

**IMPLEMENTASI *LINEAR PREDICTIVE CODING (LPC)* PADA
PENGENALAN POLA SUARA DENGAN METODE *PIECEWISE
DYNAMIC TIME WARPING (PDTW)***

SKRIPSI

oleh:

**ANISA WIMA
0610963008-96**



**PROGRAM STUDI ILMU KOMPUTER
JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS BRAWIJAYA
MALANG
2011**

**IMPLEMENTASI *LINEAR PREDICTIVE CODING* (LPC) PADA
PENGENALAN POLA SUARA DENGAN METODE *PIECEWISE*
DYNAMIC TIME WARPING (PDTW)**

SKRIPSI

Sebagai salah satu syarat untuk memperoleh gelar
Sarjana Ilmu Komputer

oleh:

ANISA WIMA
0610963008-96

UNIVERSITAS BRAWIJAYA



**PROGRAM STUDI ILMU KOMPUTER
JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS BRAWIJAYA
MALANG
2011**

LEMBAR PENGESAHAN SKRIPSI

Implementasi *Linear Predictive Coding* (LPC) pada Pengenalan Pola Suara
dengan Metode *Piecewise Dynamic Time Warping* (PDTW)

Oleh:

ANISA WIMA
0610963008-96

Setelah dipertahankan di depan Majelis Penguji
pada tanggal 09 Februari 2011

dan dinyatakan memenuhi syarat untuk memperoleh gelar
Sarjana Ilmu Komputer

Pembimbing I

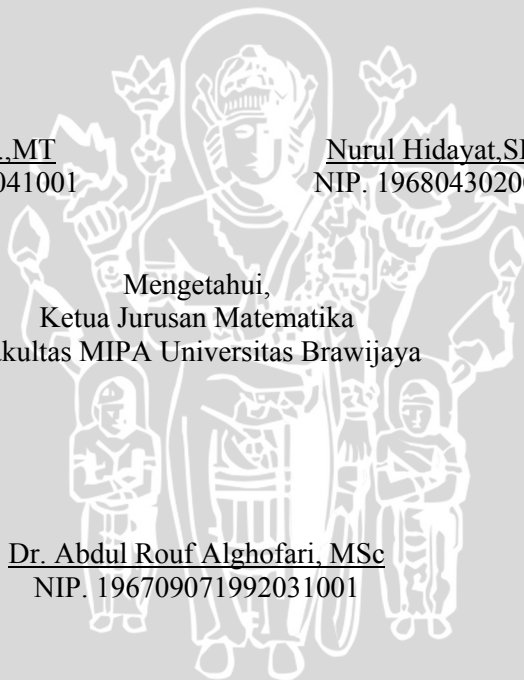
Pembimbing II

Bayu Rahayudi, ST., MT
NIP. 197407122006041001

Nurul Hidayat, SPd., MSc
NIP. 196804302002121001

Mengetahui,
Ketua Jurusan Matematika
Fakultas MIPA Universitas Brawijaya

Dr. Abdul Rouf Alghofari, MSc
NIP. 196709071992031001



LEMBAR PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Anisa Wima
NIM : 0610963008-96
Jurusan : Matematika
Program studi : Ilmu komputer
Penulis skripsi berjudul : Implementasi *Linear Predictive Coding* (LPC)
pada Pengenalan Pola Suara dengan Metode
Piecewise Dynamic Time Warping (PDTW)

Dengan ini menyatakan bahwa:

1. Tugas Akhir ini adalah benar-benar karya saya sendiri, dan bukan hasil plagiat dari karya orang lain. Karya-karya yang tercantum dalam Daftar Pustaka TA ini, semata-mata digunakan sebagai acuan/referensi.
2. Apabila dikemudian hari diketahui bahwa isi TA saya merupakan hasil plagiat, maka saya bersedia menanggung akibat hukum dari keadaan tersebut.

Demikian pernyataan ini dibuat dengan segala kesadaran

Malang, 09 Februari 2011

Yang menyatakan,

Anisa Wima
NIM. 0610963008-96



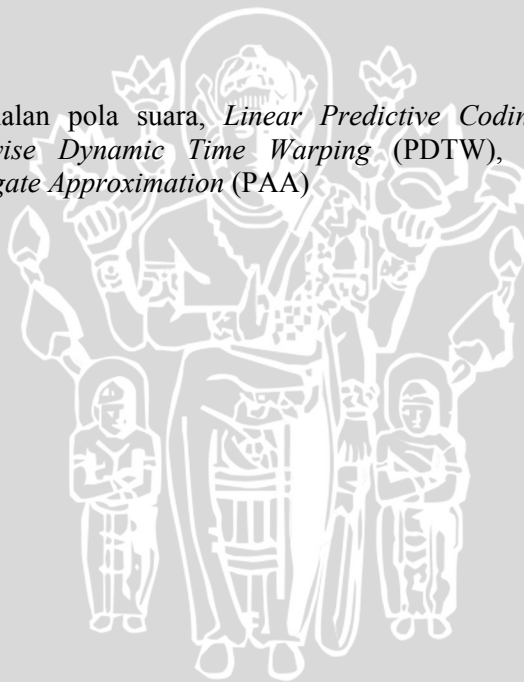
ABSTRAK

Pengenalan suara merupakan teknologi yang menggunakan peralatan dengan sumber masukannya adalah suara, seperti mikropn untuk menginterpretasikan suara manusia, untuk transkripsi atau sebagai metode alternatif interaksi dengan komputer.

Pada proses pengenalan suara, untuk mendapatkan informasi dari sinyal suara perlu dilakukan ekstraksi ciri sehingga dapat dianalisis untuk setiap variasi sinyal suara yang ada. Metode ekstraksi yang digunakan dalam penelitian ini adalah *Linear Predictive Coding* (LPC), yang menghasilkan koefisien-koefisien *cepstral* dari pola suara yang di-input-kan. Dari koefisien tersebut kemudian dilakukan pengenalan dengan menggunakan metode *Piecewise Dynamic Time Warping* (PDTW). PDTW merupakan sebuah modifikasi dari DTW yang menjalankan *Piecewise Aggregate Approximation* (PAA) sebagai metode pembandingan antara data *input* dan data referensi. Dari hasil perbandingan tersebut akan didapatkan suatu keputusan yang kemudian diubah menjadi *output* berupa simulasi gerak arah.

Hasil dari penelitian ini didapatkan *Word Error Rate* (WER) sebesar 48.75% untuk 80 data uji dari 4 kata. Dari hasil tersebut dapat disimpulkan bahwa metode LPC dan PDTW dapat digunakan untuk mengenali pola suara dengan keakuratan 51.25%.

Kata kunci: Pengenalan pola suara, *Linear Predictive Coding* (LPC), *Piecewise Dynamic Time Warping* (PDTW), *Piecewise Aggregate Approximation* (PAA)





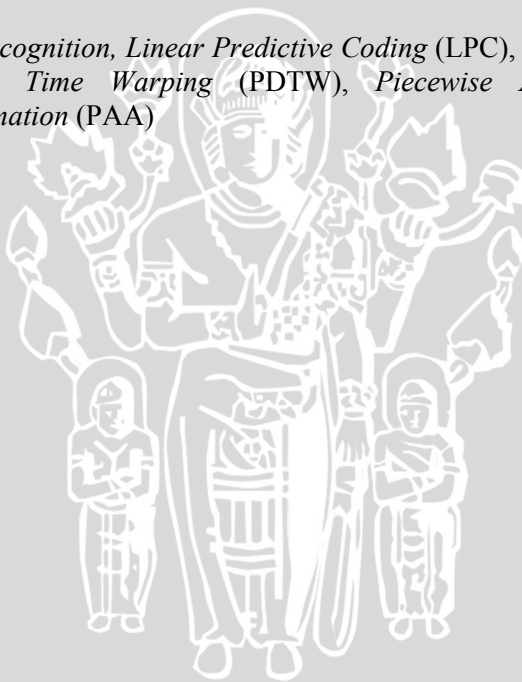
ABSTRACT

Speech recognition is a technology that uses equipment with input source is a sound, like a microphone to interpret the human voice, for transcription or as an alternative method of interaction with computers.

In the speech recognition process, to obtain information from the speech signal, feature extraction needs to be done so it can be analyzed for any variation of the existing sound signals. Extraction methods used in this research is Linear Predictive Coding (LPC), which produces cepstral coefficients of the sound patterns that are the inputs. Of coefficients is then performed recognition using Piecewise Dynamic Time Warping method (PDTW). PDTW is a modification of DTW that run Piecewise Aggregate Approximation (PAA) as an method of comparison between input data and reference data. From the comparison results will be obtained a decision that was later transformed into outputs in the form of simulated motion direction.

The Word Error Rate (WER) results of this study obtained by 48.75% for 80 data test from 4 words. From these results it can be concluded that the LPC and PDTW method can be used to identify patterns of sound with the accuracy of 51.25%.

Keywords: *Speech recognition, Linear Predictive Coding (LPC), Piecewise Dynamic Time Warping (PDTW), Piecewise Aggregate Approximation (PAA)*





KATA PENGANTAR

Alhamdulillah, puji syukur ke hadirat Allah SWT atas segala rahmat yang telah diberikan-Nya, sehingga skripsi yang berjudul “Implementasi *Linear Predictive Coding* (LPC) pada Pengenalan Pola Suara dengan Metode *Piecewise Dynamic Time Warping* (PDTW)” ini dapat diselesaikan. Skripsi ini disusun dan diajukan sebagai syarat untuk memperoleh gelar sarjana pada Program Studi Ilmu Komputer, Jurusan Matematika, Fakultas MIPA, Universitas Brawijaya.

Skripsi ini bertujuan untuk menerapkan metode *Linear Predictive Coding* (LPC) sebagai ekstraksi suara serta metode *Piecewise Dynamic Time Warping* (PDTW) sebagai pengenalan kata untuk proses pengenalan kata penunjuk arah dengan menggunakan perangkat lunak.

Dalam penyusunan dan penyelesaian skripsi ini, penulis mengucapkan terima kasih kepada:

1. Bayu Rahayudi, ST., MT selaku pembimbing utama dan Nurul Hidayat, SPd., MSc selaku pembimbing pendamping dalam penulisan skripsi. Terima kasih atas semua waktu dan bimbingan yang telah diberikan.
2. Drs. Mardji, MT selaku Ketua Program Studi Ilmu Komputer Universitas Brawijaya.
3. Reza Andria, ST selaku pembimbing akademik.
4. Dr. Abdul Rouf Alghofari, MSc selaku Ketua Jurusan Matematika Universitas Brawijaya.
5. Segenap bapak dan ibu dosen yang telah mendidik dan mengamalkan ilmunya kepada penulis selama menempuh pendidikan di Program Studi Ilmu Komputer.
6. Segenap staf dan karyawan di Jurusan Matematika FMIPA Universitas Brawijaya.
7. Abah Djaelani, Umi Endang Dwi, Mbak Asih, Mas Hakim, Ida, dan Hafidz, serta keluarga besar, atas dukungan materi dan doa restunya kepada penulis.
8. Teman dan sahabat Ilmu Komputer 2006 FMIPA UB yang telah banyak memberikan bantuan demi kelancaran penyusunan skripsi ini (Epi, Dinda, Obi, Tyas, Falen, Nafis, Wewo, Rina, Tiara, Hendra, Yuan, Wildan, Made, Marmut, Bapak, Yasser, Dika, Welly, Mbak Sulis, Nia, Gema, Puci, Nafsin, Yoga, Yossy, Oki, Isrofi, Lupi, Canggi, Wisnu, Fitri, Nata, Restu, Rado, Andika, Lutfi, Nita, Yuli, Riris, Asri, Anung, Ike, Yamin, Afif, Agung, Pakwo, dkk.).
9. Dan semua pihak yang telah membantu dalam penyusunan skripsi ini yang tidak dapat disebutkan satu per satu.

Penulis sadari bahwa masih banyak kekurangan dalam penyusunan skripsi ini. Oleh karena itu penulis mengharapkan kritik dan saran yang membangun dari pembaca.

Penulis berharap semoga skripsi ini dapat memberikan manfaat kepada pembaca serta dapat dikembangkan di masa mendatang.

Malang, Februari 2011

Penulis



DAFTAR ISI

	Halaman
HALAMAN JUDUL	i
HALAMAN PENGESAHAN	iii
HALAMAN PERNYATAAN	v
ABSTRAK	vii
ABSTRACT	ix
KATA PENGANTAR	xi
DAFTAR ISI	xiii
DAFTAR GAMBAR	xvii
DAFTAR TABEL	xix
DAFTAR SOURCECODE	xxi
 BAB I PENDAHULUAN	
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan Penelitian	2
1.4 Batasan Masalah	3
1.5 Manfaat	3
1.6 Metodologi Pemecahan Masalah	3
1.7 Sistematika Penulisan	4
 BAB II TINJAUAN PUSTAKA	
2.1 Pengenalan Pola	7
2.1.1 Pengenalan pola suara	7
2.1.2 Mekanisme produksi kata	9
2.1.3 Proses kerja alat pengenalan pola suara	10
2.2 Metode Pengenalan Suara	11
2.2.1 Penelitian terdahulu	11
2.2.1.1 Pengenalan suara menggunakan <i>Linear Predictive Coding</i> (LPC), Vektor Kuantisasi dan metode <i>Hidden Markov Model</i> (HMM)	12
2.2.1.2 Penggunaan prosesor sinyal digital keluarga TMS320 sebagai alat pengenalan suara manusia dengan algoritma DTW (<i>Dynamic Time Warping</i>)	14
2.2.2 Pencuplikan sinyal suara	15
2.2.3 Normalisasi	16
2.2.4 Ekstraksi ciri dengan <i>Linear Predictive Coding</i> (LPC)	17

2.2.4.1 <i>Preemphasis</i>	18
2.2.4.2 <i>Frame blocking</i>	19
2.2.4.3 <i>Windowing</i>	20
2.2.4.4 Analisa autokorelasi	20
2.2.4.5 Analisa LPC	21
2.2.4.6 Pengubahan parameter LPC menjadi koefisien <i>cepstral</i>	22
2.2.5 Pengenalan suara dengan <i>Piecewise Dynamic Time Warping</i> (PDTW)	22
2.2.5.1 Representasi <i>Piecewise Aggregate Approximation</i> (PAA)	23
2.2.5.2 <i>Warping</i> menggunakan representasi PAA	25
BAB III METODOLOGI DAN PERANCANGAN	
3.1 Deskripsi Sistem	29
3.1.1 Tujuan sistem	30
3.1.2 Batasan sistem	30
3.2 Desain Sistem	31
3.2.1 Data input	34
3.2.2 Pengolahan sinyal suara	34
3.2.2.1 Pemrosesan awal (<i>preprocessing</i>)	35
3.2.3 Ekstraksi ciri menggunakan metode LPC	36
3.2.3.1 <i>Preemphasize</i>	38
3.2.3.2 <i>Frame blocking</i>	39
3.2.3.3 <i>Windowing</i>	40
3.2.3.4 <i>Autocorrelation</i>	42
3.2.3.5 <i>LPC analysis</i>	43
3.2.3.6 <i>LPC parameter conversion</i>	45
3.2.4 Sistem pengenalan kata menggunakan metode <i>Piecewise Dynamic Time Warping</i> (PDTW)	47
3.2.5 Perancangan antarmuka	48
3.2.6 Perancangan uji coba	49
3.2.6.1 Skenario pengujian	49
3.3 Contoh Perhitungan Proses Pembelajaran	50
BAB IV HASIL DAN PEMBAHASAN	
4.1 Lingkungan Implementasi	59
4.1.1 Lingkungan implementasi perangkat keras	59
4.1.2 Lingkungan implementasi perangkat lunak	59

4.2 Implementasi Perangkat Lunak	59
4.2.1 Implementasi Antarmuka	60
4.2.1.1 Pemrosesan awal	60
4.2.1.2 Sistem pengenalan	61
4.2.1.2.1 <i>Training</i>	61
4.2.1.2.2 <i>Recognizing</i>	62
4.2.2 Deskripsi perangkat lunak	63
4.2.2.1 Pemrosesan awal	63
4.2.2.2 Ekstraksi ciri (LPC)	64
4.2.2.3 <i>Recognizing</i>	69
4.3 Implementasi Pengujian	71
4.4.1 Hasil Pengujian	71
4.4.2 Analisa Hasil Pengujian	76

BAB V KESIMPULAN DAN SARAN

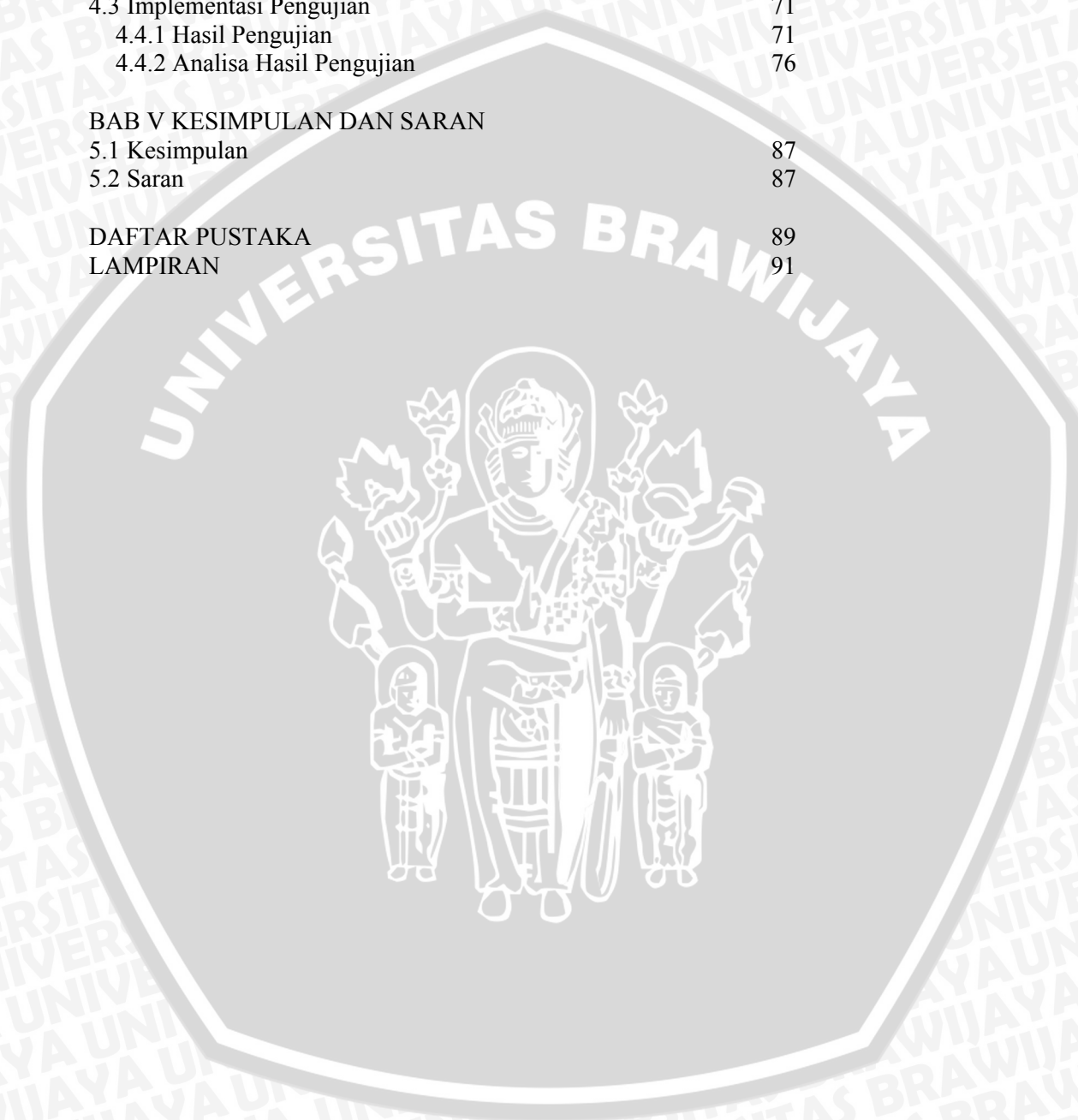
5.1 Kesimpulan	87
5.2 Saran	87

DAFTAR PUSTAKA

89

LAMPIRAN

91





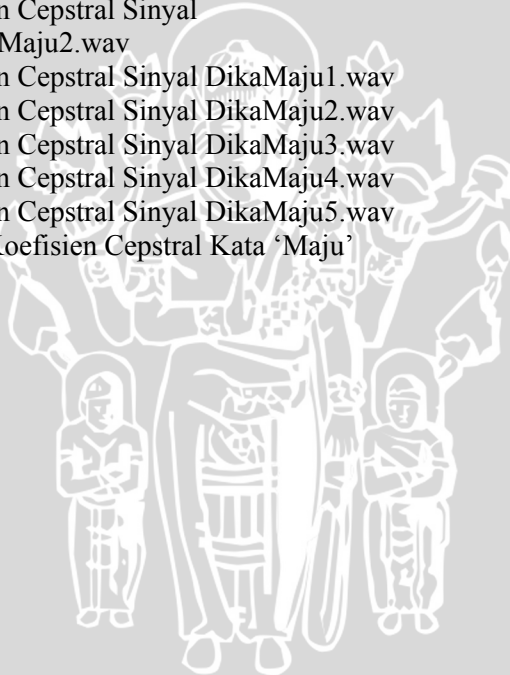
DAFTAR GAMBAR

	Halaman
Gambar 2.1	Komponen-komponen Sistem Pengenalan Pola 7
Gambar 2.2	Organ-organ Vokal 9
Gambar 2.3	Antarmuka Program <i>Speech To Text</i> saat Proses <i>Recognizing</i> 13
Gambar 2.4	Blok Diagram dari Proses LPC 18
Gambar 2.5	Perbandingan <i>Alignment</i> DTW dan PDTW 23
Gambar 2.6	Contoh Ilustrasi Teknik Pengurangan Data 24
Gambar 2.7	Rangkaian <i>Time Series X</i> dan Pendekatan <i>Piecewise Aggregate X'</i> 24
Gambar 2.8	Contoh <i>Warping Path</i> 26
Gambar 3.1	Langkah Pembuatan Perangkat Lunak 29
Gambar 3.2	Diagram Blok Pengenalan Suara secara Umum 30
Gambar 3.3	Diagram Blok Penyimpanan Data Training 31
Gambar 3.4	Diagram Blok Sistem Pengenalan Kata 32
Gambar 3.5	Diagram Alir Penyimpanan Data Training 32
Gambar 3.6	Diagram Alir Sistem Pengenalan Kata 32
Gambar 3.7	Diagram Alir Ekstraksi Ciri 37
Gambar 3.8	Diagram Alir Proses <i>Preemphasize</i> 38
Gambar 3.9	Diagram Alir Proses Frame Blocking 40
Gambar 3.10	Diagram Alir Proses Windowing 41
Gambar 3.11	Diagram Alir Proses Autokorelasi 43
Gambar 3.12	Diagram Alir Analisa LPC 44
Gambar 3.13	Diagram Alir Konversi Parameter LPC 46
Gambar 3.14	Diagram Alir Rekursi PDTW 47
Gambar 3.15	Antarmuka Program 49
Gambar 3.16	Contoh Signal Input 50
Gambar 4.1	Tampilan Utama 61
Gambar 4.2	Tampilan Sinyal Suara 61
Gambar 4.3	Tampilan Pemrosesan Sinyal 62
Gambar 4.4	Tampilan Penyimpanan Data Training 62
Gambar 4.5	Tampilan Proses Testing 63



DAFTAR TABEL

	Halaman
Tabel 2.1	Nilai Parameter Analisa LPC yang Sering Digunakan pada Pengenalan Suara
Tabel 3.1	Contoh Tabel Pengujian
Tabel 3.2	Contoh <i>Signal Input</i>
Tabel 3.3	Hasil Perhitungan <i>Preprocessing</i>
Tabel 3.4	Hasil Perhitungan <i>Preemphasize</i>
Tabel 3.5	Hasil Perhitungan <i>Frame Blocking</i>
	Hasil Perhitungan Hamming Window
Tabel 3.6	Hasil Perhitungan <i>Windowing</i>
Tabel 3.7	Hasil Perhitungan Koefisien Autokorelasi
Tabel 3.8	Koefisien LPC
Tabel 3.9	Koefisien <i>Cepstral</i>
Tabel 4.1	Hasil Pengujian Kata Maju
Tabel 4.2	Hasil Pengujian Kata Mundur
Tabel 4.3	Hasil Pengujian Kata Kanan
Tabel 4.4	Hasil Pengujian Kata Kiri
Tabel 4.5	Koefisien Cepstral Sinyal TrainingMaju1.wav
Tabel 4.6	Koefisien Cepstral Sinyal TrainingMaju2.wav
Tabel 4.7	Koefisien Cepstral Sinyal DikaMaju1.wav
Tabel 4.8	Koefisien Cepstral Sinyal DikaMaju2.wav
Tabel 4.9	Koefisien Cepstral Sinyal DikaMaju3.wav
Tabel 4.10	Koefisien Cepstral Sinyal DikaMaju4.wav
Tabel 4.11	Koefisien Cepstral Sinyal DikaMaju5.wav
Tabel 4.12	Selisih Koefisien Cepstral Kata 'Maju'





DAFTAR SOURCECODE

	Halaman
<i>Sourcecode 4.1</i>	Proses Normalisasi 64
<i>Sourcecode 4.2</i>	Kode <i>Preemphasis</i> 65
<i>Sourcecode 4.3</i>	Kode <i>Frame Blocking</i> 66
<i>Sourcecode 4.4</i>	Kode <i>Windowing</i> 66
<i>Sourcecode 4.5</i>	Kode Analisa Autokorelasi 67
<i>Sourcecode 4.6</i>	Kode <i>Levinson-Durbin</i> 68
<i>Sourcecode 4.7</i>	Kode Konversi Koefisien Cepstral 69
<i>Sourcecode 4.8</i>	<i>Piecewise Dynamic Time Warping</i> 71





BAB I PENDAHULUAN

1.1 Latar Belakang

Pengenalan pola merupakan bidang dalam pembelajaran mesin dan dapat diartikan sebagai tindakan mengambil data mentah dan bertindak berdasarkan klasifikasi data. Salah satu aplikasinya adalah pengenalan suara (*speech recognition*), yang merupakan teknologi yang menggunakan peralatan dengan sumber masukannya adalah suara, seperti mikropon untuk menginterpretasikan suara manusia, untuk transkripsi atau sebagai metode alternatif interaksi dengan komputer (Irfani, 2006).

Banyak aplikasi yang dikembangkan menggunakan pengenalan pola suara, antara lain di bidang kesehatan terdapat *Medical Transcriptionist* (MT), di bidang militer terdapat *High-performance Fighter aircraft*, *Training air traffic controllers*, sampai pada alat yang membantu seseorang yang memiliki kesulitan dalam menggunakan tangan, maka diciptakannya komputer yang dapat dioperasikan menggunakan deteksi pengucapan *user*.

Pada proses pengenalan suara, untuk mendapatkan informasi dari sinyal suara perlu dilakukan ekstraksi ciri sehingga dapat dianalisis untuk setiap variasi sinyal suara yang ada. Dari ciri yang ada tersebut kemudian dikenali dan diubah menjadi *output* yang diinginkan. Dalam proses pengenalan diperlukan suatu metode pemodelan yang dapat mendukung proses pengenalan sehingga tidak terjadi kesalahan pengertian.

Penelitian dalam bidang pengenalan suara dengan menggunakan metode *Linear Predictive Coding* (LPC) sebagai ekstraksi ciri dan *Dynamic Time Warping* (DTW) sebagai metode pengenalan pernah dilakukan oleh Frederikus David (1996) dengan mengenali pengucapan angka 1, 2, 3, dan 5. Penelitian serupa juga dilakukan oleh Ari Rachmawati (2009) pada database *diphone* dengan struktur KV (konsonan-vokal), akan tetapi metode pengenalan yang digunakan adalah metode *Hidden Markov Models* (HMM). Dari penelitian perbandingan HMM dan DTW yang dilakukan oleh Levy et al menunjukkan bahwa kompleksitas yang dimiliki metode HMM sebagai pembaharuan pengenalan DTW secara signifikan lebih tinggi, serta dalam pelatihan pengenalan suara HMM menunjukkan hasil yang lebih efisien pada kondisi yang sama.

Baik metode DTW maupun HMM dapat bekerja dengan baik pada penelitian dengan data *clustering time series*. DTW menghasilkan inisial *clustering* dalam bentuk garis-garis besar dan HMM menghapus rangkaian dari *clusters* yang seharusnya tidak termasuk didalamnya. Namun

kekurangan HMM adalah saat menghapus rangkaian yang dianggap buruk, beberapa rangkaian yang bagus ikut terbuang (Oates, 2007). Sehingga pada skripsi ini metode yang digunakan merupakan salah satu modifikasi dari DTW yang memiliki performa paling baik menurut penelitian yang dilakukan oleh Helwig (2009) yang dapat bekerja lebih efisien lagi dengan intensitas-normalisasi, yaitu *Piecewise Dynamic Time Warping* (PDTW). Dari metode modifikasi DTW ini diharapkan juga dapat menutup kekurangan dari metode HMM dengan menghapus rangkaian *clusters* lebih teliti. Metode PDTW ini digunakan untuk proses pengenalan ciri suara yang telah ditemukan oleh metode LPC sehingga akan dihasilkan pengambilan keputusan kata yang paling tepat. Sedangkan untuk data penelitian pada skripsi ini dikhususkan pada penggunaan kata-kata penunjuk arah.

1.2 Rumusan Masalah

Rumusan masalah yang akan dijadikan obyek penelitian, antara lain:

1. Bagaimana implementasi metode *Linear Predictive Coding* (LPC) sebagai ekstraksi ciri dan *Piecewise Dynamic Time Warping* (PDTW) sebagai pengenalan kata pada pengenalan pola suara.
2. Bagaimana keakuratan sistem dalam mengenali kata yang di-input-kan, yang dihasilkan metode *Linear Predictive Coding* (LPC) sebagai ekstraksi ciri dan *Piecewise Dynamic Time Warping* (PDTW) sebagai pengenalan kata.

1.3 Tujuan Penelitian

Sesuai dengan rumusan masalah tersebut, maka tujuan penelitian yang ingin dicapai dalam pembuatan skripsi ini adalah sebagai berikut:

1. Mengimplementasikan metode *Linear Predictive Coding* (LPC) sebagai ekstraksi ciri dan *Piecewise Dynamic Time Warping* (PDTW) sebagai pengenalan kata pada pengenalan pola suara.
2. Mengetahui keakuratan sistem dalam mengenali kata yang di-input-kan, yang dihasilkan metode *Linear Predictive Coding* (LPC) sebagai ekstraksi ciri dan *Piecewise Dynamic Time Warping* (PDTW) sebagai pengenalan kata.

1.4 Batasan Masalah

Ruang lingkup yang membatasi permasalahan yang akan dibahas pada skripsi ini antara lain:

1. Masukan perangkat lunak yang akan dikenali berupa kata penunjuk arah dalam Bahasa Indonesia yang telah direkam dan disimpan dalam format file .WAV 8KHz mono 8 bit, antara lain: kata MAJU, MUNDUR, KANAN, dan KIRI.
2. *Input* kata yang dapat dikenali hanya satu kata dalam satu perintah.
3. *Input* sinyal suara melalui mikropon dengan *background-noise* seminimal mungkin.
4. Pengambilan sampel suara dilakukan dari 4 orang laki-laki yang berbeda dengan kriteria umur yang setara.

1.5 Manfaat

Manfaat yang dapat diambil dari penulisan skripsi ini adalah menghasilkan suatu sistem pengenalan pola suara dengan penerapan metode *Linear Predictive Coding* (LPC) sebagai ekstraksi ciri pola suara dan metode *Piecewise Dynamic Time Warping* (PDTW) sebagai pengenalan kata, yang diimplementasikan pada simulasi kontrol gerak arah.

1.6 Metodologi Pemecahan Masalah

Untuk mencapai tujuan yang dirumuskan sebelumnya, maka metodologi yang digunakan dalam penulisan skripsi ini adalah:

1. Studi literatur
Mempelajari teori-teori dan konsep-konsep dasar serta identifikasi masalah yang berhubungan dengan pengenalan pola suara, khususnya mengenai metode *Linear Predictive Coding* (LPC) dan *Piecewise Dynamic Time Warping* (PDTW) melalui pustaka baik berupa buku, jurnal ilmiah maupun melalui internet.
2. Pendefinisian dan analisis masalah
Mendefinisikan dan menganalisis masalah untuk mencari solusi yang tepat
3. Perancangan dan implementasi sistem
Membuat perancangan perangkat lunak dan mengimplementasikan hasil rancangan tersebut, yaitu membuat perangkat lunak sistem pengenalan pola suara
4. Uji coba dan analisis hasil implementasi
Menguji perangkat lunak, dan menganalisis hasil dari implementasi tersebut apakah telah sesuai dengan tujuan yang dirumuskan sebelumnya, untuk kemudian dievaluasi dan disempurnakan.

1.7 Sistematika Penulisan

Pembuatan skripsi ini dilakukan dengan pembagian bab sebagai berikut:

1. BAB I PENDAHULUAN

Bab I membahas mengenai latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, serta sistematika penulisan skripsi.

2. BAB II TINJAUAN PUSTAKA

Bab II mencantumkan beberapa tinjauan pustaka yang berkaitan dengan penelitian, diantaranya pengenalan suara (*speech recognition*), metode *Linear Predictive Coding* (LPC), dan metode *Piecewise Dynamic Time Warping* (PDTW).

3. BAB III METODOLOGI DAN PERANCANGAN

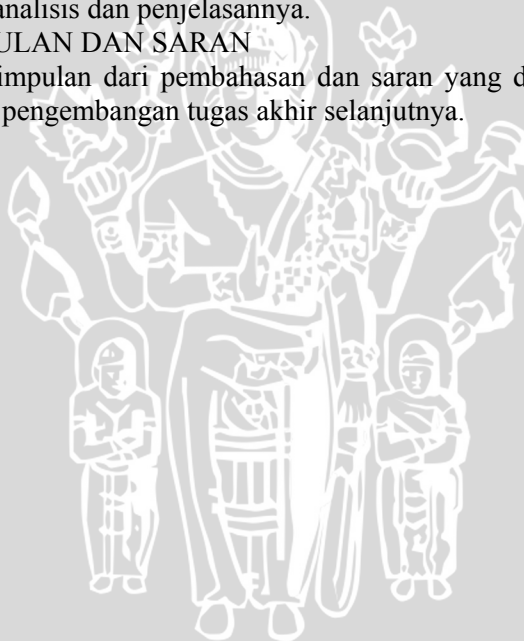
Bab III menerangkan beberapa hal mengenai metode dan perancangan yang akan dilakukan dalam membangun sistem ini. Termasuk kebutuhan sistem, metode dan langkah yang digunakan dalam pengolahan sinyal dan pengenalan suara dengan metode *Piecewise Dynamic Time Warping* (PDTW). Beberapa poin yang dibahas ialah deskripsi sistem, desain sistem, perancangan *interface*, dan perancangan uji coba.

4. BAB IV HASIL DAN PEMBAHASAN

Bab IV menerangkan proses implementasi dari rancangan penelitian yang dijelaskan pada bab III. Implementasi yang dijelaskan terdiri dari implementasi program. Selain itu, bab IV juga menjelaskan penerapan aplikasi, uji coba dan analisa hasil percobaan mengenai tingkat kesalahan berdasarkan perbedaan *signal noise* yang digunakan untuk pengenalan serta analisis dan penjelasannya.

5. BAB V KESIMPULAN DAN SARAN

Bab V berisi kesimpulan dari pembahasan dan saran yang diharapkan bermanfaat untuk pengembangan tugas akhir selanjutnya.

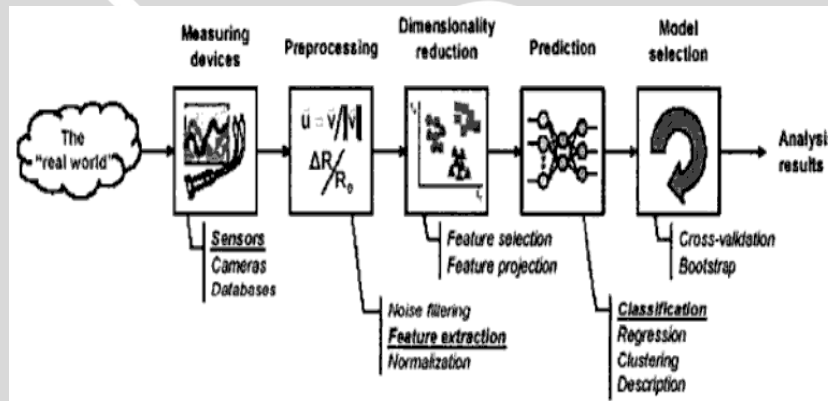


BAB II TINJAUAN PUSTAKA

2.1 Pengenalan Pola

Pengenalan obyek merupakan kegiatan sehari-hari pada makhluk hidup dan telah menjadi kemampuan bawaan serta kebutuhan untuk berhubungan dengan lingkungan. Pembangunan metode yang mampu menyaingi bentuk yang paling bervariasi dari pengenalan obyek telah dikembangkan seiring dengan kebutuhan untuk membangun sebuah sistem “kecerdasan” yang terotomasi. Pada sistem ini obyek-obyek direpresentasikan pada cara yang paling cocok untuk jenis pemrosesan yang menjadi subyek. Representasi tersebut dikenal dengan istilah *patterns* atau pola (Marques de Sa, 2001).

Menurut Al Fatta (2009), pengenalan pola adalah salah satu cabang dari bidang kecerdasan buatan yang menitik-beratkan pada metode pengklasifikasian obyek ke dalam kelas-kelas tertentu untuk menyelesaikan masalah tertentu. Pengenalan pola dapat didefinisikan juga sebagai disiplin ilmiah yang tujuannya mengklasifikasi obyek-obyek ke dalam sejumlah kategori atau kelas. Tergantung pada aplikasi yang digunakan, obyek-obyek tersebut dapat berupa gambar atau bentuk gelombang sinyal atau beberapa jenis bentuk lain yang dibutuhkan untuk diklasifikasikan (Theodoridis, 2006)



Gambar 2.1 Komponen-komponen Sistem Pengenalan Pola
Sumber: Al Fatta, 2009.

Pada Gambar 2.1 menunjukkan komponen-komponen sistem pengenalan pola menurut Al Fatta (2009). Komponen sistem tersebut terdiri dari:

1. Sensor
Sensor digunakan untuk menangkap obyek yang ciri atau fiturnya akan diekstraksi.
2. Mekanisme *pre-processing*
Mekanisme pengolahan obyek yang ditangkap oleh sensor biasanya digunakan untuk mengurangi kompleksitas ciri yang akan dipakai untuk proses klasifikasi.
3. Mekanisme pencari fitur (manual/otomatis)
Bagian ini digunakan untuk mengekstraksi ciri yang telah melalui tahapan *preprocessing* untuk memisahkannya dari fitur-fitur pada obyek yang tidak diperlukan dalam proses klasifikasi.
4. Algoritma pemilah
Pada tahapan ini proses klasifikasi dilakukan dengan menggunakan algoritma klasifikasi tertentu. Hasil dari tahapan ini adalah klasifikasi dari obyek yang ditangkap ke dalam kriteria-kriteria yang telah ditentukan.

Jangkauan dari aplikasi-aplikasi sistem pengenalan pola cukup beragam, diantaranya: *voice recognition*, *fingerprint identification*, *face identification*, *handwriting identification*, *optical character recognition* (OCR) dan *robot vision* (Al Fatta, 2009).

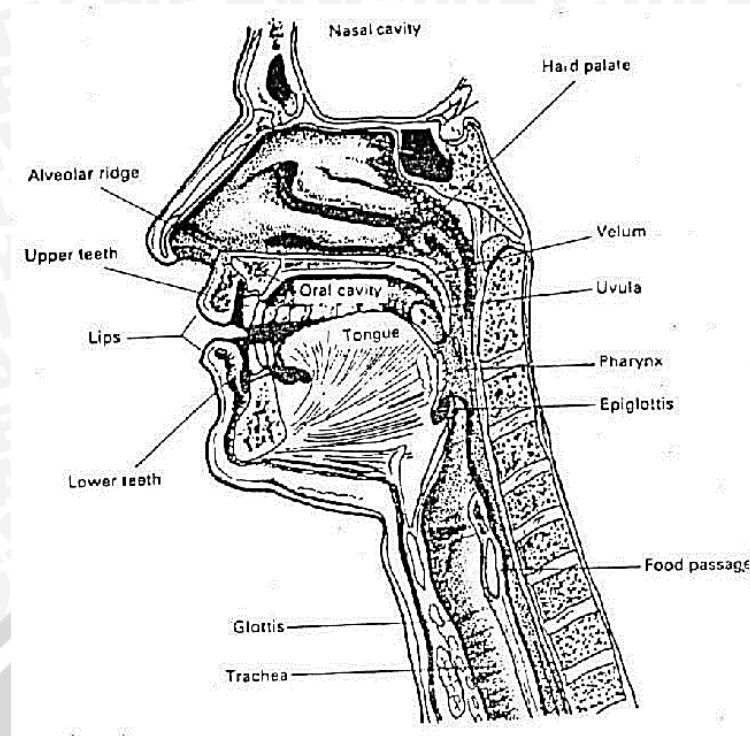
2.1.1 Pengenalan pola suara

Teknologi pengenalan suara adalah teknologi yang menggunakan peralatan dengan sumber masukannya adalah suara, seperti mikropn untuk menginterpretasikan suara manusia untuk transkripsi atau sebagai metode alternatif interaksi dengan komputer (Irfani, 2006).

Bentuk gelombang suara yang diberikan menjadi sebuah tugas dari sistem pengenalan suara untuk menghasilkan perhitungan dari *string* kata yang dihubungkan dengan bentuk gelombang tersebut.

2.1.2 Mekanisme produksi kata

Gambar organ-organ vokal seperti terlihat pada Gambar 2.2 secara garis besar terdiri dari paru-paru, batang tenggorok (*trachea*), pangkal tenggorok (*larynx*), jalur mulut (*oral track*) dan jalur hidung (*nasal track*).



Gambar 2.2 Organ-organ Vokal
Sumber: Owen, 1993.

Pangkal tenggorok terdiri dari dua lipatan kulit yaitu sepasang pita suara (*vocal cord*) yang dapat bergerak memisah maupun mengembang secara berulang-ulang apabila terdapat udara yang melewati celah diantaranya yang sering disebut sebagai celah suara (*glottis*).

Jalur mulut (*oral track*) adalah tabung akustik tak seragam kira-kira panjangnya 17 cm pada pria dewasa, ujung akhir berupa bibir dan ujung awal berupa pita suara atau pangkal tenggorok. Luas daerah irisan dapat bervariasi dari 0 sampai 20 cm² dengan kontrol otot dari pengartikulasi kata yaitu bibir, lidah, rahang, dan langit-langit lunak.

Jalur hidung (*nasal track*) juga berupa tabung akustik tak seragam dengan luasan dan panjang yang tetap (kurang lebih 12 cm pada pria dewasa), ujung akhir pada lubang hidung dan ujung awal pada sirip tergerakkan yang disebut langit-langit lunak (*velum*) yang berfungsi untuk mengontrol kopling akustik antara jalur mulut dan jalur hidung.

Saat produksi bunyi yang tidak sengau, velum menutup jalur hidung sehingga hanya ada satu jalur transmisi bunyi melalui bibir. Saat produksi bunyi sengau, velum secara akustik mengkopel jalur hidung dan jalur mulut, tetapi ujung akhir jalur mulut (bibir) tertutup sehingga hanya ada satu jalur transmisi bunyi melalui lubang hidung.

Pada saat berbicara, paru-paru terisi udara oleh pengembangan otot pada rongga dada dan tarikan diafragma. Saat rongga dada berkontraksi, udara keluar sepanjang batang tenggorok dan melalui celah suara (*glottis*). Aliran udara ini merupakan energi untuk produksi kata yang dapat dikontrol untuk menghasilkan bermacam-macam bunyi yang memiliki arti (kata/*speech*).

Bunyi bermakna (selanjutnya disebut kata) dapat dibagi menjadi tiga bagian besar berdasarkan mode eksitasinya:

1. *Voiced sounds*, sebagai contoh adalah bunyi pada kata “ah” atau “oh” saat pita suara bergetar pada frekuensi tertentu dengan jangkauan pitch dari *voiced sound* ini kurang lebih 50 Hz sampai dengan 250 Hz dengan rata-rata 120 Hz pada pria dewasa.
2. *Unvoiced sounds*, yaitu saat pita suara tidak bergetar sama sekali dan terdengar seperti bunyi pada kata “shh”.
3. *Plosive sounds*, misalnya bunyi “pah” pada ucapan kata “pagi” atau “deh” pada ucapan kata “dari”. Pada saat pita suara tertutup untuk sesaat dan kemudian tiba-tiba membuka.

2.1.3 Proses kerja alat pengenalan pola suara

Seperti yang disebutkan Cook (2002), sistem pengenalan dapat dipecah menjadi langkah-langkah sebagai berikut:

1. Perekaman suara dan pendeteksian ucapan
Perekaman suara dan pendeteksian ucapan dapat dipenuhi dengan berbagai cara. Menentukan titik awal dapat dilakukan dengan membandingkan *level* di sekitar suara dengan contoh pada rekaman suara. Sedangkan pendeteksian titik akhir lebih sulit dikarenakan pembicara cenderung meninggalkan “sesuatu” termasuk menarik/menghembuskan nafas, gemeretak gigi, dan gema.
2. *Prefiltering* (*preemphasis*, normalisasi, penjilidan, dan lain-lain)
Prefiltering dapat dilakukan dalam berbagai cara, tergantung pada ciri-ciri lain dari sistem pengenalan. Metode yang paling umum digunakan adalah metode “*Bank-of-Filters*” yang memanfaatkan rangkaian dari saringan suara untuk menyiapkan sampel, dan metode *Linear Predictive Coding* (LPC) yang digunakan sebagai fungsi prediksi untuk menghitung selisih (*error*).

3. *Framing* dan *windowing* (mencancang data menjadi *format* yang dapat dipergunakan)
Framing dan *windowing* melibatkan bagian-bagian sampel data yang terpisah menjadi ukuran yang lebih spesifik. Langkah ini juga menyiapkan batasan-batasan sampel yang akan digunakan untuk analisis (membuang suara sampling, dan lain-lain).
4. *Filtering* (*filtering* lebih lanjut dari masing-masing *window/ frame/pita* frekuensi)
Langkah ini merupakan persiapan terakhir untuk masing-masing *window* sebelum perbandingan dan pencocokan pola suara. Langkah ini terdiri dari penjajaran waktu dan normalisasi.
5. Perbandingan dan pencocokan (*comparison and matching*)
Perbandingan melibatkan aliran *window* dengan sampel yang diketahui. Pada langkah ini terdapat metode algoritma pengenalan, analisis frekuensi, analisis diferensial, teknik/ *shortcut* aljabar linier, distorsi spektral, dan metode distorsi waktu. Semua metode ini digunakan untuk menghasilkan kemungkinan dan pencocokan yang akurat.
6. *Action* (melakukan fungsi yang terkait dengan pola yang dikenali)
Actions dapat berbentuk apa saja yang *developer* inginkan.

2.2 Metode Pengenalan Suara

2.2.1 Penelitian terdahulu

Skripsi ini menggunakan dua penelitian terdahulu sebagai referensi utama, yaitu penelitian berjudul Pengenalan Suara Menggunakan *Linear Predictive Coding* (LPC), Vektor Kuantisasi dan Metode *Hidden Markov Model* (HMM) yang dilakukan oleh Ary Rachmawati (2009) sebagai referensi ekstraksi ciri menggunakan LPC, dan penelitian kedua berjudul Penggunaan Prosesor Sinyal Digital Keluarga TMS320 sebagai Alat Pengenalan Suara Manusia dengan Algoritma DTW (*Dynamic Time Warping*) yang dilakukan oleh Frederikus David (1996) sebagai referensi pengenalan kata.

2.2.1.1 Pengenalan suara menggunakan *Linear Predictive Coding* (LPC), Vektor Kuantisasi dan metode *Hidden Markov Model* (HMM)

Penelitian ini dilakukan oleh Ary Rachmawati (2009), yang menghasilkan vektor ciri sinyal suara menggunakan metode *Linear Predictive Coding* (LPC), yang diikuti dengan vektor kuantisasi. setelah

vektor dikuantisasi kemudian dikenali dengan teknik pemodelan yang disebut *Hidden Markov Model* (HMM). HMM yaitu suatu teknik yang digunakan untuk mengetahui probabilitas suatu keadaan yang belum diketahui urutannya. HMM adalah sebuah proses stokastik rangkap yang menghasilkan sebuah urutan simbol observasi. Pada tahap ini pengenalan ini digunakan algoritma *baum-welch*.

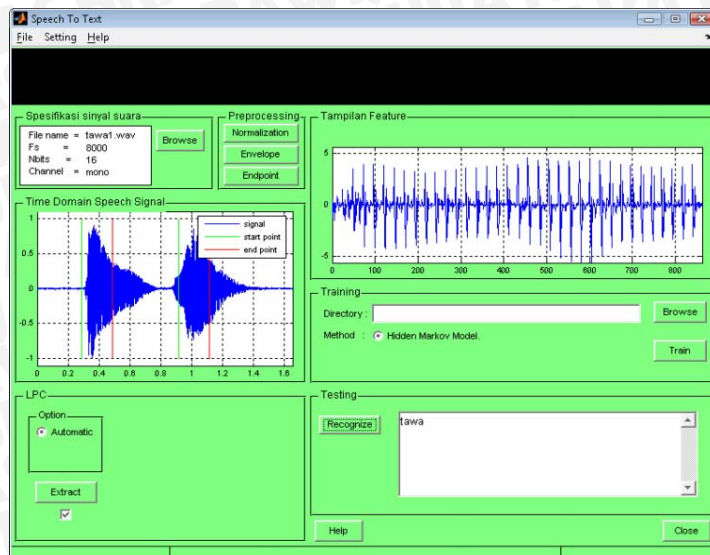
Untuk unit pengenalan digunakan *database diphone* sederhana yang berstruktur KV (konsonan-vokal).

Pada penelitian ini, proses ekstraksi ciri menggunakan metode LPC (*Linear Predictive Coding*), dan untuk pengenalan kata menggunakan metode *Hidden Markov Model* yang merupakan suatu rantai *Markov* dimana simbol keluaran atau fungsi peluang yang menggambarkan simbol keluaran berhubungan dengan *state* atau transisi antar *state*. Observasi tiap *state* digambarkan secara terpisah dengan suatu fungsi probabilitas yang didefinisikan sebagai peluang untuk menghasilkan simbol tertentu saat terjadi transisi antar *state*.

Penelitian yang disusun ini bertujuan untuk merancang dan melakukan implementasi program aplikasi *speech-to-text* yang berfungsi untuk mengubah sinyal suara menjadi representasi tekstual, serta menganalisa pengaruh parameter-parameter HMM terhadap tingkat akurasi sistem pengenalan suara.

Ruang lingkup yang membatasi permasalahan pada penelitian ini antara lain:

- Pembahasan dibatasi pada proses pembelajaran dan pengenalan suara dengan menggunakan metode *Hidden Markov Model*.
- Metode HMM digunakan untuk mencari indeks dari sinyal referensi yang mempunyai *likelihood* paling besar.
- Masukan perangkat lunak yang akan dikenali berupa kata dasar dalam Bahasa Indonesia yang telah direkam dan disimpan dalam format file *.WAV*, dan berstruktur KV-KV.
- *Database* yang digunakan adalah *database diphone* dengan struktur konsonan-vokal (KV) dari sampel 3 orang laki-laki.



Gambar 2.3 Antarmuka Program *Speech To Text* saat Proses *Recognizing*
Sumber: Rachmawati, Ari. 2009.

Gambar 2.3 menunjukkan antarmuka program *speech to text* yang menunjukkan proses *Recognizing*.

Berdasarkan uji coba yang dilakukan terhadap 12 kata yang diucapkan oleh 3 orang laki-laki, dihasilkan bahwa *codebook* optimal terdapat pada ukuran 128. Sedangkan pada *state* 6 nilai *Word Error Rate* (WER) adalah 33.33%, *state* 7 nilai WER 50%, *state* 8 nilai WER 66.67%, *state* 9 nilai WER 41,67%, dan pada *state* 10 didapat nilai WER 41,67%.

Nilai *Word Error Rate* (WER) di sini merupakan sebuah matriks umum untuk mengukur performansi sistem pengenalan suara. Adapun perhitungannya dapat dituliskan pada persamaan 2.1.

$$WER = \frac{\text{Banyak kata yang dikenali salah}}{\text{Total kata yang diujikan}} \quad (2.1)$$

Hasil uji coba jumlah *state* terhadap keakurasian sistem dapat disimpulkan bahwa semakin besar jumlah *state* tidak menjamin dapat meningkatkan keakurasian sistem.

2.2.1.2 Penggunaan prosesor sinyal digital keluarga TMS320 sebagai alat pengenalan suara manusia dengan algoritma DTW (*Dynamic Time Warping*)

Penelitian ini dilakukan oleh Frederikus David (1996), yaitu dibuatnya suatu alat pengenalan suara manusia dengan bantuan prosesor sinyal *digital* keluaran Texas Instruments keluarga TMS320. Permasalahan dibatasi pada pengenalan suara pembicaraan yang merupakan sepatah kata (*isolated word*) yang berasal dari orang tertentu saja (*speaker dependence*) dan memiliki perbendaharaan kata kecil (*small vocabulary*).

Tujuan penelitian ini adalah pengimplementasian tugas pengenalan suara otomatis menggunakan pendekatan pengenalan pola menggunakan *Dynamic Time Warping*, dengan pemberdayaan prosesor sinyal digital keluarga TMS320 dari Texas Instrument.

Pemrosesan ekstraksi ciri (*feature*) pada sistem dilakukan dengan TMS320C50 DSK (*digital signal processing starter kit*) untuk mendapatkan pengolahan yang *real time*, sedangkan proses pengenalan pola digunakan komputer IBM PC/AT untuk kemudahannya.

Ruang lingkup pengenalan suara manusia pada penelitian ini meliputi tiga hal, yaitu:

- Pengenalan suara manusia bergantung pembicara, yaitu dapat mengenali kata yang diucapkan oleh pembicara A tetapi tidak dapat mengenali kata yang sama yang diucapkan pembicara B.
- Pengenalan suara manusia terisolasi antar kata, yaitu sistem dapat mengenali kata jika antara kata yang satu dengan kata yang lain dibatasi oleh suatu *'pause'* atau diam sesaat.
- Pengenalan suara manusia berdasarkan pengenalan pola, yaitu sistem ini dapat melakukan pengenalan suara dengan menggunakan teknik membandingkan antara pola yang dikenal dengan pola yang tak dikenal.
- Pengenalan suara manusia dengan perbendaharaan kata kecil, yaitu sistem ini hanya mampu mengenali pola kata sesuai dengan pola referensi yang telah dilatihkan yaitu tidak lebih dari 1000 pola kata.

Metode yang digunakan untuk proses ekstraksi ciri adalah metode analisa LPC (*Linear Predictive Coding*) dimana perhitungan cuplikan data menuju ke vektor autokorelasi memakai algoritma Durbin, yang selanjutnya diturunkan menjadi vektor ciri yaitu koefisien prediksi, koefisien PARCOR, dan kesalahan prediksi. Dilanjutkan dengan proses perbandingan dengan menggunakan algoritma *Dynamic Time Warping*. Pada akhirnya diambil suatu keputusan yang menyatakan sampel suara tersebut dikenal atau tidak.

Kemampuan sistem pada penelitian ini terbatas pada kemampuan TMS320C50 yang merupakan prosesor *fixed point* sehingga dilakukan

emulasi *floating point* secara perangkat lunak, dan ini mempengaruhi performansi pengenalan sistem. Sistem yang didapat berkemampuan mengenali kata yang terisolasi diucapkan oleh seorang pembicara dengan 1 perbendaharaan kata berdurasi maksimum 3 detik secara baik untuk lingkungan sekitar yang bebas *noise* dimana keseluruhan proses memakan waktu sekitar 7,5 detik.

2.2.2 Pencuplikan sinyal suara

Cara kerja sistem pengenalan suara secara garis besar ialah mula-mula sinyal suara yang diterima dengan menggunakan mikropon (berupa sinyal *analog*, yaitu sinyal kontinyu dalam domain waktu dan *amplitude*) dicuplik sehingga menjadi sinyal *digital* (diskrit) dengan bantuan *sound card* pada PC.

Sinyal *digital* hasil cuplikan ini terlebih dahulu dinormalisasi kemudian diproses awal menggunakan metode *Linear Predictive Coding* (LPC) sehingga didapat beberapa koefisien LPC yang merupakan *feature* (ciri) dari suara pembicaraan. Sedangkan tujuan dari normalisasi itu sendiri untuk mendapatkan sinyal dengan ukuran yang sama walaupun kata yang diucapkan berbeda.

Untuk mengubah sinyal *analog* menjadi sinyal *digital* diperlukan proses pencuplikan, kuantisasi, dan pengkodean.

1. Pencuplikan

Proses ini merupakan penentuan jumlah sampel dalam satu detik. Jika pencuplikan dilakukan dengan frekuensi cuplik 8000Hz, maka dalam satu detik terdapat 8000 sampel. Perlu diperhatikan komponen utama frekuensi sinyal suara berada pada kisaran 300-3400Hz. Menurut Nyquist frekuensi sampling dalam pencuplikan harus lebih besar dari 2 kali frekuensi sinyal aslinya. Sesuai dengan persamaan Nyquist, $f_s \geq 2f_h$ dimana $f_h = f_{intertinggi}$. Semakin tinggi frekuensi sampling, maka sinyal *digital* yang dihasilkan semakin bagus.

2. Kuantisasi

Proses ini membatasi amplitudo atau nilai *axis* sinyal. Jika sinyal dicuplik dengan resolusi 8 bit, maka terdapat 2^8 atau 256 nilai batas sinyal.

3. Pengkodean

Karena memori komputer menyimpan data berupa tipe data biner, maka nilai amplitudo tiap sampel sinyal akan dikonversi ke dalam bentuk biner. Jika sinyal dicuplik dengan resolusi 8 bit, maka nilai amplitudo akan disimpan pada ukuran 8 bit kode biner atau 1 *byte* data.

2.2.3 Normalisasi

Proses normalisasi bertujuan untuk menyamakan amplitudo maksimum sinyal suara sehingga tidak ada pengaruh perubahan amplitudo pada pemrosesan berikutnya.

Cara kerja normalisasi yaitu dengan menambahkan beberapa data tambahan apabila data hasil pencuplikan belum memenuhi jumlah yang dibutuhkan atau dengan mengurangi jumlah data hasil pencuplikan apabila melebihi jumlah input yang dibutuhkan.

Normalisasi dilakukan dengan membagi masing-masing nilai amplitudo diskrit ($x(n)$) dengan nilai amplitudo maksimumnya ($\max(|x|)$). Fungsi normalisasi dapat didefinisikan pada persamaan 2.2. Dimana n merupakan indeks data amplitudo sinyal suara.

$$x(n)_{\text{ternormalisasi}} = \frac{x(n)}{\max(|x|)} \quad (2.2)$$

Proses normalisasi dilakukan dengan kondisi pusat letak distribusi amplitudo diskrit berada pada sumbu $y = 0$. Jika $y \neq 0$, maka perlu dilakukan proses *centering*, yaitu proses yang berfungsi untuk membuat amplitudo sinyal suara mempunyai rata-rata sama dengan nol. Fungsi *centering* dapat didefinisikan pada persamaan 2.3 (Rachmawati, 2009). Dimana n merupakan indeks data amplitudo sinyal suara.

$$x(n)_{\text{center}} = x(n) - \bar{x} \quad (2.3)$$

2.2.4 Ekstraksi ciri dengan Linear Predictive Coding (LPC)

Ciri sinyal ucapan sangat berguna pada sistem pengenalan suara. Salah satu metode yang paling umum digunakan untuk proses ekstraksi ciri adalah LPC. Analisis prediksi *linear* adalah suatu cara yang digunakan untuk mendapatkan sebuah pendekatan sinyal suara. Peramalan *linear* secara khusus merupakan metode yang cocok dalam pengolahan sinyal suara. Metode ini dapat juga diterapkan dalam pengenalan kata. Tujuan dari digunakannya metode ini adalah untuk mencari nilai koefisien LPC dari suatu sinyal. Koefisien LPC dapat digunakan sebagai vektor yang utama pada taraf *frame* pencocokan dalam pengenalan.

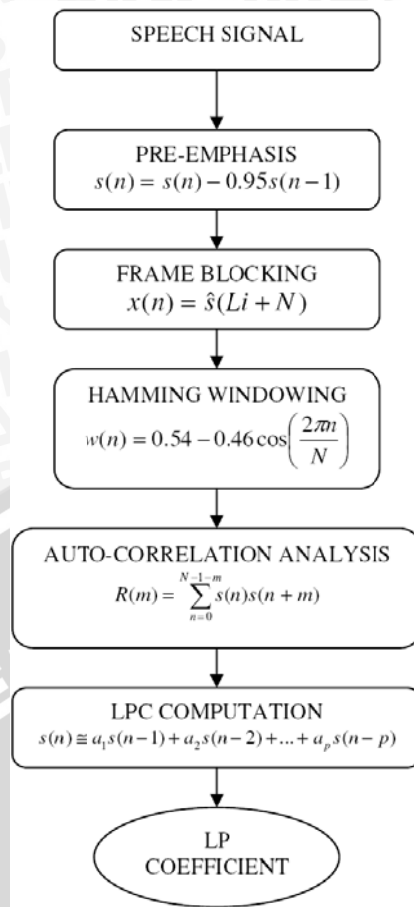
Prinsip dasar dari ekstraksi ciri sinyal dengan menggunakan LPC adalah bahwa contoh sinyal ucapan $s(n)$ pada waktu ke- n dapat diperkirakan sebagai kombinasi linear p sampel sinyal ucapan sebelumnya. Prinsip dasar ekstraksi ciri ditunjukkan pada persamaan 2.4 (Sudirman dkk, 2005).

$$s(n) \approx a_1s(n-1) + a_2s(n-2) + \dots + a_ps(n-p) \quad (2.4)$$

Keutamaan metode ekstraksi LPC seperti yang diungkapkan Sudirman dkk (2005) adalah dapat diuraikan pada banyak pekerjaan lebih awal, seperti pada blok diagram pada Gambar 2.4 yang menunjukkan proses mendapatkan nilai koefisien LPC.

Setelah ekstraksi LPC, kemudian koefisien tersebut masuk ke dalam proses *frame* normalisasi. Hal tersebut dilakukan untuk mencocokkan sampel yang belum diketahui dengan sampel referensi. Pencocokan terdiri atas kompresi *frame-frame* dan teknik ekspansi. Keuntungan lain dari teknik ini yang diuraikan oleh Sudirman dkk (2005) yaitu pengurangan jumlah *input*; dapat mengurangi jumlah kompleksitas *network* dan perhitungan bobot, tetapi akan menambah kecepatan konvergensi.





Gambar 2.4 Blok Diagram dari Proses LPC
Sumber: Sudirman dkk, 2005.

2.2.4.1 Preemphasis

Pada langkah penapisan atau *preemphase*, cuplikan kata dalam bentuk *digital* ditapis dengan menggunakan *FIR filter* orde satu untuk meratakan spektral sinyal kata yang telah dicuplik tersebut. Sinyal frekuensi tinggi akan diratakan, sehingga terjadi keseragaman amplitudo. Dengan amplitudo seragam dan diperbesar akan terjadi pengerasan suara. Fungsi ini didefinisikan pada persamaan 2.5:

$$H(z) = 1 - \tilde{\alpha}z^{-1} \quad (2.5)$$

dimana $0.9 \leq \tilde{a} \leq 1.0$

$$\tilde{s}(n) = s(n) - \tilde{a}s(n-1) \quad (2.6)$$

untuk harga \tilde{a} yang sering digunakan adalah 0.95. sedangkan untuk *fixed point*, harga \tilde{a} ialah 15/16 atau sama dengan 0.9375. (Proakis, 1995)

2.2.4.2 Frame blocking

Pada tahap ini sinyal kata yang telah ter-*emphasis* dibagi menjadi *frame-frame* sebanyak T dengan masing-masing *frame* memuat N cuplikan kata dan *frame-frame* yang berdekatan dipisahkan sejauh M cuplikan. Ukuran sampel tiap *frame* dihitung dari rata-rata cuplik tiap detik dengan waktu cuplik tiap periode, umumnya digunakan 30 ms. Jika rata-rata cuplik adalah 8000 Hz, maka jumlah sampel tiap *frame* adalah $3 \cdot 10^{-3}$ detik x 8000 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 $240 < 2^8$, maka $N = 256$ (Rabiner, 1993).

Tabel 2.1 Nilai Parameter Analisa LPC yang Sering Digunakan pada Pengenalan Suara

Parameter	$F_s = 6,67$ KHz	$F_s = 8$ KHz	$F_s = 10$ KHz
N	300 (45 msec)	240 (30 msec)	300 (30 msec)
M	100 (15 msec)	80 (10 msec)	100 (10 msec)
P	8	10	10
Q	12	12	12
K	3	3	3

Sumber: Rabiner dkk, 1993.

Untuk melakukan pembagian blok ini dapat mengacu pada *Linear Prediction*. Parameter-parameter yang biasa digunakan diantaranya adalah: N adalah jumlah sampel pada analisa *frame blocking*, dan M adalah jarak antara satu *frame* dengan *frame* selanjutnya. Nilai parameter LPC yang sering digunakan pada pengenalan suara dapat dilihat pada Tabel 2.1.

Parameter P pada Tabel 2.1 merupakan orde koefisien LPC, sedangkan parameter Q merupakan orde koefisien *cepstral*.

2.2.4.3 Windowing

Pada langkah ini dilakukan fungsi *weighting* pada setiap *frame* yang telah dibentuk pada langkah sebelumnya untuk membuat awal dan akhir tiap *frame* menuju nol. N adalah banyak sampel dalam 1 *frame*, dengan indeks n , maka sinyal terboboti adalah $h(n)$ merupakan *impuls* respon sinyal $x(n)$ dikalikan $w(n)$. Metode yang ada adalah *rectangular window*, *Bartlett window*, *Hanning window*, *Gaussian Window*, *Hamming window*, *Blackman window*, *Blackman-Harris window*, *Blackman-Nuttall window*, dan *User-defined weighting window*. Untuk sinyal suara, metode yang cocok adalah *Hamming Window*, yang didefinisikan pada persamaan 2.7 (Jones, 2009).

$$w(n) = 0,54 - 0,46 \cos\left(\frac{2\pi n}{N-1}\right) \quad (2.7)$$

dimana $0 \leq n \leq N-1$

2.2.4.4 Analisa autokorelasi

Pada tahap ini masing-masing *frame* yang telah di *windowing* diautokorelasikan dengan nilai autokorelasi yang tertinggi yaitu orde dari analisa LPC. Biasanya orde LPC tersebut 8 sampai 16 untuk menghasilkan koefisien autokorelasi. Koefisien ini secara matematis dinyatakan dengan persamaan 2.8.

$$r_t(m) = \sum_{n=0}^{N-1-m} \tilde{x}_t(n) \tilde{x}_t(n+m), m = 0, 1, 2, \dots, p \quad (2.8)$$

Dimana $r_t(m)$ adalah koefisien autokorelasi ke- m pada *frame* ke- t , sedangkan p menyatakan orde LPC. (Deller, 2002).

2.2.4.5 Analisa LPC

Langkah selanjutnya adalah analisa LPC, dimana pada tahap ini nilai autokorelasi pada setiap *frame* diubah menjadi satu set LPC parameter yaitu koefisien LPC, koefisien pantulan (*reflection coefficient*), dan koefisien perbandingan daerah logaritmis (*log area ratio coefficient*). Koefisien LPC didapat dari solusi persamaan *Yule-Walker* yang berbentuk sistem *Toeplitz* dan didefinisikan pada persamaan 2.9.

$$\begin{bmatrix} r(1) & r(2) & \dots & r(p) \\ r(2) & r(1) & \dots & r(p-1) \\ \vdots & \vdots & \ddots & \vdots \\ r(p) & r(p-1) & \dots & r(1) \end{bmatrix} \cdot \begin{bmatrix} a(2) \\ a(3) \\ \vdots \\ a(p+1) \end{bmatrix} = \begin{bmatrix} -r(2) \\ -r(3) \\ \vdots \\ -r(p+1) \end{bmatrix} \quad (2.9)$$

Solusi dari persamaan 2.9 dapat dipecahkan menggunakan algoritma *Levinson-Durbin* yang secara matematis didefinisikan pada persamaan 2.10. (Jones, 2009).

$$E_0 = r[0] \quad (2.10)$$

$$k_i = \frac{1}{E_{i-1}} (r[i] - \sum_{j=1}^{i-1} (\alpha_{j,i-1} r[|i-j|])) \quad (2.11)$$

$$\alpha_{j,i} = \alpha_{j,i-1} - k_i \alpha_{i-j,i-1} \quad (2.12)$$

untuk $j = 1$ sampai $j = i-1$

$$\alpha_{i,i} = k_i \quad (2.13)$$

$$E_i = (1 - k_i^2) E_{i-1} \quad (2.14)$$

Persamaan 2.10 sampai 2.14 direkursi untuk $i=1,2,\dots,p$ dan penyelesaian akhirnya berupa:

$$a_m = \text{koefisien LPC} = \alpha_m^{(p)}, 1 \leq m \leq p \quad (2.15)$$

Koefisien LPC yang dimaksud dinyatakan dalam persamaan 2.16.

$$a_m = \alpha_m^{(p)}; 1 \leq m \leq p \quad (2.16)$$

2.2.4.6 Pengubahan parameter LPC menjadi koefisien cepstral

Koefisien LPC yang diperoleh dari analisis LPC kemudian dikonversi menjadi set koefisien *cepstral*, $c(m)$. Koefisien *cepstral* ini merupakan koefisien transformasi Fourier yang merepresentasikan spektrum *log magnitude*. Solusi matematis dalam konversi ini dinyatakan dalam persamaan 2.17 – 2.19.

$$c_0 = l \times n \times \sigma^2; \sigma^2 = \text{gain dalam model LPC} \quad (2.17)$$

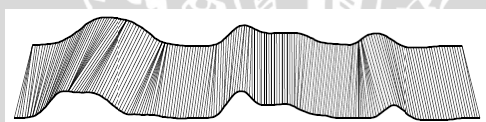
$$c_m = a_m + \sum_{k=1}^{m-1} \left(\frac{k}{m}\right) c_k a_{m-k}, 1 \leq m \leq p \quad (2.18)$$

$$c_m = \sum_{k=1}^{m-1} \left(\frac{k}{m}\right) c_k a_{m-k}, m > p \quad (2.19)$$

2.2.5 Pengenalan suara dengan *Piecewise Dynamic Time Warping* (PDTW)

Metode *Piecewise Dynamic Time Warping* (PDTW) merupakan modifikasi dari metode *Dynamic Time Warping* (DTW). Dimana DTW merubah indeks waktu dari masing-masing *point* data dalam sebuah lintasan yang diuji untuk meminimalkan jarak antara lintasan yang diuji dan lintasan konsensus (lintasan yang disepakati). Sedangkan PDTW merupakan modifikasi dari DTW dimana lintasan-lintasannya terbagi-bagi pada titik-titik dari *interest* dan klasikal DTW yang digunakan untuk meluruskan bagian-bagian yang terhubung secara intensitas-normalisasi (Helwig, 2009).

Keogh (2000) menunjukkan PDTW sebagai sebuah modifikasi dari DTW yang menjalankan *Piecewise Aggregate Approximation* (PAA) yang mengurangi representasi dimensionalitas. PDTW menunjukkan hasil penjajaran yang sangat mirip dengan yang ditunjukkan oleh DTW. Gambar 2.5 menunjukkan contoh perbandingan jajaran DTW dan PDTW. PDTW menunjukkan secara efektif dalam membangkitkan kecepatan dari besarnya orde satu sampai orde tiga, dibandingkan dengan algoritma DTW klasik, tanpa adanya kerugian yang signifikan dari keakuratan untuk klasifikasi dan *task clustering*. (Chu, 2007).



a) *Time series* dan *alignment* (jajaran) ditunjukkan oleh DTW.



b) *Time series* yang sama pada representasi PAA, dan *alignment* yang ditunjukkan oleh PDTW. Hal ini menunjukkan bukti yang kuat bahwa PDTW menemukan pendekatan *warping* yang sama dengan DTW.

Gambar 2.5 Perbandingan *Alignment* DTW dan PDTW
Sumber: Chu, 2007.

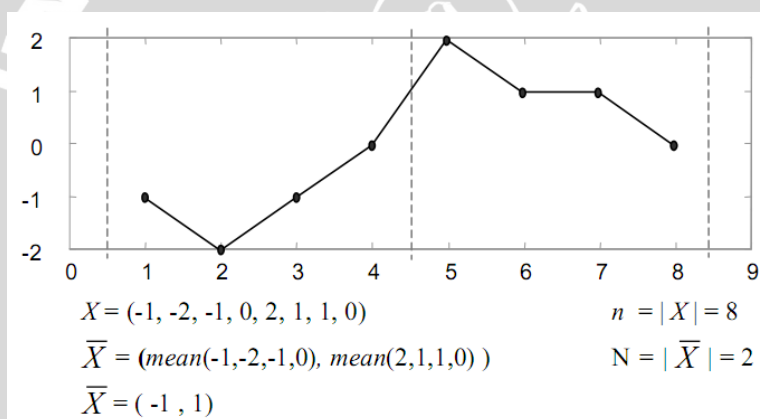
2.2.5.1 Representasi *Piecewise Aggregate Approximation* (PAA)

Sebuah *statement time series* dinotasikan sebagai $X = x_1, \dots, x_n$. Dimisalkan N adalah dimensionalitas dari *time series* yang ditransformasikan dapat digunakan dengan $(1 \leq N \leq n)$. Untuk pengefisienan waktu, dapat diasumsikan bahwa N adalah faktor dari n .

Sebuah *time series* X dengan panjang n direpresentasikan pada jarak N dengan sebuah vektor $\bar{X} = \bar{x}_1, \dots, \bar{x}_N$. Elemen ke- i dari \bar{X} dapat dihitung dengan persamaan 2.20. (Keogh, 2000).

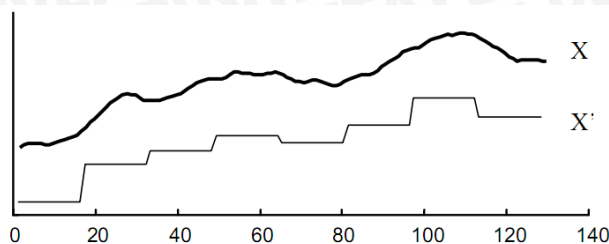
$$\bar{x}_i = \frac{N}{n} \sum_{j=\frac{n}{N}(i-1)+1}^{\frac{n}{N}i} x_j \quad (2.20)$$

Dinyatakan secara sederhana, untuk mengurangi data dari ukuran n ke ukuran N , data dibagi ke dalam N *equi-sized "frames"*. Nilai mean dari data didapat dalam sebuah *frame* yang dihitung dan sebuah vektor dari nilai ini menjadi representasi pengurangan data. Gambar 2.6 menggambarkan notasi ini. *Subscripting* yang rumit pada persamaan 2.20 digunakan hanya untuk memastikan rangkaian *time series* yang asli dibagi menjadi bilangan-bilangan yang lebih baik dari sebelumnya serta ukuran dari *frame-frame*.



Gambar 2.6 Contoh Ilustrasi Teknik Pengurangan Data
Sumber: Keogh, 2000.

Pada Gambar 2.6 sebuah *time series* terdiri dari 8 (n) titik yang diproyeksikan ke dalam 2 (N) ukuran. *Time series* tersebut dibagi menjadi 2 (N) *frame* dan mean dari masing-masing *frame* tersebut dihitung. Nilai vektor dari mean inilah yang menjadi representasi pengurangan data.



Gambar 2.7 Rangkaian *Time Series* X dan Pendekatan *Piecewise Aggregate* X'

Sumber: Keogh, 2000.

Dua kasus istimewa dapat bermanfaat ketika representasi yang ditransformasikan $N = n$ memiliki kesamaan dengan representasi asli. Jika representasi yang ditransformasikan $N = 1$ maka nilai mean yang didapat lebih sederhana dari rangkaian yang asli. Secara umum, transformasi menghasilkan sebuah penaksiran *piecewise constant* dari rangkaian yang asli, yang dapat disebut dengan pendekatan *Piecewise Aggregate Approximation* (PAA). Gambar 2.7 meng-gambarkan sebuah *time series* asli dan *time series* hasil penaksiran PAA.

Perbandingan jarak *time series* yang asli dengan *time series* hasil representasi PAA dapat dinotasikan sebagai *compression rate* c yang ditunjukkan pada persamaan 2.21. Dimana variabel n pada persamaan tersebut merupakan banyaknya data, sedangkan variabel N merupakan besar amplitudo rata-rata.

$$c = \frac{n}{N} \quad (2.21)$$

2.2.5.2 Warping menggunakan representasi PAA

Pada *Dynamic Time Warping* (DTW), representasi menggunakan notasi *time series* Q dan C . Sedangkan PDTW menggunakan DTW versi pengurangan dimensionalitas dari *time series* Q dan C tersebut, yang dinotasikan secara berturut-turut \bar{Q} dan \bar{C} . Dengan jarak n dan m berturut-turut ditunjukkan pada persamaan 2.22 dan 2.23 (Keogh, 2000).

$$\bar{Q} = \bar{q}_1, \bar{q}_2, \dots, \bar{q}_l, \dots, \bar{q}_n \quad (2.22)$$

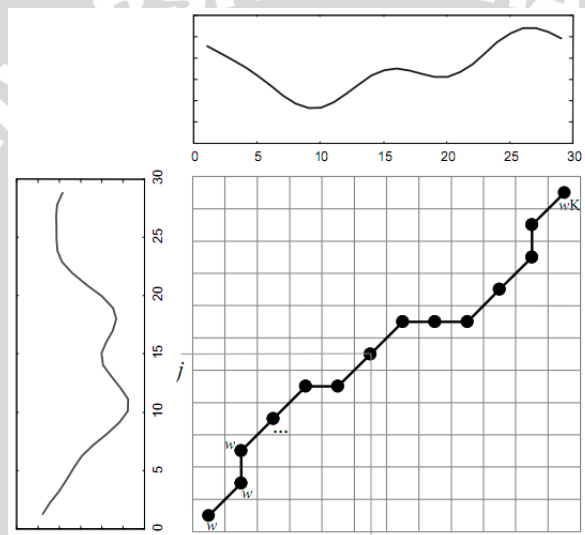
$$\bar{C} = \bar{c}_1, \bar{c}_2, \dots, \bar{c}_j, \dots, \bar{c}_m \quad (2.23)$$

Untuk mensejajarkan dua rangkaian sinyal tersebut menggunakan PDTW, maka perlu dibentuk suatu matriks $N \times M$ dimana elemen (i^{th}, j^{th}) dari matriks berisi jarak $d(\bar{Q}_i, \bar{C}_j)$ antara dua elemen \bar{Q}_i dan \bar{C}_j . Jarak antara dua elemen tersebut didefinisikan sebagai kuadrat dari jarak di antara \bar{Q}_i dan \bar{C}_j seperti pada persamaan 2.24.

$$d(\bar{Q}_i, \bar{C}_j) = (\bar{Q}_i - \bar{C}_j)^2 \quad (2.24)$$

Matriks tersebut digambarkan oleh Gambar 2.8, dimana *warping path* W dibatasi set matriks dari elemen-elemen yang ditetapkan oleh sebuah pemetaan antara \bar{Q} dan \bar{C} . Elemen ke- k dari W didefinisikan sebagai $w_k = (i, j)_k$, sehingga didapatkan persamaan 2.25.

$$W = w_1, w_2, \dots, w_k, \dots, w_K \quad \max(m, n) \leq K < m+n-1 \quad (2.25)$$



Gambar 2.8 Contoh *Warping Path*
Sumber: Keogh, 2000.

Bagian-bagian dari modifikasi algoritma pencari matriks ini keutamaannya adalah tidak dapat diubah. Definisi $\gamma(i, j)$ dari DTW dimodifikasi untuk menggambarkan ukuran jarak yang baru.

$$\gamma(i, j) = d(\bar{Q}_i, \bar{C}_j) + \min\{\gamma(i-1, j-1), \gamma(i-1, j), \gamma(i, j-1)\} \quad (2.26)$$

Untuk baris pertama menggunakan persamaan 2.27 dan kolom pertama menggunakan persamaan 2.28.

$$\gamma(1, j) = \sum_{k=1}^j c(x_1, y_k), j \in [1, M] \quad (2.27)$$

$$\gamma(i, 1) = \sum_{k=1}^i c(x_k, y_1), i \in [1, N] \quad (2.28)$$

Selanjutnya, *path* yang dapat meminimasi *warping cost* pada DTW yang memenuhi batasan *warping path* ditunjukkan persamaan 2.29. Nilai K yang terdapat pada penyebut pada persamaan 2.29 digunakan untuk mengimbangi fakta bahwa *warping path* bisa saja mempunyai jarak yang berbeda (Chu, 2007).

$$DTW(Q, C) = \min \left\{ \frac{\sqrt{\sum_{k=1}^K w_k}}{K} \right\} \quad (2.29)$$

$$PDTW(\bar{Q}, \bar{C}) = \min \left\{ \frac{\sqrt{\sum_{k=1}^K w_k}}{\sqrt{c}} \right\} \quad (2.30)$$

Pada PDTW, nilai K tidak dapat digunakan secara langsung, karena elemen pada matriks *warping* kini disamakan untuk menjumlah segmen-segmen dari data dan PDTW dapat diukur pada unit yang sama layaknya DTW untuk memudahkan perbandingan antara kedua ukuran tersebut. Untuk mengimbangi perbedaan jarak *warping path*, dapat digunakan pengukuran jarak yang sama dengan definisi 2.29 dimana penyebut merupakan akar kuadrat dari c , yaitu *compression rate* atau nilai kompresi.

Karena jarak dari *warping path* dihitung pada unit yang sama sebagai DTW, maka PDTW dapat didefinisikan seperti pada persamaan 2.31 (Keogh, 2000).

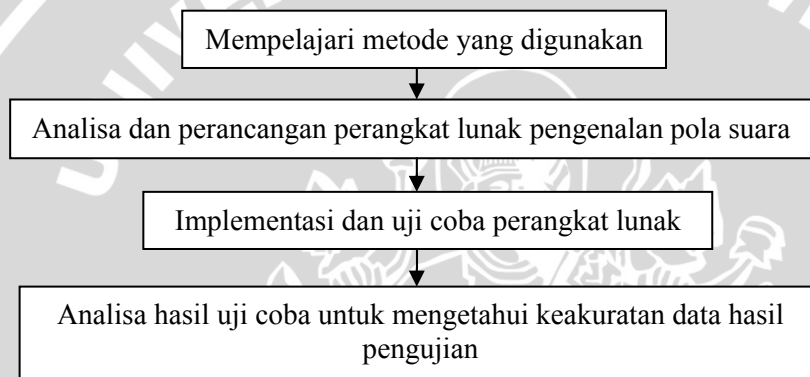
$$PDTW(\bar{Q}, \bar{C}) \cong \frac{DTW(Q, C)}{\sqrt{c}} \quad (2.31)$$

BAB III METODOLOGI DAN PERANCANGAN

Pada bab Metodologi dan Perancangan akan dibahas mengenai penggunaan metode yang digunakan dalam pembuatan perangkat lunak pengenalan pola suara. Adapun tahapan-tahapan yang dilakukan adalah sebagai berikut:

1. Mempelajari metode-metode yang akan digunakan dari berbagai referensi.
2. Menganalisa dan merancang perangkat lunak dengan menggunakan metode yang telah dipelajari sebelumnya.
3. Membuat perangkat lunak berdasarkan analisa dan perancangan yang telah dilakukan serta melakukan uji coba perangkat lunak dengan menggunakan *signal-noise* yang berbeda.
4. Evaluasi hasil pengenalan pola suara yang dilakukan oleh perangkat lunak untuk mengetahui keakuratan data hasil pengujian.

Langkah-langkah dalam pembuatan perangkat lunak tersebut ditunjukkan pada Gambar 3.1.

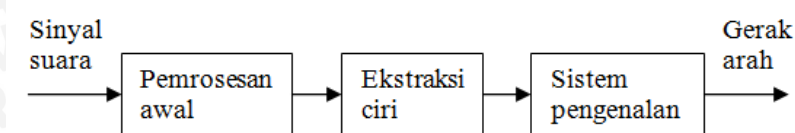


Gambar 3.1 Langkah Pembuatan Perangkat Lunak

3.1 Deskripsi Sistem

Perangkat lunak yang dibangun pada skripsi ini merupakan cabang dari aplikasi pengenalan suara yang mengolah sinyal suara, mengenalinya, dan mengubahnya menjadi *output* yang diharapkan (dalam hal ini berupa simulasi gerak sebuah grafis). Prinsip kerja perangkat lunak ini adalah

dengan membandingkan informasi suara yang ada pada data referensi dengan informasi suara yang menjadi masukan sistem pengenalan suara tersebut. Pengenalan suara dengan *Linear Predictive Coding* (LPC) dan *Piecewise Dynamic Time Warping* (PDTW) dapat dibagi menjadi 3 tahap yaitu pemrosesan awal (*preprocessing*), ekstraksi ciri (*feature extraction*), dan tahap sistem pengenalan PDTW. Diagram blok pengenalan suara secara umum ditunjukkan pada Gambar 3.2.



Gambar 3.2 Diagram Blok Pengenalan Suara secara Umum

Tahap pemrosesan awal dilakukan untuk menyetarakan semua sinyal suara dan mengubah sinyal suara dari *analog* menjadi *digital*. Tahap *feature extraction* bertujuan untuk mendapatkan parameter-parameter yang dapat merepresentasikan sinyal suara tersebut. Dan tahap yang ketiga yaitu sistem pengenalan dilakukan untuk mengenali sinyal yang diubah menjadi representasi animasi sebagai simulasi gerak suatu grafis.

3.1.1 Tujuan sistem

Sistem ini dirancang dengan tujuan untuk mengenali suara manusia dan mengubahnya menjadi representasi visual sebagai simulasi pengontrolan gerak suatu grafis. Disediakan referensi data suara yang diucapkan oleh beberapa orang yang berbeda, dan sistem akan dapat mengenalinya sebagai kata yang sama dalam bentuk simulasi grafis.

3.1.2 Batasan sistem

Perancangan dibatasi dengan unit pengenalan sebagai data referensi dari sampel suara 4 orang laki-laki. Dimana setiap orang mengucapkan KANAN, KIRI, MAJU, dan MUNDUR masing-masing sebanyak 5 kali. Dan pengambilan data referensi dilakukan secara random dengan masing-masing kata diambil 2 sampel suara yang berbeda dengan sinyal untuk data uji.

Untuk keterangan keluaran dari masing-masing kata dapat dijabarkan sebagai berikut:

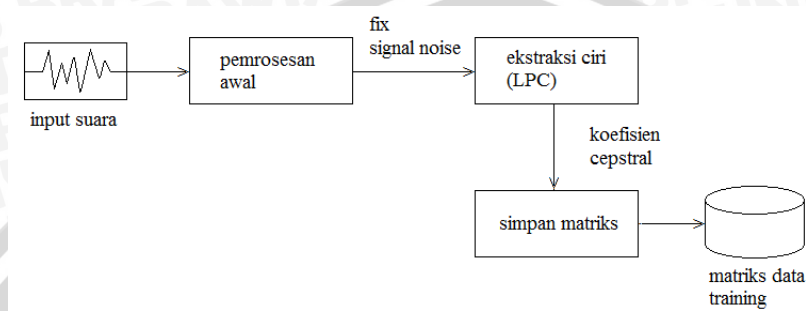
1. Kanan: animasi grafis bergerak ke kanan.

2. Kiri: animasi grafis bergerak ke kiri.
3. Maju: animasi grafis bergerak ke atas.
4. Mundur: animasi grafis bergerak ke bawah.

3.2 Desain Sistem

Sistem pengenalan suara yang akan dibangun ini menggunakan metode pengenalan *Piecewise Dynamic Time Warping* (PDTW) dengan *Linear Predictive Coding* (LPC) sebagai ekstraksi cirinya.

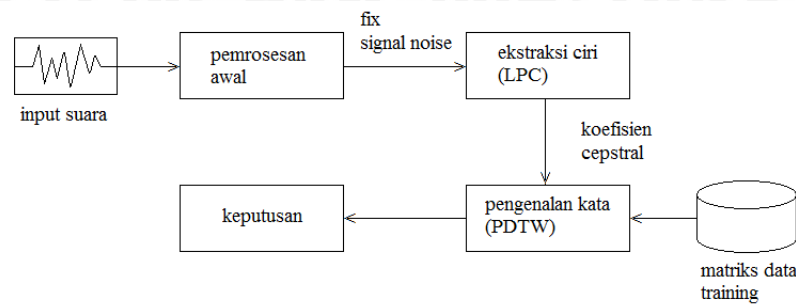
Pada skripsi ini terdapat dua proses, yaitu training dan recognizing. Diagram blok untuk penyimpanan data training ditunjukkan pada Gambar 3.3.



Gambar 3.3 Diagram Blok Penyimpanan Data Training

Penyimpanan data training pada Gambar 3.3 dimulai dari *input* sinyal suara berbentuk file *.WAV* yang mengalami proses pemrosesan awal yaitu normalisasi. Setelah pemrosesan awal, kemudian dikenali cirinya pada proses ekstraksi ciri menggunakan metode LPC. Dari ekstraksi tersebut akan didapatkan koefisien *cepstral* dari sinyal dalam bentuk matriks yang disimpan sebagai data *training*.

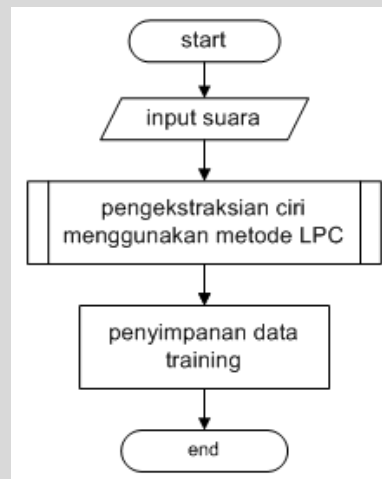
Sedangkan diagram blok untuk pengenalan kata ditunjukkan pada Gambar 3.4.



Gambar 3.4 Diagram Blok Sistem Pengenalan Kata

Sistem pengenalan kata pada Gambar 3.4 dimulai dari *input* sinyal suara berbentuk file *.WAV* yang mengalami proses pemrosesan awal yaitu normalisasi. Setelah pemrosesan awal, kemudian dikenali cirinya pada proses ekstraksi ciri menggunakan metode LPC. Dari ekstraksi tersebut akan didapatkan koefisien *cepstral* dari sinyal dalam bentuk matriks yang kemudian akan dibandingkan dengan matriks data *training* menggunakan metode pengenalan PDTW. Hasil dari pengenalan itu akan ditemukan keputusan kata yang paling tepat.

Diagram alir dari sistem penyimpanan data training ditunjukkan pada Gambar 3.5.



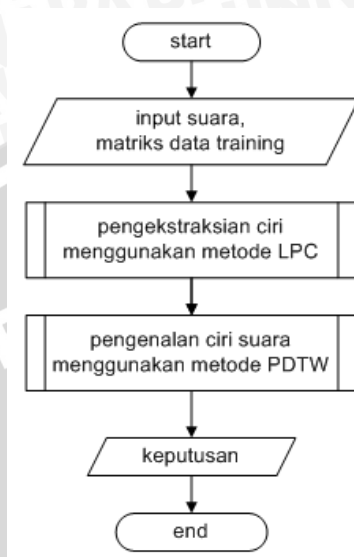
Gambar 3.5 Diagram Alir Penyimpanan Data Training

Langkah-langkah proses sistem pengenalan kata dapat dijabarkan sebagai berikut:

1. *Input* sinyal suara yang akan di-*training*.
2. Ekstraksi ciri

Sinyal suara yang diterima akan dilakukan proses ekstraksi ciri menggunakan LPC. Hasil dari proses ini adalah koefisien *cepstral* dari sinyal suara yang kemudian disimpan sebagai data *training* dalam bentuk matriks.

Adapun diagram alir dari sistem pengenalan kata secara umum ditunjukkan pada Gambar 3.6.



Gambar 3.6 Diagram Alir Sistem Pengenalan Kata

Langkah-langkah proses sistem pengenalan kata dapat dijabarkan sebagai berikut:

1. *Input* sinyal suara yang akan diuji dan matriks data yang telah di-*training* sebelumnya.
2. Ekstraksi ciri
Sinyal suara yang diuji akan dilakukan proses ekstraksi ciri menggunakan LPC. Hasil dari proses ini adalah koefisien *cepstral* dari sinyal suara.
3. Pengenalan ciri
Setelah didapatkan koefisien *cepstral* dari sinyal yang diuji dari proses ekstraksi ciri, koefisien tersebut kemudian dibandingkan dengan

koefisien-koefisien dari matriks data *training* menggunakan metode PDTW. Hasil dari proses ini adalah pengambilan keputusan berdasarkan nilai jarak yang paling minimum dari perbandingan yang telah dilakukan.

Dalam metode PDTW terdapat parameter dua *time series* yang dibandingkan. Pada subbab 3.2.1 sampai 3.2.4 akan dibahas tahap-tahap pengenalan suara dari pemrosesan awal, ekstraksi ciri, sampai sistem pengenalan beserta diagram alir sistem pengenalan suara.

3.2.1 Data input

Sinyal yang digunakan untuk data referensi diambil secara random dari beberapa masukan yang dilakukan oleh 4 orang laki-laki. Dari masukan tersebut ditentukan dua data dari masing-masing kata uji yang menjadi data referensi (*supervisor*). Sedangkan sinyal untuk data uji diambil dari 4 orang laki-laki yang masing-masing mengucapkan 4 kata (MAJU, MUNDUR, KANAN, dan KIRI) sebanyak 5 kali. Sehingga total sinyal suara yang digunakan sebanyak 88 sinyal.

3.2.2 Pengolahan sinyal suara

Sinyal suara merupakan sinyal *analog*. Agar dapat diproses dalam vektor numerik maka diperlukan proses ADC (*Analog-to-Digital Converter*). Peralatan yang digunakan dalam proses ini diantaranya adalah *transducer* dan *soundcard*. *Transducer* merupakan peralatan yang dapat mengubah tekanan udara (yang terdengar oleh telinga manusia) ke dalam tegangan elektrik yang dapat dimengerti oleh perangkat elektronik. *Transducer* yang dipergunakan dalam skripsi ini adalah mikropon. Sedangkan *soundcard* adalah peralatan untuk mengubah gelombang suara (sinyal audio) menjadi data *digital*. Proses pencuplikan sinyal ini dilakukan dengan menggunakan *soundcard* yang terdapat pada *personal computer* (PC).

Data berupa sinyal suara diperoleh dengan cara merekam suara melalui mikropon yang dihubungkan dengan komputer pribadi. Perekaman suara dilakukan dengan bantuan program aksesoris Windows yaitu *sound recorder* dengan frekuensi sampling 8000Hz, 8 bit mono (1 byte). Kecepatan pencuplikan tersebut dilakukan dengan didasarkan asumsi bahwa sinyal percakapan (*speech*) berada pada daerah frekuensi 300-3400 Hz sehingga memenuhi kriteria *Nyquist* yang menyatakan bahwa frekuensi

pencuplikan harus lebih besar atau sama dengan dua kali frekuensi maksimum sinyal.

Suara diucapkan oleh 4 orang pria yang mempunyai variasi suara yang berbeda, dimana untuk setiap kata diulang sebanyak 5 kali. Perekaman suara ini dilakukan pada ruangan yang bebas *noise*. Dengan *sound recorder* suara direkam kemudian berkas suara dibuka kembali dengan Cool Edit Pro 2.1 untuk dipotong dan disimpan dalam berkas dengan format penamaan *ABC.WAV*. Huruf A menunjukkan nama responden, huruf B menunjukkan kata yang diucapkan, dan huruf C menunjukkan iterasi suara (1-5).

3.2.2.1 Pemrosesan awal (*preprocessing*)

Tujuan dari pemrosesan awal ini adalah untuk menyetarakan semua sinyal suara masukan sistem agar sesuai dengan spesifikasi yang dibutuhkan oleh sistem.

Sinyal suara merupakan realisasi dari beberapa kode pesan yang berupa urutan vektor ciri. Vektor ciri ini merepresentasikan amplitudo diskrit dalam sinyal suara. Sinyal suara hasil pencuplikan dilakukan proses normalisasi. Proses normalisasi bertujuan untuk mendapatkan sinyal dengan ukuran yang sama walau-pun kata yang diucapkan berbeda. Cara kerja proses normalisasi ini dilakukan dengan menambahkan beberapa data tambahan apabila data hasil pencuplikan belum memenuhi jumlah yang dibutuhkan atau dengan mengurangi jumlah data hasil pencuplikan apabila melebihi jumlah *input* yang dibutuhkan. Jumlah data *output* (vektor numerik) dari proses normalisasi ini ditetapkan sebanyak 3360 buah (0,42 detik) dengan asumsi bahwa untuk pengucapan satu kata dibutuhkan waktu kurang dari 0,5 detik.

Normalisasi yang dilakukan adalah dengan membagi nilai amplitudo diskrit dengan nilai amplitudo maksimum dari amplitudo diskrit, sehingga dalam hal ini amplitudo-nya berada pada interval -1 sampai dengan 1. Akan tetapi sebelum normalisasi dilakukan, perlu adanya proses *centering* yang bertujuan untuk menggeser letak distribusi amplitudo diskrit sehingga pusatnya berada pada sumbu $y = 0$. Dengan kata lain *centering* berfungsi untuk membuat amplitudo sinyal suara mempunyai rata-rata sama dengan nol. *Centering* dilakukan dengan cara mengurangi amplitudo sinyal suara dengan rata-rata amplitudo sinyal suara.

3.2.3 Ekstraksi ciri menggunakan metode LPC

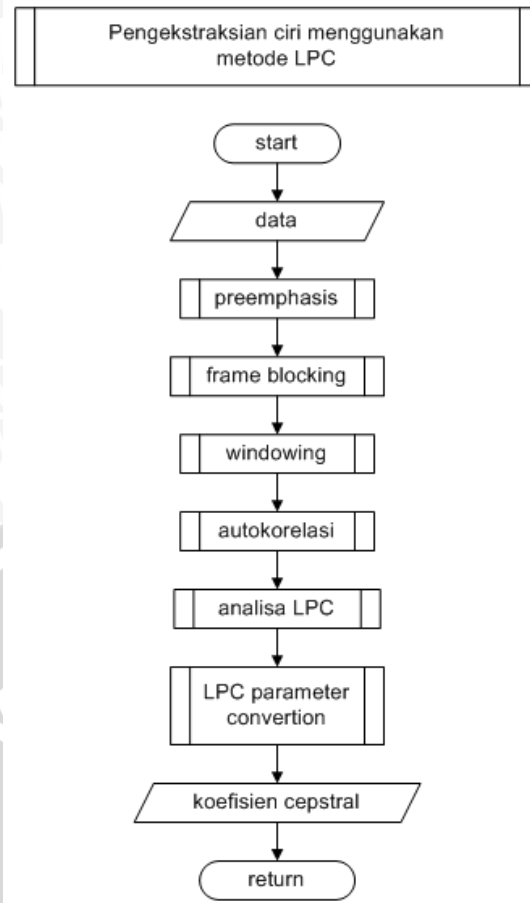
Setelah melalui proses perekaman, sinyal suara akan dibaca atau dipanggil kembali dengan fungsi *wavread*. Hasil pembacaan data untuk

keseluruhan sinyal suara digunakan untuk proses selanjutnya yaitu ekstraksi ciri. Proses ekstraksi ciri digunakan untuk mencari nilai koefisien-koefisien LPC dari sinyal suara. LPC merupakan teknik untuk merepresentasikan sinyal dalam bentuk parameter-parameter yang diperoleh dari kombinasi sinyal sekarang dan sinyal sebelumnya.

Sinyal suara merupakan sinyal yang dilihat dalam selang waktu singkat antara 10 ms sampai 30 ms. Pada keadaan tersebut vektor-vektor ciri sinyal suara yang akan diekstrak cenderung konstan. Vektor numerik yang merepresentasikan amplitudo diskrit dalam domain waktu singkat diubah menjadi representasi energi dan spektral dalam domain frekuensi. Proses ini bertujuan untuk mendapatkan karakteristik sinyal suara. Diagram alir proses ini terlihat pada Gambar 3.7.

Langkah-langkah proses ekstraksi ciri adalah:

1. *Input* sinyal suara.
2. *Preemphasize*
Sinyal *input* yang diterima dicuplik dan dilakukan proses *preemphasize*.
3. *Frame Blocking*
Dari sinyal yang telah ter-*emphasis* dilakukan *framing*, yaitu membagi sinyal *input* menjadi beberapa *frame* waktu yang sangat singkat untuk memperoleh kondisi sinyal yang stabil dalam domain waktu.
4. *Windowing*
Frame yang telah terbentuk dilakukan pembobotan dengan memperkecil sinyal menuju nol pada ujung-ujungnya.
5. *Autokorelasi*
Frame hasil *windowing* diautokorelasikan untuk menghasilkan koefisien autokorelasi.
6. *Analisa LPC*
Pada langkah analisa LPC, autokorelasi pada setiap *frame* diubah menjadi satu set parameter LPC.
7. *LPC parameter conversion*
Dari sekelompok parameter LPC hasil analisa LPC akan terbentuk suatu *output*, yaitu koefisien *cepstral*.

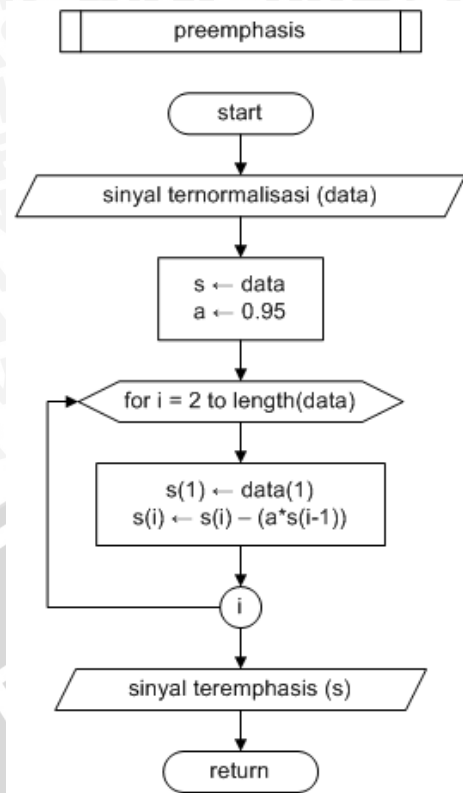


Gambar 3.7 Diagram Alir Ekstraksi Ciri

3.2.3.1 Preemphasize

Sinyal suara digital $s(n)$ dilewatkan pada suatu *filter digital* yang mempunyai orde rendah, biasanya merupakan *filter* orde satu. Proses ini bertujuan untuk meratakan spektral sinyal, menghilangkan harga-harga puncak pada spektrum sinyal sehingga memudahkan dalam menentukan batas-batas ketelitian sinyal pada pemrosesan sinyal. Nilai koefisien \tilde{a} yang digunakan dalam skripsi ini adalah 0.95.

Diagram alir proses *preemphasize* ditunjukkan oleh Gambar 3.8.



Gambar 3.8 Diagram Alir Proses *Preemphasize*

Langkah-langkah proses *preemphasize* adalah:

1. *Input sampling data* sinyal suara yang telah ternormalisasi.
2. Inisialisasi sinyal $s = data$ dan nilai $a = 0.95$.
3. Dilakukan iterasi i dimulai dari 2 sampai dengan panjang $data$, dengan rumus $s(i) = s(i) - (a*s(i-1))$. Iterasi dimulai dari 2 karena untuk $i = 1$ nilai $s(1)$ sama dengan $data(1)$. Sehingga hasil dari perhitungan ini akan didapatkan set sinyal *ter-emphasis*.

3.2.3.2 Frame Blocking

Sinyal keluaran *preemphasize* kemudian dibagi menjadi beberapa *frame* waktu yang sangat singkat untuk memperoleh kondisi sinyal yang stabil dalam domain waktu. Tiap *frame* berisi N sampel dengan jarak antar *frame* M sampel. Nilai M lebih kecil dari N sehingga ada bagian *frame* yang

overlapping sepanjang $N-M$ sampel. Tujuannya adalah supaya estimasi spektral LPC saling berkorelasi antara *frame* satu dengan *frame* yang lain.

Pada skripsi ini proses *frame blocking* yang dilakukan ditetapkan tiap 30 ms dengan jarak antar *frame* 10 ms. Hal ini dikarenakan sifat dari sinyal kata yang dapat diasumsikan tetap pada jangka waktu pendek, jadi dengan kecepatan cuplik sebesar 8 KHz maka tiap *frame* akan berisi (N) 240 byte data dengan jarak antar *frame* (M) 80 byte data atau dengan kata lain *overlap* yang terbentuk sebesar 160 byte data. Dengan ketentuan *frame* tersebut, maka untuk data hasil cuplik sebanyak 3360 data maka akan terbentuk $frame = \frac{3360-160}{80} = 40$ buah.

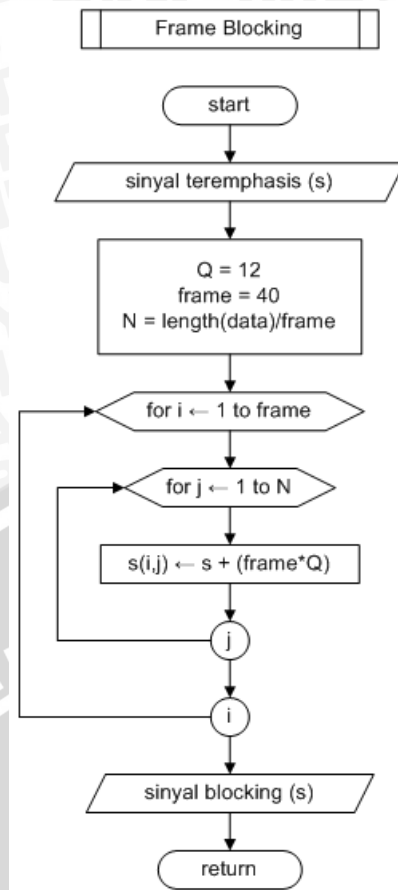
Setelah diketahui banyak *frame*, tiap *frame* diisi data dengan aturan pengindekan. Dimana data ke- i *frame* t adalah data ke- $i + tQ$ dari sinyal setelah perataan. Sehingga data tiap blok saling tumpang tindih, data akhir *frame* t akan terulang pada *frame* $t + 1$ sejauh $N - M$ atau *overlap* sebesar 160. Nilai Q pada setiap data ke- i tersebut merupakan orde koefisien *cepstral*.

Diagram alir proses *frame blocking* ditunjukkan pada Gambar 3.9.

Langkah-langkah proses *frame blocking* adalah:

1. Input data sinyal yang telah ter-*emphasis*.
2. Inisialisasi koefisien *cepstral* $Q = 12$, *frame* = 40, dan nilai N yang merupakan panjang *sample* data setiap *frame*.
3. Dilakukan iterasi i dimulai dari 1 sampai dengan *frame* dan iterasi j yang dimulai dari 1 sampai N . Untuk mengisi nilai $s(i,j)$ dilakukan perhitungan aturan pengindekan $s + (frame * Q)$.



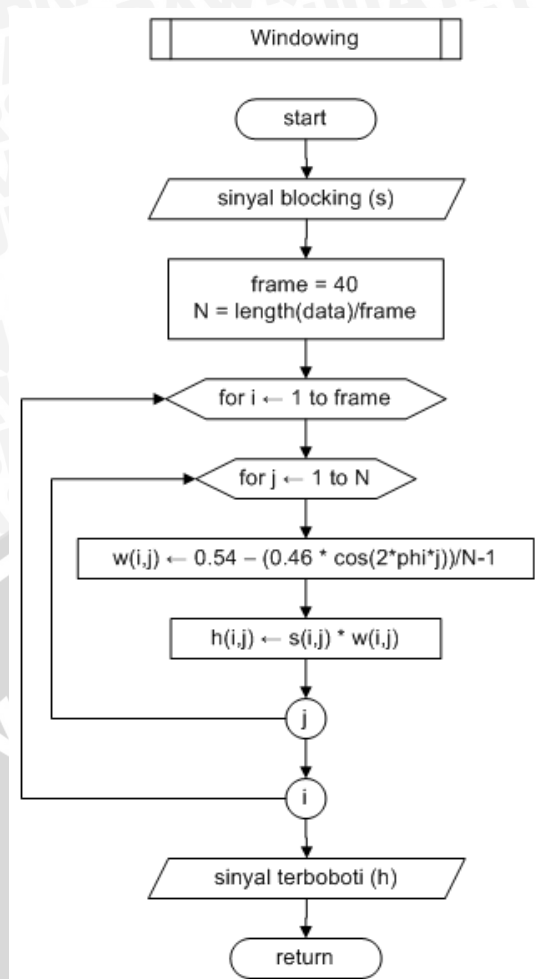


Gambar 3.9 Diagram Alir *Frame Blocking*

3.2.3.3 Windowing

Langkah berikutnya adalah memboboti 40 *frame* hasil *frame blocking* dengan *Hamming window*. Pada langkah ini dilakukan fungsi *weighting* pada setiap *frame* yang telah dibentuk pada langkah sebelumnya dengan tujuan untuk meminimalkan *discontinuities* pada ujung awal dan ujung akhir setiap *frame* yaitu dengan memperkecil sinyal menuju nol pada ujung-ujungnya.

Diagram alir proses *windowing* ditunjukkan oleh Gambar 3.10.

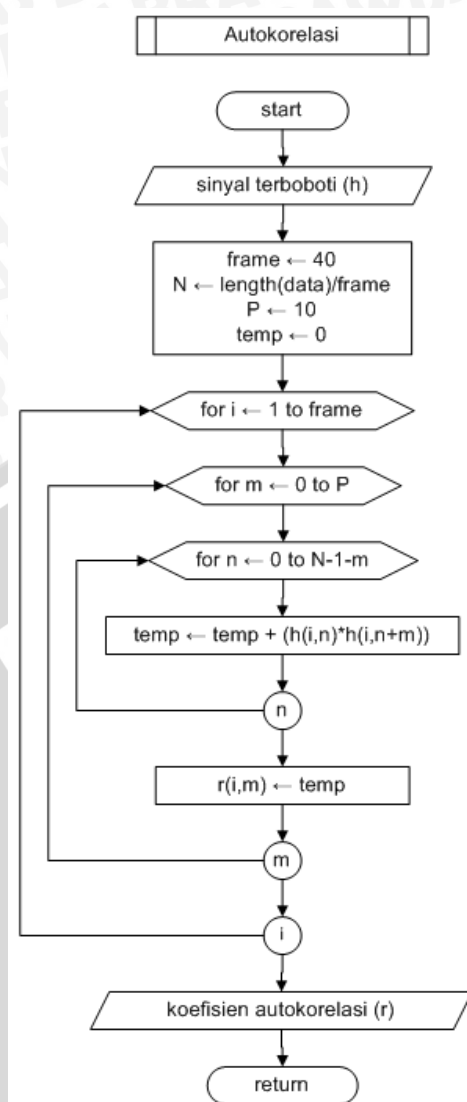


Gambar 3.10 Diagram Alir Proses *Windowing*

Langkah-langkah proses *windowing* adalah:

1. Input data sinyal s .
2. Inisialisasi jumlah $frame = 40$, dan N yang merupakan jumlah sample data pada setiap $frame$.
3. Dilakukan iterasi i dimulai dari 1 sampai $frame$ dan j dimulai dari 1 sampai N . Perhitungan *hamming window* $w(i,j) = 0.54 - 0.46 \cdot \cos((2 \cdot \pi \cdot n)/(N-1))$. Hasil dari *windowing* adalah sinyal terboboti $h(i,j)$ yang merupakan impuls sinyal $s(i,j)$ dikalikan $w(i,j)$.

3.2.3.4 Autocorrelation



Gambar 3.11 Diagram Alir Proses Autokorelasi

Setiap *frame* keluaran *windowing* diautokorelasikan untuk menghasilkan koefisien autokorelasi, dimana nilai autokorelasi yang tertinggi pada $m = p$. Dimana p adalah orde dari analisa LPC, biasanya antara 8 sampai 16. Metode autokorelasi mempunyai salah satu keuntungan yaitu

autokorelasi ke-0 melambangkan energi dari *frame* yang bersangkutan. Pada skripsi ini menggunakan orde LPC (p) 10.

Diagram alir proses *autokorelasi* ditunjukkan oleh Gambar 3.11.

Langkah-langkah proses *autokorelasi* adalah:

1. Input sinyal hasil *windowing* h .
2. Inisialisasi $frame = 40$, $P = 10$, $temp = 0$, dan N = panjang data dibagi dengan $frame$.
3. Iterasi i dimulai dari 1 sampai dengan $frame$ dan m dimulai dari 0 sampai dengan P . Kemudian dilakukan iterasi n dimulai dari 0 sampai dengan $N-1-m$ untuk perhitungan $temp = temp + (h(i,n) * h(i,n+m))$. Hasil dari proses autokorelasi ini adalah koefisien autokorelasi r .

3.2.3.5 LPC analysis

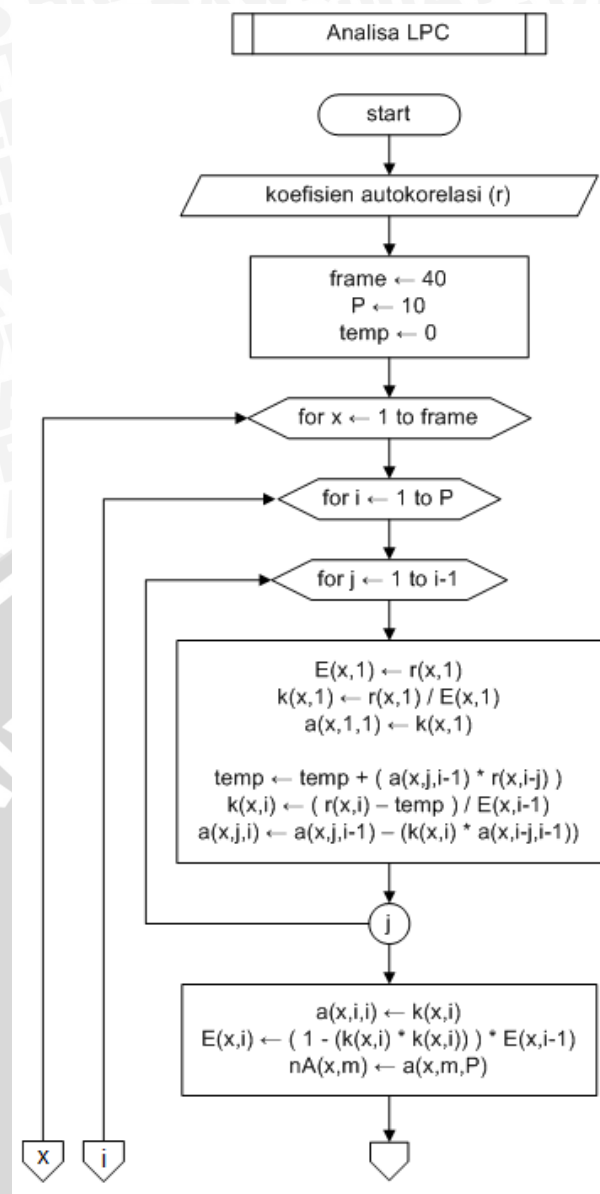
Pada tahap ini $p + 1$ autokorelasi pada setiap *frame* diubah menjadi satu set LPC parameter yaitu koefisien LPC, koefisien pantulan (*reflection coefficient*), koefisien perbandingan daerah logaritmis (*log area ratio coefficient*). Metode yang digunakan dalam skripsi ini adalah metode *Durbin*. Keluaran dari proses ini adalah satu set LPC parameter.

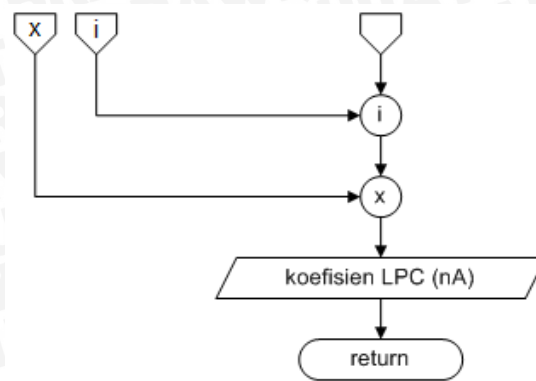
Diagram alir proses analisa LPC ditunjukkan pada Gambar 3.12.

Langkah-langkah proses analisa LPC adalah:

1. Input sinyal hasil autokorelasi r .
2. Inisialisasi $frame = 40$, $P = 10$, dan $temp = 0$.
3. Iterasi x dimulai dari 1 sampai dengan $frame$ dan i dimulai dari 1 sampai dengan P . Kemudian dilakukan iterasi j dimulai dari 1 sampai dengan $i-1$ untuk perhitungan:
 - $E(x,i) = (r(x,i)-temp)/E(x,i-1)$ dari $temp = temp + (a(x,j,i-1)*r(x,i-j))$
 - $k(x,i) = a(x,j,i-1)-(k(x,i)*a(x,i-j,i-1))$
 - $a(x,i,i) = k(x,i)$, dan
 - $a(x,i,j) = (1-(k(x,i)*k(x,i))) * E(x,i-1)$.

Hasil dari proses analisa LPC ini adalah koefisien LPC nA yang diambil dari variabel $a(x,m,P)$.





Gambar 3.12 Diagram Alir Analisa LPC

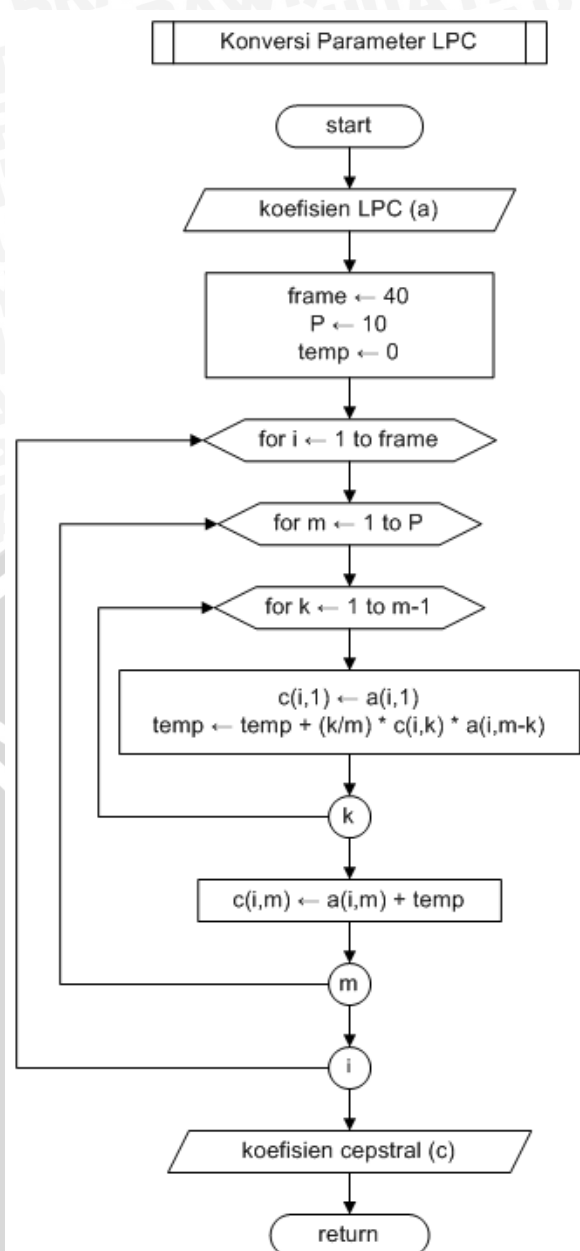
3.2.3.6 LPC parameter conversion

Sekelompok LPC parameter hasil analisa LPC yang sangat penting yang dapat diperoleh dari penurunan koefisien LPC adalah koefisien *cepstral* $c(m)$. Koefisien *cepstral* ini merupakan koefisien transformasi *Fourier* yang merepresentasikan spektrum *log magnitude*. Koefisien *cepstral* ini lebih tahan terhadap *noise* jika digunakan pada pengenalan suara. Dalam skripsi ini telah diketahui bahwa *frame* hasil pencuplikan adalah 40 *frame*, dan orde dari set koefisien *cepstral* yang digunakan dalam skripsi ini adalah orde LPC 10 sehingga didapat data *output* berukuran 40 x 10.

Diagram alir proses konversi parameter LPC menjadi koefisien *cepstral* ditunjukkan pada Gambar 3.13.

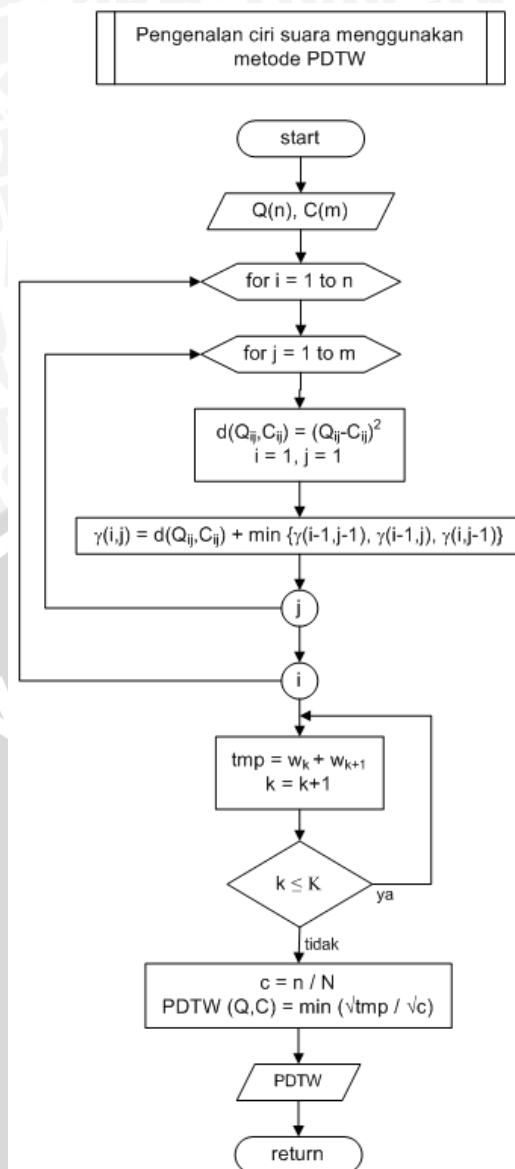
Langkah-langkah proses konversi parameter LPC adalah:

1. Input sinyal hasil analisa LPC a .
2. Inisialisasi $frame = 40$, $P = 10$, dan $temp = 0$;
3. Iterasi i dimulai dari 1 sampai dengan $frame$, dan iterasi m dimulai dari 1 sampai dengan P . Kemudian dilakukan iterasi k dimulai dari 1 sampai dengan $m-1$ untuk mencari nilai kumulatif $temp = temp + (k/m) * c(i,k) * a(i,m-k)$. Hasil dari konversi ini adalah nilai $c(i,m)$ yang didapat dari perhitungan $a(i,m) + temp$.



Gambar 3.13 Diagram Alir Konversi Parameter LPC

3.2.4 Sistem pengenalan kata menggunakan metode *Piecewise Dynamic Time Warping* (PDTW)



Gambar 3.14 Diagram Alir Rekursi PDTW-
Pengenalan pola suara dilakukan setelah ekstraksi ciri dengan menggunakan algoritma PDTW.

Pengenalan kata ini membutuhkan masukan berupa koefisien *cepstral* dari sinyal yang diuji, yang kemudian dibandingkan dengan koefisien sinyal hasil *training*. Setelah dibandingkan dan didapatkan hasil yang optimal, selanjutnya akan ditentukan keluaran berupa animasi gerak suatu grafis sesuai dengan pilihan *output* yang ditentukan.

Diagram alir rekursi PDTW ditunjukkan pada Gambar 3.14.

Langkah-langkah proses rekursi PDTW dapat dijabarkan sebagai berikut:

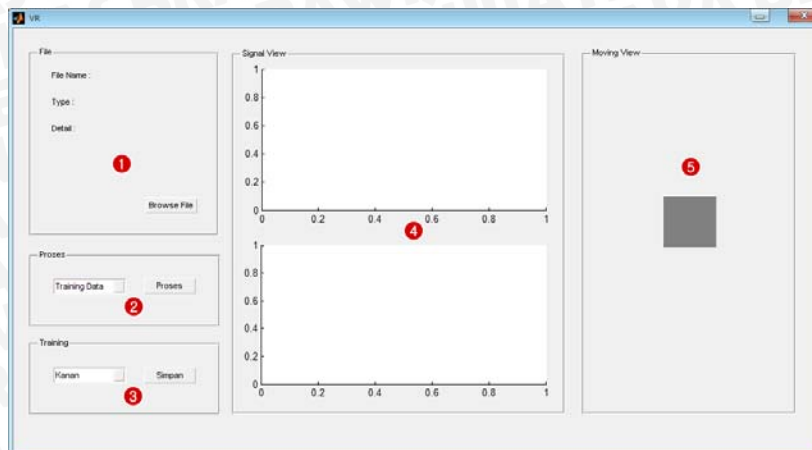
1. Inisialisasi *input* berupa matriks \bar{Q} dan matriks \bar{C} yang berisi koefisien *cepstral* dari masing-masing sinyal sepanjang n dan m .
2. Perhitungan matriks $N \times M$
Untuk mensejajarkan dua rangkaian sinyal tersebut dilakukan pembentukan matriks $N \times M$ dengan rumus $d(\bar{Q}_{ij}, \bar{C}_{ij}) = (\bar{Q}_{ij} - \bar{C}_{ij})^2$.
3. Perhitungan akumulasi jarak
Setelah ditemukan matriks $N \times M$, kemudian dilakukan perhitungan akumulasi jarak menggunakan rumus $\gamma(i,j) = d(\bar{Q}_{ij}, \bar{C}_{ij}) + \min\{\gamma(i-1,j-1), \gamma(i-1,j), \gamma(i,j-1)\}$ dengan tujuan untuk optimalisasi. Dimana $\gamma(i,j)$ merupakan pembebanan distorsi terakumulasi (jarak lokal) antara titik (i,j) .
4. Meminimasi *warping cost* PDTW
Untuk minimum *warping cost* dihitung menggunakan akar kumulasi dari w_k dibagi dengan akar c atau *compression rate* yang didapat dari perhitungan n/N , dimana n merupakan panjang dari rangkaian sinyal.

3.2.5 Perancangan antarmuka

Rancangan program untuk mensimulasikan program pengenalan suara dengan LPC dan metode PDTW ditunjukkan pada Gambar 3.15.

Keterangan Gambar:

1. Tempat untuk mengetahui informasi tentang spesifikasi sinyal suara yang akan dikenali.
2. Menu untuk memilih proses *training* atau *recognizing*.
3. Menu untuk memilih jenis kata dari matriks data *training* yang akan disimpan. *Frame* ini akan muncul jika memilih menu *Training Data* pada *frame* 2.
4. Tempat untuk melihat tampilan sinyal suara, baik sinyal suara asli maupun sinyal suara hasil LPC (*preemphasis*).
5. *Frame* untuk menunjukkan representasi grafis hasil pengenalan suara.



Gambar 3.15 Antarmuka Program

3.2.6 Perancangan uji coba

Tujuan dilakukannya uji coba adalah untuk mengetahui pengaruh *signal noise* yang berbeda terhadap tingkat akurasi kata.

3.2.6.1 Skenario pengujian

Pengujian dilakukan untuk mengetahui pengaruh *signal noise* yang berbeda terhadap tingkat akurasi kata. Pengaruh *signal noise* tersebut ditunjukkan pada Tabel 3.1.

Pengujian dilakukan pada 5 x 4 sinyal kata uji yang masing-masing diucapkan oleh 4 orang laki-laki. Kata uji ini mempunyai format .wav 8KHz mono 8 bit.

Hasil pengujian tersebut kemudian dihitung keakuratannya dengan menggunakan *Word Error Rate* (WER) yang didefinisikan pada persamaan 2.1 dengan membagi jumlah kata yang salah dengan jumlah data.

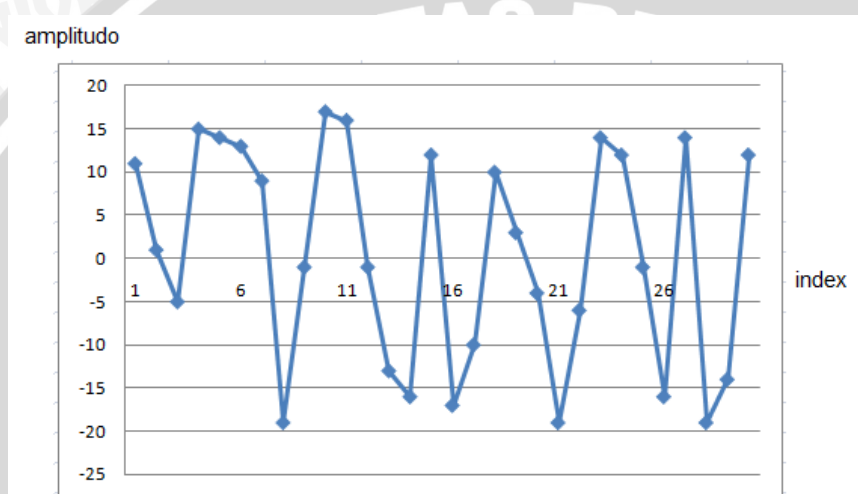
Tabel 3.1 Contoh Tabel Pengujian



File	Dikenali sebagai
Maju1.wav	
Maju2.wav	
...	
Mundur1.wav	
Mundur2.wav	
...	
WER(%)	

3.3 Contoh Perhitungan Proses Pembelajaran

Pada contoh perhitungan manual proses pembelajaran, digunakan data *signal input* yang ditunjukkan pada Gambar 3.16.



Gambar 3.16 Contoh *Signal Input*

Dari gambar *signal input* tersebut didapatkan data seperti pada Tabel 3.2 dimana i merupakan indeks dan $x(i)$ merupakan amplitudo pada *signal noise* dengan asumsi titik pusat sudah berada pada sumbu $y = 0$ sehingga tidak perlu dilakukan perhitungan untuk proses *centering*.

Tabel 3.2 Contoh *Signal Input*

i	$x(i)$	i	$x(i)$	i	$x(i)$
1	11	11	16	21	-19
2	1	12	-1	22	-6
3	-5	13	-13	23	14
4	15	14	-16	24	12
5	14	15	12	25	-1
6	13	16	-17	26	-16
7	9	17	-10	27	14
8	-19	18	10	28	-19
9	-1	19	3	29	-14
10	17	20	-4	30	12

Tabel 3.3 Hasil Perhitungan *Preprocessing*

i	$x(i)$ ternormalisasi	i	$x(i)$ ternormalisasi	i	$x(i)$ ternormalisasi
1	0.578947368	11	0.842105263	21	-1
2	0.052631579	12	-0.052631579	22	-0.315789474
3	-0.263157895	13	-0.684210526	23	0.736842105
4	0.789473684	14	-0.842105263	24	0.631578947
5	0.736842105	15	0.631578947	25	-0.052631579
6	0.684210526	16	-0.894736842	26	-0.842105263
7	0.473684211	17	-0.526315789	27	0.736842105
8	-1	18	0.526315789	28	-1
9	-0.052631579	19	0.157894737	29	-0.736842105
10	0.894736842	20	-0.210526316	30	0.631578947

Untuk *preprocessing*, yaitu normalisasi, dilakukan dengan membagi nilai amplitudo diskrit dengan nilai amplitudo maksimum dari amplitudo diskrit, sehingga dalam hal ini amplitudonya berada pada interval -1 sampai dengan 1. Rumus perhitungan normalisasi dapat dilihat pada persamaan 2.2.

Sebagai contoh normalisasi $x(1)$ dapat dijabarkan sebagai berikut:

$$| \text{maks} | = | -19 |$$

$$x(1) = \frac{11}{|-19|} = \frac{11}{19} = 0.578947368$$

Hasil perhitungan *preprocessing* pada semua indeks contoh sinyal *input* ditunjukkan pada Tabel 3.3.

Setelah dilakukan pemrosesan awal, kemudian dilakukan proses ekstraksi ciri. Dimulai dari *preemphasize*, *frame blocking*, *windowing*, autokorelasi, analisis LPC, dan konversi parameter LPC.

Untuk *preemphasize*, sinyal suara digital $s(i)$ dilewatkan pada suatu *filter digital* yang mempunyai orde rendah (*filter* orde satu). Nilai koefisien \tilde{a} yang digunakan dalam skripsi ini adalah 0.95. Rumus perhitungan *preemphasize* dapat dilihat pada persamaan 2.6 dan hasil perhitungan ditunjukkan oleh Tabel 3.4.

Tabel 3.4 Hasil Perhitungan *Preemphasize*

i	$s(i)$	i	$s(i)$	i	$s(i)$
1	0.578947368	11	-0.007894737	21	-0.8
2	-0.497368421	12	-0.852631579	22	0.634210526
3	-0.313157895	13	-0.634210526	23	1.036842105
4	1.039473684	14	-0.192105263	24	-0.068421053
5	-0.013157895	15	1.431578947	25	-0.652631579
6	-0.015789474	16	-1.494736842	26	-0.792105263
7	-0.176315789	17	0.323684211	27	1.536842105
8	-1.45	18	1.026315789	28	-1.7
9	0.897368421	19	-0.342105263	29	0.213157895
10	0.944736842	20	-0.360526316	30	1.331578947

Contoh perhitungan *preemphasize* indeks pertama dan kedua dapat dijabarkan sebagai berikut:

Untuk $s(1) = x(1)$, sehingga:

$$s(1) = 0.578947368$$

$$s(2) = 0.052631579 - (0.95 * 0.052631579) = -0.497368421$$

Setelah dilakukan proses *preemphasize*, kemudian sinyal keluaran tersebut dibagi menjadi beberapa *frame*. Dengan kecepatan cuplik sebesar 8 Khz maka tiap *frame* akan berisi (N) 240 *byte* data dengan jarak antar *frame* (M) 80 *byte* data, dan nilai $Q = 12$. Untuk jumlah *frame*, sebelumnya telah dilakukan perhitungan dengan membagi selisih data dan *overlap* dengan jarak antar *frame* sehingga terbentuk 40 *frame*. Untuk contoh perhitungan ini, *frame* yang digunakan adalah 2 *frame*, sehingga banyak data pada setiap *frame* = $\frac{30}{2} = 15$.

Tabel 3.5 Hasil Perhitungan *Frame Blocking*

Frame 1		Frame 2	
<i>i</i>	Frame Blocking	<i>i</i>	Frame Blocking
1	12.57894737	1	22.50526316
2	11.50263158	2	24.32368421
3	11.68684211	3	25.02631579
4	13.03947368	4	23.65789474
5	11.98684211	5	23.63947368
6	11.98421053	6	23.2
7	11.82368421	7	24.63421053
8	10.55	8	25.03684211
9	12.89736842	9	23.93157895
10	12.94473684	10	23.34736842
11	-0.094736844	11	23.20789474
12	-10.23157895	12	25.53684211
13	-7.610526312	13	22.3
14	-2.305263156	14	24.2131579
15	17.17894736	15	25.33157895

Setelah diketahui banyak *frame*, tiap *frame* diisi data dengan aturan pengindekan. Dimana data ke-*i* *frame* *t* adalah data ke-*i* + *tQ* dari sinyal setelah perataan, dengan *Q* adalah orde koefisien *cepstral* sebesar 12.

Contoh perhitungan *frame blocking* indeks pertama *frame* 1 dijabarkan sebagai berikut:

$t = 1$, karena terletak pada *frame* 1. Sehingga:

$$s_1^{(1)} = 0.578947368 + (1 * 12) = 12.57894737$$

Hasil *frame blocking* ditunjukkan pada Tabel 3.5.

Langkah berikutnya adalah memboboti masing-masing *frame* dengan *Hamming Window* ($w(n)$). Untuk rumus *Hamming Window* dapat dilihat pada persamaan 2.7. Hasil perhitungan *Hamming Window* ditunjukkan pada Tabel 3.6 dan hasil perhitungan *windowing* ditunjukkan pada Tabel 3.7.

Tabel 3.6 Hasil Perhitungan *Hamming Window*

i	$w(i)$
1	0.08
2	0.125508921
3	0.253031068
4	0.437334286
5	0.641951443
6	0.826395991
7	0.954172831
8	0.999999417
9	0.954808275
10	0.827541146
11	0.643379724
12	0.438763086
13	0.254177678
14	0.126146467
15	0.080002334

Sebagai contoh perhitungan *Hamming Window* untuk indeks pertama dapat dijabarkan sebagai berikut:

$$w(1) = 0.54 - 0.46 \cos\left(\frac{2\pi \cdot 0}{9}\right) = 0.08$$

sehingga

$$h(1) = 12.57894737 * 0.08 = 1.006315789.$$

Tabel 3.7 Hasil Perhitungan *Windowing*

<i>Frame 1</i>		<i>Frame 2</i>	
i	$h(i)$	i	$h(i)$
1	1.006315789	1	1.800421053
2	1.44368288	2	3.052839364
3	2.957134136	3	6.332435406
4	5.702608915	4	10.34640851
5	7.694970588	5	15.17539425
6	9.903703529	6	19.17238698
7	11.28183824	7	23.5052944
8	10.54999385	8	25.0368275
9	12.31451409	9	22.85006961

10	10.71230236	10	19.32090802
11	-0.060951765	11	14.93148892
12	-4.489239157	12	11.20462366
13	-1.934425903	13	5.66816221
14	-0.290800802	14	3.05440432
15	1.374355878	15	2.02658543

Setelah dilakukan *windowing*, kemudian masing-masing *frame* diautokorelasikan untuk menghasilkan koefisien autokorelasi. Orde LPC yang digunakan menggunakan orde 10. Untuk rumus per-hitungan koefisien autokorelasi dapat dilihat pada persamaan 2.8.

Sebagai contoh perhitungan autokorelasi dengan m ke-0 pada *frame* ke-1 dijabarkan sebagai berikut:

$$\begin{aligned}
 r_1^{(0)} &= (1.006315789 * 1.006315789) + (1.44368288 * 1.44368288) \\
 &\quad + (2.957134136 * 2.957134136) + \dots + (1.374355878 * \\
 &\quad 1.374355878) \\
 &= 1.012671468 + 2.084220259 + 8.7446423 + \dots + 1.88885408 \\
 &= 732.5125279.
 \end{aligned}$$

Hasil perhitungan koefisien autokorelasi ditunjukkan pada Tabel 3.8.

Tabel 3.8 Hasil Perhitungan Koefisien Autokorelasi

Frame 1		Frame 2	
m	$r(m)$	m	$r(m)$
0	732.5125279	0	3226.404478
1	643.7332736	1	3133.837706
2	483.6071954	2	2883.012866
3	351.6657831	3	2509.428464
4	241.347542	4	2061.215037
5	149.5898519	5	1594.940784
6	71.19126072	6	1160.774082
7	17.24114401	7	792.9108326
8	-0.183254939	8	502.8159825
9	-2.241083227	9	295.1731538
10	-3.345426499	10	159.3385019

Setelah didapatkan koefisien autokorelasi, langkah berikutnya adalah analisis LPC, dimana nilai autokorelasi pada setiap *frame* diubah menjadi satu set LPC parameter yaitu koefisien LPC yang didefinisikan pada persamaan 2.10 sampai 2.16.

Contoh perhitungan analisa LPC pada *frame* ke-1 dapat dijabarkan sebagai berikut:

$$E_0 = 732.5125279$$

untuk $i = 1$

$$k_1 = \frac{643.7332736}{732.5125279} = 0.878801726$$

$$\alpha_{1,1} = 0.878801726$$

$$E_1 = (1 - (0.878801726)^2) * 732.5125279 = 166.7986162.$$

Iterasi dilakukan dengan nilai $i = 1$ sampai dengan p (dalam skripsi ini p sebagai orde LPC digunakan orde 10). Sehingga akan ditemukan nilai $\alpha_{1,10}$ sampai $\alpha_{10,10}$ yang merupakan hasil akhir dari analisa LPC, yaitu koefisien LPC. Koefisien LPC yang diperoleh pada *frame* ke-1 dapat dilihat pada Tabel 3.9.

Tabel 3.9. Koefisien LPC

Frame 1		Frame 2	
m	$a(m)$	m	$a(m)$
1	1.622255409	1	1.764155148
2	-1.353302315	2	-0.44642114
3	1.110013985	3	-0.271913075
4	-0.900751441	4	-0.439367157
5	0.689295399	5	0.081989916
6	-0.50252347	6	0.180968082
7	0.214828224	7	0.607310892
8	-0.019575477	8	-0.515133434
9	-0.00765212	9	-0.141532214
10	0.002790781	10	0.151485451

Langkah terakhir dalam proses ekstraksi ciri adalah pengubahan parameter LPC menjadi koefisien *cepstral*. Koefisien LPC yang diperoleh

dari analisis LPC dikonversi menjadi set koefisien *cepstral*, $c_{(m)}$ yang dinyatakan dalam persamaan 2.18.

Tabel 3.10 Koefisien *Cepstral*

Frame 1		Frame 2	
m	$c(m)$	m	$c(m)$
1	1.622255409	1	1.764155148
2	-0.037446009	2	1.109700553
3	0.337715324	3	0.770690837
4	-0.014336624	4	0.212725392
5	0.087594691	5	-0.099936419
6	0.013874936	6	-0.272456743
7	-0.113850433	7	0.120616199
8	0.007522828	8	-0.05964688
9	0.023186695	9	-0.100133447
10	-0.013720527	10	0.075087998

Contoh perhitungan koefisien *cepstral* untuk *frame* ke-1 iterasi $m = 1$ dan $m = 2$ dapat dijabarkan sebagai berikut:

$$c_1 = a_1 = 1.622255409$$

$$c_2 = -1.353302315 + \left(\frac{1}{2} * 1.622255409 * 1.622255409\right) \\ = -0.037446009$$

Hasil perhitungan koefisien *cepstral* dapat dilihat pada Tabel 3.10.

Setelah diperoleh koefisien *cepstral* dari suatu *signal noise* dalam proses pembelajaran, langkah berikutnya adalah proses pengenalan dengan menggunakan metode *Piecewise Dynamic Time Warping* (PDTW). Metode ini memerlukan dua inputan koefisien *signal noise* dan kemudian membandingkan kedua rangkaian koefisien sinyal tersebut dengan beberapa tahapan. Hasil akhir dari metode ini berupa pengambilan keputusan untuk kata yang paling tepat.

BAB IV HASIL DAN PEMBAHASAN

4.1 Lingkungan Implementasi

Pada bab ini akan dibahas mengenai implementasi dari pe-rancangan pada bab 3. Berikut ini uraian spesifikasi perangkat keras dan perangkat lunak yang digunakan dalam sistem pengenalan suara serta implementasi metode yang digunakan.

4.1.1 Lingkungan implementasi perangkat keras

Perangkat keras yang digunakan dalam menerapkan sistem pe-ngenalan suara dengan metode *Piecewise Dynamic Time Warping* (PDTW) adalah sebagai berikut:

1. *Processor Intel® Atom™* 1.67 GHz
2. Memori 2.00 GB
3. *HardDisk* 298 GB

4.1.2 Lingkungan implementasi perangkat lunak

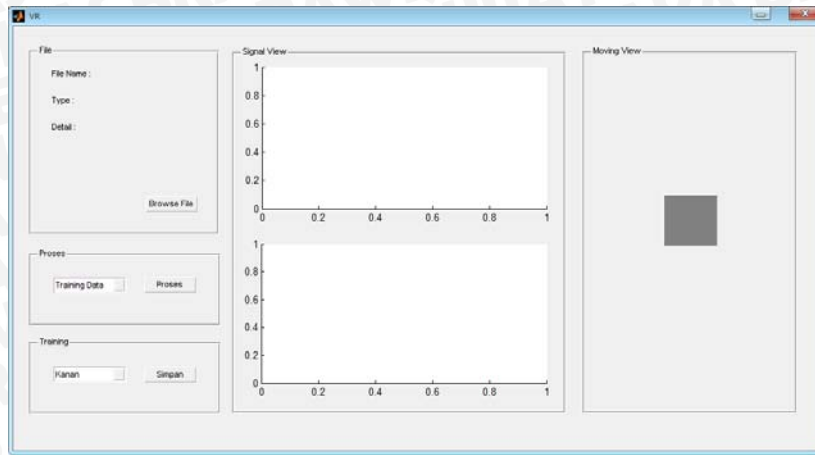
Perangkat lunak yang digunakan dalam menerapkan sistem pe-ngenalan suara dengan metode *Piecewise Dynamic Time Warping* (PDTW) adalah sebagai berikut:

1. Sistem operasi Microsoft Windows 7 Starter
2. Bahasa pemrograman yang digunakan dalam menerapkan sistem pengenalan suara adalah skrip Matlab versi 7.7.0.471 (R2008b)

4.2 Implementasi Perangkat Lunak

Perangkat lunak pengenalan suara dengan metode *Piecewise Dynamic Time Warping* (PDTW) ini memiliki prinsip kerja dapat mengolah sinyal suara, mengenalinya dengan membandingkan informasi suara yang ada pada data referensi dengan informasi suara yang menjadi masukan sistem pengenalan suara tersebut, dan mengubahnya menjadi representasi grafis gerak arah.

Secara umum sistem pengenalan suara dengan metode *Piecewise Dynamic Time Warping* (PDTW) dapat dibagi menjadi 3 tahap yaitu pemrosesan awal, ekstraksi ciri, dan sistem pengenalan kata yang terdiri dari *training* dan *recognize*. Tampilan utama dari implemen-tasi program ini ditunjukkan pada Gambar 4.1.



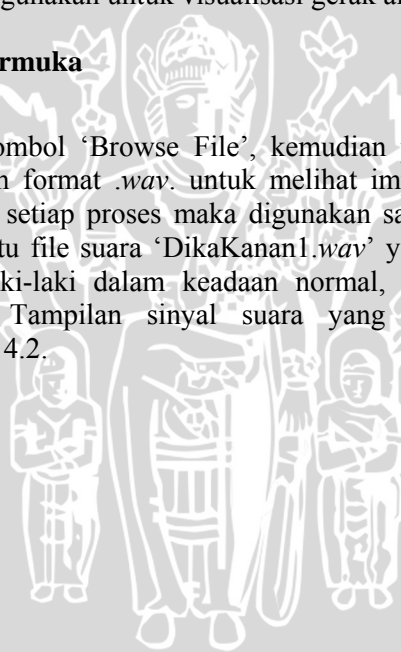
Gambar 4.1 Tampilan Utama

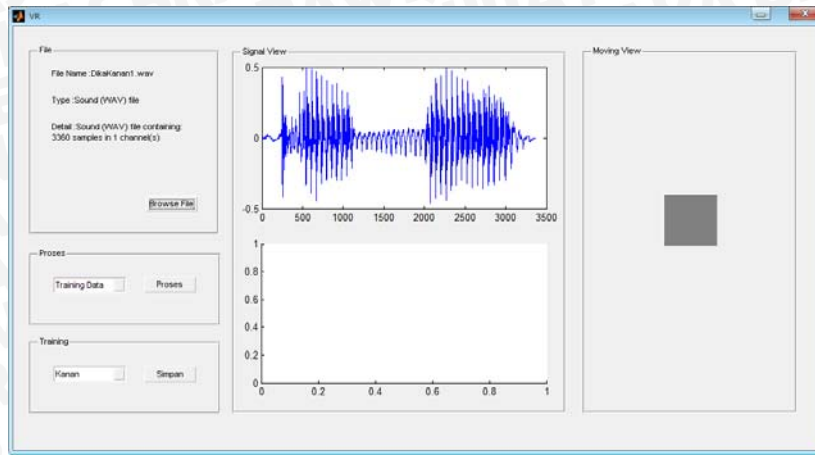
Pada tampilan utama di atas terdapat 4 *frame*, yaitu spesifikasi sinyal suara, tampilan sinyal, *training*, dan *moving view*. Spesifikasi sinyal suara berisi informasi tentang sinyal suara yang akan dikenali, terdiri dari nama file, panjang file, ukuran data dan *audio format* dari file suara. Tampilan sinyal digunakan untuk menampilkan sinyal hasil normalisasi dan sinyal hasil ekstraksi ciri (*preemphasis*) dengan metode LPC. *Training* digunakan untuk melatih sinyal yang ada pada data referensi dan menyimpannya, sedangkan *moving view* digunakan untuk visualisasi gerak arah grafis.

4.2.1 Implementasi Antarmuka

4.2.1.1 Pemrosesan awal

Pertama-tama tekan tombol 'Browse File', kemudian pilih file suara yang akan dikenali dalam format *.wav*. untuk melihat implementasi dan bentuk sinyal suara pada setiap proses maka digunakan salah satu sinyal suara sebagai contoh, yaitu file suara 'DikaKanan1.wav' yang merupakan sampel suara seorang laki-laki dalam keadaan normal, pada frekuensi 8000Hz, resolusi 8bit. Tampilan sinyal suara yang akan dikenali ditunjukkan pada Gambar 4.2.





Gambar 4.2 Tampilan Sinyal Suara

Pada Gambar 4.2 dapat dilihat informasi spesifikasi sinyal suara DikaMaju1.wav beserta tampilan sinyalnya.

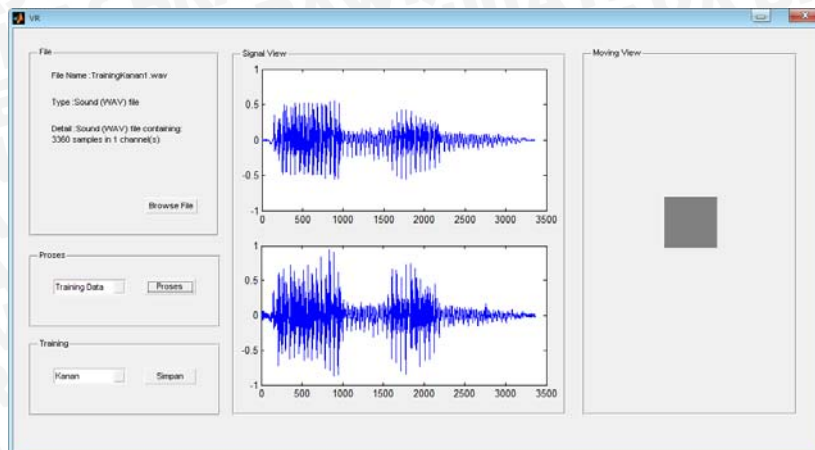
4.2.1.2 Sistem pengenalan

Pada sistem pengenalan ini dibagi menjadi 2 tahap yaitu *training* dan *recognize*.

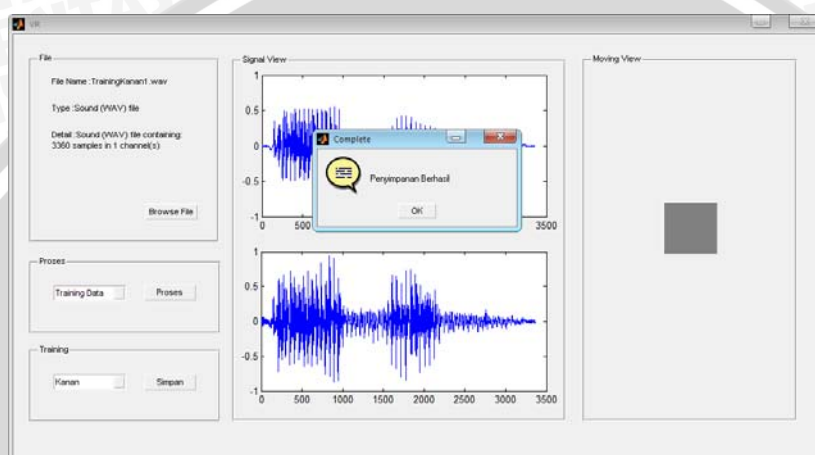
4.2.1.2.1 Training

Untuk melakukan proses *training*, dilakukan pemrosesan terlebih dahulu dengan memilih *Training Data* > *Proses*. Sebagai contoh file *training* yang digunakan adalah 'TrainingKanan1.wav'. Tampilan sinyal hasil normalisasi dan LPC akan tampak seperti pada Gambar 4.3.

Kemudian hasil pemrosesan tersebut disimpan dengan memilih jenis *Kanan* pada panel *Training*, kemudian menekan tombol *Simpan*. Jika tidak terjadi kesalahan maka akan muncul pemberitahuan 'Penyimpanan Berhasil'. Tampilan penyimpanan data *training* dapat dilihat pada Gambar 4.4.



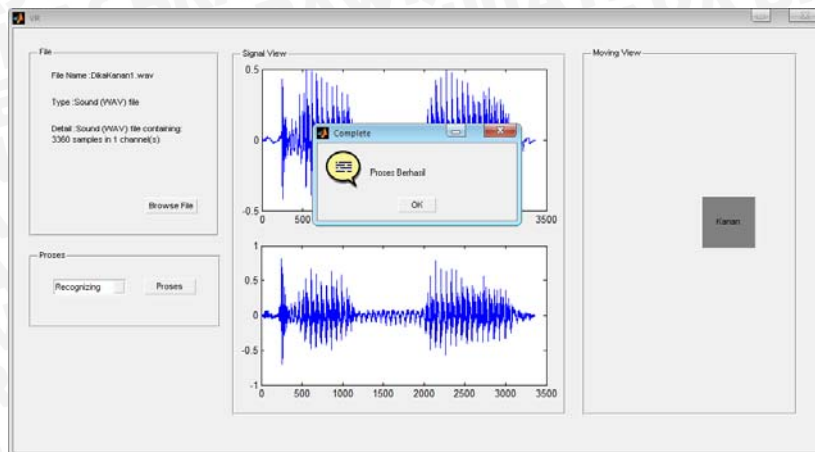
Gambar 4.3 Tampilan Pemrosesan Sinyal



Gambar 4.4 Tampilan Penyimpanan Data *Training*

4.2.1.2.2 *Recognizing*

Setelah training kata 'Kanan' tersimpan, kemudian akan dilakukan proses *recognizing* pada data latih. Tampilan proses *Recognizing* untuk file 'DikaKanan1.wav' ditunjukkan pada Gambar 4.5.



Gambar 4.5 Tampilan Proses *Recognizing*

Jika proses berhasil, maka grafis pada *moving view* akan bergerak ke kanan sesuai dengan hasil perhitungan metode PDTW dan muncul pemberitahuan 'Proses Berhasil'.

4.2.2 Deskripsi perangkat lunak

4.2.2.1 Pemrosesan awal

Tahap pemrosesan awal dilakukan untuk menyetarakan semua sinyal suara dan mengubah sinyal suara dari *analog* menjadi *digital*. Pada tahap ini terdiri dari normalisasi.

Proses normalisasi bertujuan untuk menyamakan amplitudo maksimum sinyal suara sehingga tidak ada pengaruh perubahan amplitudo pada pemrosesan berikutnya. Normalisasi dilakukan dengan membagi masing-masing nilai amplitudo diskrit dengan nilai amplitudo maksimumnya. Baris kode yang digunakan dalam proses normalisasi ditunjukkan pada *Sourcecode* 4.1.

```
function [normal] = Normalisasi(data)

if ~isempty(data)
    rata = 0;
    maks = -1;

    for i = 1:length(data)
        rata = (rata + data(i))/length(data);
```

```

end;

for i = 1:length(data)
    normal(i) = data(i); - rata;
end;

for i = 1:length(data)
    if normal(i) >= maks
        maks = normal(i);
    else
        maks = maks;
    end;
end;

for i = 1:length(data)
    normal(i) = normal(i)/maks;
end;

else
    normal = [];
end;

end

```

Sourcecode 4.1 Proses Normalisasi

4.2.2.2 Ekstraksi ciri (LPC)

Proses ekstraksi ciri mengubah representasi amplitudo diskrit dalam domain waktu menjadi representasi energi dalam domain frekuensi untuk mendapatkan karakteristik sinyal suara. Ekstraksi ciri pada skripsi ini menggunakan metode *Linear Predictive Coding* (LPC).

Baris kode yang menunjukkan fungsi *preemphasis* ditunjukkan pada *Sourcecode 4.2*.

```

function [s]=Preemphasize(data)

if ~isempty(data)
    s = data;
    a = 0.95;

    for i = 2:length(data)
        s(1) = s(1);
        s(i) = s(i) - (a * s(i-1));
    end
end

```

```
end;  
disp(s);  
  
else  
s = 0;  
end;  
  
end
```

Sourcecode 4.2 Kode Preemphasis

Baris kode yang menunjukkan fungsi *frame blocking* untuk membagi sinyal suara menjadi beberapa *frame* serta pengisian dengan aturan pengindekan ditunjukkan pada Sourcecode 4.3.

```
function [s] = FrameBlocking(data)  
  
if ~isempty(data)  
Q = 12;  
frame = 40;  
  
tempN = abs(length(data)/frame);  
N = uint32(tempN);  
disp(N);  
  
y = 1;  
  
for i = 1:length(data)  
if y <= frame  
for j = 1:N  
y = 1;  
s(i) = data(i) + (y * Q);  
end;  
y = y+1;  
end;  
end;  
  
else  
s = [];  
end;  
  
end
```

Sourcecode 4.3 Kode Frame Blocking

Baris kode yang menunjukkan fungsi *windowing* ditunjukkan pada *Sourcecode 4.4*.

```
function [h] = Windowing(data)

if ~isempty(data)

    frame = 40;

    tempN = abs(length(data)/frame);
    N = uint32(tempN);
    disp(N);

    y = 1;

    for i = 1:length(data)
        if y <= frame
            for j = 1:N
                y = 1;
                hitung = (2*pi*j)/N-1;
                tempC = double(hitung);
                w(j) = 0.54 - (0.46* cos(tempC));
            end;
            y = y+1;
        end;
        h(i) = data(i) * w(j);
    end;
else
    h=[];
end;

end
```

Sourcecode 4.4 Kode Windowing

Baris kode yang menunjukkan fungsi analisa autokorelasi ditunjukkan pada *Sourcecode 4.5*.

```
function [r] = Autokorelasi(data)

if ~isempty(data)
    frame = 40;
    P = 10;
```

```

tempN = length(data)/frame;
N = uint32(tempN);
disp(N);

x = 1;

if x <= length(data)
    for i = 1:frame
        for j = 1:N
            h(i,j) = data(x);
            x = x+1;
        end;
    end;
end;

for i = 1:frame
    for m = 1:(P+1)
        temp = 0;
        for n = 1:(N-1-m)
            temp = temp + (h(i,n)*h(i,n+m));
        end;
        r(i,m) = temp;
    end;
end;

else
    r=[];
end;

end

```

Sourcecode 4.5 Kode Analisa Autokorelasi

Baris kode yang menunjukkan fungsi *levinson-durbin* untuk mendapatkan koefisien LPC ditunjukkan pada *Sourcecode 4.6*.

```

function [nA] = AnalisaLPC(data)

if ~isempty(data)

    P = 10;
    frame = 40;
    temp = 0;

```

```

for x = 1:frame
    E(x,1) = data(x,1);
    k(x,1) = data(x,1)/E(x,1);
    a(x,1,1) = k(x,1);

    for i = 2:P
        for j = 1:i-1
            temp = temp + (a(x,j,i-1)*data(x,i-j));
            k(x,i) = (data(x,i)-temp)/E(x,i-1);
            a(x,j,i) = a(x,j,i-1)-(k(x,i)*a(x,i-j,i-1));
        end;
        a(x,i,i) = k(x,i);
        E(x,i) = (1-(k(x,i)*k(x,i)))*E(x,i-1);
    end;
end;

for x = 1:frame
    for m = 1:P
        nA(x,m) = a(x,m,P);
    end;
end;

else
    nA = [];
end;

end

```

Sourcecode 4.6 Kode Levinson-Durbin

Baris kode yang menunjukkan fungsi untuk mengkonversi koefisien LPC menjadi koefisien cepstral ditunjukkan pada *Sourcecode 4.7*.

```

function [c] = KonversiLPC(data)

if ~isempty(data)

    frame = 40;
    P = 10;
    temp = 0;

    for i = 1:frame
        for m = 2:P;
            c(i,1) = data(i,1);
            for k = 1:m-1

```



```

        temp = temp + (k/m)*(c(i,k)*data(i,m-k));
    end;
    c(i,m) = data(i,m) + temp;
end;
end;

else
    c = [];
end;

end

```

Sourcecode 4.7 Kode Konversi Koefisien Cepstral

4.2.2.3 Recognizing

Proses pengenalan dilakukan dengan metode *Piecewise Dynamic Time Warping* yang ditunjukkan pada Sourcecode 4.8.

```

function [Dist] = PiDTW(t,r)

[rows,N] = size(t);
[rows,M] = size(r);

for n=1:N
    for m=1:M
        d(n,m)=(t(n)-r(m))^2;
    end
end

D = zeros(size(d));
D(1,1) = d(1,1);

for n = 2:N
    D(n,1) = d(n,1)+D(n-1,1);
end

for m = 2:M
    D(1,m) = d(1,m)+D(1,m-1);
end

for n = 2:N
    for m = 2:M
        D(n,m) = d(n,m)+min([D(n-1,m),D(n-1,m-1),
            D(n,m-1)]);
    end
end

```

```

end
end

Dist = D(N,M);
n = N;
m = M;
k = 1;
w = [];
w(1,:) = [N,M];

while ((n+m)~=2)
    if (n-1) == 0
        m = m-1;
    elseif (m-1) == 0
        n = n-1;
    else
        [values,number] = min([D(n-1,m),D(n,m-1),
                               D(n-1,m-1)]);
        switch number
            case 1
                n = n-1;
            case 2
                m = m-1;
            case 3
                n = n-1;
                m = m-1;
            end;
        end;
        k = k+1;
        w = cat(1,w,[n,m]);
    end;
end;

```

Sourcecode 4.8 Piecewise Dynamic Time Warping

4.3 Implementasi Pengujian

Pada subbab ini akan dibahas mengenai pengujian sistem dan metode PDTW yang dipakai dalam implementasi pengenalan pola suara. Dari pengujian ini dapat diketahui tingkat ketepatan dalam mengatur gerak arah sesuai dengan perintah kata yang diberikan.

4.4.1 Hasil Pengujian

Proses *training* dilakukan pada 8 sinyal kata, yaitu 2 sinyal untuk masing-masing kata. Sedangkan uji coba dilakukan pada 80 sinyal suara

yang diucapkan oleh 4 orang laki-laki dimana 1 orang mengucapkan 4 kata masing-masing sebanyak 5 kali. Sehingga total sinyal kata yang digunakan dalam skripsi ini adalah sebanyak 88 sinyal kata. Pengujian dilakukan untuk mengetahui tingkat ketepatan dalam mengatur gerak arah sesuai dengan perintah kata yang diberikan dari *sample* orang yang berbeda.

Untuk hasil pengujian kata 'Maju' ditunjukkan pada Tabel 4.1.

Untuk perhitungan tingkat akurasi menggunakan *Word Error Rate* (WER) yaitu dihitung berapa jumlah kata yang dikenali salah dibagi dengan jumlah data sesuai dengan persamaan 2.1. Pada Tabel 4.1, data yang dicetak tebal merupakan data kata yang dikenali salah.

Dari Tabel 4.1 didapatkan:

Jumlah data uji salah = 7

Jumlah data uji = 20

Jadi didapatkan tingkat akurasi WER sebesar:

$$\text{WER (\%)} = \frac{7}{20} \times 100\% = 35\%$$

Tabel 4.1 Hasil Pengujian Kata Maju

File	Dikenali sebagai
DikaMaju1.wav	Mundur
DikaMaju2.wav	Maju
DikaMaju3.wav	Maju
DikaMaju4.wav	Mundur
DikaMaju5.wav	Maju
GitaMaju1.wav	Mundur
GitaMaju2.wav	Kiri
GitaMaju3.wav	Maju
GitaMaju4.wav	Maju
GitaMaju5.wav	Maju
RuliskMaju1.wav	Kanan
RuliskMaju2.wav	Maju
RuliskMaju3.wav	Maju
RuliskMaju4.wav	Kiri
RuliskMaju5.wav	Maju
YasserMaju1.wav	Maju
YasserMaju2.wav	Maju
YasserMaju3.wav	Maju

YasserMaju4.wav	Maju
YasserMaju5.wav	Kanan

Untuk hasil pengujian kata ‘Mundur’ ditampilkan pada Tabel 4.2.
Dari Tabel 4.2 didapatkan:

Jumlah data uji salah = 8
Jumlah data uji = 20

Jadi didapatkan tingkat akurasi WER sebesar:

$$\text{WER (\%)} = \frac{8}{20} \times 100\% = 40\%$$



Tabel 4.2 Hasil Pengujian Kata Mundur

File	Dikenali sebagai
DikaMundur1.wav	Mundur
DikaMundur2.wav	Mundur
DikaMundur3.wav	Maju
DikaMundur4.wav	Mundur
DikaMundur5.wav	Mundur
GitaMundur1.wav	Mundur
GitaMundur2.wav	Kanan
GitaMundur3.wav	Kiri
GitaMundur4.wav	Mundur
GitaMundur5.wav	Mundur
RuliskMundur1.wav	Mundur
RuliskMundur2.wav	Mundur
RuliskMundur3.wav	Mundur
RuliskMundur4.wav	Mundur
RuliskMundur5.wav	Mundur
YasserMundur1.wav	Kanan
YasserMundur2.wav	Kanan
YasserMundur3.wav	Maju
YasserMundur4.wav	Maju
YasserMundur5.wav	Kanan

Untuk hasil pengujian kata ‘Kanan’ ditampilkan pada Tabel 4.3.
Dari Tabel 4.3 didapatkan:

Jumlah data uji salah = 11

Jumlah data uji = 20

Jadi didapatkan tingkat akurasi WER sebesar:

$$\text{WER (\%)} = \frac{11}{20} \times 100\% = 55\%$$

Tabel 4.3 Hasil Pengujian Kata Kanan

File	Dikenali sebagai
DikaKanan1.wav	Maju
DikaKanan2.wav	Kanan
DikaKanan3.wav	Maju
DikaKanan4.wav	Kanan
DikaKanan5.wav	Kanan
GitaKanan1.wav	Kanan
GitaKanan2.wav	Kanan
GitaKanan3.wav	Maju
GitaKanan4.wav	Maju
GitaKanan5.wav	Mundur
RuliskKanan1.wav	Maju
RuliskKanan2.wav	Maju
RuliskKanan3.wav	Maju
RuliskKanan4.wav	Maju
RuliskKanan5.wav	Kiri
YasserKanan1.wav	Mundur
YasserKanan2.wav	Kanan
YasserKanan3.wav	Kanan
YasserKanan4.wav	Kanan
YasserKanan5.wav	Kanan

Untuk hasil pengujian kata ‘Kiri’ ditampilkan pada Tabel 4.4.
 Dari Tabel 4.4 didapatkan:

Jumlah data uji salah = 13

Jumlah data uji = 20

Jadi didapatkan tingkat akurasi WER sebesar:

$$\text{WER (\%)} = \frac{13}{20} \times 100\% = 65\%$$

Tabel 4.4 Hasil Pengujian Kata Kiri

File	Dikenali sebagai
DikaKiri1.wav	Kiri
DikaKiri2.wav	Kiri
DikaKiri3.wav	Mundur
DikaKiri4.wav	Maju
DikaKiri5.wav	Kiri
GitaKiri1.wav	Kiri
GitaKiri2.wav	Mundur
GitaKiri3.wav	Maju
GitaKiri4.wav	Kanan
GitaKiri5.wav	Kanan
RuliskKiri1.wav	Maju
RuliskKiri2.wav	Kanan
RuliskKiri3.wav	Maju
RuliskKiri4.wav	Kiri
RuliskKiri5.wav	Kiri
YasserKiri1.wav	Maju
YasserKiri2.wav	Kiri
YasserKiri3.wav	Maju
YasserKiri4.wav	Kanan
YasserKiri5.wav	Maju

Sehingga didapatkan nilai total WER:

Jumlah data uji salah = 39

Jumlah data uji = 80

Jadi didapatkan tingkat akurasi WER sebesar:

$$\text{WER (\%)} = \frac{39}{80} \times 100\% = 48.75\%$$

Dengan nilai WER 48.75% untuk keseluruhan data dengan rincian 35% untuk kata 'Maju', 40% untuk kata 'Mundur', 55% untuk kata 'Kanan' dan 65% untuk kata 'Kiri', maka membuktikan bahwa metode LPC dan PDTW dapat digunakan untuk pengenalan pola suara.

4.4.2 Analisa Hasil Pengujian

Berdasarkan uji coba yang dilakukan pada sinyal kata ‘Maju’ pada Tabel 4.1 didapatkan hasil:

Jumlah kata yang benar : 13

Jumlah kata yang salah : 7

Salah satu kata yang dikenali salah pada Tabel 4.1 tersebut adalah kata *DikaMaju1.wav*. Kata tersebut memiliki nilai PDTW sebagai berikut:

$h1 = 10716.3$

$h2 = 9347.27$

$h3 = 9197.18$

$h4 = 15833.8$

$h5 = 23800.3$

$h6 = 18104.1$

$h7 = 7774.09$

$h8 = 10076.5$

Dimana $h1$ merupakan hasil dari perbandingan koefisien *cepstral* data uji *DikaMaju1.wav* dengan data *training TrainingKanan1.wav*, $h2$ merupakan hasil perbandingan koefisien *cepstral* data uji *DikaMaju1.wav* dengan data *training TrainingKanan2.wav*, $h3$ merupakan hasil dari perbandingan koefisien *cepstral* data uji *DikaMaju1.wav* dengan data *training TrainingKiri1.wav*, $h4$ merupakan hasil dari perbandingan koefisien *cepstral* data uji *DikaMaju1.wav* dengan data *training TrainingKiri2.wav*, $h5$ merupakan hasil dari perbandingan koefisien *cepstral* data uji *DikaMaju1.wav* dengan data *training TrainingMaju1.wav*, $h6$ merupakan hasil dari perbandingan koefisien *cepstral* data uji *DikaMaju1.wav* dengan data *training TrainingMaju2.wav*, $h7$ merupakan hasil dari perbandingan koefisien *cepstral* data uji *DikaMaju1.wav* dengan data *training TrainingMundur1.wav*, dan $h8$ merupakan hasil dari perbandingan koefisien *cepstral* data uji *DikaMaju1.wav* dengan data *training TrainingMundur2.wav*.

Untuk menentukan *output* dilakukan pemilihan nilai paling minimum dari $h1$ sampai dengan $h8$. Jika nilai minimum tersebut $h1$ atau $h2$ maka *output* ‘Kanan’, jika nilai minimum $h3$ atau $h4$ maka *output* ‘Kiri’, jika

nilai minimum $h5$ atau $h6$ maka *output* ‘Maju’, dan jika nilai minimum $h7$ atau $h8$ maka *output* ‘Mundur’.

Pada kata *DikaMaju1.wav* nilai paling minimum adalah $h7$, yaitu 7774.09, sehingga *output* yang muncul adalah ‘Mundur’. Hal ini dikarenakan pada sinyal kata tersebut memiliki perbedaan amplitudo yang cukup besar dengan sinyal dari data training, sehingga koefisien cepstral yang dihasilkan dari metode LPC juga memiliki disparitas yang cukup besar. Dan saat dilakukan proses pengenalan menggunakan metode PDTW, didapatkan hasil jarak terdekat atau paling minimum dengan kata ‘Mundur’ pada data *training* sehingga sinyal suara tersebut dikenali sebagai kata ‘Mundur’.

Sebagai contoh, berikut adalah analisa untuk kata ‘Maju’ dari data training *TrainingMaju1.wav*, serta *TrainingMaju2.wav* dan data uji *DikaMaju1.wav*, *DikaMaju2.wav*, *DikaMaju3.wav*, *DikaMaju4.wav*, dan *DikaMaju5.wav*.

Untuk sinyal *TrainingMaju1.wav* memiliki koefisien *cepstral* seperti pada Tabel 4.5. Data yang ditampilkan adalah data indeks pertama dari setiap *frame*.

Untuk sinyal *TrainingMaju2.wav* memiliki koefisien *cepstral* seperti pada Tabel 4.6. Data yang ditampilkan adalah data indeks pertama dari setiap *frame*.

Untuk sinyal *DikaMaju1.wav* memiliki koefisien *cepstral* seperti pada Tabel 4.7. Data yang ditampilkan adalah data indeks pertama dari setiap *frame*.

Untuk sinyal *DikaMaju2.wav* memiliki koefisien *cepstral* seperti pada Tabel 4.8. Data yang ditampilkan adalah data indeks pertama dari setiap *frame*.

Untuk sinyal *DikaMaju3.wav* memiliki koefisien *cepstral* seperti pada Tabel 4.9. Data yang ditampilkan adalah data indeks pertama dari setiap *frame*.

Untuk sinyal *DikaMaju4.wav* memiliki koefisien *cepstral* seperti pada Tabel 4.10. Data yang ditampilkan adalah data indeks pertama dari setiap *frame*.

Untuk sinyal *DikaMaju5.wav* memiliki koefisien *cepstral* seperti pada Tabel 4.11. Data yang ditampilkan adalah data indeks pertama dari setiap *frame*.

Tabel 4.5 Koefisien *Cepstral* Sinyal *TrainingMaju1.wav*

Frame	Koefisien <i>cepstral</i>	Frame	Koefisien <i>cepstral</i>
-------	---------------------------	-------	---------------------------

1	183.7061162	21	-1010.707692
2	-7302.332801	22	-897.2829505
3	-6546.121375	23	-793.418268
4	-5925.482632	24	-704.4568955
5	-5315.479033	25	-619.068494
6	-4790.556955	26	-542.9887638
7	-4298.923221	27	-475.0315745
8	-3862.765635	28	-415.3057755
9	-3479.904445	29	-357.0995988
10	-3113.557638	30	-304.0450211
11	-2787.339427	31	-266.4487435
12	-2506.183722	32	-225.6656849
13	-2231.986102	33	-183.9326442
14	-2021.126202	34	-151.3171856
15	-1791.632724	35	-121.8149602
16	-1608.724366	36	-94.96025482
17	-1429.306624	37	-70.18843357
18	-1274.334994	38	-48.1431488
19	-1142.098166	39	-27.91520046
20	183.7061162	40	-9.627032255
total			-68552.52459

Tabel 4.6 Koefisien *Cepstral* Sinyal *TrainingMaju2.wav*

Frame	Koefisien <i>cepstral</i>	Frame	Koefisien <i>cepstral</i>
1	183.4493134	21	-1014.684361
2	-7276.986906	22	-905.3259601
3	-6586.605953	23	-803.2776177
4	-5932.083514	24	-716.2053906
5	-5328.171197	25	-623.2337899
6	-4786.000738	26	-552.1810918

7	-4313.281716	27	-479.3005252
8	-3891.09198	28	-418.6707627
9	-3477.151579	29	-365.2689923
10	-3127.584415	30	-313.5761618
11	-2819.816697	31	-265.4852787
12	-2522.9233	32	-225.8155356
13	-2257.93289	33	-188.8482592
14	-2028.117008	34	-154.6381183
15	-1814.030908	35	-124.9062185
16	-1617.350604	36	-96.9161157
17	-1449.387245	37	-72.41295945
18	-1290.53431	38	-49.78235514
19	-1154.051027	39	-29.68201926
20	183.4493134	40	-11.12089347
total			-68763.83338

UNIVERSITAS BRAWIJAYA

Tabel 4.7 Koefisien Cepstral Sinyal *DikaMajul.wav*

Frame	Koefisien cepstral	Frame	Koefisien cepstral
1	184.3695435	21	-1024.242709
2	-7323.854936	22	-920.913416
3	-6648.737169	23	-806.0301385
4	-5915.572891	24	-723.5120403
5	-5368.237925	25	-625.4721706
6	-4819.4026	26	-556.3346627
7	-4360.611647	27	-486.8190792
8	-3862.123104	28	-419.798158

9	-3552.557726	29	-369.0096487
10	-3110.634336	30	-316.3012347
11	-2856.917021	31	-268.0421971
12	-2502.27472	32	-227.3284208
13	-2296.866415	33	-191.9123242
14	-2035.115969	34	-155.7848087
15	-1809.191671	35	-126.4023298
16	-1621.829597	36	-98.25319175
17	-1452.414585	37	-73.89789823
18	-1290.696589	38	-50.75373411
19	-1161.968214	39	-30.93364262
20	184.3695435	40	-12.1189557
total			-69294.70001

Tabel 4.8 Koefisien *Cepstral* Sinyal *DikaMaju2.wav*

<i>Frame</i>	Koefisien <i>cepstral</i>	<i>Frame</i>	Koefisien <i>cepstral</i>
1	183.7064027	21	-1021.848109
2	-7229.309105	22	-897.743322
3	-6548.149478	23	-804.7188362
4	-5865.657859	24	-700.1348447
5	-5313.312072	25	-625.4554079
6	-4794.640199	26	-542.5856864
7	-4298.245623	27	-482.1761775
8	-3870.57518	28	-414.5535024
9	-3430.642251	29	-363.380941
10	-3115.099002	30	-307.7203689

11	-2793.878574	31	-266.8562723
12	-2539.177774	32	-221.9916896
13	-2215.518651	33	-188.5904031
14	-2019.913537	34	-153.495269
15	-1787.631253	35	-123.4776132
16	-1625.740642	36	-95.97400106
17	-1428.327881	37	-71.45641076
18	-1288.412612	38	-49.46276618
19	-1135.88966	39	-28.96983009
20	183.7064027	40	-10.64919432
total			-68477.20035

Tabel 4.9 Koefisien Cepstral Sinyal DikaMaju3.wav

Frame	Koefisien cepstral	Frame	Koefisien cepstral
1	184.2854463	21	-1026.442601
2	-7254.977235	22	-896.6062977
3	-6607.027728	23	-806.5308286
4	-5892.203358	24	-702.1529333
5	-5351.158552	25	-628.027862
6	-4736.083762	26	-542.4890225
7	-4319.733814	27	-484.382411
8	-3817.374957	28	-413.8101927
9	-3512.126263	29	-363.1240768
10	-3101.428672	30	-309.590145
11	-2823.155783	31	-263.4625162
12	-2485.893999	32	-225.5111255

13	-2269.181469	33	-184.6720905
14	-2010.649439	34	-155.2562867
15	-1808.120794	35	-121.7517589
16	-1597.883388	36	-96.75267461
17	-1448.895308	37	-70.3624
18	-1285.833562	38	-49.01743147
19	-1137.194443	39	-28.47007876
20	184.2854463	40	-10.0739175
total			-68657.79228

Tabel 4.10 Koefisien *Cepstral* Sinyal *DikaMaju4.wav*

Frame	Koefisien cepstral	Frame	Koefisien cepstral
1	184.818545	21	-1023.757979
2	-7380.200969	22	-898.7322137
3	-6628.215806	23	-801.8701645
4	-5848.09768	24	-704.637347
5	-5334.132456	25	-621.7875195
6	-4709.922643	26	-547.7242238
7	-4345.673745	27	-475.8860324
8	-3862.216021	28	-417.4329875
9	-3506.100238	29	-358.1434595
10	-3095.943381	30	-313.9920996
11	-2791.064581	31	-262.3805499
12	-2519.848025	32	-224.2677303
13	-2263.273453	33	-187.9561183
14	-2027.761087	34	-152.3969477

15	-1792.947447	35	-124.0059685
16	-1621.383679	36	-94.81100318
17	-1425.361657	37	-71.30344472
18	-1294.659376	38	-48.75631915
19	-1133.691277	39	-28.63267761
20	184.818545	40	-10.19257562
total		-68769.53033	

Tabel 4.11 Koefisien *Cepstral* Sinyal *DikaMaju5.wav*

Frame	Koefisien <i>cepstral</i>	Frame	Koefisien <i>cepstral</i>
1	183.8837637	21	-1006.966752
2	-7231.423701	22	-902.5588687
3	-6564.781908	23	-811.7900828
4	-5870.204513	24	-708.5051392
5	-5286.826157	25	-631.7683243
6	-4745.735193	26	-551.9131892
7	-4283.558325	27	-485.3093735
8	-3814.859063	28	-425.1519404
9	-3475.370804	29	-364.928971
10	-3055.257559	30	-313.3823047
11	-2799.459047	31	-269.8553327
12	-2449.267136	32	-225.0289993
13	-2246.384088	33	-191.2735005
14	-1962.370491	34	-154.7327897
15	-1795.203287	35	-126.3026633
16	-1582.772516	36	-97.20930632

17	-1419.436166	37	-73.60399767
18	-1263.154545	38	-50.47914203
19	-1136.121992	39	-30.34446189
20	183.8837637	40	-11.77306809
total		-68226.63919	

Dari jumlah total koefisien *cepstral* tersebut dapat diketahui selisih dari masing-masing sinyal kata, yang ditampilkan pada Tabel 4.12.

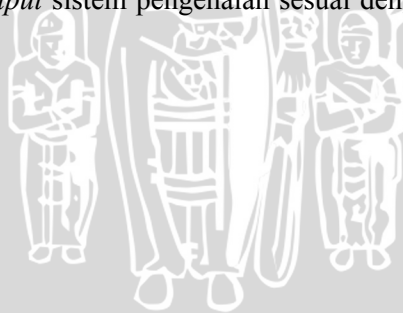
Tabel 4.12 Selisih Koefisien *Cepstral* Kata ‘Maju’

Sinyal	<i>TrainingMaju1.wav</i>	<i>TrainingMaju2.wav</i>
<i>DikaMaju1.wav</i>	742.1754107	530.8666228
<i>DikaMaju2.wav</i>	-75.32423995	-286.6330278
<i>DikaMaju3.wav</i>	105.2676885	-106.0410994
<i>DikaMaju4.wav</i>	217.0057395	5.696951637
<i>DikaMaju5.wav</i>	-325.885401	-537.1941889

Dari Tabel 4.12 terlihat bahwa sinyal *DikaMaju1.wav* dan sinyal *DikaMaju4.wav* memiliki selisih yang cukup besar dengan data training. Hal inilah yang menyebabkan metode PDTW tidak menghasilkan keputusan kata ‘Maju’, melainkan menghasilkan keputusan yang salah, yang merupakan kata dari data *training* yang memiliki selisih paling minimum.

Pada sebagian besar sinyal kata yang dikenali salah juga memiliki masalah perbedaan amplitudo yang cukup besar dengan sinyal dari data *training*, sehingga koefisien *cepstral* yang dihasilkan dari metode LPC juga memiliki disparitas yang cukup besar sehingga membuat perhitungan PDTW menghasilkan keputusan yang salah.

Pada sinyal kata uji coba yang dikenali benar, memiliki jarak terdekat atau paling minimum sesuai dengan data *training*. Hal ini berarti koefisien *cepstral* sinyal kata uji coba yang dihasilkan oleh metode LPC memiliki selisih paling minimum dengan koefisien *cepstral* data *training* masing-masing kata. Sehingga *output* sistem pengenalan sesuai dengan sinyal kata yang di-*input*-kan.





BAB V KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil uji coba yang telah dilakukan, maka dapat diambil kesimpulan antara lain:

1. Metode *Linear Predictive Coding* (LPC) sebagai metode ekstraksi ciri serta metode *Piecewise Dynamic Time Warping* (PDTW) sebagai metode pengenalan untuk menyelesaikan permasalahan sistem pengenalan pola suara telah berhasil diimplementasikan.
2. Dari uji coba terhadap 80 data uji dari 4 kata didapatkan nilai *Word Error Rate* (WER) untuk kata 'Maju' sebesar 35%, kata 'Mundur' sebesar 40%, kata 'Kanan' sebesar 55%, dan kata 'Kiri' sebesar 65%. Total kata yang dikenali salah terdapat 39 sinyal kata, sehingga total WER didapatkan 48.75% dan akurasi metode PDTW sebesar 51.25% untuk empat jenis kata yang berbeda. Dari 4 kata yang dikenali salah tersebut memiliki perbedaan amplitudo yang cukup besar, sehingga koefisien *cepstral* yang dihasilkan berbeda jauh dengan koefisien *cepstral* pada data *training* dan mengakibatkan pengambilan keputusan yang salah dari perhitungan nilai minimum metode PDTW.

5.2 Saran

Saran untuk pengembangan penelitian lebih lanjut adalah:

1. Penelitian dapat dikembangkan untuk pengontrolan gerak arah dengan pengenalan sinyal suara kalimat.
2. Penelitian dapat dikembangkan untuk pengenalan kata yang hampir mirip.
3. Penelitian dapat dikembangkan untuk pengenalan suara pada kondisi *background noise* yang tinggi.

DAFTAR PUSTAKA

- Al Fatta, Hanif. 2009. *Rekayasa Sistem Pengenalan Wajah*. ANDI OFFSET. Yogyakarta.
- Chu, S. Keogh, E. Hart, D. dan Pazzani, M. 2007. *Iterative Deepening Dynamic Time Warping for Time Series*. University of California. California.
- Cook, S.C. 2002. *Speech Recognition HOWTO*. GNU Free Documentation License.
- David, Frederikus. 1996. *Penggunaan Prosesor Sinyal Digital Keluarga TMS320 sebagai Alat Pengenalan Suara Manusia dengan Algoritma DTW (Dynamic Time Warping)*. Universitas Kristen Petra. Surabaya.
- Deller. 2002. *Linear Prediction*. Speech Processing Handbook.
- Helwig, N.E. Hong, S. dan Hsiao-Wecksler, E.T. 2009. *Time-Normalization Techniques for Gait Data*. University of Illinois at Urbana-Champaign.
- Irfani, A. 2006. *Algoritma Viterbi dalam Metode Hidden Markov Models pada Teknologi Speech Recognition*. Institut Teknologi Bandung. Bandung.
- Jones, D.L. Appadwedula, S. Berry, M. Haun, M. Janovetz, J. Kramer, M. Moussa, D. Sachs, D. dan Wade, B. 2009. *Speech Processing: Theory of LPC Analysis and Synthesis*. The Connexions Project.
- Keogh, E.J. dan Michael J. Pazzani. 2000. *Scaling up Dynamic Time Warping for Datamining Applications*. University of California. California.
- Levy, C. Linares, G. dan Nocera, P. *Comparaison of Several Acoustic Modeling Techniques and Decoding Algorithms for Embedded Speech Recognition Systems*. University of Avignon. France.

Marques de Sá, J.P. 2001. *Pattern Recognition: Concept, Methods and Applications*. Springer-Verlag Berlin Heidelberg. Germany.

Oates, T. Firoiu, L. dan Cohen, P.R. *Clustering Time Series with HMM and DTW*. University of Massachusetts. Massachusetts.

Owen, F. J. 1993. *Signal Processing of Speech*. McGraw-Hill Inc. New York.

Proakis. 1995. *Pemrosesan Sinyal Digital (Prinsip-prinsip, Algoritma, dan Aplikasi)*. Prentice-Hall International, Inc. Singapore.

Rabiner. 1993. *Fundamentals of Speech Recognition*. PTR Prentice Hall: Englewood Clift. New Jersey.

Rachmawati, A. 2009. *Pengenalan Suara Menggunakan Linear Predictive Coding (LPC), Vektor Kuantisasi dan Metode Hidden Markov Model (HMM)*. Universitas Brawijaya. Malang.

Sudirman, R. Salleh, S. dan Ming, T.C. 2005. *NN Speech Recognition Utilizing Aligned DTW Local Distance Scores*. Universiti Teknologi Malaysia. Malaysia.

Theodoridis, S. dan Konstantinos Koutroumbas. 2006. *Pattern Recognition Third Edition*. Academic Press. UK.

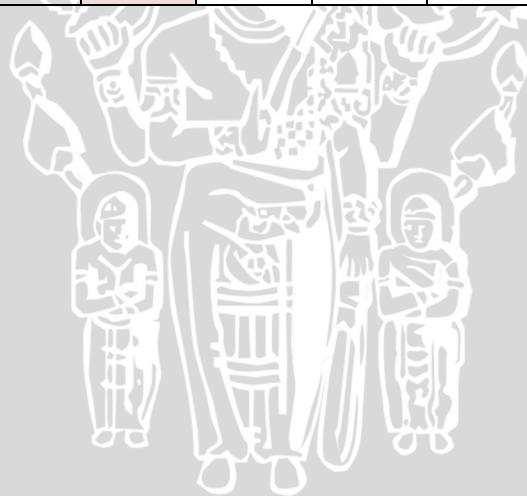


LAMPIRAN

Hasil perbandingan kata 'Kanan'

Sinyal Suara	Kanan		Kiri		Maju		Mundur		Keputusan
	<i>h1</i>	<i>h2</i>	<i>h3</i>	<i>h4</i>	<i>h5</i>	<i>h6</i>	<i>h7</i>	<i>h8</i>	
DikaKanan1.wav	15047	29068.9	17429.6	51623.5	7305.38	10361.3	14724.2	23396.6	Maju
DikaKanan2.wav	10496.6	22360.2	16481.1	30882.3	13523	15217.1	13778.8	21823.6	Kanan
DikaKanan3.wav	37956.3	74843.2	43132.3	106707	18597.8	22757	42140.4	61023.7	Maju
DikaKanan4.wav	10129.6	21502.2	15821.8	30130.8	13515.8	14994.4	13483.6	21220.6	Kanan
DikaKanan5.wav	6954.44	20064.6	13400	33141	7800.81	12984.1	12964.9	16899	Kanan
GitaKanan1.wav	4233.14	1638.27	4485.31	8178.72	14298.4	10154.4	4324.56	2227.26	Kanan
GitaKanan2.wav	3464.5	9672.55	4180.88	19839.9	8729.17	3845.05	4049.28	7299.23	Kanan
GitaKanan3.wav	7645.01	22303.3	13608.6	39615.6	1497.54	5570.71	7880.59	18626.3	Maju
GitaKanan4.wav	3821.49	13970.6	7813.29	27758.9	2598.22	3013.98	4114.53	11630.1	Maju
GitaKanan5.wav	5449.9	1558.22	4724.51	9426.27	14603	9510.43	3029.45	1304.5	Mundur
RuliskKanan1.wav	17509.2	45436.8	26615	68501	6019.8	10959.8	18415.5	38952.3	Maju
RuliskKanan2.wav	5247.05	25252.2	11431.8	43785.4	334.871	4883.24	8385.19	21448.5	Maju
RuliskKanan3.wav	3717.7	17930.9	8011.43	32916.2	2570.78	1234.34	3686.29	14409.6	Maju

Sinyal Suara	Kanan		Kiri		Maju		Mundur		Keputusan
	<i>h1</i>	<i>h2</i>	<i>h3</i>	<i>h4</i>	<i>h5</i>	<i>h6</i>	<i>h7</i>	<i>h8</i>	
RuliskKanan4.wav	25126.5	56337.3	32320.8	84485.1	11205.4	13423.8	25910.6	47572.8	Maju
RuliskKanan5.wav	34825.4	10796.8	30408.4	3804.94	60557.6	51948	34151.5	17776.5	Kiri
YasserKanan1.wav	2404.94	11019.6	4613.3	25210.2	4344.12	3890.72	2027.28	8991.52	Mundur
YasserKanan2.wav	4949.72	6880.31	7826.5	13065.8	11583.4	11172.4	7140.55	8107.51	Kanan
YasserKanan3.wav	9008.27	3857.1	10841.3	5960.13	20107.4	16239.3	9037.48	6189.15	Kanan
YasserKanan4.wav	1730.75	5621.27	3282.12	15295.7	6918.68	4654.46	1800.28	4629.12	Kanan
YasserKanan5.wav	2667.1	3811.86	4380.04	11459.5	9616.61	7799.92	2944.56	3926.12	Kanan



Hasil perbandingan kata 'Kiri'

Sinyal Suara	Kanan		Kiri		Maju		Mundur		Keputusan
	<i>h1</i>	<i>h2</i>	<i>h3</i>	<i>h4</i>	<i>h5</i>	<i>h6</i>	<i>h7</i>	<i>h8</i>	
DikaKiri1.wav	6689.28	18345.8	6390.59	36934.8	8864.39	9137.09	10250.8	18569.2	Kiri
DikaKiri2.wav	39513.3	13511.6	35613.6	5515.67	65140.7	57362.8	36390	20355.3	Kiri
DikaKiri3.wav	35186.5	42044.8	38842.2	51367.6	41202.5	29712.6	27058	37138.9	Mundur
DikaKiri4.wav	16014.2	34397	18135.1	51780.8	10870.2	14359	22157	28272.3	Maju
DikaKiri5.wav	62837.4	25347.2	55065.7	12722.7	93634.7	77825.1	55536.6	30922.1	Kiri
GitaKiri1.wav	17094.1	4365.03	17222.7	2014.59	32762.7	27914.6	16148.3	8173.04	Kiri
GitaKiri2.wav	2184.9	10618.9	5757.87	23817.4	3201.91	2859.35	1797.21	8086.43	Mundur
GitaKiri3.wav	7084.01	28438.5	12476.7	47703.2	1151.63	3700.02	10384.5	23552	Maju
GitaKiri4.wav	12674.4	2469.59	12137.8	3593.82	27916.6	19477.9	8520.16	5187.32	Kanan
GitaKiri5.wav	5581.74	2276.99	5831.71	7069.04	16494	10752.2	4263.59	3577.57	Kanan
RuliskKiri1.wav	22661.5	51538.3	29367.4	77916.8	8904.11	13883.1	24580	43460.9	Maju
RuliskKiri2.wav	8651.17	3747.71	8945.13	7303.73	21702.3	13946.9	6401.76	4942.44	Kanan
RuliskKiri3.wav	118586	187847	136357	233402	80611.1	98481.8	126986	171062	Maju

Sinyal Suara	Kanan		Kiri		Maju		Mundur		Keputusan
	<i>h1</i>	<i>h2</i>	<i>h3</i>	<i>h4</i>	<i>h5</i>	<i>h6</i>	<i>h7</i>	<i>h8</i>	
RuliskKiri4.wav	229276	154409	214320	116036	287426	264140	222471	172796	Kiri
RuliskKiri5.wav	116979	68968	108510	43178	160089	145988	115383	82519.6	Kiri
YasserKiri1.wav	18252.9	46198.2	22997.6	70928.9	8485.21	9047.47	18644.5	37922.6	Maju
YasserKiri2.wav	38125.7	15177.5	35133.2	6563.97	62552.3	52913.1	36970.2	23461.3	Kiri
YasserKiri3.wav	9813.8	33632.6	16931.5	55115.4	3345.52	6770.4	11349	30482.6	Maju
YasserKiri4.wav	6363.18	1537.08	5475.97	5481.14	19408.2	13854.2	6639.88	3763.89	Kanan
YasserKiri5.wav	6711.95	22246.8	10771.3	41712.7	6348.46	5677.17	5900.34	18799.2	Maju

Hasil perbandingan kata 'Maju'

Sinyal Suara	Kanan		Kiri		Maju		Mundur		Keputusan
	<i>h1</i>	<i>h2</i>	<i>h3</i>	<i>h4</i>	<i>h5</i>	<i>h6</i>	<i>h7</i>	<i>h8</i>	
DikaMaju1.wav	10716.3	9347.27	9197.18	15833.8	23800.3	18104.1	7774.09	10076.5	Mundur
DikaMaju2.wav	23039.4	50380.9	29729.3	74025.7	11427.5	11426	22249.3	43875.4	Maju
DikaMaju3.wav	16973.7	33725.3	17042.8	58048.5	14979.1	12895.9	15376.5	27429.1	Maju
DikaMaju4.wav	24749.2	31277.4	27603.8	46251.2	28824.9	28985.7	23191.9	27466.7	Mundur
DikaMaju5.wav	32211.2	64882.5	37678.6	94256.1	17212.6	21641.5	31929.8	54854	Maju
GitaMaju1.wav	4026.67	3772.62	5455.05	12628	9781.09	7243.14	2015.82	3879.73	Mundur
GitaMaju2.wav	15517.1	7558.92	14877.4	6880.96	32430.4	24162.5	16986.9	10320.3	Kiri
GitaMaju3.wav	4828.44	22040	10072.3	39368.8	453.378	2401.54	5952.37	18255.1	Maju
GitaMaju4.wav	8365.73	28922.6	15140.1	48708.8	785.659	5226.91	9809.34	24039.1	Maju
GitaMaju5.wav	7281.24	20304.3	10449.4	38530.3	4442.91	3384.19	5122.16	15445.2	Maju
RuliskMaju1.wav	2552.23	4517.28	4711.7	11084.4	10471.6	8896.11	2696.68	5488.94	Kanan
RuliskMaju2.wav	13381	36153	21951.2	56353.1	3225.1	6288.79	13096.2	29535.9	Maju
RuliskMaju3.wav	3227.88	15595.2	7004.25	29729.8	2748.69	2140.78	2655.26	13233.3	Maju

Sinyal Suara	Kanan		Kiri		Maju		Mundur		Keputusan
	<i>h1</i>	<i>h2</i>	<i>h3</i>	<i>h4</i>	<i>h5</i>	<i>h6</i>	<i>h7</i>	<i>h8</i>	
RuliskMaju4.wav	2724.51	5438.53	2410.51	16392.3	8670.3	4199.62	3158.99	4452.07	Kiri
RuliskMaju5.wav	6983.18	25174.9	10001.7	45807.9	2280.79	2142.49	9000.62	20372.7	Maju
YasserMaju1.wav	6869.24	22775.3	8784.09	37515.2	5436.62	5571.67	12114.2	18103.5	Maju
YasserMaju2.wav	5128.4	18819.1	11393.8	36259.6	1716.68	4826.7	4731.35	16408	Maju
YasserMaju3.wav	18273.8	32384.8	23748.7	48429.5	13898.7	14988.3	14137.1	32683.6	Maju
YasserMaju4.wav	18887.8	45517.9	25067	71118.1	7245.35	13234.9	24109.2	40636.2	Maju
YasserMaju5.wav	4360.7	4177.42	6721.74	11022.5	11051.4	10274.7	4209.47	5194.3	Kanan



Hasil perbandingan kata 'Mundur'

Sinyal Suara	Kanan		Kiri		Maju		Mundur		Keputusan
	<i>h1</i>	<i>h2</i>	<i>h3</i>	<i>h4</i>	<i>h5</i>	<i>h6</i>	<i>h7</i>	<i>h8</i>	
DikaMundur1.wav	17012.5	16591.7	20986.8	31504.9	20278	17014.7	11537.1	13483.6	Mundur
DikaMundur2.wav	16302	5605.12	12274.8	14558.2	28707	19342.4	12477	4729.63	Mundur
DikaMundur3.wav	5912.11	26356.9	13843.5	43046.9	2266.81	6767.79	9490.35	25038.2	Maju
DikaMundur4.wav	17195.6	21602.6	21603.7	39630.8	17321.8	14423.4	11972.9	16989	Mundur
DikaMundur5.wav	18347.6	14239.1	19621.7	26900.3	27355.4	19279.4	11775.6	13686.9	Mundur
GitaMundur1.wav	2967.3	9068.88	3382.98	22239	6698.86	2222.59	1316.82	6151.89	Mundur
GitaMundur2.wav	4305.58	2865.07	5001.5	10695.7	13359.1	11475.8	4122.25	3579.59	Kanan
GitaMundur3.wav	21091.5	3954.28	18291.7	1651.27	40589.8	30122.4	15940.7	8483.45	Kiri
GitaMundur4.wav	4431.36	17987.6	8227.14	34279.1	4394.36	3637.33	3345.39	16153.1	Mundur
GitaMundur5.wav	7907.45	2933.36	8556.22	8383.02	18541.1	11613.2	3956.84	2867.92	Mundur
RuliskMundur1.wav	3113.4	4343.69	3749.77	14807.4	9673.89	6153.17	3088.15	3250.63	Mundur
RuliskMundur2.wav	3573.9	12204.3	8634.37	23666.2	4881.13	3461.58	2076.94	9875.34	Mundur
RuliskMundur3.wav	4007.35	16121.4	7184.55	31840.2	5080.23	5357.15	3781.15	15170.1	Mundur

Sinyal Suara	Kanan		Kiri		Maju		Mundur		Keputusan
	$h1$	$h2$	$h3$	$h4$	$h5$	$h6$	$h7$	$h8$	
RuliskMundur4.wav	7197.15	2883.65	7474	10089.4	14924.5	9096.97	4650.08	2291.18	Mundur
RuliskMundur5.wav	1329.96	9130.86	4425.79	21182.2	5344.93	4763.99	1197.45	9189.49	Mundur
YasserMundur1.wav	4457.36	9976.06	8342.78	18954.6	7626.22	9555.11	6225.09	10694.1	Kanan
YasserMundur2.wav	9287.53	1851.23	7763.84	6645.26	19609.9	14688.5	8776.94	2207.28	Kanan
YasserMundur3.wav	6134.68	20102.3	9889.09	33363.1	3310.43	5174.31	9857.54	15425.8	Maju
YasserMundur4.wav	3755.09	15835.4	8708.4	28984.2	1686.81	3005.63	3654.99	12576.7	Maju
YasserMundur5.wav	13935.5	1391.8	13468.3	1860.69	29277.4	23525.2	11156	3792.2	Kanan