (DSP).

2.2 Sistem Pengenalan Suara

Sistem pengenalan suara yang dibuat digambarkan pada blok diagram gambar 2.1.



Gambar 2.1 Blok diagram sistem pengenalan suara

Secara garis besar, cara kerja sistem pengenalan suara ini ialah mula-mula sinyal suara manusia yang diterima dengan menggunakan *microphone* (sinyal analog) dicuplik sehingga menjadi sinyal digital dengan bantuan *sound card* pada Personal Computer.

Sinyal digital hasil cuplikan ini terlebih dulu dinormalisasi (disamakan panjang sinyal yang satu dengan yang lain) kemudian diproses awal menggunakan metode *Linier Predictive Coding* (LPC) sehingga didapat beberapa koefisien LPC yang merupakan *feature* (ciri) dari suara pembicaraan. Kemudian koefisien LPC tersebut diproses dengan *Fast Fourier Transform* (FFT) untuk mendapatkan sinyal pada domain frekuensi. Hal ini bertujuan agar perbedaan antar pola kata yang satu dengan yang lain terlihat lebih jelas sehingga ekstraksi parameter sinyal memberikan hasil yang lebih baik. Hasil keluaran FFT ini akan di *compare* sebagai fungsi utama dari sistem untuk proses pengenalan

2.2.1 Normalisasi

Satu masalah yang cukup rumit dalam *speech recognition* (pengenalan suara) adalah proses perekaman yang terjadi sering kali berbeda durasinya, biarpun kata atau kalimat yang diucapkan sama. Bahkan untuk satu suku kata atau *vocal* yang sama seringkali proses perekaman terjadi dalam durasi yang berbeda. Sebagai akibatnya proses *matching* antara sinyal uji dengan sinyal referensi (*template*) seringkali tidak menghasilkan nilai yang optimal. Sebuah teknik yang cukup populer

di awal perkembangan teknologi pengolahan sinyal suara adalah dengan cara memanfaatkan sebuah teknik *dynamic-programing* yang juga lebih dikenal sebagai *dynamic time warping* (DTW). Teknik ini ditujukan untuk mengakomodasi perbedaan waktu antara proses perekaman saat pengujian dengan yang tersedia pada *template* sinyal referensi. Prinsip dasarnya adalah dengan memberikan sebuah rentang 'step' dalam ruang (dalam hal ini sebuah *frame-frame* waktu dalam *sampel*, *frame-frame* waktu dalam (*template*) dan digunakan untuk mempertemukan lintasan yang menunjukkan local match terbesar (kemiripan) antara *time frame* yang lurus. Total '*similarity cost*' yang diperoleh dengan algoritma ini merupakan sebuah indikasi seberapa bagus *sampel* dan *template* ini memiliki kesamaan, yang selanjutnya akan dipilih best-matching *template* suara. (Tri Budi Santoso dan Miftahul Huda, pengolahan sinyal wicara: hal 54)

2.2.2 Linier Predictive Coding (LPC)

Linier Predictive Coding (LPC) adalah salah satu metode parametrik yang digunakan untuk merepresentasikan sinyal. Lebih tepatnya *Linier Predictive Coding* (LPC) di sini adalah sebagai pemfilteran suara yang dijadikan ciri khas untuk membedakan suara yang sudah ada di dalam *database* dengan suara yang di-*input*-kan.

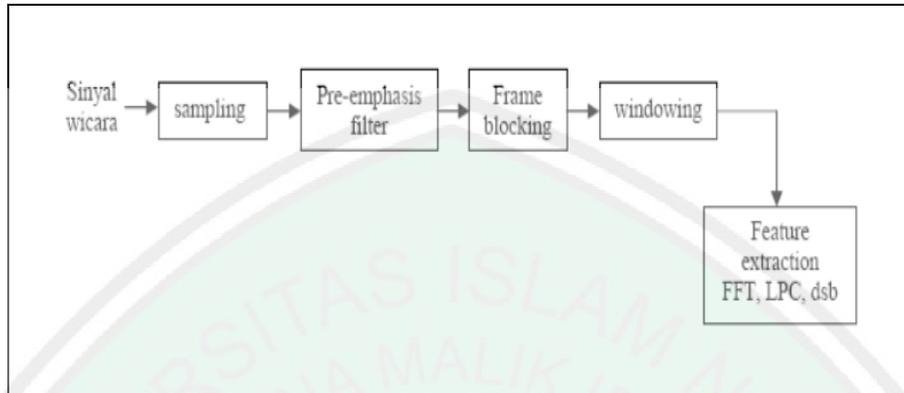
LPC umumnya digunakan untuk pengenalan suara karena terdapat beberapa keuntungan dari LPC, yaitu;

1. LPC menyediakan pemodelan yang baik untuk sinyal suara (*speech signal*), hal ini terutama untuk bagian *voice* dimana pemodelan *all pole* model LPC menghasilkan pendekatan selubung spektakuler jalur *vocal* yang baik, sedangkan untuk bagian *invoice*, pemodelan LPC ini tidak seefektif sebelumnya tapi masih dapat digunakan keperluan pengenalan suara.
2. LPC dapat dengan mudah dan langsung diterapkan baik secara perangkat lunak maupun perangkat keras karena perhitungan matematis yang dilibatkan realtif lebih singkat dari metode-metode yang dikenal sebelumnya seperti *filter bank*.
3. Hasil pengenalan suara yang didapat dengan menerapkan LPC cukup baik bahkan lebih baik dari metode-metode yang dikenal sebelumnya.

Dalam proses LPC terdapat beberapa hal yang harus dilakukan, yaitu:

1. Preemphasis/ Penapisan

Dalam pengolahan sinyal suara preemphasis filter diperlukan setelah proses sampling. Tujuan dari pemfilteran ini adalah untuk mendapatkan bentuk *spectral* frekuensi sinyal suara yang lebih halus. Dimana bentuk *spectral* yang relatif bernilai tinggi untuk daerah rendah dan cenderung turun secara tajam untuk daerah frekuensi di atas 2000Hz.



Gambar.2.2 Posisi Filter Pre-Emphasis pada sistem pengolah suara

2. Frame Blocking

Pada tahap ini sinyal kata yang telah diratakan dibagi sebanyak T *frame* dengan masing-masing *frame* memuat N cuplikan dan *frame-frame* yang berdekatan dipisahkan sejauh M cuplikan. Ukuran sampel tiap *frame* dihitung dari rata-rata cuplik tiap detik dengan waktu cuplik tiap detik dengan waktu cuplik tiap periode, umumnya digunakan 30 ms. Jika rata-rata cuplik adalah 8000Hz, maka jumlah sampel tiap *frame* adalah $3 \cdot 10^{-3} \text{ detik} \cdot 8000 \text{ sampel/detik}$, sama dengan 240 sampel. Dengan menggunakan aturan N point DFT, bahwa jumlah sampel N harus bernilai 2^p , dimana $240 < 2^p$, sedangkan $240 < 28$, maka $N=256$ [Rabiner,1993]. Untuk melakukan pembagian blok ini dapat mengacu pada *linier prediction*). Parameter-parameter yang biasanya digunakan antara lain; N adalah jumlah sampel pada analisa *frame* blocking, dan M adalah jarak antara satu *frame* dengan *frame* selanjutnya.

3. Windowing

Untuk pengukuran nilai energi pada sinyal wicara kita harus melibatkan fungsi *window*. Hal ini karena dalam pengukuran energi sinyal wicara kita harus menyusunnya dalam *frame-frame* tertentu. Ini merupakan standar dalam teknologi *speech processing*, sebab secara umum dalam pengolahan sinyal wicara kita terlibat dengan sinyal dengan durasi yang terlalu panjang bila dihitung dalam total waktu pengukuran. Fenomena ini juga dikenal sebagai *short term speech signal energy*.

2.2.3 Fast Fourier Transform (FFT)

Dalam pertengahan tahun 1960, J. W. Cooley dan J. W. Tukey, berhasil merumuskan suatu teknik perhitungan *Fourier Transform* yang efisien. *Fast Fourier transform* (FFT) menjadi penting untuk bermacam – macam aplikasi, dari pengolahan sinyal digital dan memecahkan persamaan *diferensial parsial* menjadi algoritma-algoritma untuk penggandaan bilangan *integer* dalam jumlah yang banyak. Algoritma FFT merupakan penyederhanaan dari *Diskrit Fourier Transform* (DFT). Dengan metode FFT, laju komputasi dari perhitungan *transformasi Fourier* dapat ditingkatkan, Komputasi DFT adalah komputasi yang memerlukan waktu untuk proses looping dan memerlukan banyak memori. Dengan menerapkan FFT, perhitungan DFT dapat dipersingkat, dalam hal ini proses looping dapat direduksi. DFT memiliki persyaratan jumlah data harus merupakan bilangan 2^n untuk $n=0,1,2,3,\dots$. Waktu komputasi DFT memiliki kompleksitas N^2 sedangkan FFT

memiliki kompleksitas $Np/2$ dengan $p = 2 \log N$, sehingga FFT lebih cepat daripada DFT dengan rasio kecepatan FFT terhadap DFT adalah: (New York: McGraw-Hill, 2004, hal: 554).

2.3 Pemrosesan Sinyal Digital

Lazimnya sinyal yang dijumpai dalam bidang sains dan teknologi merupakan sinyal analog, dengan demikian sinyal tersebut merupakan fungsi dari satu atau beberapa pengubah kontinyu. Sinyal *analog* yang ada yang diproses secara langsung dengan menggunakan sistem pengolah sinyal analog khusus (semisal untuk penapis, penganalisis frekuensi atau pengali frekuensi). Digitalisasi merupakan proses konversi sinyal *analog* menjadi sinyal digital, yang terdiri dari pencuplikan, kuantisasi, dan pengkodean. Pencuplikan adalah proses yang menggambarkan sinyal kontinyu sebagai nilai runtun periodis. Kuantisasi meliputi pendekatan representasi nilai gelombang oleh salah satu nilai tak terhingga. Sedangkan pengkodean merupakan penandaan nilai aktual untuk suatu nilai, pengkodean yang sering digunakan adalah pengkodean *biner*. Dalam proses ini sinyal analog yang *continue* mampu diubah menjadi runtun kode yang terpilih dari sinyal terhingga. Untuk pengenalan suara terdapat tiga langkah yang digunakan yaitu *spectral analisis*, *Filtering*, dan *auto correlation*. Spektral analisis merupakan proses analisa spektrum frekuensi suara yang sering terkait dengan transformasi. *Filtering* berguna untuk membuang bagian yang tidak berguna dan untuk memodifikasi frekuensi

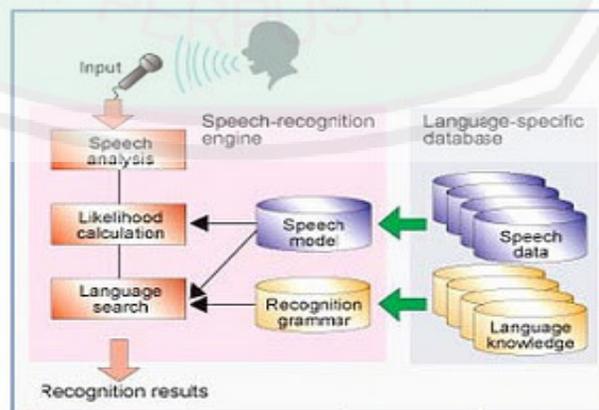
untuk mendapat informasi. Sedangkan *auto correlation* berguna untuk mencari karakteristik rongga suara manusia.

Analisa spektrum suara dapat dilakukan dengan menggunakan transformasi fourier waktu diskrit (DFT), *Linier Predictive Coding* (LPC), Mel Scala dan sebagainya. Dari proses ini dapat dianalisa *magnitude*, energi, atau *phase* dari suatu sinyal. (Menurut hasil laporan tugas akhir pengolahan suara. Brawijaya: 2006).

2.4 Skema Utama Dan Algoritma Voice recognition

Terdapat 4 langkah utama dalam sistem pengenalan suara:

- Penerimaan data input
- Ekstraksi, yaitu penyimpanan data masukan sekaligus pembuatan *database* untuk *template*.
- Perbandingan / pencocokan, yaitu tahap pencocokan data baru dengan data suara (pencocokan tata bahasa) pada *template*.
- Validasi identitas pengguna.



Gambar 2.3 Skema pengenalan



Gambar 2.4 Spektrum Suara

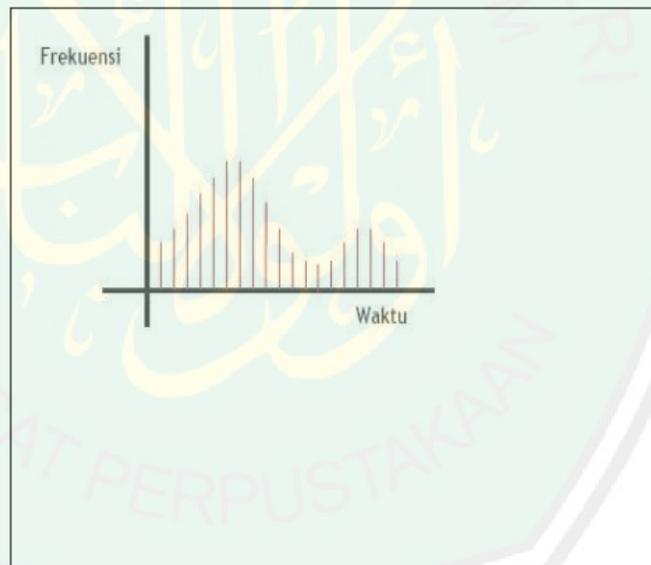
Secara umum, *speech recognizer* memproses sinyal suara yang masuk dan menyimpannya dalam bentuk digital. Hasil proses digitalisasi tersebut kemudian dikonversi dalam bentuk spektrum suara yang akan dianalisa dengan membandingkannya dengan *template* suara pada *database* sistem.

Sebelumnya, data suara masukan dipilah-pilah dan diproses satu per satu berdasarkan urutannya. Pemilahan ini dilakukan agar proses analisis dapat dilakukan secara paralel. Proses yang pertama kali dilakukan ialah memproses gelombang kontinu spektrum suara ke dalam bentuk *diskrit*. Langkah berikutnya ialah proses kalkulasi yang dibagi menjadi dua bagian :

- Transformasi gelombang *diskrit* menjadi *array* data.

- Untuk masing-masing elemen pada *array* data, hitung ketinggian gelombang (frekuensi). Objek permasalahan yang akan dibagi adalah masukan berukuran n , berupa data diskrit gelombang suara.

Ketika mengkonversi gelombang suara ke dalam bentuk *diskrit*, gelombang diperlebar dengan cara memperinci berdasarkan waktu. Hal ini dilakukan agar proses algoritma selanjutnya (pencocokan) lebih mudah dilakukan. Namun, efek buruknya ialah data *array* yang terbentuk akan lebih banyak.



Gambar 2.5 Contoh Hasil Konversi Sinyal Diskrit

Dari tiap elemen *array* data tersebut, dikonversi ke dalam bentuk bilangan *biner*. Data *biner* tersebut yang nantinya akan dibandingkan dengan *template* data suara.

2.5 Java

Java adalah bahasa pemrograman berorientasi objek yang dikembangkan oleh Sun Microsystems sejak tahun 1991. Bahasa ini dikembangkan dengan model yang mirip dengan bahasa C++ dan Smalltalk, namun dirancang agar lebih mudah dipakai dan ber-*platform independent*, yaitu dapat dijalankan di berbagai jenis sistem operasi dan arsitektur komputer. Bahasa ini juga dirancang untuk pemrograman di internet sehingga dirancang agar aman dan portabel.

Pemrograman berorientasi objek adalah teknik untuk mengorganisasi program dan dapat dilakukan dengan hampir semua bahasa pemrograman. Namun Java telah mengimplementasikan berbagai fasilitas agar seorang pemrogram dapat mengoptimalkan teknik pemrograman berorientasi objek. Selain itu, Java juga memiliki fasilitas perpustakaan (*library*) yang luas untuk memudahkan pemrogram membuat aplikasi yang diinginkan.

2.3.1 Karakteristik Java

Java memiliki karakteristik yang diantaranya adalah sebagai berikut:

- **Sederhana:** Bahasa pemrograman Java menggunakan sintaks mirip dengan C++ namun sintaks pada Java telah banyak diperbaiki terutama menghilangkan penggunaan *pointer* yang rumit dan pewarisan majemuk (*multiple inheritance*). Java juga menggunakan *automatic memory allocation* dan *memory garbage collection* yaitu sistem pengalokasian memori dan pengumpulan sampah otomatis untuk membebaskan memori yang tidak terpakai lagi.
- **Berorientasi objek:** Java merupakan pemrograman berorientasi objek yang membuat program dapat dibuat secara modular dan dapat dipergunakan kembali. Pemrograman berorientasi objek memodelkan dunia nyata ke dalam objek dan melakukan interaksi antar objek-objek tersebut.
- **Terdistribusi:** Java dibuat untuk membuat aplikasi terdistribusi secara mudah dengan adanya *libraries networking* yang terintegrasi pada Java.
- **Terinterpretasi:** Program Java dijalankan menggunakan *interpreter* yaitu *Java Virtual Machine* (JVM). JVM merupakan aplikasi sederhana yang ditulis dalam bahasa C untuk mengeksekusi program yang ditulis dalam bahasa Java. JVM mengubah kode sumber (*source code*) Java yang telah dikompilasi menjadi Java *bytecodes* sehingga dapat dijalankan pada platform yang berbeda-beda.
- **Kokoh (*robust*):** Java mempunyai reliabilitas atau keandalan yang tinggi. *Compiler* pada Java mempunyai kemampuan mendeteksi eror secara lebih

teliti dibandingkan bahasa pemrograman lain. Java mempunyai *runtime-Exception handling* untuk membantu mengatasi eror atau kesalahan pada pemrograman.

- **Aman:** Java sebagai bahasa pemrograman untuk aplikasi internet dan terdistribusi, memiliki beberapa mekanisme keamanan untuk menjaga aplikasi tidak digunakan untuk merusak sistem komputer yang menjalankan aplikasi tersebut.
- **Berarsitektur netral:** Program Java merupakan *independent platform*. Program cukup mempunyai satu buah versi yang dapat dijalankan pada *platform* berbeda dengan *Java Virtual Machine*.
- **Portabel:** Kode sumber (*source code*) maupun program Java dapat dengan mudah dibawa ke *platform* yang berbeda-beda tanpa harus dikompilasi ulang.
- **Berkinerja Tinggi:** Java dapat melakukan optimasi pada *byte code* dengan baik sehingga eksekusi program dapat dilakukan dengan cepat.
- **Berkas berganda (*multi-threaded*):** Berkas yang dimaksud di sini yaitu bahwa Java mempunyai kemampuan untuk membuat suatu program yang dapat melakukan beberapa pekerjaan sekaligus secara simultan.
- **Dinamis:** Java didesain untuk dapat dijalankan pada lingkungan yang dinamis. Perubahan pada suatu *class* dengan menambahkan *properties* ataupun *method* dapat dilakukan tanpa mengganggu program yang menggunakan *class* tersebut.

2.6 Pengenalan Suara Pada Al-Qur'an

Manusia dianugerahi akal dan pikiran oleh Allah SWT supaya bisa membedakan antara yang baik dan yang buruk, berbeda dengan makhluk Allah SWT yang lainnya, mereka tidak disertai dengan akal. Kesempurnaan manusia ini terdapat pada Surat al-mu'minun ayat 12-14, yaitu:

وَلَقَدْ خَلَقْنَا الْإِنْسَانَ مِنْ سُكَّالَةٍ مِنْ طِينٍ ﴿١٢﴾
 ثُمَّ جَعَلْنَاهُ نُطْفَةً فِي قَرَارٍ مَكِينٍ ﴿١٣﴾
 ثُمَّ خَلَقْنَا النُّطْفَةَ عَلَقَةً فَخَلَقْنَا الْعَلَقَةَ مُضْغَةً فَخَلَقْنَا
 الْمُضْغَةَ عِظْلًا فَكَسَوْنَا الْعِظْمَ لَحْمًا ثُمَّ أَنْشَأْنَاهُ خَلْقًا آخَرَ
 فَتَبَارَكَ اللَّهُ أَحْسَنُ الْخَالِقِينَ ﴿١٤﴾

Artinya: “Dan sesungguhnya Kami telah menciptakan manusia dari suatu sari pati (berasal) dari tanah (12). Kemudian Kami jadikan saripati itu air mani (yang disimpan) dalam tempat yang kokoh (rahim) (13). Kemudian air mani itu Kami jadikan segumpal darah, lalu segumpal darah itu Kami jadikan segumpal daging, dan segumpal daging itu Kami jadikan tulang belulang, lalu tulang belulang itu Kami bungkus dengan daging. Kemudian Kami jadikan dia makhluk yang (berbentuk) lain. Maka Maha Suci Allah, Pencipta Yang Paling Bagus (14)”

Dari ayat tersebut, diketahui bahwa manusia merupakan ciptaan Allah SWT yang paling sempurna. Mempunyai kelebihan akal pikiran, suara dan pendengaran, seperti yang terdapat pada surat At-tiin ayat 4, yaitu:

لَقَدْ خَلَقْنَا الْإِنْسَانَ فِي أَحْسَنِ تَقْوِيمٍ ﴿٤﴾

Artinya: “*Sesungguhnya Kami telah menciptakan manusia dalam bentuk yang sebaik-baiknya.*”

Kesempurnaan manusia yang diberikan oleh Allah SWT sangatlah menonjol jika dibandingkan dengan makhluk lain, salah satunya adalah manusia diberi kelebihan mempunyai suara yang merdu, seperti yang tertuang dalam surat Luqman ayat 19, yaitu:

وَأَقْصِدْ فِي مَشْيِكَ وَأَعْظُضْ مِنْ صَوْتِكَ إِنَّ أَنْكَرَ الْأَصْوَاتِ
لَصَوْتُ الْحَمِيرِ ﴿١٩﴾

Artinya: “*Dan sederhanalah kamu dalam berjalan dan lunakkanlah suaramu. Sesungguhnya seburuk-buru suara ialah suara keledai*

Dari keterangan di atas, jelas bahwa manusia diberi suara yang lunak dan merdu. Maka dari itu sejak kecil manusia harus diajarkan cara berbicara yang merdu pula supaya terlihat begitu besar kebesaran Allah SWT.

Pengenalan suara bertujuan untuk mempermudah *user* dalam mengakses komputer yaitu mengubah suara manusia menjadi suatu perintah pada komputer, sehingga *user* bisa dengan mudah membuka aplikasi. *User* bisa membuka aplikasi Al-‘Quran tanpa harus memegang *mouse* dan mengucapkan dengan merdu ayat Al-‘Quran yang ingin dibaca.

Manusia mempunyai pedoman hidup yaitu Al-Qu'ran yang menjadi mu'jizat Nabi Muhammad SAW. Nabi yang mempunyai sifat dan akhlak paling sempurna dan menjadi panutan umat muslim di dunia. Hal tersebut dijelaskan dalam Al-Qur'an surat AL-anbiya' ayat 107 yang berbunyi:

وَمَا أَرْسَلْنَاكَ إِلَّا رَحْمَةً لِّلْعَالَمِينَ ﴿١٠٧﴾

Artinya: “Dan tiadalah Kami mengutus kamu, melainkan untuk (menjadi) rahmat bagi semesta alam”

Dengan akhlak mulia yang dimiliki oleh Nabi Muhammad SAW, manusia dianjurkan untuk mencontoh Nabi Muhammad SAW supaya bisa memberi manfaat kepada makhluk Allah SWT yang lainnya. Maka dari itu, penulis mengharapkan supaya skripsi ini bisa berguna bagi manusia dalam membaca dan membuka aplikasi serta mempermudah pekerjaan yang berhubungan dengan komputer.

BAB III

ANALISIS DAN PERANCANGAN SISTEM

3.1. Analisa Permasalahan Sistem

Berdasarkan latar belakang masalah, maka dibutuhkan perangkat lunak (*software*) yang dapat membantu mempermudah *user* dalam mengakses komputer. Berikut adalah proses utama pada aplikasi:

1. Perekaman

Proses ini berfungsi untuk mendapatkan *sample* suara. Suara dideteksi melalui *microphone*, lalu sistem akan melakukan perekaman terhadap suara. Hasil dari rekaman ini adalah *file* suara dengan ekstensi WAV(*Waveform Audio Format*)

```
private void
ButtonRecordActionPerformed(java.awt.event.ActionEvent
evt) {
try
{
ButtonRecord.setText("Start Recording");
if(TextRecord.getText().trim().isEmpty()){
setSource("source");
}
else{
setSource(TextRecord.getText());
}
VText.dataAcq(1, source);
```

```

        //Sistem.out.println("recording complete");
    }
    catch (Exception e)
    {
        Sistem.out.println("Exception: " + e.toString());
    } // TODO add your handling code here:
    }

```

2. Normalisasi suara

Normalisasi berfungsi untuk mendapatkan *sample* suara *user* saja karena ada kalanya saat melakukan rekaman sumber suara yang berasal dari sekitar juga ikut terekam. Oleh karena itu proses ini dibutuhkan agar

```

private void
ButtonLoginActionPerformed(java.awt.event.ActionEvent
evt) {
    // TODO add your handling code here:
    ButtonLogin.setText("Recording...");
    double a[]=new double[100];
    try
    {
        vTText.dataAcq(1, compare);
        Sistem.out.println("Recording Complete");
        ButtonLogin.setText("Checking...");
        ButtonLogin.setEnabled(false);
        insertArray(a, compare);
        valueArray(a[0], a);
        ButtonLogin.setText("Start Recording");
        ButtonLogin.setEnabled(true);
    }
    catch (Exception e)

```

pencocokan suara *user* mendapatkan hasil yang maksimal.

```
        ButtonLogin.setEnabled(true);  
    }  
    catch (Exception e)  
    {  
        Sistem.out.println("Exception: " + e.toString());  
    }  
}
```

3. Ekstraksi suara

Suara yang telah didapatkan akan diekstraksi sehingga hanya didapatkan kumpulan angka yang merepresentasikan suara tersebut. Tahap pertama pada ekstraksi adalah LPC, LPC disini berfungsi untuk melakukan ekstraksi terhadap suara menjadi parameter-parameter sinyal suara. Parameter-parameter tersebut akan dibagi kedalam blok-blok *frame*, beberapa blok *frame* yang jaraknya akan dikumpulkan menjadi sebuah kumpulan dari beberapa blok *frame* Hasil dari proses LPC ini adalah sinyal suara berbasis waktu.

Proses yang selanjutnya adalah FFT (*Fast Fourier Transform*), proses ini akan mentransformasikan sinyal suara yang berbasis waktu menjadi sinyal suara berbasis frekuensi. Hal ini dilakukan agar suara satu dengan yang lainnya terlihat jelas sehingga ekstraksi parameter memberikan hasil yang lebih baik. *Source* LPC adalah sebagai berikut:

```

public void insertArray(double[] array,String source){
    clearArray(array);
    try
    {
        result = VText.complexmodulus(1, source);
        double[][] ret = new double[result.length][];
        ret[0] = (double[]) ((MWArray) result[0]).getData();
        for (int i = 0; i < 10000; i++)

```

4. Pencocokan suara

Tahap ini merupakan tahap yang terpenting karena menentukan apakah suara tersebut valid atau tidak. Di sini akan membandingkan parameter *input* dengan data yang sudah ada pada *database*. Cara yang dilakukan adalah dengan membandingkan setiap parameter sinyal masuk dengan parameter data yang sudah ada sampai batas iterasi tertentu. Jika hasil setiap perbandingan cocok maka sinyal *input* tersebut adalah sinyal suara yang valid. *Source compare* sebagai berikut:

```

public void valueArray(double val,double[] array){
    double setengah,satusetengah;
    setengah=val/2;
    satusetengah=val+(val/2);
    int kounter=0;
    double[] source=new double[100];

```

```
        from complex_modulus_user where field1
between "+ setengah +" and "
        + ""+satusetengah+"");
        while(rs.next()){
for(int i=0;i<100;i++){
        source[i]=rs.getDouble("field"+(i+1));
        }
kounter=compareArray(source, array);
if (kounter>=40){
        break;
        }
        }
        stm.close();
        System.out.println("Frekuensi Matched :"+
kounter +"%");
        if (kounter >=40){
            isMatch= true;
            this.dispose();
            System.out.println("matched");
        }
        else{
            LabelStatus.setIcon(new
javax.swing.ImageIcon(getClass().getResource("image/access
-denied1.png")));
        }
        }
        catch (Exception e) {
            System.out.print (e);
        }
    }
}
```

3.2.PERANCANGAN SISTEM

3.2.1. Diskripsi Umum sistem

Teknik yang digunakan adalah *speech recognition*, pada prinsipnya terdapat empat langkah sistem pengenalan suara, yaitu:

1. Penerimaan data *input*.

Menerima *input* suara yang masuk dan menyimpannya ke dalam *file* suara berekstensi WAV

2. Ekstraksi yaitu penyimpanan data masukan sekaligus pembuatan *database* untuk *template*

Ekstraksi data sehingga menghasilkan sinyal suara berbasis frekuensi, hasil dari ekstraksi tersebut akan disimpan pada *database* sebagai media perbandingan.

3. Perbandingan atau pencocokan, yaitu tahap data baru dengan data suara.

Proses perbandingan ini dilakukan dengan menggunakan jaringan saraf tiruan dengan metode *backpropagation*, dengan cara membandingkan setiap parameter sinyal yang masuk dengan parameter data yang ada pada *database*

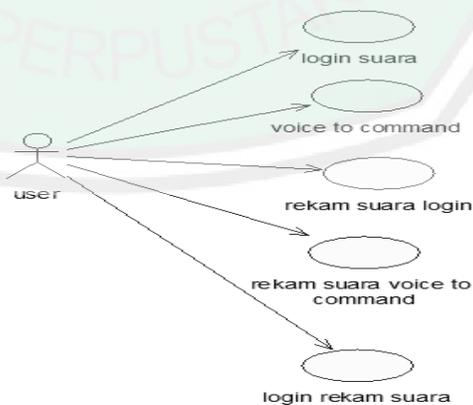
4. *Validasi* identitas pengguna.

Ini adalah hasil akhir dari hasil perbandingan sesuai dengan batas yang ditentukan oleh sistem. Jika hasil perbandingan kurang dari batas yang ditentukan maka suara dianggap tidak valid sebaliknya suara akan dianggap valid.

3.2.2. Analisis Use Case Diagram

Langkah pertama yang harus dilakukan untuk menangkap kebutuhan pengguna adalah memodelkan sistem dengan menggunakan *use case diagram*. Untuk memperoleh gambaran mengenai sistem yang digunakan proses dan data model dari sistem, dapat dimodelkan dengan *use case diagram*. Dengan *use case diagram* ini dapat diketahui proses yang terjadi pada aktivitas sistem. Dengan diagram ini juga dapat diketahui fungsi yang digunakan oleh sistem yang digunakan. Berikut adalah gambar secara global *use case diagram* dari sistem yang akan dibuat:

- *Actor* : Pengguna *software* aplikasi, mungkin berupa manusia, hardware atau sistem *informasi* yang lain. Aktor dapat memasukan *informasi* ke sistem, menerima *informasi* dari sistem, atau juga keduanya. Dalam sistem ini, aktor meliputi pengguna *software*.
- *Use case* : Perilaku (apa yang pengguna kerjakan) *software* aplikasi, termasuk interaksi antara aktor dengan *software* aplikasi tersebut. Dalam hal ini meliputi operator-operator *text to speech*.



Gambar 3.1 Use case diagram

Dari gambar *use case diagram* diatas, maka deskripsi dari masing-masing *use case* dapat dijelaskan sebagai berikut:

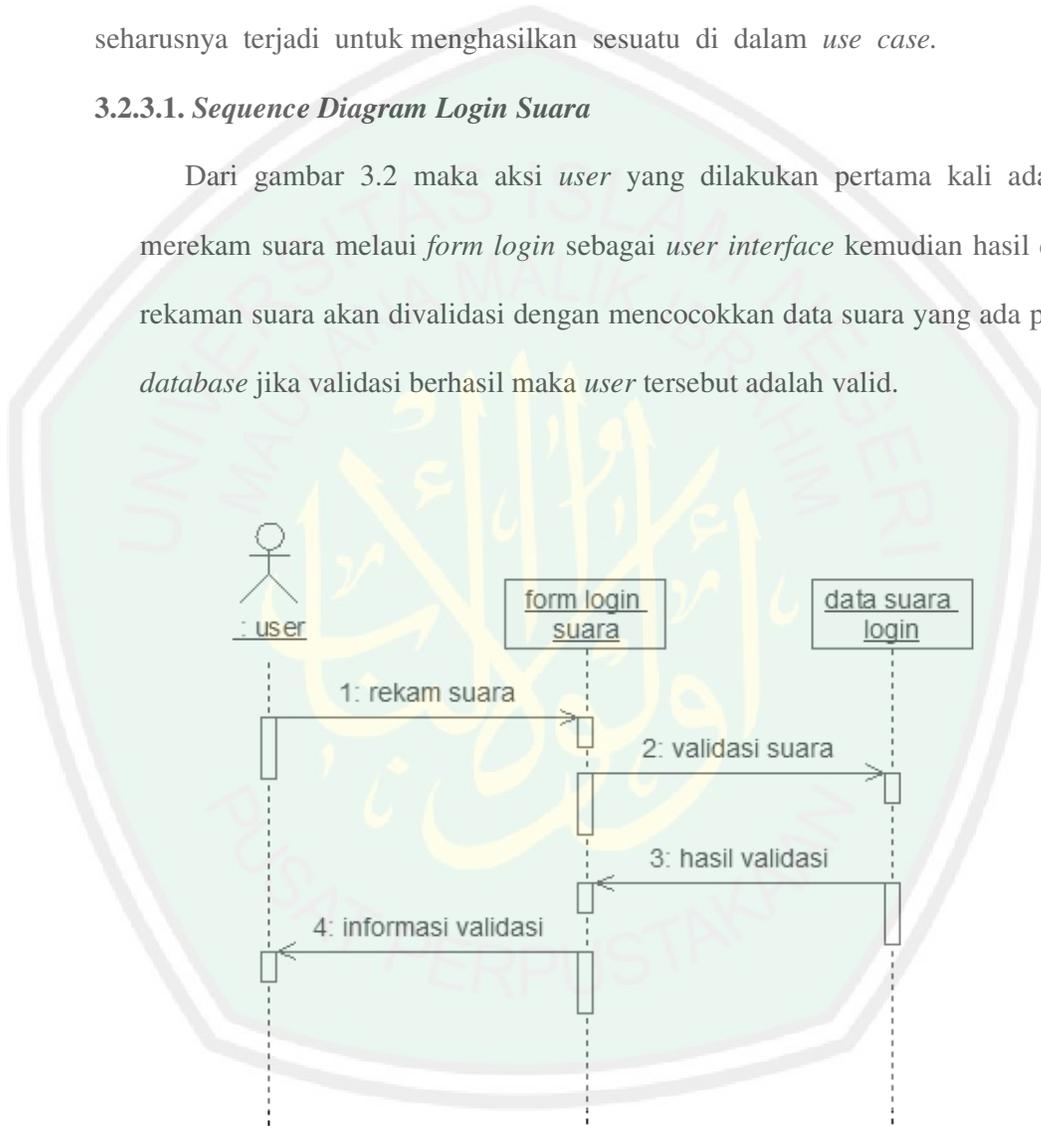
- a. *Use Case* : *login user*
Actor : *User*
Deskripsi : *Use Case* ini digunakan untuk melakukan *login* dengan menggunakan suara
- b. *Use Case* : *speech recognition*
Actor : *User*
Deskripsi : *Use Case* ini digunakan untuk melakukan *speech recognition*
- c. *Use Case* : *rekam suara login*
Actor : *User*
Deskripsi : *Use Case* ini digunakan merekam suara *user* yang akan digunakan sebagai identifikasi saat *login*
- d. *Use Case* : *rekam suara speech recognition*
Actor : *User*
Deskripsi : *Use Case* ini digunakan untuk merekam suara yang dijadikan identifikasi *speech recognition*
- e. *Use Case* : *login rekam suara*
Actor : *User*
Deskripsi : *Use Case* ini digunakan untuk melakukan *login* sebelum melakukan perekaman suara untuk *source* identifikasi suara

3.2.3. Analisis Sequence Diagram

Sequence diagram menjelaskan interaksi obyek yang disusun dalam suatu urutan waktu. Dalam *sequence diagram* diperlihatkan tahap demi tahap apa yang seharusnya terjadi untuk menghasilkan sesuatu di dalam *use case*.

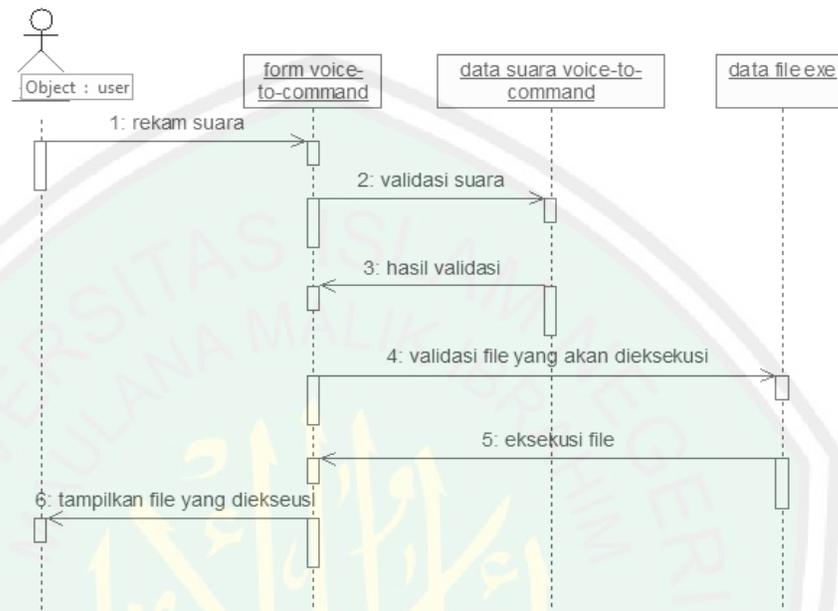
3.2.3.1. Sequence Diagram Login Suara

Dari gambar 3.2 maka aksi *user* yang dilakukan pertama kali adalah merekam suara melalui *form login* sebagai *user interface* kemudian hasil dari rekaman suara akan divalidasi dengan mencocokkan data suara yang ada pada *database* jika validasi berhasil maka *user* tersebut adalah valid.



Gambar 3.2 Sequence Diagram Login Suara

3.2.3.2. Sequence Diagram Speech recognition



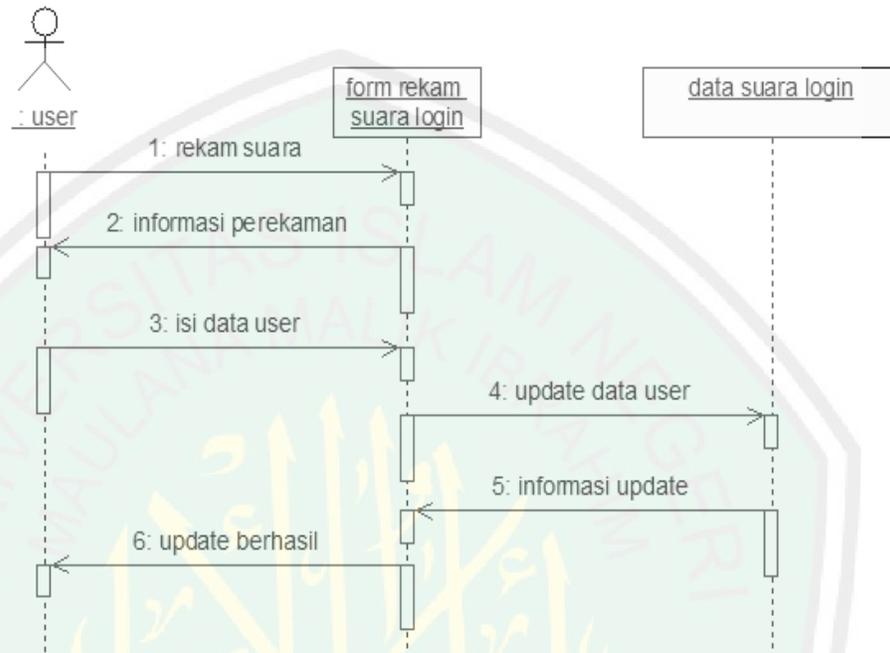
Gambar 3.3 Sequence Diagram Speech recognition

Dari gambar 3.3 aksi yang dilakukan *user* pertama kali adalah merekam suara melalui *form voice to command*, hasil suara akan divalidasi. Jika validasi berhasil, kemudian *sistem* akan mencari nama *file* yang sudah di *input*-kan ke *database*, kemudian akan dieksekusi sesuai dengan *input* suara, dan selanjutnya sistem akan mengeksekusi *file* sesuai hasil validasi.

3.2.3.3. Sequence Diagram Rekam Suara Login

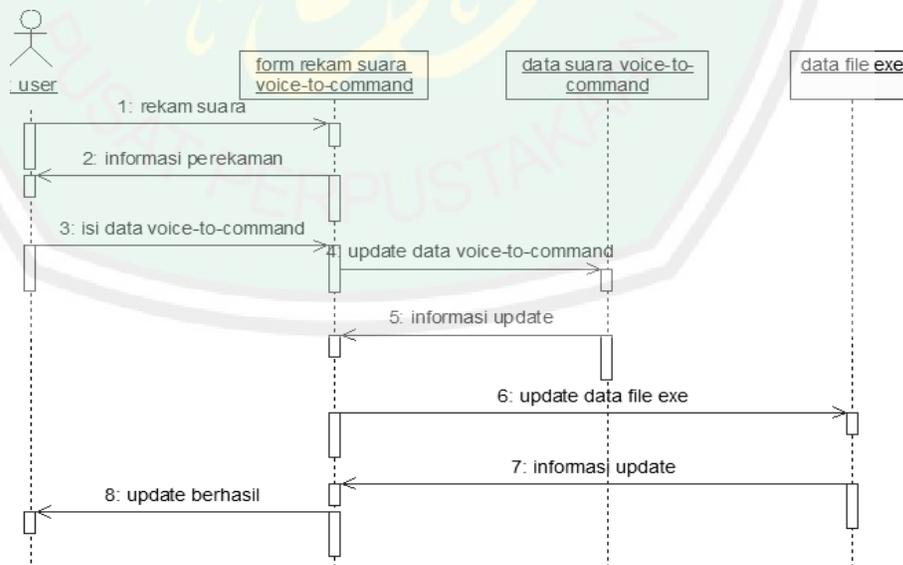
Dari gambar 3.4 menjelaskan proses perekaman suara yang akan digunakan untuk validasi data suara saat *user login* pertama kali aksi yang pertama adalah *user* merekam suara. Hasil dari perekaman suara berupa *file*

WAV kemudian *user* akan mengisi *field-field* pada *form* sebagai informasi *user* yang nantinya akan disimpan di *database* pada *sistem*.



Gambar 3.4 Sequence Diagram Rekam Suara Login

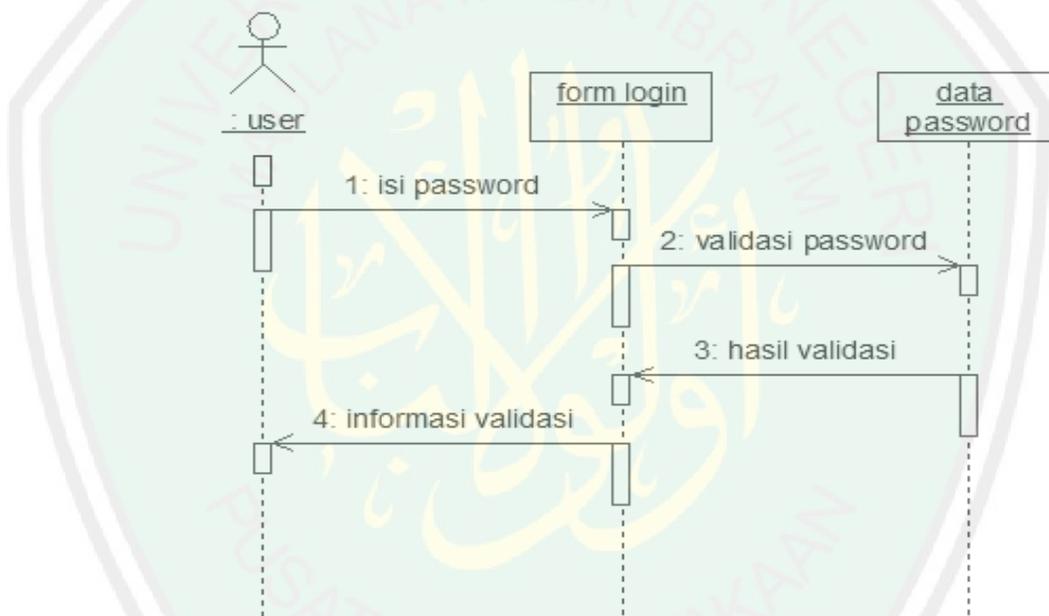
3.2.3.4. Sequence Diagram Rekam Suara Speech recognition



Gambar 3.5 Sequence Diagram Rekam Suara Speech recognition

Dari gambar 3.5 menjelaskan proses perekaman suara yang akan digunakan untuk validasi data suara saat *user* melakukan *voice to command* pertama kali aksi yang pertama adalah *user* merekam suara. Hasil dari perekaman suara berupa *file* WAV kemudian *user* akan mengisi *field-field* pada *form* sebagai *informasi user* untuk *voice to command* yang nantinya akan disimpan di *database* pada *sistem*.

3.2.3.5. Sequence Diagram Login Rekam Suara

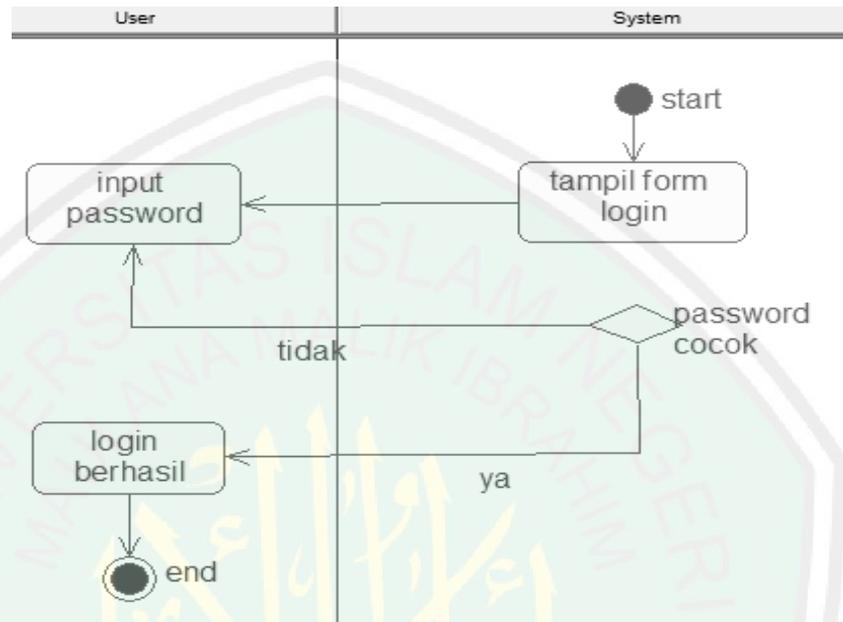


Gambar 3.6 Sequence Diagram Login Rekam Suara

Diagram ini menjelaskan tentang *login* untuk proses perekaman suara sehingga hanya *user* yang mengetahui *password* saja yang dapat melakukan proses perekaman suara. Dari gambar 3.6 maka proses pertama adalah *user* mengisi *password* kemudian *password* tersebut akan divalidasi jika validasi berhasil maka *user* tersebut valid.

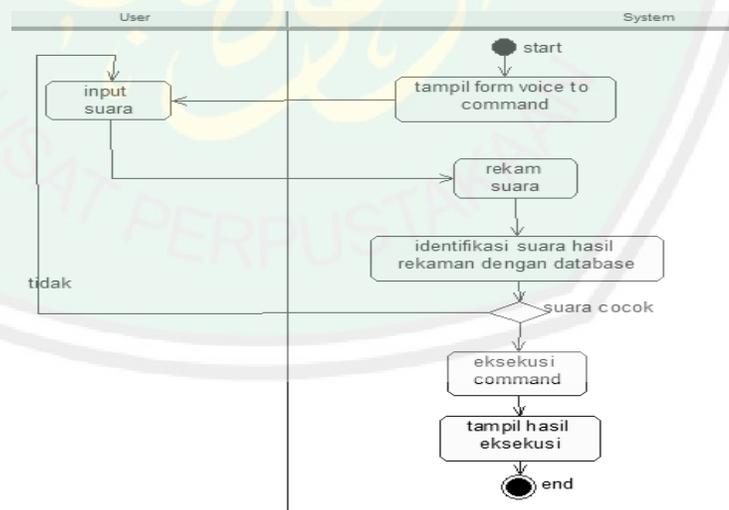
3.2.4. Diagram Activity

3.2.4.1. Activity Diagram Login Suara



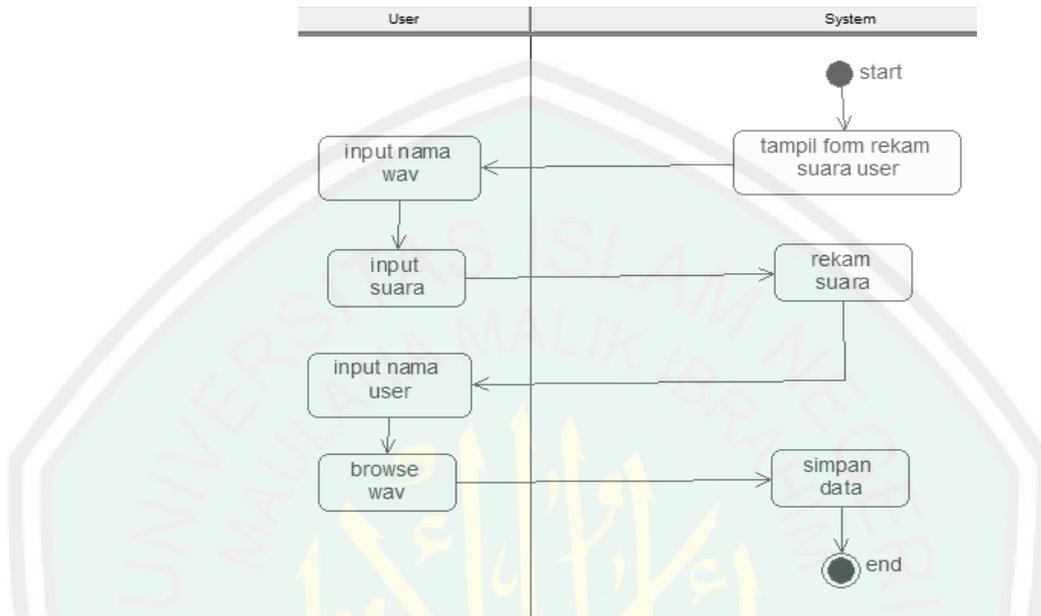
Gambar 3.7 Activity Diagram Login Suara

3.2.4.2. Activity Diagram Speech recognition



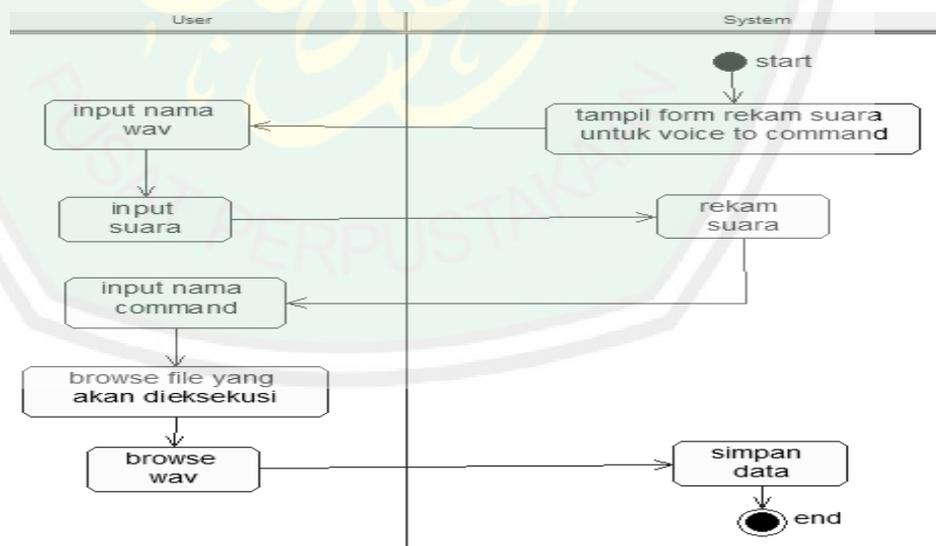
Gambar 3.8 Activity Diagram Speech recognition

3.2.4.3. Activity Diagram Rekam Suara Login



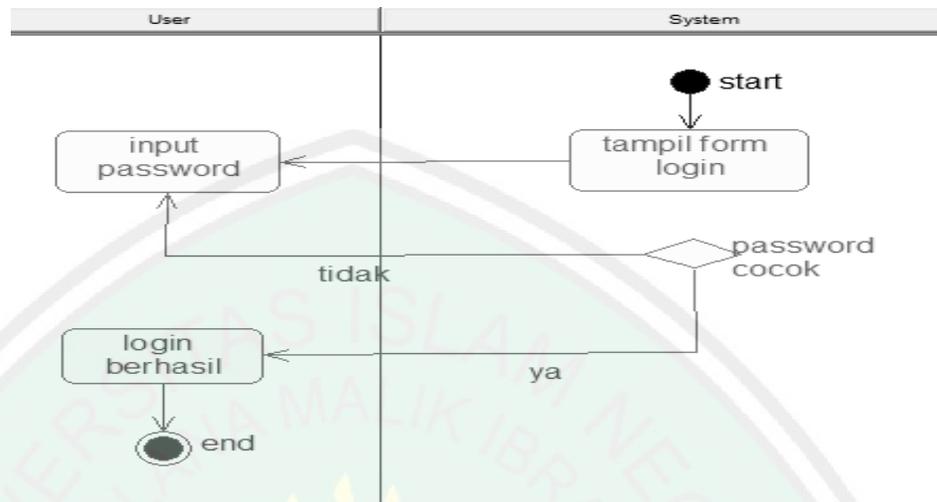
Gambar 3.9 Activity Diagram Rekam Suara Login

3.2.4.4. Activity Diagram Rekam Suara Speech recognition



Gambar 3.10 Activity Diagram Rekam Suara Speech recognition

3.2.4.5. Activity Diagram Login Rekam Suara



Gambar 3.11 Activity Diagram Login Rekam Suara

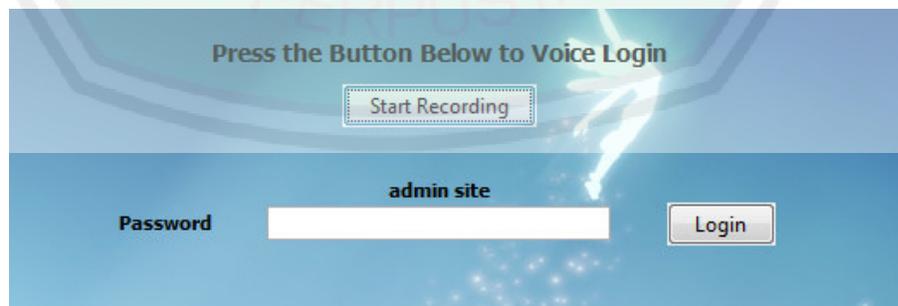
3.2.5. Rancangan Antar Muka

Antarmuka merupakan media yang digunakan untuk menjembatani antara sistem dengan *user*.

Bagian-bagian tersebut antara lain adalah:

1. Rancangan Desain Form Login

Desain ini merupakan antarmuka utama, yaitu halaman untuk melakukan *login* dengan suara. Berikut merupakan rancangan desain halaman utama:



Gambar 3.12 Rancangan Antar Muka Form Login

Gambar diatas merupakan rancangan desain *form login*, pada *form* ini terdapat beberapa komponen penting yaitu:

a. Tombol Record

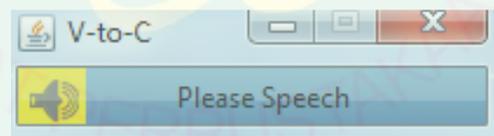
Tombol record berfungsi untuk merekam suara *user* dan mencocokkannya pada *database* yang telah tersedia pada *sistem*

b. Tombol *Login*

Tombol *login* berfungsi sebagai *security* sebelum *user* melakukan perekaman suara. Jadi hanya *user* yang mengetahui *password* yang dapat melakukan perekaman suara sebagai *source* untuk melakukan perbandingan suara.

Selain itu *form login* ini akan tampil *fullscreen* dan mempunyai fungsi tidak bisa berpindah ke window yang lain (*alt + tab*) dan tidak dapat dilakukan *task manager* untuk menjaga eksklusifitas dari *form* sehingga hanya *user* dengan suara yang valid saja yang dapat mengakses

2. **Rancangan Desain Form Voice to Command**



Gambar 3.13 Rancangan Antar Muka Form Voice to Command

Form ini akan berfungsi melakukan perintah *Voice to Command* jika *user* berhasil melakukan *login* dengan suara. Pada *form* ini *user* sekali lagi akan diminta untuk menginputkan suara, jika suara valid maka aplikasi akan mengeksekusi perintah sesuai dengan suara tersebut

3. Rancangan Desain *Form Rekam Suara User*

Gambar 3.14 Rancangan Antar Muka Form Rekam Suara User

Form ini berfungsi melakukan perekaman suara untuk identifikasi suara yang nantinya sebagai media pembandingan dengan suara *user* saat *login*. Pada *form* ini terdapat beberapa fungsi utama yaitu:

a. Merekam Suara

Fungsi ini berfungsi untuk melakukan perekaman suara untuk identifikasi suara yang nantinya sebagai media pembandingan dengan suara *user* saat *login*.

b. Browse WAV

Fungsi ini berfungsi untuk melakukan pengaksesan terhadap suara wav hasil rekaman.

c. *Insert*

Fungsi ini berfungsi untuk memasukkan ekstraksi *frekuensi* suara hasil dari browse wav dengan *username* sebagai *unique key* ke *database sistem*

4. Rancangan Desain *Form* Rekam Suara untuk *Voice to Command*

Gambar 3.15 Rancangan Antar Muka *Form* Rekam Suara *Voice to command*

Form ini berfungsi untuk melakukan perekaman suara *source Speech to Command* sebagai media pembanding terhadap suara pada *form speech command*. Pada *form* ini terdapat beberapa fungsi utama yaitu:

a. Merekam WAV

Fungsi ini berfungsi untuk melakukan perekaman suara untuk identifikasi suara yang nantinya sebagai media pembanding dengan suara *user* saat melakukan *speech recognition*.

b. Browse WAV

Fungsi ini berfungsi untuk melakukan pengaksesan terhadap suara wav hasil rekaman.

c. Browse Executable *File*

Fungsi ini berfungsi melakukan pengaksesan terhadap *file* yang akan dieksekusi jika suara valid pada saat *speech recognition*

d. Insert

Fungsi ini berfungsi untuk memasukkan ekstraksi frekuensi suara hasil dari browse wav dan *file path* dari browse executable *file* dengan process name sebagai *unique key* ke *database sistem*



BAB IV

HASIL DAN PEMBAHASAN

4.1. Analisa dan Implementasi

Pada aplikasi ini sistem dapat mengenali suara yang masuk, suara yang masuk tersebut akan disimpan dalam format WAV dan diekstraksi dengan menggunakan metode FFT sehingga suara tersebut menjadi frekuensi, selanjutnya hasil ekstraksi tersebut akan di *compare* untuk membandingkan data asli dengan data yang ada pada *database*.

4.2 Instalasi Program

4.2.1 Kebutuhan Perangkat Keras

1. Laptop Acer ASPIRE 4930
2. Processor Intel Core 2 T5800
3. RAM 1GB DDR2
4. 250 GB HDD

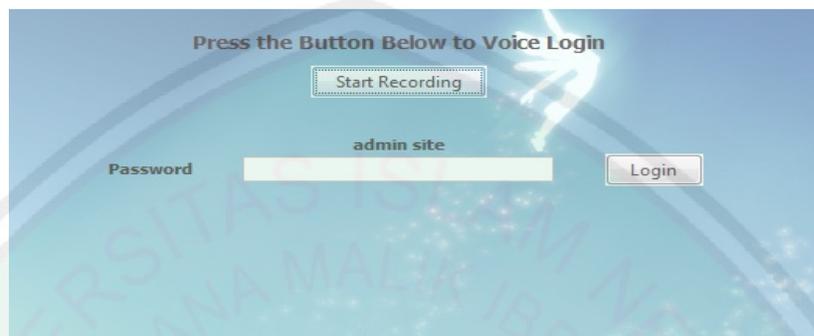
4.2.2 Kebutuhan Perangkat Lunak

1. Instalasi Microsoft Windows 7 Ultimate
2. Instalasi JDK 1.7 dan JRE.7
3. Instalasi Phyton 2.7
4. VText library
5. Instalasi Matlab compiler runtime
6. Instalasi XAMPP

4.2.3. Penjelasan Penggunaan Program

1. Form Utama

Tampilan utama program adalah sebagai berikut.



Gambar 4.1 Tampilan Utama

Untuk menginputkan suara maka *user* harus meng-klik tombol start recording, setelah tombol diklik maka aplikasi akan merekam suara *user* selama 5 detik hasil rekaman tersebut akan dibandingkan dengan data pada *database*. Jika *user* ingin merekam *source* suara baru yang akan dijadikan pembanding dengan suara yang masuk maka *user* harus masuk menjadi *admin* dengan mengisi *password* dan kemudian klik tombol *login* jika berhasil maka akan muncul tampilan seperti dibawah ini.



Gambar 4.2 Tampilan Login Admin Saat Berhasil

Pada tampilan tersebut terdapat dua tombol yang baru yaitu :

a. *Record voice for user login*

Tombol ini berfungsi untuk menampilkan *form record user* yang berfungsi untuk merekam suara *user* yang nanti akan dijadikan *source* untuk *login* pertama kali dengan suara pada tampilan utama

b. *Record voice for voice to command*

Tombol ini berfungsi untuk menampilkan *form record process* yang berfungsi untuk merekam suara *user* yang nanti akan dijadikan *source* untuk *voice to command* dan sekaligus memilih *file* yang akan dieksekusi jika suara berhasil.

2. *Record User Form*

Gambar 4.3 Tampilan Record User Form

Form ini berfungsi melakukan perekaman suara untuk identifikasi suara yang nantinya sebagai media pembanding dengan suara *user* saat *login*. Pada *form* ini terdapat beberapa fungsi utama yaitu:

a. Merekam Suara

Fungsi ini berfungsi untuk melakukan perekaman suara untuk identifikasi suara yang nantinya sebagai media pembandingan dengan suara *user* saat *login*.

b. *Browse WAV*

Fungsi ini berfungsi untuk melakukan pengaksesan terhadap suara wav hasil rekaman.

c. *Insert*

Fungsi ini berfungsi untuk memasukkan ekstraksi frekuensi suara hasil dari *browse wav* dengan *username* sebagai *unique key* ke *database* sistem

3. *Record Procces Form*

The screenshot shows a Windows-style application window titled "Record Process Form". It is divided into two main sections. The top section, titled "Insert Process Freq to DB", contains three text input fields: "Process Name", "Data Record", and "Executable File". The "Data Record" and "Executable File" fields have "Choose File" buttons to their right. An "Insert" button is centered below these fields. The bottom section, titled "Record Proses Freq", contains a "Record Name" input field and a "Start Record" button. Below the "Record Name" field, there is a text label: "wav file will be stored in c:\wav\source\record_name".

Gambar 4.4 Tampilan Record Procces Form

Form ini berfungsi untuk melakukan perekaman suara *source voice to command* sebagai media pembanding terhadap suara pada *form voice to command*. Pada *form* ini terdapat beberapa fungsi utama yaitu:

a. Merekam WAV

Fungsi ini berfungsi untuk melakukan perekaman suara untuk identifikasi suara yang nantinya sebagai media pembanding dengan suara *user* saat melakukan *voice to command*.

b. *Browse* WAV

Fungsi ini berfungsi untuk melakukan pengaksesan terhadap suara wav hasil rekaman.

c. *Browse Executable File*

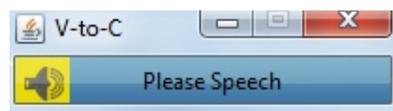
Fungsi ini berfungsi melakukan pengaksesan terhadap *file* yang akan dieksekusi jika suara valid pada saat *voice to command*

d. Insert

Fungsi ini berfungsi untuk memasukkan ekstraksi frekuensi suara hasil dari *browse wav* dan *file path* dari *browse executable file* dengan process name sebagai unique key ke *database* sistem

4. *Voice to command Form*

Jika *user* berhasil melakukan *login* pada tampilan utama maka akan muncul *voice to command form*.



Gambar 4.5 Tampilan *Voice to command Form*

Form ini akan berfungsi melakukan perintah *voice to command* jika *user* berhasil melakukan *login* dengan suara .Pada *form* ini *user* sekali lagi akan diminta untuk menginputkan suara, jika suara valid maka aplikasi akan mengeksekusi perintah sesuai dengan suara tersebut

4.2.4 Deskripsi Program

1. *Form* Utama

- a. Untuk melakukan perekaman suara dan mengecek kevalidan dari suara :

```
private void
ButtonLoginActionPerformed(java.awt.event.ActionEvent
evt) {
// TODO add your handling code here:
ButtonLogin.setText("Recording...");
double a[]=new double[100];
try
{
vTText.dataAcq(1, compare);
Sistem.out.println("Recording Complete");
ButtonLogin.setText("Checking...");
ButtonLogin.setEnabled(false);
insertArray(a, compare);
valueArray(a[0], a);
ButtonLogin.setText("Start Recording");
ButtonLogin.setEnabled(true);
}
catch (Exception e)
{
Sistem.out.println("Exception: " + e.toString());
}
}
```

b. Untuk melakukan *login* sebagai *admin* :

```
private void ButtonLoginAdminActionPerformed
(java.awt.event.ActionEvent evt) {
    try {
        if
(ButtonLoginAdmin.getText().trim().equalsIgnoreCase(
"login")){
            stm = (Statement) con.createStatement();
            rs=stm.executeQuery("select * from
admin_password");
            rs.first();
            if(Arrays.equals(TPassword.getPassword(),rs.getStrin
g(1).trim().toCharArray())){
                ButtonLoginAdmin.setText("Log Out");
                ButtonVoiceCommand.setVisible(true);
                ButtonSuaraLogin.setVisible(true);
            }
        }
    else
        if(ButtonLoginAdmin.getText().trim().equalsIgnoreCase(
"log out")){
            ButtonLoginAdmin.setText("Login");

            ButtonVoiceCommand.setVisible(false);
            ButtonSuaraLogin.setVisible(false);
        }
    }
    catch (Exception e) {
        System.out.print(e);
    }
    TPassword.setText("");
}
```

2. Record User Form

a. Untuk melakukan perekaman suara :

```
private void
ButtonRecordActionPerformed(java.awt.event.Action
Event evt) {
try
{
    ButtonRecord.setText("Start Recording");
    if(TextRecord.getText().trim().isEmpty()){
        setSource("source");
    }
    else{
        setSource(TextRecord.getText());
    }
    VText.dataAcq(1, source);
    ButtonRecord.setText("Recording Completed");
    //Sistem.out.println("recording compelete");
}
catch (Exception e)
{
    Sistem.out.println("Exception: " +
e.toString());
} // TODO add your handling code here:
}
```

c. Untuk melakukan *input ke database*

```

private void
InsertToDBActionPerformed(java.awt.event.ActionEvent
evt) {
// TODO add your handling code here:
    try{
double a[]=new double[100];
if (cekVal()){
    if
(selectProcess(TextUser.getText().trim().toLowerCase())
){
        int confirm =
JOptionPane.showConfirmDialog(this,
        "Are You Sure Want to Update This
Data?",
        "Confirm",
        JOptionPane.YES_NO_OPTION);
        if (confirm == JOptionPane.YES_OPTION) {
deleterecordModulus(TextUser.getText().trim().toLowerCa
se());
        insertArray(a, TextWav.getText());

insertToDBase(TextUser.getText().trim().toLowerCase(),
a);
        JOptionPane.showMessageDialog(this, "Inserting
Data Success", "Success",1);
        }
        }
        else{
            insertArray(a, TextWav.getText());

insertToDBase(TextUser.getText().trim().toLowerCase(),
a);
            JOptionPane.showMessageDialog(this, "Inserting
Data Success", "Success",1);
        }
    }
}
catch(Exception e){
    JOptionPane.showMessageDialog(this, e.getMessage(),
"Error",0);
}
}
}

```

d. untuk koneksi Mysql

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class KoneksiMysql {
    String url, usr, pwd, dbn;
    public KoneksiMysql (String dbn) {
        this.url = "jdbc:mysql://localhost/" + dbn;
        this.usr = "root";
        this.pwd = "root";
    }

    public KoneksiMysql (String host, String user,
String pass, String dbn) {
        this.url = "jdbc:mysql://" + host + "/" + dbn;
        this.usr = user;
        this.pwd = pass;
    }

    public Connection getConnection() {
        Connection con = null;
        try {
            Class.forName("com.mysql.jdbc.Driver");
            con =
DriverManager.getConnection(this.url, this.usr,
this.pwd);
        } catch (ClassNotFoundException e) {
            System.out.println ("Error #1 : " +
e.getMessage());
            System.exit (0);
        } catch (SQLException e) {
            System.out.println ("Error #2 : " +
e.getMessage());
            System.exit (0);
        }
        return con;
    }
}
```

3. Record Procces Form

a. Untuk merekam suara *user*:

```
private void
ButtonRecordActionPerformed(java.awt.event.Ac
tionEvent evt) {
try
{
    ButtonRecord.setText("Start Recording");

    if(TextRecord.getText().trim().isEmpty()){
        setSource("source");
    }
    else{
        setSource(TextRecord.getText());
    }
    VText.dataAcq(1, source);
    ButtonRecord.setText("Recording
Completed");
    //Sistem.out.println("recording
compolete");
}
catch (Exception e)
{
    Sistem.out.println("Exception: " +
e.toString());
} // TODO add your handling code here:
}
```

b. Untuk memasukkan data ke *database*:

```

private void
InsertToDBaseActionPerformed(java.awt.event.ActionEvent
evt) {
// TODO add your handling code here:
String quotes="";
try{
double a[]=new double[100];
if (cekVal()){
quotes=sqlQuote(TextProses.getText().toLowerCase());
Sistem.out.println(quotes);
if
(selectProcess(TextUser.getText().trim().toLowerCase())){
int confirm = JOptionPane.showConfirmDialog(this,
"Are you sure want to update data?",
"Confirm",
JOptionPane.YES_NO_OPTION);
if (confirm == JOptionPane.YES_OPTION) {
deleterecordModulus(TextUser.getText());
deleterecordTasklist(TextUser.getText());
insertArray(a, TextWav.getText());
insertToDBase(TextUser.getText().toLowerCase(), a);

insertTaskList(TextUser.getText().toLowerCase(), quotes);
JOptionPane.showMessageDialog(this, "Inserting Data
Success", "Success",1);
}
}
else{
insertArray(a, TextWav.getText());
insertToDBase(TextUser.getText(), a);
insertTaskList(TextUser.getText(), quotes);
JOptionPane.showMessageDialog(this, "Inserting Data
Success", "Success",1);
}
}
}
catch(Exception e){
JOptionPane.showMessageDialog(this, e.getMessage(),
"Error",0);
}
}
}

```

4. *Voice to command Form*

a. Untuk merekam suara dan membandingkan dengan *database*, jika suara cocok maka *file* akan dieksekusi:

```
public static void main(String[] args) {
    double []c=new double [100];
    FormLogin fn=new FormLogin();
    fn.setAlwaysOnTop(true);
    fn.setVisible(true);
    AltTabStopper alt=new AltTabStopper(fn);
    alt.create(fn);
    while(!fn.getMatch()){
    }
    alt.stop();
    tester t=new tester();
    t.setVisible(true);
    while(true && t.isShowing()){
        t.recordVtext();
        t.insertArray(c,compare);
        t.valueArray(c[0], c);
    }
}
```

4.2.5 Analisa Hasil

Analisa hasil pada aplikasi ini akan difokuskan pada pengujian kemampuan untuk mengenali suara *user* satu dengan yang lainnya. Pengujian pada aplikasi terdiri dari dua jenis yaitu pengujian untuk *login* dengan suara dan pengujian untuk *voice to command*.

Pengujian menggunakan skenario yang diantaranya:

1. *User* yang valid telah menginputkan *sample* suara yang hasil ekstraksi disimpan pada *database*
2. Selanjutnya diambil *sample* suara dari 5 *user* yang dipilih secara acak untuk menguji apakah aplikasi dapat mengenali suara *user* yang sebenarnya atau tidak.
3. Pengambilan *sampling* suara dilakukan sebanyak 3 kali untuk setiap jenis pengujian.
4. Pengambilan suara dilakukan ditempat yang cukup sepi (tidak ada intervensi suara dari sumber lain) .
5. Pengambilan *sampling* suara dilakukan selama 5 detik.
6. Jika akurasi *user* mencapai 70% atau lebih maka suara dianggap valid

Berikut adalah *table* hasil pengujian.

1. Pengujian *login* suara

Suara yang diuji: “coba”

No	<i>User</i>	Percobaan	Akurasi
1	<i>User 1</i>	1	2%
2	<i>User 1</i>	2	47%
3	<i>User 1</i>	3	39%
4	<i>User 2</i>	1	50%
5	<i>User 2</i>	2	43%
6	<i>User 2</i>	3	68%
7	<i>User 3</i>	1	58%

8	User 3	2	67%
9	User 3	3	66%
10	User 4	1	64%
11	User 4	2	56%
12	User 4	3	61%
13	User 5	1	24%
14	User 5	2	32%
15	User 5	3	34%

Tabel 4.1 Hasil Pengujian untuk *Login Suara*

2. Pengujian *Voice to command*

Suara yang diuji: "Krom " untuk mengeksekusi Google Chrome.exe

No	User	Percobaan	Akurasi
1	User 1	1	14%
2	User 1	2	29%
3	User 1	3	37%
4	User 2	1	62%
5	User 2	2	33%
6	User 2	3	41%
7	User 3	1	67%
8	User 3	2	53%
9	User 3	3	49%

10	User 4	1	32%
11	User 4	2	44%
12	User 4	3	59%
13	User 5	1	53%
14	User 5	2	49%
15	User 5	3	37%

Table 4.2 Hasil Pengujian untuk *Voice to command*

Dari hasil percobaan diatas maka dapat disimpulkan bahwa aplikasi bisa mengenali suara *user* yang asli dan tidak. Selain itu juga dapat disimpulkan bahwa dari beberapa pengujian beberapa *sampling* suara *user* memiliki beberapa kemiripan dengan suara *user* yang asli yang dibuktikan dengan tingkat akurasi diatas 60% atau mendekati 70%

4.2.5.1 Perhitungan Manual

Cara penghitungan manual untuk mendapatkan hasil pengujian *Login* suara dan *Voice to Command* adalah dengan membagi setiap *field* yang ada di *database* dalam hal ini adalah *source login* dengan setiap *field* yang ada pada suara rekam *user*. Bisa dikatakan sesuai jika hasil rekaman antara 0,5 – 1,5.

BAB V

PENUTUP

5.1 Kesimpulan

Dari hasil pengujian program pengenalan suara menggunakan metode *Linier Predictive Coding* diperoleh kesimpulan sebagai berikut:

1. Sistem ini berfungsi untuk mendeteksi suara *user* dan membedakan suara *user* yang satu dengan yang lainnya
2. Proses pendeteksian suara dilakukan melalui *microphone* lalu suara *user* akan direkam dalam format WAV
3. Proses membedakan suara dilakukan dengan mengekstraksi hasil rekaman suara. Rekaman suara pertama akan diekstraksi menjadi data suara berbasis waktu dengan metode normalisasi lalu hasil dari normalisasi akan diubah lagi menjadi data suara berbasis *frekuensi* yang kemudian akan dibandingkan dengan data *frekuensi* yang ada pada *database*
4. Untuk proses *voice to command* kecepatan eksekusi *file* tergantung seberapa besar proses *file* yang dieksekusi untuk *file* berukuran kecil seperti *notepad* eksekusi *file* terhitung cepat tetapi untuk *file* berukuran besar seperti *Mozilla Firefox* eksekusi *file* sedikit lebih lambat
5. Dari hasil percobaan ketepatan sistem ini untuk membedakan suara satu dengan yang lainnya sekitar 70% dan waktu yang dibutuhkan dari melakukan perekaman sampai mencocokkan data sekitar 5 detik

5.2 Saran

Dari penelitian ini masih banyak hal yang bisa dikembangkan seperti:

1. Dari segi metode tidak menutup kemungkinan bisa menggunakan metode yang lain yang mungkin saja dapat meningkatkan waktu eksekusi.
2. Dari segi keakuratan bisa ditingkatkan lagi menjadi lebih akurat lagi dan selain itu proses filtering noise dapat lebih ditingkatkan agar sumber suara selain *user* saat perekaman dapat diminimalisir.
3. Sistem ini dapat menjadi rujukan untuk pengembangan sistem yang lebih baik berbasis pendeteksian suara yang lebih murah karena hanya menggunakan sebuah komputer dan *microphone* yang bisa saja diimplementasikan untuk sistem presensi pegawai , sistem keamanan dan sebagainya.

DAFTAR PUSTAKA

- Adipranata, Rudy and Resmana. 1999. “*Pengenalan Suara Manusia dengan Metode LPC dan Jaringan Saraf Tiruan Propagasi Balik*”. Surabaya.
- Al-Qur’an dan Terjemahannya. 1971. Jakarta: Yayasan Penyelenggara Penerjemah/Pentafsir Al-Qur’an.
- Bradbury, Jeremy. 2002. “*Linear Predictive Coding*”. Paper.
- Fausett, Laurene. 1994. ” *Fundamental of Neural Network*”. New Jersey : Prentice Hall Inc.
- Ian Mcloughlin. 2009. “*Applied Speech And Audio Processing*”. Singapore: Cambridge University Press.
- Kusumadewi, S. 2003. “*Artificial Intellegence (Teknik dan Aplikasinya)*”. Yogyakarta: Graha Ilmu.
- Lasse L Mølgaard, Kasper W Jørgensen. 2005. “*Speaker Recognition: Special Course*”. IMM-DTU.
- Pelton, Gordon E. 1993. “*Voice Processing*”. Singapore : McGraw-Hill, Inc.
- Rabiner, Lawrence, and Biing-Hwang Juang. 1993. “*Fundamentals of Speech Recognition*”. New Jersey : Prentice Hall.
- Rich, Eleine, Knight, K. 1991. ” *Artificial Intelligence, 2nd Ed*”. Singapore : McGraw-Hill, Inc.
- www.basic-signalprocessing.com/sp/voiceToText.htm