



TUGAS AKHIR - SM141501

**PENERAPAN ALGORITMA ANTS CLUSTERING DAN JARINGAN
LVQ PADA SISTEM PENGENALAN INDIVIDU BERBASIS SUARA
UCAPAN**

METTA ANDRIANI
NRP 1213 100 097

Dosen Pembimbing
Drs. Nurul Hidayat, M.Kom
Drs. Soetrisno, Ml.Komp

DEPARTEMEN MATEMATIKA
Fakultas Matematika dan Ilmu Pengetahuan Alam
Institut Teknologi Sepuluh Nopember
Surabaya 2017



FINAL PROJECT - SM141501

**APPLICATION OF ANTS CLUSTERING ALGORITHM AND LVQ
NETWORK IN THE INDIVIDUAL IDENTIFICATION SYSTEM
BASED ON SPEECH**

METTA ANDRIANI
NRP 1213 100 097

Supervisors
Drs. Nurul Hidayat, M.Kom
Drs. Soetrisno, Ml.Komp

DEPARTMENT OF MATHEMATICS
Faculty of Mathematics and Natural Science
Sepuluh Nopember Institute of Technology
Surabaya 2017

LEMBAR PENGESAHAN

**PENERAPAN ALGORITMA ANTS CLUSTERING DAN
JARINGAN LVQ PADA SISTEM PENGENALAN
INDIVIDU BERBASIS SUARA UCAPAN**

***APPLICATION OF ANTS CLUSTERING ALGORITHM
AND LVQ NETWORK IN THE INDIVIDUAL
IDENTIFICATION SYSTEM BASED ON SPEECH***

TUGAS AKHIR

Diajukan untuk memenuhi salah satu syarat
Untuk memperoleh gelar Sarjana Sains
Pada bidang studi Ilmu Komputer
Program Studi S-1 Departemen Matematika
Fakultas Matematika dan Ilmu Pengetahuan Alam
Institut Teknologi Sepuluh Nopember Surabaya

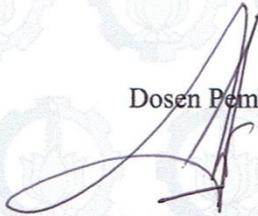
Oleh :

METTA ANDRIANI
NRP. 1213 100 097

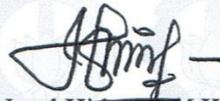
Menyetujui,

Dosen Pembimbing II

Dosen Pembimbing I



Drs. Soetrisno, MI.Komp
NIP. 19571103 198603 1 003



Drs. Nurul Hidayat, M.Kom
NIP. 19630404 198903 1 002

Mengetahui,
Kepala Departemen Matematika
FMIPA ITS



Dr. Imam Mukhlash, S.Si, MT
NIP. 19700831 199403 1 003
Surabaya, 31 Juli 2017

PENERAPAN ALGORITMA ANTS CLUSTERING DAN JARINGAN LVQ PADA SISTEM PENGENALAN INDIVIDU BERBASIS SUARA UCAPAN

Nama : Metta Andriani
NRP : 1213 100 097
Departemen : Matematika
Dosen Pembimbing : 1. Drs. Nurul Hidayat, M.Kom
2. Drs. Soetrisno, MI.Komp

ABSTRAK

Biometrika merupakan cabang dari matematika terapan yang membahas tentang identifikasi individu berdasarkan ciri-ciri biologis yang bersifat unik seperti wajah, sidik jari dan suara ucapan.. Salah satu bidang dari biometrika adalah pengenalan individu melalui suara ucapan. Penelitian ini membuat sistem pengenalan individu berbasis suara ucapan yang diolah menggunakan Estimasi Trispektrum pada tahap prapengolahan dan Algoritma *Ants Clustering* pada tahap pengelompokan ciri. Hasil dari proses pengelompokan ciri berupa data magnitudo perwakilan dari masing-masing kelompok untuk dijadikan masukan pada tahap klasifikasi dengan menggunakan Jaringan *Learning Vector Quantization (LVQ)*. Dataset suara pada penelitian ini diperoleh dari 30 individu yang berbeda. Hasil uji coba pengenalan individu dengan kata sama diperoleh rata-rata akurasi 83,33% menggunakan 5 referensi dan untuk uji pengenalan kata diperoleh rata-rata akurasi 85% dengan menggunakan nilai *chunk*=128, vektor ciri=8, epoch=100 dan *learning rate*=0.001.

Kata Kunci : Sistem Pengenalan Suara, Estimasi Trispektrum, Algoritma *Ants Clustering*, *Learning Vector Quantization*

APPLICATION OF ANTS CLUSTERING ALGORITHM AND LVQ NETWORK IN THE INDIVIDUAL IDENTIFICATION SYSTEM BASED ON SPEECH

Name of Student : Metta Andriani
NRP : 1213 100 097
Department : Mathematics
Supervisors : 1. Drs. Nurul Hidayat, M.Kom
2. Drs. Soetrisno, MI.Komp

ABSTRACT

Biometrics is a branch of applied mathematics that discuss individual identification based on biological characteristics which are unique such as face, finger print, and speech. One of the fields of biometrics is the introduction of individual through speech. This research make individual identification system based on speech which are processed using the estimation of Trispektrum on stage preprocessing and the Ants Clustering Algorithm on stage grouping characteristics. The result of the process grouping characteristics is a magnitude agent from each group to provide input in the classification using the Learning Vector Quantization (LVQ). The dataset of this research is retrieved from 30 different individuals. Trial results indentification of individuals with the same words obtained average accuracy 83.33% use 5 reference and to test the introduction of speechs obtained the average accuracy 85% using value chunk=128, vector characteristics=8, epoch=100 and learning rate=0.001.

Keywords : Voice Recognition, Estimation Trispectrum, Ants Clustering Algorithm, Learning Vector Quantization

KATA PENGANTAR

Assalamu'alaikum Wr. Wb.

Segala puji dan syukur bagi Allah SWT yang memiliki apa yang ada di langit dan di bumi atas segala rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul

“PENERAPAN ALGORITMA ANTS CLUSTERING DAN JARINGAN LVQ PADA SISTEM PENGENALAN INDIVIDU BERBASIS SUARA UCAPAN”

sebagai salah satu syarat kelulusan Program Sarjana Departemen Matematika FMIPA Institut Teknologi Sepuluh Nopember (ITS) Surabaya.

Dalam menyelesaikan Tugas Akhir ini, penulis telah banyak mendapat bantuan serta masukan dari berbagai pihak. Oleh karena itu, dalam kesempatan ini penulis menyampaikan terima kasih kepada :

1. Kedua orang tua, Bapak Santosa dan Ibu Emi serta adik Langga yang telah memberikan do'a dan dukungan.
2. Bapak Drs. Nurul Hidayat, M.Kom dan Bapak Drs. Soetrisno, Ml.Komp selaku Dosen Pembimbing Tugas Akhir yang telah banyak memberikan bimbingan, arahan serta motivasi sehingga Tugas Akhir ini dapat terselesaikan.
3. Bapak Dr. Imam Mukhlas, S.Si, MT selaku Ketua Departemen Matematika ITS.
4. Bapak Dr. Chairul Imron, Ml.Komp dan Ibu Dian Winda Setyawati, S.Si, M.Si selaku Dosen Penguji yang telah memberikan saran demi perbaikan Tugas Akhir ini.
5. Bapak Dr. Didik Khusnul Arif, S.Si, M.Si selaku Ketua Program Studi S1 Departemen Matematika ITS dan Bapak Drs. Iis Herisman, M.Si selaku Sekretaris Program Studi S1 Departemen Matematika ITS.
6. Seluruh jajaran dosen dan staf Departemen Matematika ITS.

7. Teman-teman Departemen Matematika ITS Angkatan 2013 yang saling mendukung dan memotivasi.
8. Seluruh sahabat, khususnya Gina Faaiza yang telah mendukung setiap proses dalam pengerjaan Tugas Akhir.
9. Untuk orang yang selalu menyayangiku
10. Semua pihak yang tidak bisa penulis sebutkan satu-persatu, terima kasih telah membantu sampai terselesaikannya Tugas Akhir ini.
Apabila dalam Tugas Akhir ini ada kekurangan, penulis mengharapkan kritik dan saran dari pembaca. Semoga Tugas Akhir ini dapat bermanfaat bagi semua pihak.

Wassalamu'alaikum Wr. Wb.

Surabaya, 19 Mei 2017
Penulis

DAFTAR ISI

KATA PENGANTAR	xi
DAFTAR ISI.....	xiii
DAFTAR GAMBAR.....	xvii
DAFTAR TABEL.....	xix
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah.....	3
1.4 Tujuan.....	3
1.5 Manfaat.....	4
1.6 Sistematika Penulisan Tugas Akhir.....	4
BAB II TINJAUAN PUSTAKA	7
2.1 Penelitian Terdahulu.....	7
2.2 Sistem Pengenalan Suara.....	8
2.3 Estimasi Trispektrum.....	9
2.4 Algoritma <i>Ants Clustering</i>	14
2.5 Jaringan Syaraf Tiruan	17
2.5.1 Struktur Jaringan Syaraf Tiruan	18
2.5.2 Proses Pelatihan atau Pembelajaran	20
2.5.3 Learning Vector Quantization.....	20
BAB III METODOLOGI PENELITIAN	23
3.1 Diagram Metodologi	23

3.2	Studi Literatur.....	23
3.3	Pengumpulan Data Suara	23
3.4	Data Suara	23
3.5	PemotonganData Suara	25
3.6	Data Latih dan Data Uji.....	25
3.7	Prapengolahan Data.....	25
3.8	Pengelompokan Ciri	25
3.9	Klasifikasi Data	25
3.10	Implementasi Sistem	26
3.11	Pengujian dan Evaluasi.....	26
BAB IV PERANCANGAN SISTEM.....		27
4.1	Akuisisi Data Sinyal Suara.....	27
4.2	Pemotongan Sinyal Suara.....	27
4.3	Prapengolahan Data Sinyal Suara.....	28
4.3.1	<i>Frame Blocking</i>	28
4.3.2	<i>Windowing</i>	29
4.3.3	<i>Fast Fourier Transform</i>	29
4.3.4	Estimasi Trispektrum.....	30
4.4	Pengelompokan Ciri Sinyal Suara.....	33
4.5	Klasifikasi Data Sinyal Suara.....	37
4.5.1	Proses <i>Training</i>	37
4.5.2	Proses <i>Testing</i>	39
4.6	Pengujian dan Evaluasi Sistem.....	40
4.7	Perancangan <i>User Interface</i>	41

BAB V IMPLEMENTASI SISTEM.....	45
5.1 Lingkungan Hardware dan Software	45
5.2 Implementasi <i>User Interface</i>	45
5.3 Implementasi Prapengolahan Data Sinyal Suara...	49
5.4 Implementasi Pengelompokan Ciri	51
5.5 Implementasi Klasifikasi Data.....	51
5.5.1 Proses <i>Training</i>	51
5.5.2 Proses <i>Testing</i>	51
5.5 Implementasi Klasifikasi Data.....	56
5.5.1 Proses <i>Training</i>	56
5.5.2 Proses <i>Testing</i>	58
BAB VI UJI COBA DAN EVALUASI SISTEM.....	59
6.1 Dataset Uji Coba <i>Training</i>	59
6.2 Uji Coba Estimasi Trispektrum	59
6.3 Uji Coba Proses Pengelompokan Ciri	61
6.4 Uji Coba Pengenalan Individu dengan Kata Sama	61
6.5 Uji Coba Pengenalan Individu dengan Kata Berbeda.....	63
6.6 Uji Coba Pengenalan Kata.....	64
BAB VII KESIMPULAN DAN SARAN	67
7.1 Kesimpulan.....	67
7.2 Saran	67
DAFTAR PUSTAKA	69
LAMPIRAN.....	71

BIODATA PENULIS 85

DAFTAR GAMBAR

Gambar 2. 1 Contoh Algoritma Ant Clustering.....	14
Gambar 2. 2 Sel Syaraf Biologis.....	17
Gambar 2. 3 Jaringan Lapis Tunggal.....	19
Gambar 2. 4 Jaringan Lapis Jamak.....	19
Gambar 3. 1 Diagram Alir Metodologi Penelitian.....	24
Gambar 4. 1 Pemotongan Sinyal Suara.....	27
Gambar 4. 2 Contoh Proses Frame Blocking.....	29
Gambar 4. 3 Contoh Dekomposisi Sinyal.....	30
Gambar 4. 4 Diagram Alir Proses Prapengolahan Data Sinyal Suara.....	31
Gambar 4. 5 Diagram Alir Proses Prapengolahan Data Sinyal Suara (Lanjutan).....	32
Gambar 4. 6 Contoh Peletakkan Item Data.....	33
Gambar 4. 7 Neighboring Cells dalam koordinat Cartesius .	35
Gambar 4. 8 Diagram Alir Algoritma Ant Clustering.....	36
Gambar 4. 9 Halaman Utama.....	41
Gambar 4. 10 Halaman Prapengolahan Data.....	42
Gambar 4. 11 Halaman Pengelompokan Ciri.....	42
Gambar 4. 12 Halaman Training.....	43
Gambar 4. 13 Halaman Testing.....	43
Gambar 5. 1 Running Halaman Utama.....	46
Gambar 5. 2 Running Halaman Prapengolahan Data.....	46
Gambar 5. 3 Running Halaman Pengelompokan Ciri.....	47
Gambar 5. 4 Running Halaman Training.....	48
Gambar 5. 5 Running Halaman Testing.....	48
Gambar 6. 1 Visualisasi Hasil Estimasi Trispektrum dari Kata “Maju”.....	60
Gambar 6. 2 Diagram Prosentase Pengenalan Individu dengan Kata Sama (chunk = 128, vektor ciri = 8, 1 kata referensi).....	62

Gambar 6. 3 Diagram Prosentase Pengenalan Individu dengan Kata Sama (chunk = 128, vektor ciri = 8, 3 kata referensi).....	62
Gambar 6. 4 Diagram Prosentase Pengenalan Individu dengan Kata Sama (chunk = 128, vektor ciri = 8, 5 kata referensi).....	63
Gambar 6. 5 Diagram Prosentase Pengenalan Individu dengan Kata Berbeda	64
Gambar 6. 6 Diagram Prosentase Uji Pengenalan Kata.....	65

DAFTAR TABEL

Tabel 4. 1 Tabel Representasi Label Nama	37
Tabel 4. 2 Contoh Data Training	38
Tabel 5. 1 Spesifikasi Hardware dan Software	45

BAB I

PENDAHULUAN

Pada bab ini dijelaskan hal-hal yang melatarbelakangi munculnya permasalahan yang dibahas dalam Tugas Akhir ini. Kemudian permasalahan tersebut disusun ke dalam suatu rumusan masalah. Selanjutnya dijabarkan juga batasan masalah yang digunakan untuk mendapatkan tujuan yang diinginkan serta manfaat yang dapat diperoleh. Adapun sistematika penulisan Tugas Akhir ini akan diuraikan di bagian akhir bab ini.

1.1 Latar Belakang

Kemajuan teknologi terutama dalam bidang komputer telah berkembang dengan sangat pesat. Hal tersebut memberikan dampak yang besar berupa kemudahan pengenalan informasi pada suatu objek oleh mesin komputer. Salah satu teknologi dibidang komputer yang banyak diteliti adalah biometrika. Biometrika merupakan cabang dari matematika terapan yang membahas tentang identifikasi individu berdasarkan ciri-ciri biologis yang bersifat unik seperti wajah, iris mata, sidik jari dan suara ucapan. Faktor penting yang menyebabkan teknologi biometrika terus dikembangkan adalah kebutuhan dalam penerapan sistem keamanan yang tangguh. Saat ini penerapan biometrika yang berkembang adalah mengenali individu melalui suara ucapan.

Pada dasarnya setiap individu memiliki suatu ciri khas yang mampu membedakan identitas antara individu satu dengan individu lainnya. Salah satu ciri khas yang dimiliki oleh individu adalah suara. Hal ini dikarenakan suara merupakan alat komunikasi yang sangat sederhana dan bersifat alami sehingga memudahkan manusia dalam memberikan perintah. Selain itu suara yang diucapkan individu mengandung informasi penting mengenai identitas individu. Perbedaan suara dari individu disebabkan oleh adanya perbedaan spektrum suara, frekuensi dan amplitudo.

Dalam membangun sistem pengenalan suara, secara umum terdapat empat tahapan yang dapat dilakukan yaitu akuisisi data suara, estimasi sinyal suara, pengelompokan ciri, serta klasifikasi sinyal suara [1]. Data sinyal suara yang digunakan untuk proses pengenalan individu diperoleh dengan bantuan alat perekam. Namun, sinyal tersebut belum bisa digunakan untuk proses pengenalan karena belum menunjukkan pola atau karakteristik dari sinyal tersebut. Oleh karena itu, perlu dilakukan prapengolahan untuk mendapatkan karakteristik dari data sinyal suara tersebut. Pada tahap prapengolahan, dilakukan proses estimasi sinyal suara yang kemudian menghasilkan data magnitudo dan fase. Magnitudo dan fase berisi informasi sinyal suara yang digunakan untuk proses pengenalan. Hasil pengamatan yang dilakukan terhadap visualisasi dari magnitudo menunjukkan bahwa, untuk setiap jenis kata yang sama dari satu individu mempunyai pola yang mirip, sedangkan untuk jenis kata yang berbeda mempunyai pola yang tidak mirip [2].

Dalam proses klasifikasi suara, data yang digunakan merupakan data magnitudo dan fase dari hasil prapengolahan dengan estimasi trispektrum. Mengingat bahwa data yang didapatkan dari hasil prapengolahan masih berada dalam jumlah yang banyak, maka hal ini sangat berpengaruh terhadap lamanya proses klasifikasi. Untuk itu, diperlukan suatu metode pengelompokan ciri untuk mendapatkan data magnitudo dan fase perwakilan dari masing-masing kelompok sehingga diperoleh data yang lebih sedikit untuk proses klasifikasi.

Pada Tugas Akhir ini, dikembangkan sistem pengenalan individu berbasis suara ucapan dengan menggunakan Algoritma *Ants Clustering* pada tahap pengelompokan ciri. Hasil pengelompokan ciri yang diperoleh dari metode tersebut akan digunakan sebagai masukan pada tahap klasifikasi menggunakan metode LVQ (*Learning Vector Quantization*).

1.2 Rumusan Masalah

Rumusan masalah dari Tugas Akhir ini adalah :

1. Melakukan prapengolahan terhadap data sinyal suara individu menggunakan estimasi trispektrum
2. Menerapkan algoritma *Ants Clustering* untuk pengelompokan ciri sinyal suara individu hasil estimasi trispektrum
3. Menerapkan metode jaringan LVQ untuk klasifikasi data sinyal suara individu
4. Mengintegrasikan program prapengolahan, pengelompokan ciri, dan klasifikasi data untuk membuat sistem perangkat lunak pengenalan individu berbasis suara ucapan
5. Membuat *interface* sistem perangkat lunak pengenalan individu berbasis suara ucapan yang *user friendly*.

1.3 Batasan Masalah

Berdasarkan rumusan masalah diatas, batasan masalah yang digunakan sebagai acuan dari Tugas Akhir ini adalah :

1. Pengucapan kata satu persatu dan dilakukan dengan suara normal
2. Keadaan ruangan saat pengambilan suara ucapan dalam keadaan normal dan dianggap tidak ada *noise*
3. Data suara diperoleh dari individu dengan jenis kelamin laki-laki dan perempuan
4. Usia dari pengucap adalah usia dewasa (15-50 tahun)
5. Kata yang telah dipilih sebagai referensi diucapkan dengan bahasa Indonesia.

1.4 Tujuan

Tujuan umum dari Tugas Akhir ini adalah untuk mendapatkan perangkat lunak yang dapat digunakan untuk mengenali individu berbasis suara ucapan. Sedangkan tujuan khusus untuk Tugas Akhir ini adalah sebagai berikut :

1. Mendapatkan program prapengolahan data sinyal suara individu menggunakan estimasi trispektrum
2. Mendapatkan program pengelompokan ciri sinyal suara individu menggunakan algoritma *Ants Clustering*
3. Mendapatkan program klasifikasi data sinyal suara individu menggunakan LVQ
4. Mengintegrasikan program untuk mengenali individu berbasis suara ucapan dengan menerapkan estimasi trispektrum, algoritma *Ants Clustering* dan LVQ

1.5 Manfaat

Manfaat yang diperoleh dari Tugas Akhir ini adalah sebagai berikut :

1. Hasil penelitian ini dapat digunakan untuk membangun sistem keamanan berbasis suara ucapan
2. Sebagai pemberi informasi bagi pembacamengenai pengembangan sistem pengenalan individu berbasis suara ucapan menggunakan algoritma *Ants Clustering* dan jaringan LVQ.

1.6 Sistematika Penulisan Tugas Akhir

Sistematika penulisan dalam Tugas Akhir ini adalah sebagai berikut :

1. BAB I PENDAHULUAN

Bab ini menjelaskan tentang latar belakang pembuatan Tugas Akhir, rumusan dan batasan permasalahan yang dihadapi dalam penelitian Tugas Akhir, tujuan dan manfaat pembuatan Tugas Akhir serta sistematika penulisan Tugas Akhir.

2. BAB II TINJAUAN PUSTAKA

Bab ini berisi teori dasar yang mendukung dalam Tugas Akhir ini, antara lain penelitian terdahulu, pengolahan sinyal suara, estimasi trispektrum,

algoritma *Ants Clustering*, jaringan syaraf tiruan, *Learning Vector Quantization*.

3. BAB III METODOLOGI PENELITIAN

Bab ini menjelaskan tahap pengerjaan dalam menyelesaikan Tugas Akhir yang terdiri dari studi literatur, pengumpulan data, perancangan sistem, implementasi sistem, pengujian dan evaluasi sistem, serta penarikan kesimpulan dan penulisan laporan Tugas Akhir.

4. BAB IV PERANCANGAN SISTEM

Bab ini menjelaskan tahap persiapan pengolahan data hingga proses konstruksi sistem menggunakan algoritma *Ants Clustering* dan Jaringan *Learning Vector Quantization* sebagai acuan dalam implementasi sistem.

5. BAB V IMPLEMENTASI SISTEM

Bab ini membahas proses implementasi dengan menggunakan *software* Matlab berdasarkan rancangan sistem yang telah dibuat pada bab sebelumnya.

6. BAB VI UJI COBA DAN EVALUASI SISTEM

Bab ini membahas tahap-tahap uji coba berdasarkan implementasi sistem yang telah dibuat beserta pengujian kinerja sistem.

7. BAB VII KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan Tugas Akhir yang diperoleh dari bab uji coba dan evaluasi sistem serta saran untuk pengembangan penelitian selanjutnya.

BAB II

TINJAUAN PUSTAKA

Pada bab ini diuraikan mengenai dasar teori yang digunakan dalam penyusunan Tugas Akhir ini. Dasar teori yang dijelaskan dibagi menjadi beberapa sub bab yaitu penelitian terdahulu, pengolahan sinyal suara, estimasi trispektrum, algoritma *Ants Clustering*, jaringan syaraf tiruan, *Learning Vector Quantization*.

2.1 Penelitian Terdahulu

Sistem pengenalan individu berbasis suara ucapan merupakan salah satu penerapan ilmu biometrika yang cukup berkembang dalam bidang keamanan seperti sistem kunci otomatis, detektor elektronis, dan komponen robotik.

Salah satu penelitian tentang sistem pengenalan individu berbasis suara ucapan adalah “Klasterisasi Sinyal Suara Menggunakan Metode *Particle Swarm Optimization* Pada Pengembangan Sistem Pengenalan Individu Berbasis Suara Ucapan” oleh Moh. Bagus Hadi pada tahun 2010. Dalam penelitian ini menjelaskan bahwa metode *Particle Swarm Optimization* digunakan untuk mencari data magnitude perwakilan dari masing-masing kluster untuk dijadikan masukan pada proses klasifikasi dengan JST-LVQ. Dalam penelitian tersebut uji coba dilakukan terhadap 10 kata dari pembicara yang sama, 1 kata dari 20 pembicara yang berbeda, dan 20 kata dari pembicara yang berbeda. Dengan mengambil lima referensi untuk masing-masing kata, metode Particle Swarm Optimization mampu mengenali kata yang diucapkan hingga 81% [3]. Akan tetapi dalam penelitian tersebut proses perekaman dan estimasi sinyal suara memanfaatkan hasil dari penelitian sebelumnya yang dikerjakan oleh Amin (2006) dan Riduwan (2007).

Penelitian lain yang membahas tentang pengelompokan ciri dan klasifikasi sinyal suara adalah “*Learning Vector*

Quantization (LVQ) Neural Network Approach for Multilingual Speech Recognition” oleh Rajat Haldar pada tahun 2016. Dalam penelitian ini menjelaskan bahwa penerapan teknik *LVQ dan Particle Swarm Optimization* untuk pengenalan kata dan identifikasi bahasa memberikan hasil yang sedikit lebih baik dibandingkan tanpa menggunakan teknik PSO. Dalam penelitian tersebut uji coba sistem dilakukan terhadap bahasa Bengali, Chhattisgarhi, Inggris, dan India. Dengan menggunakan LVQ, sistem mampu mengenali bahasa hingga 88% dan persentase error 10-12 %. Sedangkan dengan menggunakan LVQ dan PSO, sistem mampu mengenali bahasa hingga 90 % dan persentase error hanya 8-10 % [5].

Selain itu terdapat penelitian lain yang membahas tentang pengelompokan ciri yaitu “Segmentasi Pasar dengan *Ant-Kmeans Clustering*” yang dilakukan oleh Budi Santosa dan Patdono Suwignyo. Dalam penelitian tersebut menjelaskan penerapan metode *hybrid ant-kmeans clustering*. Dalam hal ini digunakan pendekatan *ant colony* untuk melakukan proses clustering pada permasalahan strategi produksi dan pemasaran yang tepat bagi perusahaan. Hasil dari penelitian ini menunjukkan bahwa algoritma *ant-kmeans clustering* mampu membagi data kedalam tiga buah klaster[6].

2.2 Sistem Pengenalan Suara

Komunikasi merupakan hal yang sangat penting dalam kehidupan individu. Salah satu cara komunikasi yang dilakukan oleh individu adalah dengan berbicara. Berbicara memberikan paling banyak informasi penting dan efektif dalam berkomunikasi. Informasi tersebut berupa mengidentifikasi jenis kelamin, keadaan kesehatan, emosi, serta identitas pembicara [10]. Pengenalan suara dapat dikategorikan menjadi dua bagian yaitu :

1. *Speech Recognition*

Speech recognition adalah suatu teknologi yang mampu mengubah *input* suara menjadi suatu *output* tertentu, seperti

contoh *output* penulisan *text* maupun *output running* program. *Speech recognition* merupakan teknologi yang dapat mengenali pembicaraan tanpa memperdulikan siapa pembicaranya.

2. *Speaker Recognition*

Speaker recognition adalah suatu proses yang bertujuan mengenali siapa yang sedang berbicara berdasarkan informasi yang terkandung dalam gelombang suara yang di inputkan. *Speaker recognition* dibagi menjadi 2 bagian, yaitu :

a. *Speaker Verification*

Speaker verification adalah proses verifikasi seorang pembicara, dimana sebelumnya telah diketahui identitas pembicara tersebut berdasarkan data yang telah diinputkan. *Speaker verification* melakukan perbandingan *one to one* (1:1) dalam arti bahwa fitur-fitur suara dari seorang pembicara dibandingkan secara langsung dengan fitur-fitur seorang pembicara tertentu yang ada dalam sistem.

b. *Speaker Identification*

Speaker identification adalah proses mendapatkan identitas dari seorang pembicara dengan membandingkan fitur-fitur suara yang diinputkan dengan semua fitur-fitur dari setiap pembicara yang ada dalam *database*. Berbeda dengan pada *speaker verification*, proses ini melakukan perbandingan *one to many* (1:N) [10].

2.3 Estimasi Trispektrum

Estimasi trispektrum merupakan proses estimasi sinyal suara yang didasarkan pada korelasi tiga parameter frekuensi yaitu f_1 , f_2 dan f_3 . Daerah estimasi trispektrum merupakan ruang tiga dimensi yang dibentuk oleh vektor frekuensi f_1 , f_2 dan f_3 . Pada setiap koordinat (f_1 , f_2 dan f_3) tersebut terdapat nilai trispektrumnya, sehingga hasil estimasi Trispektrum dapat dikatakan berada dalam ruang empat dimensi.

Proses estimasi trispektrum menggunakan analisis orde empat dengan rumus [7] :

$$M_T(f_1, f_2, f_3) = \frac{1}{T} [X_T(f_1) \cdot X_T(f_2) \cdot X_T(f_3) \cdot X_T^*(f_1 + f_2 + f_3)] \quad (2.1)$$

dengan $X_T(f_i)$ adalah spektrum fourier berdurasi T pada frekuensi f_i , dan $X_T^*(f_i)$ adalah konjugasi kompleks dari $X_T(f_i)$.

Dalam proses estimasi trispektrum terdapat tahapan-tahapan sebagai berikut [7] :

1. *Frame Blocking*

Sinyal suara kontinu diblok dalam *frame* dari N sampel (*chunk*), dengan *frame* yang berdekatan yang terpisah oleh M dimana $M < N$. *Frame* pertama terdiri dari N sampel pertama. *Frame* kedua mulai M sampel setelah *frame* pertama dan saling menyusul dengan $N - M$ sampel. Begitu pula dengan *frame* ketiga mulai sampel 2M setelah *frame* pertama (atau M sampel setelah *frame* kedua) dan menyusul $2(N - M)$ sampel. Proses ini terus berlanjut sampai semua suara dihitung dalam satu *frame* atau lebih.

2. *Windowing*

Langkah selanjutnya adalah *windowing* masing-masing *frame* untuk meminimalisir diskontinuitas sinyal pada permulaan dan akhir dari tiap *frame*. *Windowing* ini untuk meminimalisir *spectral distortion* dengan menggunakan *window* untuk men-*taper* sinyal ke nol pada permulaan dan akhir tiap *frame*. Jika kita mendefinisikan *window* sebagai $w(n)$, $0 \leq n \leq N - 1$, dengan N adalah angka sampel pada masing-masing *frame*. Hasil *windowing* dinyatakan dengan persamaan :

$$y(n) = x(n)w(n), \quad 0 \leq n \leq N - 1 \quad (2.2)$$

dengan :

$x(n)$: sinyal input

$y(n)$: hasil proses *windowing*

Pada penelitian ini menggunakan *Hamming windowing* yang dinyatakan dalam persamaan :

$$w(n) = 0.54 - 0.46 \cos\left(\frac{2\pi n}{N-1}\right), \quad 0 \leq n \leq N-1 \quad (2.3)$$

3. *Fast Fourier Transform (FFT)*

Analisis *fourier* adalah sebuah metode yang memungkinkan untuk melakukan analisa terhadap *spectral properties* dari sinyal yang diinputkan. Representasi dari *spectral properties* sering disebut sebagai *spectrogram*. Dalam *spectrogram* terdapat hubungan yang sangat erat antara waktu dan frekuensi. Sedangkan hubungan frekuensi dan waktu adalah berbanding terbalik. Bila resolusi waktu yang digunakan tinggi, maka resolusi frekuensi yang dihasilkan semakin rendah. Untuk dapat melihat perbedaan sinyal suara yang berbeda-beda maka harus dilihat dari domain frekuensi karena jika dilihat dari domain waktu sulit terlihat perbedaannya. Untuk itu dari sinyal suara yang berada pada domain waktu diubah ke domain frekuensi dengan *Fast Fourier Transform (FFT)*.

Fast Fourier Transform adalah perhitungan cepat dari Transformasi Fourier Diskrit. Perhitungan FFT menggunakan sifat periodikal dari Transformasi Fourier. Tujuan dari Transformasi Fourier adalah untuk mentransformasikan suatu fungsi dalam domain waktu ke fungsi yang lainnya dalam domain frekuensi. Dengan demikian, besaran fungsi tidak lagi bergantung terhadap waktu, melainkan terhadap frekuensi.

Pandang Transformasi Fourier Diskrit berikut :

$$X(k) = \sum_{n=0}^{N-1} x(n)W^{nk} \text{ dengan } k = 0, 1, \dots, N-1 \quad (2.4)$$

Didefinisikan W sebagai bilangan kompleks

$$W = e^{-i2\pi/N}$$

maka

$$W^{nk} = e^{-i2\pi nk/N}$$

Dalam bilangan kompleks, berdasarkan persamaan Euler dinyatakan bahwa :

Dalam aplikasinya, proses perhitungan transformasi dengan menggunakan FFT berjalan dengan cepat, karena dalam perhitungan biasa untuk $n = 2^p$ sampel dibutuhkan $n \times n$ perkalian kompleks dan $n(n - 1)$ penjumlahan kompleks, sedangkan dalam FFT hanya membutuhkan $\frac{np}{2}$ perkalian kompleks dan np penjumlahan kompleks [9]. Proses FFT memiliki tiga langkah, yaitu :

- a. Dekomposisi matriks x_0 menjadi dua bagian yaitu ganjil dan genap
- b. Masing-masing matriks yang telah didekomposisi dikalikan dengan matriks fourier dengan ordo $m \times m$ ($F_{m \times m}$, dengan $m = \left(\frac{1}{2}\right)n$) sehingga diperoleh X_m genap dan X_m ganjil
- c. Merekonstruksi X_m genap dan X_m ganjil untuk mendapatkan X_n , dengan menggunakan formula yang telah diperoleh Cooley dan Tukey sebagai berikut [9] :

$$X_j = X_m \text{genap} + W_n^j X_m \text{ganjil}, \quad j = 0, \dots, m - 1 \quad (2.5)$$

$$X_{j+m} = X_m \text{genap} - W_n^j X_m \text{ganjil}, \quad j = 0, \dots, m - 1$$

4. Proses Trispektrum

Hasil dari proses FFT selanjutnya akan diproses dengan trispektrum. Pada proses estimasi trispektrum, f_1, f_2 , dan f_3 diperoleh dari nilai frekuensi tertinggi. Selanjutnya akan diestimasi menggunakan Persamaan (2.1). Hasil estimasi trispektrum berupa magnitude dan fase yang merupakan hasil transformasi dari hasil estimasi trispektrum itu sendiri yang berbentuk kompleks dalam koordinat kartesius, ke dalam koordinat kutub yang berbentuk besaran magnitude dan fase.

5. Magnitude dan Fase

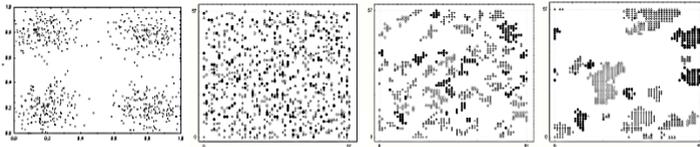
Untuk sebuah besaran X yang mempunyai nilai kompleks, magnitude dari X yaitu $M(X)$ dan fase dari X yaitu $F(X)$, ditentukan dengan menggunakan rumus sebagai berikut :

$$M(X) = \sqrt{\text{Re}(X)^2 + \text{Im}(X)^2} \quad (2.6)$$

$$F(X) = \arctan\left(\frac{\text{Im}(X)}{\text{Re}(X)}\right)$$

2.4 Algoritma Ants Clustering

Pengelompokan (*clustering*) pada lingkungan dari sebuah semut telah dirancang untuk mendapatkan sebuah algoritma baru untuk data analisis dan partisi. Algoritma *Ant Clustering* didasarkan pada perilaku beberapa spesies semut pada saat membersihkan sarang dengan mengelompokkan tumpukan larva sesuai dengan ukurannya. Dalam *Ant Clustering* hal yang pertama dilakukan adalah mendistribusikan data secara acak kedalam sebuah *workspace/grid* [12]. Setelah itu sejumlah semut ditempatkan pada posisi yang berbeda pada *workspace*. Dengan asumsi hanya terdapat satu semut dan hanya satu objek yang ditempatkan pada satu posisi dalam *workspace*. Setiap semut akan membawa satu data.



Gambar 2. 1 Contoh Algoritma Ant Clustering

Pada setiap *step*, setiap semut akan melakukan *picking/dropping* sebuah obyek, dengan probabilitas *picking/dropping* didasarkan pada *disparity* (jarak pada *feature space*) antara objek tersebut dengan objek tetangganya. Fungsi $f(o_i)$ akan meningkatkan *disparity* antara objek i dengan tetangganya. Sehingga untuk $f(o_i)$ dirumuskan sebagai berikut :

$$f(o_i) = \begin{cases} \frac{1}{s^2} \sum_{o_j \in \text{Neigh}(sxs)(r)} \left[1 - \frac{d(i,j)}{\alpha} \right], & \text{untuk } f(o_i) > 0 \\ 0, & \text{lainnya} \end{cases} \quad (2.6)$$

dengan :

- $Neigh(sxs)(r)$ merupakan lokasi yang mengelilingi r sebesar $s \times s$ pada *workspace*
- $d(i, j)$ merupakan jarak Euclidean antara dua objek pada *feature space*
- α merupakan ukuran ketidakmiripan dari setiap pasangan objek

Kemudian probabilitas *picking* dan *dropping* dirumuskan sebagai berikut :

$$p_{pick} = \left(\frac{k_p}{k_p} + f(o_i) \right)^2 \quad (2.7)$$

$$p_{drop} = \begin{cases} 2f(o_i) & \text{jika } f(o_i) < k_d \\ 1 & \text{untuk lainnya} \end{cases} \quad (2.8)$$

dengan :

- k_p merupakan parameter jika melakukan *pick up* item
- k_d merupakan parameter jika melakukan *drop* item

Probabilitas *dropping* sebuah objek akan lebih tinggi bila dekat dengan objek yang sama, dan probabilitas *picking* sebuah objek akan lebih tinggi bila objek tersebut terisolasi atau dekat dengan objek yang tidak sama.

Setelah melakukan *picking/dropping* objek, semut akan bergerak satu langkah secara acak pada jaringan yang ada. Dengan melakukan langkah *picking/dropping* berulang kali maka objek yang berdekatan pada *feature space* cenderung akan dijatukan pada *neighboring position* pada *workspace*. Hal ini akan menyebabkan terbentuknya klaster sementara berukuran kecil. Proses ini akan berulang sejumlah iterasi yang diinginkan sehingga akan meningkatkan ukuran dari klaster.

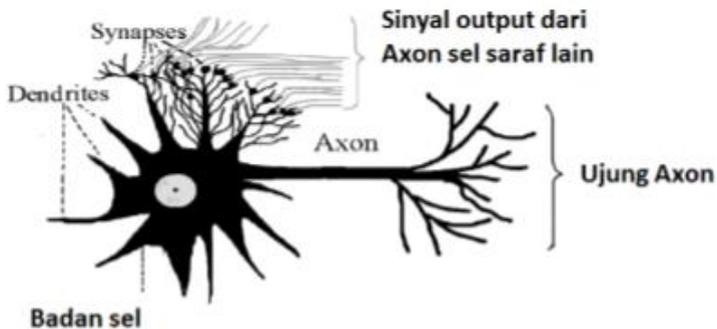
Langkah-langkah dari algoritma *Ant Clustering* adalah sebagai berikut [8] :

Algoritma *Ants Clustering*

0. /*Inisialisasi*/
1. **for** setiap objek o_i **do**
2. letakkan objek secara acak pada garis
3. **endfor**
4. **for** semua semut **do**
5. letakkan semut secara acak pada garis
6. **endfor**
7. /*Perulangan*/
8. **for** semua semut **do**
9. **for** $t = 1$ **to** t_{max} **do**
10. **if** (semut dalam keadaan kosong dan
 (wilayah ditempati oleh objek i)) **then**
11. Hitung $f(o_i)$ dan $p_{pick}(o_i)$
12. Pilih bilangan acak $R \in (0,1)$
13. **if** ($R \leq p_p(o_i)$) **then**
14. Pick up item o_i
15. **end if**
16. **else**
17. **if** (semut dalam keadaan membawa item
 o_i dan (wilayah kosong)) **then**
18. Hitung $f(o_i)$ dan $p_{drop}(o_i)$
19. Pilih bilangan acak $R \in (0,1)$
20. **if** ($R \leq p_d(o_i)$) **then**
21. Drop item
22. **end if**
23. **end if**
24. **end if**
25. Pindahkan secara acak pada wilayah
 tetangga yang tidak ditempati oleh agen
 lain
26. **end for**
27. **end for**
28. **output** lokasi dari item

2.5 Jaringan Syaraf Tiruan

Jaringan Syaraf Tiruan (JST) merupakan suatu sistem pengolahan informasi yang memiliki karakteristik performansi sebagaimana jaringan syaraf pada makhluk hidup. Sistem JST disusun dengan menggunakan model matematis dari jaringan biologi manusia. Salah satu pengambilan ide dari jaringan syaraf biologis adalah adanya elemen-elemen pemrosesan pada jaringan syaraf tiruan yang saling terhubung dan beroperasi secara paralel. Jaringan syaraf biologis yang ditunjukkan pada Gambar 2. merupakan kumpulan dari sel-sel syaraf (*neuron*) yang mempunyai tugas mengolah informasi.



Gambar 2. 2 Sel Syaraf Biologis

Komponen-komponen utama dari sebuah *neuron* adalah :

- Dendrit yang bertugas untuk menerima informasi
- Badan sel (*soma*) berfungsi sebagai tempat pengolahan informasi
- Akson (*neurite*) berfungsi mengirimkan impuls-impuls ke sel syaraf lainnya.

Seperti halnya otak manusia, JST memiliki kemampuan belajar, beradaptasi terhadap perubahan informasi yang diterima, serta belajar dari karakteristik sinyal masukan. JST dapat mensintesis sebuah memori asosiatif yang dapat

menghasilkan sebuah keluaran yang sesuai saat diberi sinyal masukan dan menggeneralisasi saat diberi sinyal masukan yang baru [10].

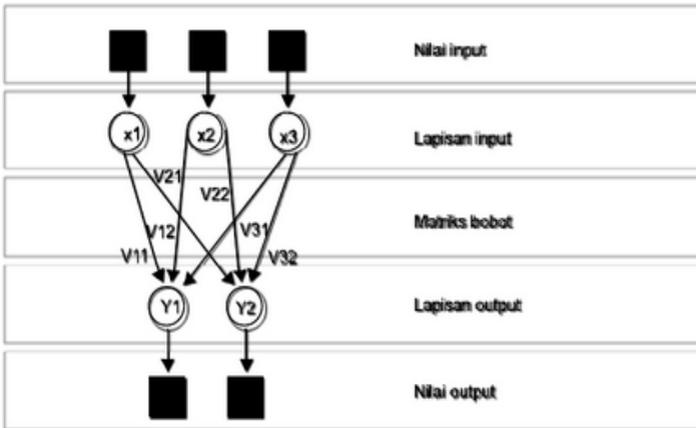
JST telah dikembangkan dengan menggunakan model matematis untuk menirukan cara kerja jaringan syaraf biologis, dengan berdasarkan asumsi-asumsi :

- Pengolah informasi terdiri dari elemen-elemen sederhana yang disebut *neuron*
- Sinyal dilewatkan dari satu *neuron* ke *neuron* yang lain melalui suatu hubungan tertentu
- Tiap hubungan mempunyai ukuran tersendiri yang disebut bobot
- Tiap *neuron* mempergunakan fungsi aktivasi terhadap masukan yang diterimanya untuk menentukan sinyal keluaran.

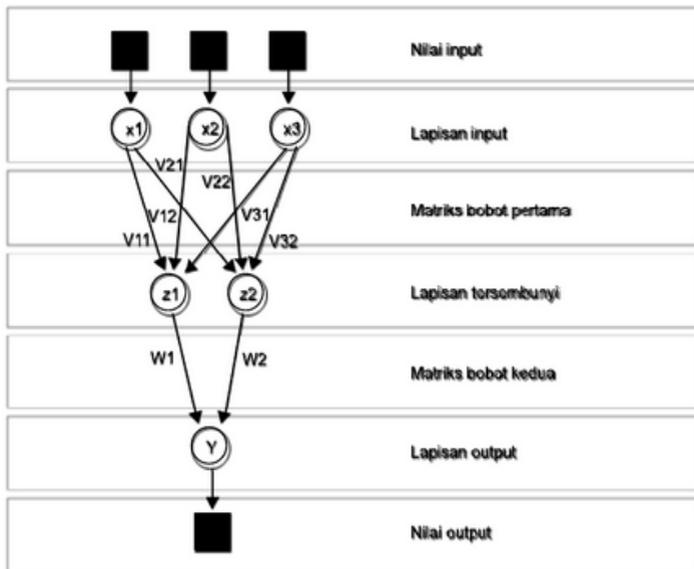
2.5.1 Struktur Jaringan Syaraf Tiruan

Struktur JST mengacu pada struktur jaringan biologi khususnya jaringan syaraf manusia. Seperti jaringan syaraf manusia, JST juga terdiri dari neuron dan keterhubungan antar neuron-neuron. Pada JST istilah untuk menggantikan hubungan tersebut adalah bobot. Informasi berupa sinyal listrik disimulasikan sebagai harga spesifik pada bobot. Dengan cara mengubah-ubah harga bobot artinya kita juga mengubah-ubah struktur hubungan antar neuron.

Semua arsitektur jaringan syaraf tiruan mempunyai struktur yang sama yaitu, satu lapisan masukan (*input layer*), satu lapisan keluaran (*output layer*), dan nol atau lebih lapisan tersembunyi (*hidden layer*). Berdasarkan jumlah lapisannya, jaringan syaraf tiruan dibedakan menjadi beberapa jenis antara lain : jaringan lapis tunggal (*single layer net*) dan jaringan lapis banyak/jamak (*multi layer net*) seperti pada Gambar 2.3 dan 2.4.



Gambar 2. 3 Jaringan Lapis Tunggal



Gambar 2. 4 Jaringan Lapis Jamak

2.5.2 Proses Pelatihan atau Pembelajaran

Target utama dari proses pembelajaran adalah untuk menentukan nilai bobot yang diperoleh berdasarkan masukan yang diberikan. Pada saat pembelajaran diberikan data masukan yang berbeda, maka nilai bobot akan berubah secara dinamis hingga mencapai suatu nilai yang cukup seimbang. Nilai tersebut mengindikasikan bahwa tiap-tiap masukan telah berhubungan dengan *neuron* keluaran, sesuai yang diharapkan.

Proses pembelajaran dalam jaringan syaraf tiruan terdapat 2 kategori yaitu :

1. Pembelajaran Terawasi (*supervised learning*)

Metode pembelajaran dalam jaringan syaraf tiruan dikatakan terawasi, jika keluaran yang diharapkan telah diketahui sebelumnya. Pada proses pembelajaran, satu pola masukan akan diberikan pada satu *neuron* lapisan masukan. Pola ini dirambatkan di sepanjang jaringan syaraf tiruan hingga sampai ke *neuron* pada lapisan keluaran. Lapisan keluaran ini akan membangkitkan pola keluaran yang nantinya akan dicocokkan dengan pola keluaran targetnya. Apabila terjadi perbedaan antara pola keluaran hasil pembelajaran dengan pola target, maka disini akan muncu *error*. Apabila nilai ini masih cukup besar, maka mengindikasikan bahwa masih perlu dilakukan lebih banyak pembelajaran lagi.

2. Pembelajaran Tak Terawasi (*unsupervised learning*)

Pada metode pembelajaran tak terawasi ini tidak memerlukan target keluaran. Pada metode ini, tidak dapat ditentukan seperti hasil apakah yang diharapkan selama proses pembelajaran. Selama proses pembelajaran, nilai bobot disusun dalam suatu *range* tertentu tergantung dari nilai masukan yang diberikan.

2.5.3 Learning Vector Quantization

Motivasi untuk sebuah algoritma yang diterapkan pada jaringan syaraf LVQ adalah untuk menentukan unit *output*

yang terdekat dengan vektor *input*. Hal tersebut akan berakhir, jika x dan w berada dalam kelas yang sama, maka bobot dipindahkan ke vektor *input* yang baru dan jika x dan w berada pada kelas yang berbeda, maka bobot akan dipindahkan dari vektor *input*.

Algoritma dari pembelajaran LVQ secara umum adalah sebagai berikut [4]:

```

Step 0: Inisialisasi vektor referensi
        Inisialisasi alpha
        Inisialisasi decalpha
Step 1: Bila kondisi STOP belum terpenuhi,
        kerjakan Step 2-6
    Step 2: Untuk setiap vektor training x, kerjakan
            Step 3-4
        Step 3: Dapatkan j sedemikian hingga
                ||x-wj|| minimum
        Step 4: Update Wj sebagai berikut:
                Jika T=Cj maka
                    Wj (baru) = Wj (lama) + alpha (x - Wj (lama))
                Jika T != Cj maka
                    Wj (baru) = Wj (lama) - alpha (x - Wj (lama))
        Step 5: Reduksi learning rate (alpha)
                alpha = alpha - alpha * decalpha
        Step 6: Tes kondisi STOP
                - dengan membatasi jumlah iterasi
                - setelah alpha mencapai nilai toleransi

```

dengan :

x : vektor pelatihan (x_1, x_2, \dots, x_n)
 T : kategori atau kelas yang benar untuk vektor pelatihan
 W_j : vektor bobot untuk unit *output*
 $j(w_{1j}, \dots, w_{ij}, \dots, w_{nj})$
 $\|x - w_j\|$: jarak *Euclidean* antara vektor input (vektor bobot) dan unit output ke j
 α : *learning rate*
 decalpha : pengurangan *learning rate*

BAB III

METODOLOGI PENELITIAN

Bab ini menjelaskan tentang tahap pengerjaan dalam menyelesaikan Tugas Akhir ini sehingga penelitian ini dapat dirancang sistematis dan diatur dengan sebaik-baiknya.

3.1 Diagram Metodologi

Gambaran tahap-tahap dalam penelitian pada Tugas Akhir ini disajikan sebagai diagram alir pada Gambar 3.1.

3.2 Studi Literatur

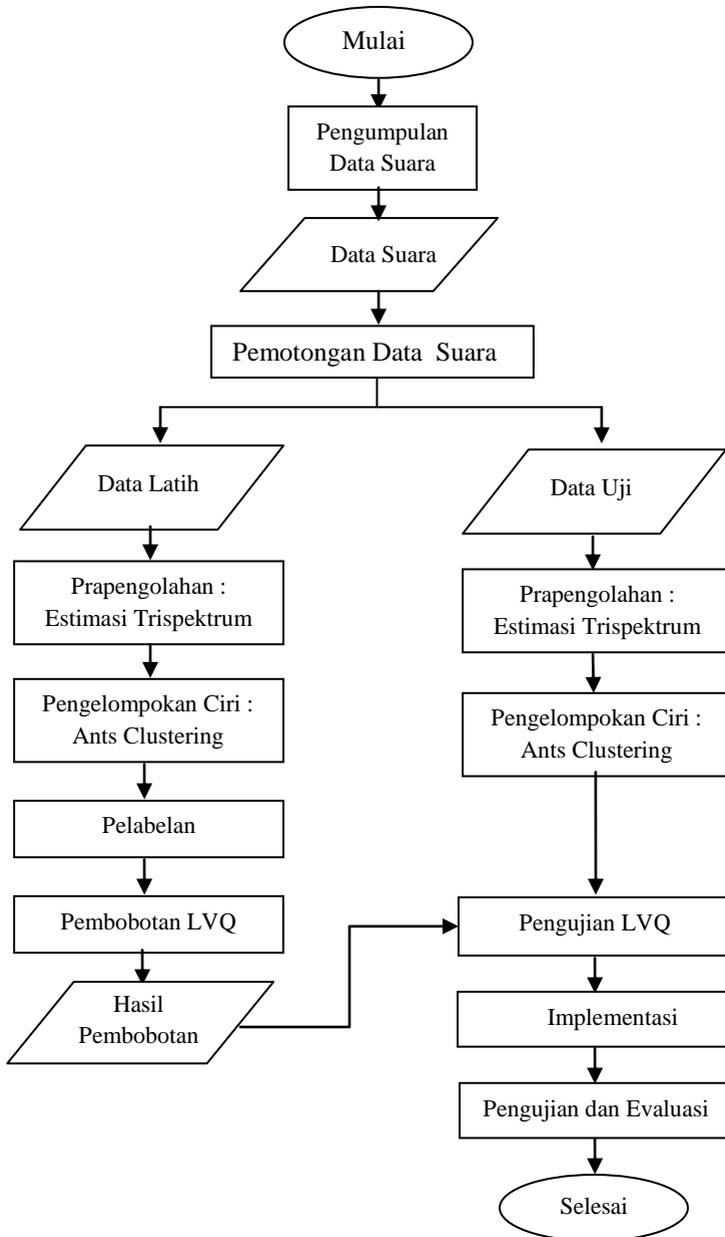
Studi literatur merupakan tahap untuk mengidentifikasi permasalahan dan mencari referensi yang menunjang penelitian. Referensi yang dipakai berupa buku-buku literatur, jurnal ilmiah, tugas akhir atau thesis yang berkaitan dengan permasalahan, maupun artikel dari internet. Tahap ini diperlukan sebagai landasan dalam melakukan penelitian Tugas Akhir ini.

3.3 Pengumpulan Data Suara

Pengumpulan data merupakan tahap mengumpulkan data yang diperlukan dalam penelitian. Data yang digunakan dalam penelitian ini adalah data primer berupa suara individu yang direkam menggunakan *software* GoldWave dan dengan bantuan alat *headphone*.

3.4 Data Suara

Data suara diperoleh dari hasil perekaman diperoleh oleh 30 individu dengan mengucapkan 10 jenis berbeda yaitu maju, mundur, kanan, kiri, atas, bawah, naik, turun, depan dan belakang. Perekaman suara menggunakan *sampling rate* sebesar 48000Hz dan menggunakan format berkas audio *.wav* dengan durasi perekaman sekitar 1-2 detik.



Gambar 3. 1 Diagram Alir Metodologi Penelitian

3.5 Pemotongan Data Suara

Pemotongan data suara merupakan proses memotong sinyal suara dan mengambil bagian dari sinyal suara yang dibutuhkan. Proses ini dilakukan secara manual dengan menggunakan software GoldWave.

3.6 Data Latih dan Data Uji

Data suara individu dibagi menjadi dua bagian, yaitu data latih sebanyak 1500 kata yang diucapkan oleh 30 individu dan data uji sebanyak 300 kata yang diucapkan oleh 30 individu.

3.7 Prapengolahan Data

Pada tahap ini akan dilakukan proses pengolahan terhadap data suara yang bertujuan untuk mendapatkan data magnitude dan fase sinyal suara. Terdapat empat tahapan yang dilakukan dalam prapengolahan data sinyal suara yaitu *frame blocking*, *windowing*, FFT (*Fast Fourier Transform*), Estimasi Trispektrum.

3.8 Pengelompokan Ciri

Pengelompokan ciri merupakan proses untuk menentukan suatu nilai atau vektor yang digunakan sebagai penciri atau perwakilan suatu objek. Dalam Tugas Akhir ini algoritma yang digunakan untuk pengelompokan ciri adalah algoritma *Ants Clustering*.

3.9 Klasifikasi Data

Data suara individu yang telah diperoleh vektor pencirinya melalui proses pengelompokan ciri akan diklasifikasikan menggunakan LVQ. Dalam proses klasifikasi data menggunakan LVQ terdapat dua tahapan, yaitu tahapan pembelajaran pola (*training*) dan tahapan pengenalan pola / pengukuran kemiripan (*testing*). Pada proses *training*, data sinyal suara akan dilabeli dengan nama pemilik suara. Data hasil dari proses *training* akan dikumpulkan dan disimpan sebagai bobot yang nantinya dapat digunakan untuk proses *testing*, sehingga suara individu tersebut dapat dikenali.

3.10 Implementasi Sistem

Pada tahap ini dilakukan pembuatan *interface* sistem dan mengintegrasikannya menjadi sebuah sistem pengenalan individu berbasis suara ucapan menggunakan *software* MATLAB. Input sistem berupa data suara individu. Output sistem berupa label nama individu dari hasil pengenalan.

3.11 Pengujian dan Evaluasi

Pada tahap ini dilakukan simulasi dengan menggunakan data yang telah diperoleh dan menguji output dari sistem apakah sudah sesuai dengan yang diharapkan. Dalam proses pengujian terdapat dua tahap, yaitu tahap *training* dan *testing*. Pada tahap *training* diperlukan input berupa data suara dan label nama individu pemilik suara untuk memperoleh bobot dan bias. Kemudian dilakukan tahap *testing* menggunakan bobot yang diperoleh pada tahap *training* dengan *input* berupa data suara dan akan memberikan *output* berupa label nama individu pemilik suara. Hasil keluaran tersebut kemudian akan dianalisis dengan perhitungan tingkat akurasi sebagai berikut:

$$AC = \frac{n}{N} \times 100\% \quad (3.1)$$

dengan :

AC : persentase akurasi

n : jumlah kebenaran pengenalan individu

N : jumlah data suara pada proses pengenalan.

BAB IV PERANCANGAN SISTEM

Bab ini menjelaskan rancangan desain sistem yang digunakan sebagai acuan untuk implementasi sistem. Desain sistem menggambarkan proses rancang bangun secara terperinci dari awal tahap pengumpulan data hingga proses konstruksi sistem menggunakan metode Estimasi Trispektrum pada tahap prapengolahan, algoritma *Ants Clustering* pada tahap pengelompokan ciri, dan algoritma *Learning Vector Quantization* pada tahap klasifikasi data, serta penjelasan mengenai cara untuk mendapatkan data keluaran yang sesuai dengan tujuan dari penelitian Tugas Akhir ini.

4.1 Akuisisi Data Sinyal Suara

Sinyal suara yang digunakan dalam penelitian Tugas Akhir ini diperoleh dari perekaman 30 individu dengan bantuan *software Goldwave*. Dalam proses perekaman sinyal suara, setiap individu mengucapkan 10 jenis kata. Data sinyal suara yang telah direkam selanjutnya disimpan dalam format *.wav yang dapat dibaca dan diproses oleh perangkat lunak yang akan dibuat dalam penelitian ini.

4.2 Pemotongan Sinyal Suara

Pada saat proses perekaman sinyal suara terdapat jeda atau *silent* yang tersimpan pada data sinyal suara. Jeda tersebut biasanya terdapat pada bagian awal dan akhir dari data sinyal suara. Proses pemotongan sinyal ini bertujuan untuk menghilangkan bagian yang tidak termasuk dari sinyal suara. Proses dari pemotongan sinyal dapat dilihat pada Gambar 4.1.



Gambar 4. 1 Pemotongan Sinyal Suara

4.3 Prapengolahan Data Sinyal Suara

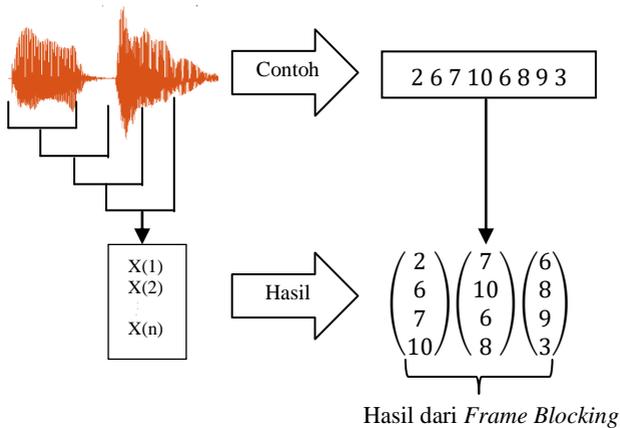
Tujuan dari proses prapengolahan data sinyal suara adalah untuk mengubah bentuk getaran suara menjadi berbagai tipe parameter yang merepresentasikan sinyal suara untuk dianalisa pada proses selanjutnya. Setelah sinyal suara mengalami proses pemotongan sinyal, sinyal suara tersebut akan diolah melalui beberapa tahap dalam proses prapengolahan data.

Adapun tahapan dalam proses prapengolahan data sinyal suara adalah sebagai berikut :

4.3.1 *Frame Blocking*

Frame blocking yang merupakan proses pembagian sinyal menjadi beberapa *frame* yang lebih kecil (*subframe*) dengan ukuran yang sama. Tujuan dari membagi sinyal suara adalah untuk mengurangi waktu komputasi yang sangat lama karena data sinyal suara yang tidak dibagi kedalam *subframe* bernilai sangat besar.

Dalam proses pembagian frame, data sinyal suara sebanyak titik N sampel dibagi menjadi beberapa sub sinyal suara atau dinamakan *chunk* dengan ukuran 128, 256, 512 atau sebanyak 2^7 , 2^8 , 2^9 untuk setiap *chunk*. Hal tersebut dikarenakan dalam proses selanjutnya yaitu proses *Fast Fourier Transform* memerlukan jumlah sampel sebanyak 2^p . Pada Tugas Akhir ini *chunk* yang digunakan dalam proses *frame blocking* adalah 128 dengan *overlapping* sebesar 50%. Ilustrasi dari proses *frame blocking* dapat dilihat pada Gambar 4.2. Pada gambar tersebut terdapat sebuah contoh sinyal dengan 8 titik sampel. Sinyal tersebut akan dibagi menjadi beberapa *subframe* dengan ukuran 4 titik sampel dan ukuran *overlapping* sebanyak 2 titik sampel. Hasil *frame blocking* dari contoh sinyal tersebut menghasilkan 3 *subframe*.



Gambar 4. 2 Contoh Proses Frame Blocking untuk $N = 4$ dan $M = 2$

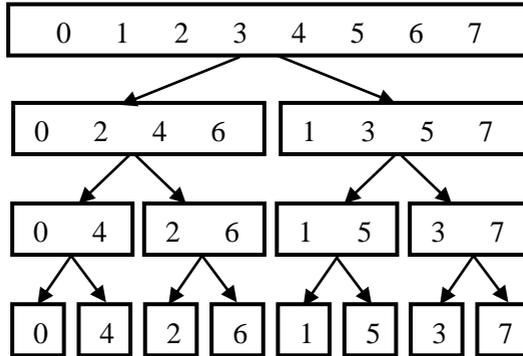
4.3.2 Windowing

Proses selanjutnya adalah menghaluskan masing-masing *frame* untuk meminimalkan sinyal yang tidak kontinu pada awal dan akhir tiap *frame* dengan proses *windowing*. Pada penelitian ini proses *windowing* yang digunakan adalah *Hamming Windowing*. Setiap *frame* akan dikalikan dengan vektor *window* pada Persamaan (2.2).

4.3.3 Fast Fourier Transform

Setelah sinyal suara mengalami proses *windowing*, maka setiap sinyal suara akan dirubah dari domain waktu menjadi domain frekuensi menggunakan FFT. Langkah pertama dari proses FFT dilakukan dengan mendekomposisi sinyal suara berdomain waktu yang memiliki n titik sampel. Hasil dekomposisinya berupa sinyal suara sebanyak n , masing-masing sinyal suara memiliki satu titik sampel. Tahapan untuk proses dekomposisi sebanyak $\log 2^n$. Misalnya untuk sinyal dengan 128 titik sampel (2^7) maka

memerlukan 7 tahap, sinyal dengan 256 titik sampel (2^8) maka memerlukan 8 tahap untuk proses dekomposisi.

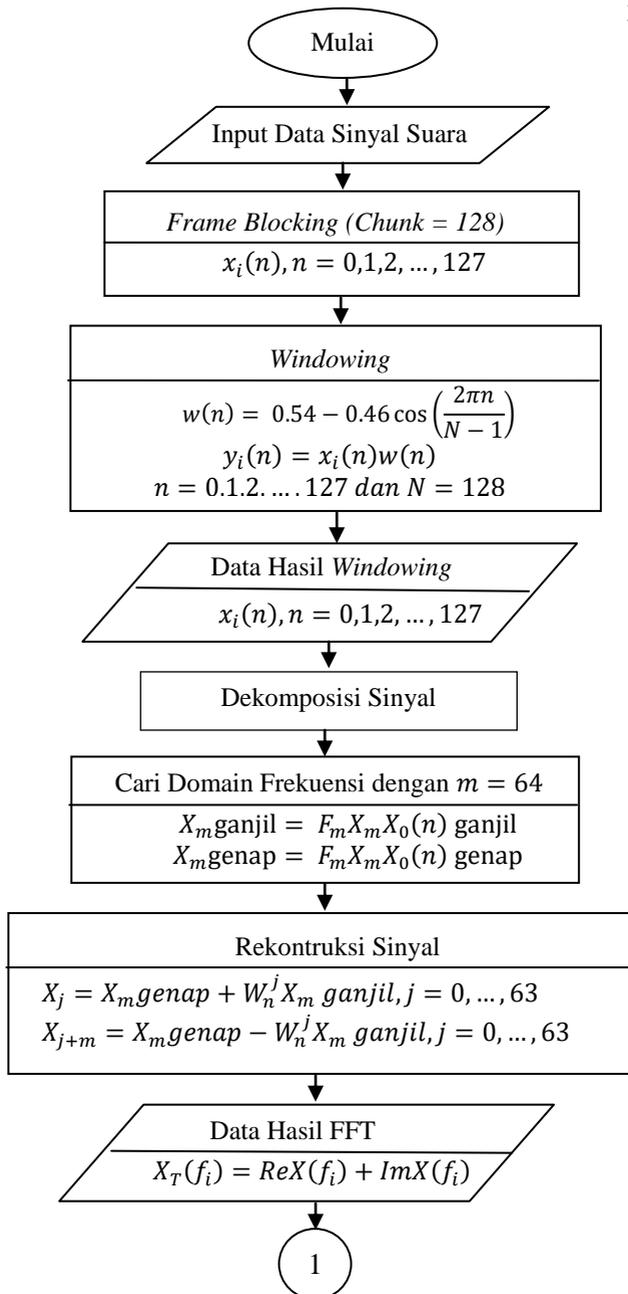


Gambar 4. 3 Contoh Dekomposisi Sinyal

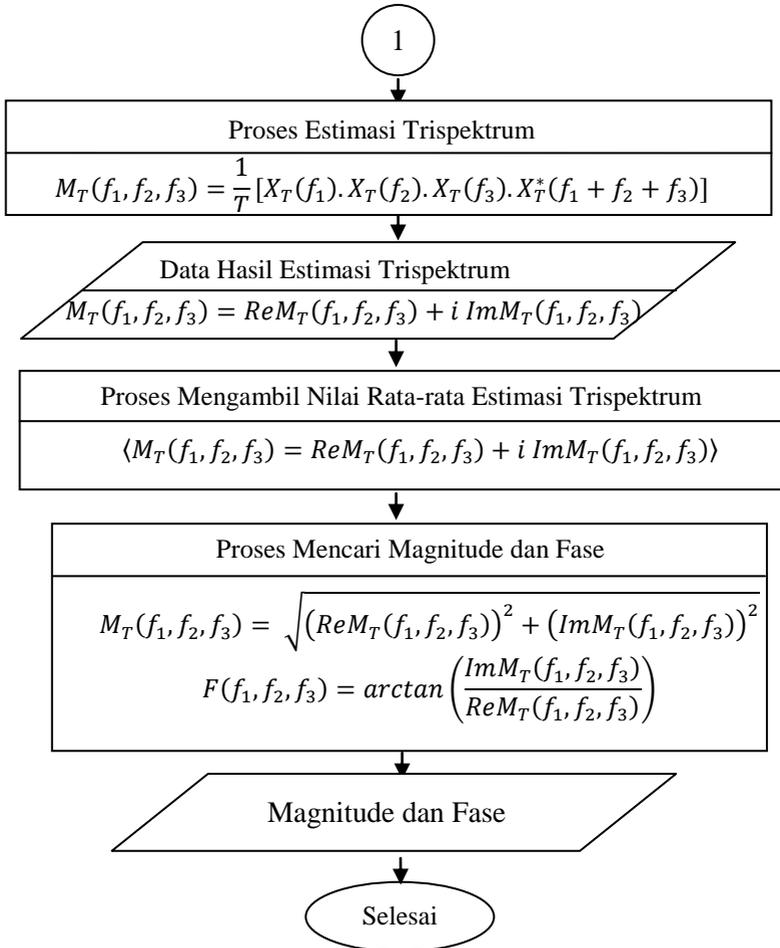
Setelah memperoleh hasil dekomposisi dari sinyal suara maka langkah kedua pada proses FFT adalah mencari domain frekuensi untuk setiap sinyal. Domain frekuensi sinyal diperoleh dengan cara mengalikan sinyal (berupa matriks x) dengan matriks *fourier* F , sehingga diperoleh matriks X_m ganjil dan X_m genap yang masing-masing memiliki elemen sebanyak m ($m = \frac{1}{2}n$). Kemudian dilanjutkan dengan langkah ketiga, yaitu merekonstruksi X_m ganjil dan X_m genap menjadi X_n dengan menggunakan Persamaan (2.5).

4.3.4 Estimasi Trispektrum

Setiap *chunk* yang telah melalui proses FFT, selanjutnya akan dilakukan proses estimasi trispektrum. Nilai sinyal suara setiap *chunk* hasil dari proses FFT akan dimasukkan kedalam Persamaan (2.1) sehingga menghasilkan data estimasi trispektrum berupa ruang empat dimensi yakni $M_T(f_1, f_2, f_3)$. Hasil akhir dari proses trispektrum akan di rata-rata dan kemudian dicari magnitude dan fase sesuai dengan Persamaan (2.6). Diagram alir dari proses prapengolahan data sinyal suara akan ditunjukkan pada Gambar 4.4 dan 4.5.



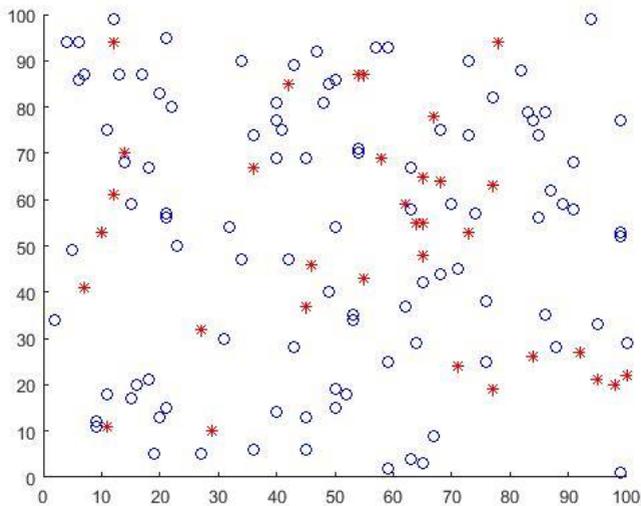
Gambar 4. 4 Diagram Alir Proses Prapengolahan Data Sinyal Suara



Gambar 4. 5 Diagram Alir Proses Prapengolahan Data Sinyal Suara (Lanjutan)

4.4 Pengelompokan Ciri Sinyal Suara

Proses pengelompokan ciri sinyal suara bertujuan untuk mencari magnitudo perwakilan dari data sinyal suara yang telah mengalami proses prapengolahan. Dengan menggunakan metode *Ant Clustering*, magnitudo data sinyal suara akan diletakkan secara acak pada *workspace/grid* sebagai item. Item tersebut akan dipindahkan oleh semut yang diletakkan pada posisi yang berbeda. Dengan asumsi hanya terdapat satu semut dan hanya satu objek yang ditempatkan pada satu posisi dalam *workspace*.



Gambar 4. 6 Contoh Peletakkan Item Data

Pada setiap iterasi, setiap semut akan melakukan *picking/dropping* sebuah item, dengan probabilitas *picking/dropping* didasarkan pada *disparity* (jarak pada *feature space*) antara item tersebut dengan item tetangganya. Fungsi $f(o_i)$ yang digunakan untuk meningkatkan *disparity* adalah :

$$f(o_i) = \begin{cases} \frac{1}{s^2} \sum_{o_j \in \text{Neigh}(sxs)(r)} \left[1 - \frac{d(i,j)}{\alpha} \right], & \text{untuk } f(o_i) > 0 \\ 0, & \text{lainnya} \end{cases}$$

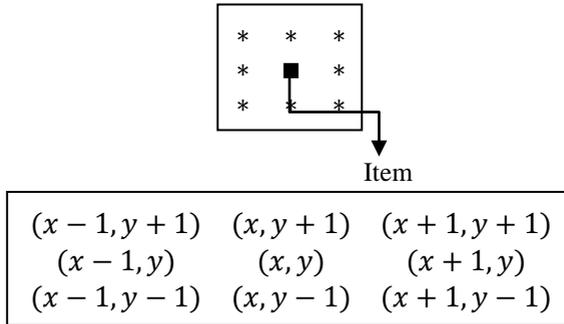
karena $d(i,j)$ merupakan jarak Euclidan antara 2 item pada *feature space* maka apabila disubtitusikan ke dalam $f(i)$ adalah sebagai berikut :

$$d(i,j) = \sqrt{\sum_{k=0}^{N-1} (x_{ik} - x_{jk})^2}$$

dengan N : jumlah elemen vektor dari $data_i$ dan $data_j$
 x_i : elemen dari $data_i$
 x_j : elemen dari $data_j$
 k : indeks $0 < k < N - 1$

$$f(o_i) = \frac{1}{s^2} \sum_{o_j \in \text{Neigh}(sxs)(r)} \left[1 - \frac{\sqrt{\sum_{k=0}^{N-1} (x_{ik} - x_{jk})^2}}{\alpha} \right]$$

Ketetanggaan dari item didefinisikan dalam 8 cell tetangga (*neighboring cells*) yakni apabila item berada pada posisi (x,y) maka tetangganya seperti yang ditunjukkan oleh Gambar 4.6.

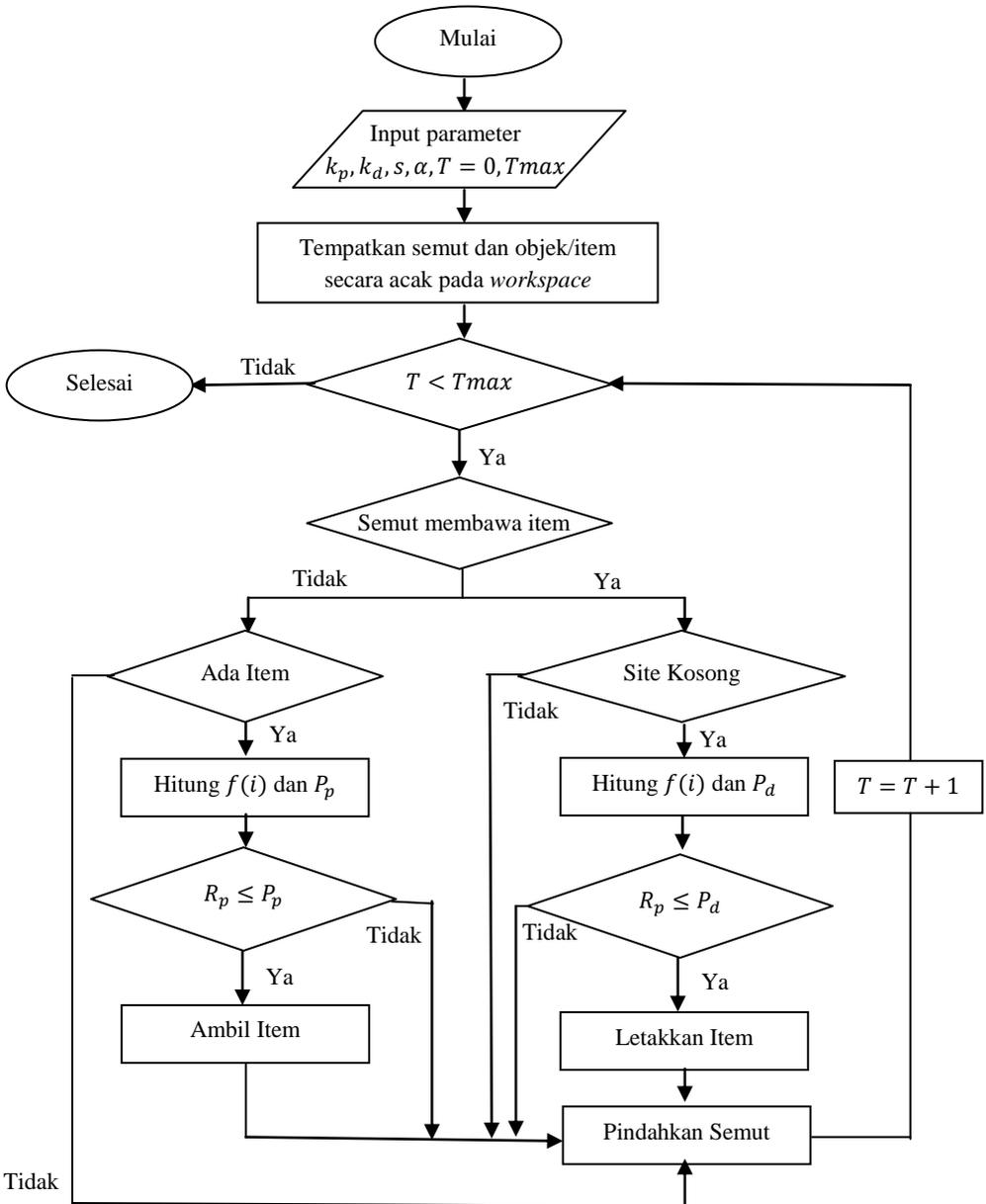


Gambar 4. 7 Neighboring Cells dalam koordinat Cartesius

Selanjutnya setelah mengitung $f(o_i)$, maka akan dihitung probabilitas *picking* dan *dropping* seperti pada Persamaan (2.7) dan (2.8). Data yang telah dipindahkan oleh semut akan dikelompokkan dengan data yang mirip. Proses perpindahan yang dilakukan oleh semut akan dilakukan sampai iterasi maksimum.

Dalam penelitian ini penentuan hasil pengelompokan diperoleh dengan mencari nilai rata-rata sekumpulan item yang telah berkelompok pada setiap *workspace*. Selanjutnya akan dilakukan dengan melakukan pembagian pada *workspace* sesuai dengan jumlah pengelompokan yang telah ditentukan. Jumlah pengelompokan yang digunakan adalah sebesar 8 pengelompokan. Selanjutnya data tersebut akan menjadi perwakilan magnitude yang akan digunakan untuk proses klasifikasi.

Adapun *flowchart* dari algoritma *Ant Clustering* adalah sebagai berikut :



Gambar 4. 8 Diagram Alir Algoritma Ant Clustering

4.5 Klasifikasi Data Sinyal Suara

Dalam proses klasifikasi menggunakan metode LVQ secara umum terbagi menjadi dua tahap yaitu proses *training* dan proses *testing*. Pada proses *training* memerlukan input berupa data perwakilan magnitude dari sinyal suara yang telah melalui tahap pengelompokan ciri. Data tersebut akan diberi label berupa nama dari individu pemilik suara. Output dari proses *training* adalah berupa bobot akhir dari tiap individu pemilik suara. Bobot ini selanjutnya akan dipakai untuk proses pengenalan objek pada proses *testing*. Proses *testing* memerlukan input berupa data suara yang telah dilakukan proses pemotongan dan akan memberikan output berupa hasil pengenalan objek. Berikut penjabaran dari proses berjalannya sistem pada tiap tahap.

4.5.1 Proses *Training*

Proses *training* memerlukan input berupa data perwakilan magnitude dari sinyal suara dan label dari nama individu pemilik suara yang masing-masing akan mewakili matriks input x dan matriks target t . Sebelum diproses menggunakan metode LVQ, sinyal suara akan melewati proses prapengolahan dan pengelompokan ciri yang menghasilkan data magnitude dengan ukuran matriks 1×8 tergantung jumlah kelompok yang diinginkan. Data magnitude tersebut akan menjadi elemen input x yang terdiri dari 8 elemen. Label nama individu pemilik suara akan dirubah menjadi matriks t yang terdiri dari 1 elemen. Representasi label nama objek dalam matriks t dapat dilihat pada Tabel 4.1.

Tabel 4. 1 Tabel Representasi Label Nama

No	Label Nama Individu Pemilik Suara	Representasi dalam Matriks t
1	Airin	1
2	Astika	2
3	Ayu Enita	3
4	Azaria	4

5	Candra	5
⋮	⋮	⋮
30	Vicky	30

Berikut adalah contoh proses *training* dengan LVQ :

Misal akan dilakukan proses *training* untuk pengenalan 2 jenis objek yaitu A dan B. Arsitektur LVQ yang digunakan terdiri dari 8 node pada *input layer*, dan 1 node pada *output layer* sehingga diperoleh

$$x_i = (x_{i1}, x_{i2}, x_{i3}, x_{i4}, x_{i5}, x_{i6}, x_{i7}, x_{i8}), t_i = (t_1).$$

Misal diberikan data *training* seperti pada Tabel 4.2 sebagai berikut :

Tabel 4. 2 Contoh Data Training

No	Mangnitude perwakilan hasil pengelompokan ciri	Objek Sinyal Suara Individu	
		Label Nama Individu	Representasi dalam Matriks t
1	$x_1 = (0,1,1,1,0,0,0,1)$	A	1
2	$x_2 = (0,0,1,1,1,1,1,0)$	A	1
3	$x_3 = (0,1,1,1,0,0,1,1)$	B	2
4	$x_4 = (1,1,1,1,0,0,0,1)$	B	2

Dari data *training* ini didapatkan matriks input :

$$x_1 = (0,1,1,1,0,0,0,1)$$

$$x_2 = (0,0,1,1,1,1,1,0)$$

$$x_3 = (0,1,1,1,0,0,1,1)$$

$$x_4 = (1,1,1,1,0,0,0,1)$$

dan matriks target $t_1 = [1]$ dan $t_2 = [2]$.

Dua vektor pertama digunakan untuk inialisasi dua vektor referensi. Kemudian, *unit output* pertama merepresentasikan target 1, yang kedua target 2 (secara simbolik, $C_1 = 1$ dan $C_2 = 2$). Tersisa vektor $(0,0,1,1,1,1,1,0)$ dan $(1,1,1,1,0,0,0,1)$ sebagai vektor *training*. Akan ditunjukkan perhitungan dari algoritma LVQ hanya satu iterasi yaitu sebagai berikut :

Step 0. Inisialisasi bobot

$$w_1 = (0,1,1,1,0,0,0,1)$$

$$w_2 = (0,1,1,1,0,0,1,1)$$

Inisialisasi *learning rate* : $\alpha = 0.1$

Step 1. Mulai iterasi

Step 2. Untuk *input* vektor $x = (0,0,1,1,1,1,1,0)$ dengan $t = 1$, lakukan langkah 3-4

Step 3. $J = 2$, karena $\|x - w_2\|$ minimum

Step 4. Karena $t = 1$ dan $C_J = 2$, *update*

w_2 sebagai berikut :

$$w_2 = (0,1,1,1,0,0,1,1) - 0.1[(0,0,1,1,1,1,1,0) - (0,1,1,1,0,0,1,1)]$$

$$w_2 = (0,1,1,1,1, -0.1, -0.1, 1, 1, 1)$$

Step 3. Untuk *input* vektor $x = (1,1,1,1,0,0,0,1)$ dengan $t = 2$, lakukan langkah 3-4

Step 3. $J = 1$, karena $\|x - w_1\|$ minimum

Step 4. Karena $t = 2$ dan $C_J = 1$, *update*

w_1 sebagai berikut :

$$w_1 = (0,1,1,1,0,0,0,1) - 0.1[(1,1,1,1,0,0,0,1) - (0,1,1,1,0,0,0,1)]$$

$$w_1 = (-0.1, 1, 1, 1, 0, 0, 0, 1)$$

Step 5. Iterasi pertama selesai

Reduksi *learning rate*

Step 6. Cek kondisi berhenti.

Proses *training* akan menghasilkan nilai bobot akhir yang akan digunakan dalam proses *testing* atau pengujian.

4.5.2 Proses *Testing*

Setelah dilakukan proses *training*, maka selanjutnya akan dilakukan *testing* dengan bobot akhir yang dihasilkan dari proses *training*. Proses *testing* memerlukan input berupa sinyal suara dan akan memberikan output berupa hasil pengenalan individu. Sebelum dilakukan pengenalan oleh LVQ, sinyal suara input akan melalui tahap prapengolahan sehingga akan dihasilkan magnitude dan fase berukuran

besar. Kemudian data tersebut akan melalui proses pengelompokan ciri untuk mengurangi ukuran data.

Hasil pengenalan objek sinyal suara akan ditentukan dari Tabel 4.1 dengan melihat label nama objek yang bersesuaian dengan indeks hasil. Berikut adalah contoh proses *testing* atau pengenalan objek suara dengan LVQ :

Misal diberikan matriks dari hasil prapengolahan data dan telah direduksi yaitu : $x_1 = (0,1,0,1,0,0,0,1)$. Dengan menggunakan bobot yang telah didapatkan pada contoh proses *training*, maka dapat dihitung jarak terdekat antara input pada *testing* dan bobot akhir :

$$w_1 = (-0.1, 1, 1, 1, 0, 0, 0, 1)$$

$$w_2 = (0, 1, 1, 1, 1, -0.1, -0.1, 1, 1, 1)$$

$$\begin{aligned} \|x - w_1\| &= \left((0 - (-0.1))^2 + (1 - 1)^2 + (0 - 1)^2 \right. \\ &\quad \left. + (1 - 1)^2 + (0 - 0)^2 + (0 - 0)^2 \right. \\ &\quad \left. + (0 - 0)^2 + (1 - 1)^2 \right)^{\frac{1}{2}} \end{aligned}$$

$$= \sqrt{0.01 + 0 + 1 + 0 + 0 + 0 + 0 + 0}$$

$$= \sqrt{1.01} = 1.004$$

$$\begin{aligned} \|x - w_2\| &= \left((0 - 0)^2 + (1 - 1.1)^2 + (0 - 1)^2 + (1 - 1)^2 \right. \\ &\quad \left. + (0 - (-0.1))^2 + (0 - (-0.1))^2 + (0 - 1)^2 \right. \\ &\quad \left. + (1 - 1.1)^2 \right)^{\frac{1}{2}} \end{aligned}$$

$$= \sqrt{0 + 0.01 + 1 + 0 + 0.01 + 0.01 + 1 + 0.01}$$

$$= \sqrt{2.04} = 1.42$$

Karena $\|x - w_1\| < \|x - w_2\|$, maka jarak terdekat matriks x_1 adalah dengan w_1 sehingga dapat disimpulkan bahwa x_1 termasuk dalam kelas $t=1$. Berdasarkan pada Tabel 4.2, matriks x_1 merupakan representasi dari label nama objek A sehingga sinyal suara x_1 dikenali sebagai individu A.

4.6 Pengujian dan Evaluasi Sistem

Pada proses pengujian sistem terdapat dua tahap yaitu tahap *training* dan tahap *testing*. Hasil keluaran dari proses *testing* kemudian akan dianalisis sesuai dengan Persamaan

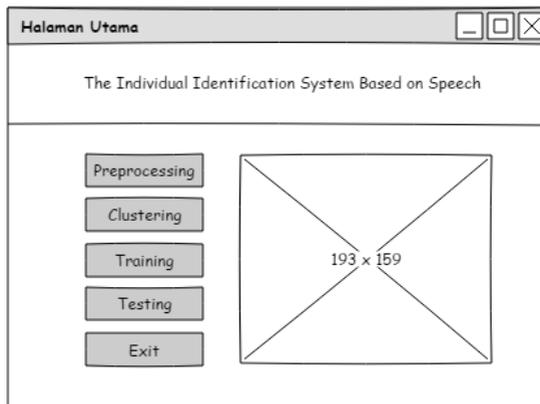
(4.1). Dalam penelitian ini akan dilakukan proses uji coba (*testing*) terhadap data training dan data testing.

4.7 Perancangan *User Interface*

Perancangan *user interface* mendeskripsikan rancangan tampilan yang digunakan pada perangkat lunak yang akan dibuat. Perancangan *user interface* terdiri dari perancangan halaman utama, halaman prapengolahan data, halaman pengelompokan ciri, halaman *training* dan halaman *testing*. Adapun gambaran dari rancangan *user interface* adalah sebagai berikut :

1. Halaman Utama

Halaman utama merupakan tampilan awal pada saat program dijalankan. Pada halaman utama terdapat menu-menu yang akan digunakan untuk proses pengenalan individu.



Gambar 4. 9 Halaman Utama

2. Halaman Prapengolahan Data

Halaman prapengolahan data merupakan halaman untuk melakukan proses prapengolahan data sinyal suara.

Gambar 4. 10 Halaman Prapengolahan Data

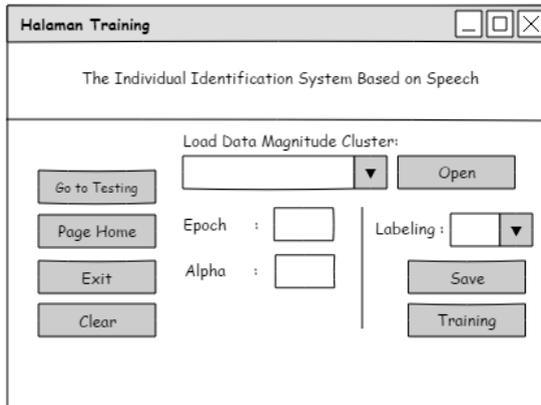
3. Halaman Pengelompokan Ciri

Halaman pengelompokan ciri merupakan halaman untuk melakukan proses pengelompokan ciri sinyal suara menggunakan metode *Ants Clustering*.

Gambar 4. 11 Halaman Pengelompokan Ciri

4. Halaman *Training*

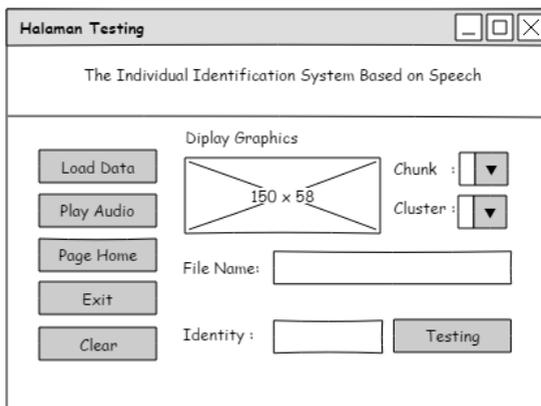
Halaman *training* digunakan untuk melakukan proses pelatihan atau pembelajaran.



Gambar 4. 12 Halaman Training

5. Halaman *Testing*

Halaman *testing* merupakan halaman untuk melakukan proses pengenalan individu berbasis suara ucapan.



Gambar 4. 13 Halaman Testing

BAB V IMPLEMENTASI SISTEM

Pada bab ini, dibahas mengenai langkah-langkah dalam pengimplementasian sistem berdasarkan desain sistem yang telah dirancang.

5.1 Lingkungan Hardware dan Software

Lingkungan perancangan sistem dibangun dari dua lingkungan yaitu lingkungan *software* dan lingkungan *hardware*. Spesifikasi lingkungan perancangan sistem secara lengkap dapat dilihat pada Tabel 5.1.

Tabel 5. 1 Spesifikasi *Hardware* dan *Software*

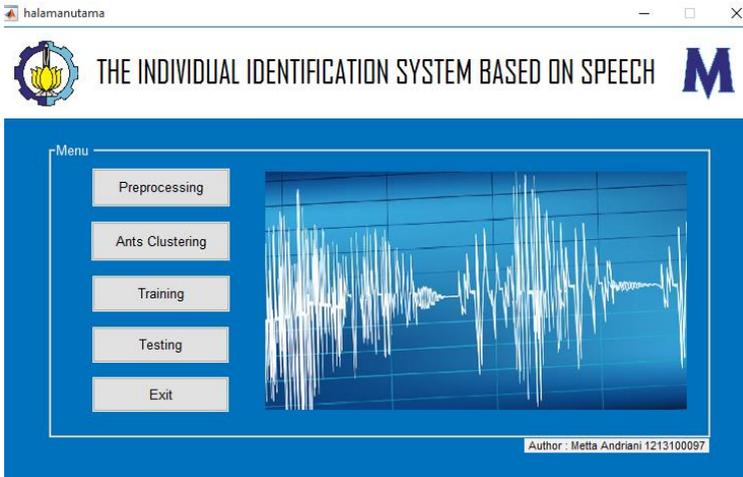
Lingkungan	Spesifikasi	
<i>Hardware</i>	<i>Processor</i>	Intel® Celeron® CPU 847 @1.10GHz
	RAM	4 GB
<i>Software</i>	Sistem Operasi	Windows 10 Pro 64-bit
	<i>Tools</i>	Matlab R2015 GoldWave 6.26

5.2 Implementasi *User Interface*

Penjelasan dari desain implementasi *user interface* pengenalan individu berbasis suara ucapan adalah sebagai berikut :

1. Halaman Utama

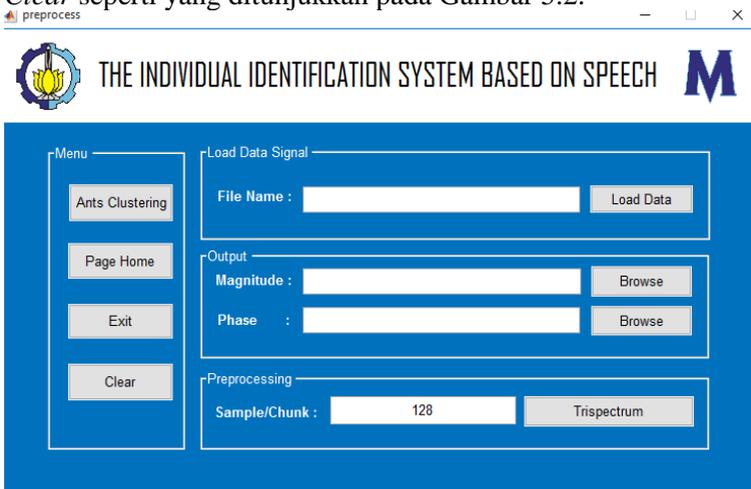
Halaman utama merupakan tampilan awal pada saat program dijalankan. Pada halaman utama terdapat tombol *preprocessing*, *ants clustering*, *training*, *testing*, dan *exit* seperti yang ditunjukkan pada Gambar 5.1. Tombol tersebut digunakan untuk membuka halaman yang diinginkan.



Gambar 5. 1 *Running* Halaman Utama

2. Halaman Prapengolahan Data

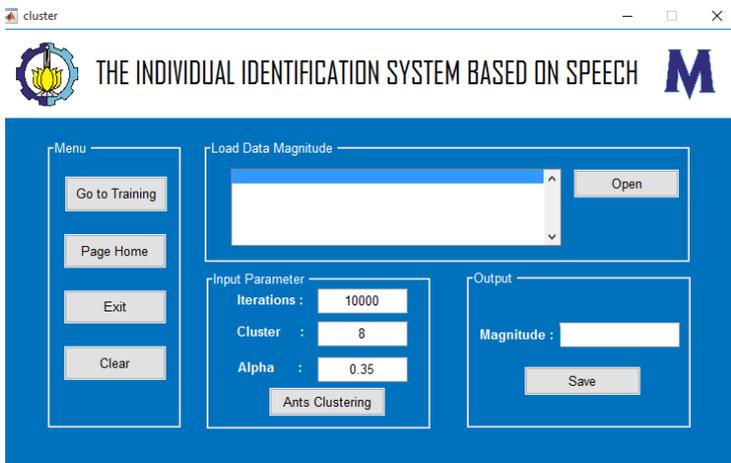
Pada halaman prapengolahan terdapat tombol *Load Data*, *Browse*, *Trispectrum*, *Ants Clustering*, *Page Home*, *Exit*, *Clear* seperti yang ditunjukkan pada Gambar 5.2.



Gambar 5. 2 *Running* Halaman Prapengolahan Data

3. Halaman Pengelompokan Ciri

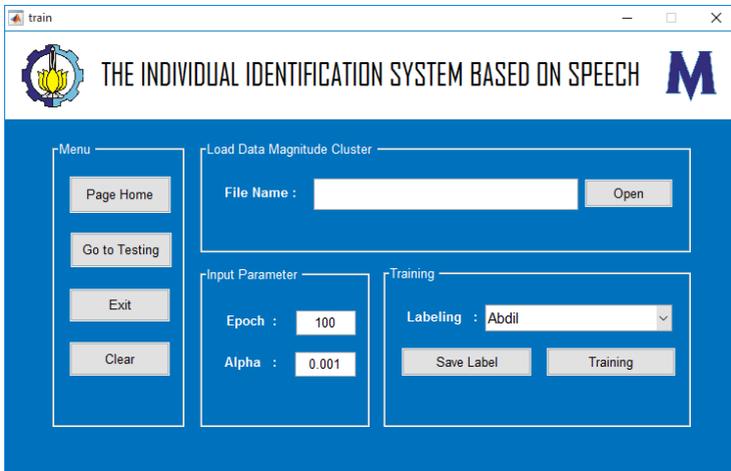
Pada halaman pengelompokan ciri terdapat tombol *Open*, *Ants Clustering*, *Save*, *Go to Training*, *Page Home*, *Exit*, *Clear* seperti yang ditunjukkan pada Gambar 5.3.



Gambar 5. 3 *Running* Halaman Pengelompokan Ciri

4. Halaman *Training*

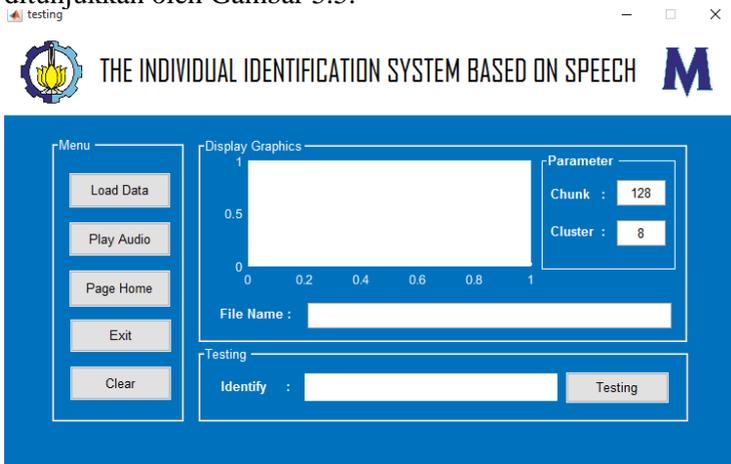
Pada halaman *training* terdapat tombol *Open*, *Save Label*, *Training*, *Go to Testing*, *Page Home*, *Exit*, *Clear* seperti yang ditunjukkan oleh Gambar 5.4.



Gambar 5. 4 *Running Halaman Training*

5. Halaman *Testing*

Pada halaman *testing* terdapat tombol *Load Data*, *Play Audio*, *Page Home*, *Exit*, *Clear*, *Testing* seperti yang ditunjukkan oleh Gambar 5.5.



Gambar 5. 5 *Running Halaman Testing*

5.3 Implementasi Prapengolahan Data Sinyal Suara

Langkah pertama yang dilakukan dalam proses prapengolahan data adalah *frame blocking*. Sinyal suara yang dijadikan *input* dalam proses *frame blocking* adalah sinyal suara yang telah melalui proses *cropping*. Sinyal suara hasil *frame blocking* dapat dilihat pada Lampiran 2(a). Implementasi program untuk *frame blocking* adalah sebagai berikut :

```

nx = length(x);
M = 64;
nbFrames = ceil((length(x)-N)/M);
Frames = zeros(nbFrames+1,N);
for i = 0:nbFrames-1
temp = x(i*M+1:i*M+N);
Frames(i+1,1:N) = temp;
end
temp = zeros(1,N);
lastLength = length(x) - nbFrames*M;
temp(1:lastLength) =
x(nbFrames*M+1:(nbFrames*M + 1 + lastLength-
1));
Frames(nbFrames+1, 1:N) = temp;

```

Langkah selanjutnya adalah proses *windowing* dengan menerapkan *Hamming window*. Sinyal suara hasil dari *windowing* dapat dilihat pada Lampiran 2(b). Implementasi program untuk proses *windowing* adalah sebagai berikut :

```

frameSize = size(Frames);
nbFrames = frameSize(1);
nbSamples = frameSize(2);
w = hamming(nbSamples);
Windows = zeros(nbFrames,nbSamples);
for i = 1:nbFrames
temp = Frames(i,1:nbSamples);
Windows(i, 1:nbSamples) = w'.*temp;
end

```

Setelah dilakukan *Hamming windowing*, sinyal suara akan diproses dengan FFT. Sinyal suara hasil dari proses ini dapat dilihat pada Lampiran 2(c). Implementasi program untuk proses FFT adalah sebagai berikut:

```
fffts = fft(Windows');
```

Kemudian dilakukan proses estimasi sinyal suara yang didasarkan pada korelasi tiga parameter frekuensi yaitu f_1 , f_2 dan f_3 dengan estimasi trispektrum. Setelah itu dilakukan perhitungan untuk mendapatkan nilai magnitude dan fase hasil estimasi trispektrum. Visualisasi hasil dari proses estimasi trispektrum dapat dilihat pada Lampiran 2(d). Implementasi program untuk proses estimasi trispektrum adalah sebagai berikut :

```
for i=1:nbFrames
for j=1:(nbSamples-2)
    f1(j,i) = ffts(j,i);
    f2(j,i) = ffts(j+1,i);
    f3(j,i) = ffts(j+2,i);
    sumft(j,i)=f1(j,i)+f2(j,i)+f3(j,i);
    mulft(j,i)=f1(j,i)*f2(j,i)*f3(j,i);
    tsp(j,i)=(nx/(fs*nbFrames))*(mulft(j,i)
    )* conj(sumft(j,i)));
    Re(j,i) = real(tsp(j,i));
    Im(j,i) = imag(tsp(j,i));
    sudut(j,i)= Im(j,i)/Re(j,i);
end
end
meantsp = mean(tsp);
magnitude = abs(meantsp);
meansud = mean(sudut);
phase = atan(meansud);
m = magnitude;
ph = phase;
```

5.4 Implementasi Pengelompokan Ciri

Tahap awal dari proses pengelompokan ciri adalah inisialisasi awal dari parameter yang dibutuhkan oleh algoritma *Ants Clustering*. Implementasi program untuk inisialisasi awal adalah sebagai berikut :

```
% Inisialisasi
ukuranKoloni = jumData;
xAxis = 1:(ukuranKoloni);
yAxis = 1:(ukuranKoloni);
kp = 0.2;
kd = 0.1;
s=8;
antCount = 50;
itemMat = zeros(ukuranKoloni);
antMat = zeros(ukuranKoloni);
itemLoc = zeros(jumData,2);
antLoc = zeros(antCount,2);
antMouths = zeros(antCount,1);
```

Proses selanjutnya adalah meletakkan setiap semut pada sebuah *workspace*. Implementasi program untuk peletakkan semut adalah sebagai berikut :

```
for i=1:antCount
    exit = 0;
    while(exit == 0)
        xPos = ceil(rand*ukuranKoloni);
        yPos = ceil(rand*ukuranKoloni);
        flag = 0;
        if(antMat((xPos),(yPos))== 0 && xPos ~=0 &&
        yPos ~= 0)
            antMat((xPos),(yPos)) = i;
            antLoc(i,1) = xPos;
            antLoc(i,2) = yPos;
            exit = 1;
        end
    end
end
```

Selanjutnya adalah meletakkan setiap data item pada sebuah *workspace*. Implementasi program untuk peletakkan data item adalah sebagai berikut :

```

for i=1:jumData
    exit = 0;
while (exit == 0)
    xPos = ceil(rand*ukuranKoloni);
    yPos = ceil(rand*ukuranKoloni);
if (itemMat(ceil(xPos),ceil(yPos))== 0 &&
xPos ~=0 && yPos ~= 0)
    itemMat(ceil(xPos),ceil(yPos))
= i;
        itemLoc(i,1) = xPos;
        itemLoc(i,2) = yPos;
        exit = 1;
end
end
end

```

Selanjutnya pada setiap iterasi, setiap semut akan melakukan *picking/dropping* sebuah item yang didasarkan pada $f(o_i)$. Pada saat perhitungan $f(o_i)$, setiap item dan semut akan dihitung *cell* ketetanggaannya. Implementasi program untuk perhitungan *cell* ketetangaan dari item dan semut adalah sebagai berikut :

```

xPos = antLoc(ant,1);
yPos = antLoc(ant,2);
% menghitung ketetanggaan semut
dirOptions = [xPos yPos+1; xPos+1
yPos+1; xPos+1 yPos; xPos+1 yPos-1; xPos
yPos-1; xPos-1 yPos-1; xPos-1 yPos; xPos-1
yPos+1];
dirLim = ones(1,8);
if (xPos+1 > ukuranKoloni)
    dirLim([2 3 4]) = 0;
end
if (xPos-1 <= 0)
    dirLim([6 7 8]) = 0;
end
if (yPos+1 > ukuranKoloni)
    dirLim([1 2 8]) = 0;
end
if (yPos-1 <= 0)
    dirLim([4 5 6]) = 0;
end
% meninjau posisi semut yang akan
dipindahkan
for t = 1:8
if (dirLim(t) == 1)
if (find(antLoc(:,1) == dirOptions(t,1)))
if (find(antLoc(:,2) == dirOptions(t,2)))
    dirLim(t) = 0;
end
end
end
end

```

Proses selanjutnya adalah mengecek setiap semut berada dalam kondisi membawa item atau tidak. Implementasi dari pengecekan kondisi semut yang akan dipindahkan adalah sebagai berikut :

```

% jika semut mempunyai makanan mulutnya,
maka tidak bisa berpindah
if(antMouths(ant,1) ~= 0)
for t = 1:8
if(dirLim(t) == 1)
if(itemMat(dirOptions(t,1),dirOptions(t,2))
~= 0)
dirLim(t) = 0;
end
end
end
end
% mengatur posisi perpindahan
move2 =
dirOptions(validDirIndeces(maxProbIndex(1),
1),:);

```

Setelah semut dicek kondisinya, maka semut akan dipindahkan untuk mencari item. Implementasi dari perpindahan semut adalah sebagai berikut :

```

antMat(move2(1,:)) = ant;
antLoc(ant,:) = move2(1,:);
if(antMouths(ant,1) ~= 0)
itemMat(xPos,yPos) = 0;

itemMat(move2(1,1),move2(1,2)) = antMouths(an
t,1);
itemLoc(antMouths(ant,1),:)
= move2(1,:);
end
nxPos = move2(1,1);
nyPos = move2(1,2);

```

Semut yang siap untuk melakukan perpindahan, akan dihitung $f(o_i)$ untuk menentukan probabilitas *picking/dropping*. Implementasi dari perhitungan $f(o_i)$ adalah :

```

if (antMouths (ant,1) ~= 0)
    len = 0;
    loc = zeros (8,1);
    distance (1)=0;
    f = 0;
for t=1:8
if (foodDirLim (1,t) ~=0)
if (itemMat (foodOptions (t,1), foodOptions (t,2)
)) ~= 0)
loc (t,1) = 1;
distance (len+1) =
sum ( (data (itemMat (foodOptions (t,1), foodOptions (t,2)), :) -
data (antMouths (ant,1), :)) .^2) .^0.5;
f = f + (1 - (distance (len+1) / alpha));
len = len+1;
end
end
end

```

Implementasi dari proses *picking/dropping* adalah sebagai berikut :

```

f = f / (s*s);
ppick = (kp / (kp+f))^2;
if (f < kd)
    pdrop = 2*f;
else pdrop = 1; end
if (rand <= pdrop)
% drop
itemMat (nxPos, nyPos) = antMouths (ant,1);
itemLoc (antMouths (ant,1), :) = [nxPos
nyPos];
antMouths (ant,1) = 0;
end

```

```

if (length(distance) >= 1)
    f = f/(s*s);
    ppick = (kp/(kp+f))^2;
    if (rand <= ppick)
        % pick
        antMouths(ant,1) = itemMat(nxPos,nyPos);
        itemMat(nxPos,nyPos) = 0;
    end
end

```

Proses algoritma *Ants Clustering* dijalankan sampai iterasi maksimum. Apabila iterasi telah selesai, maka data akan digrid menjadi beberapa bagian sesuai jumlah kluster yang diinginkan. Dari pembagian tersebut dicari nilai rata-rata dari tiap item dalam satu bagian grid.

5.5 Implementasi Klasifikasi Data

Dalam proses klasifikasi menggunakan LVQ terdapat dua tahap yaitu proses *training* dan *testing*. Implementasi dari proses *training* dan *testing* adalah sebagai berikut :

5.5.1 Proses *Training*

Proses *training* memerlukan *input* berupa data perwakilan magnitude dari sinyal suara dan label dari nama individu pemilik suara yang masing-masing akan mewakili matriks *input x* dan matriks target *t*. Daftar dari label pemilik individu dapat dilihat pada Lampiran 3. *Output* yang dihasilkan pada proses ini berupa bobot akhir yang dijadikan *input* pada proses *testing*. Implementasi program untuk proses *training* adalah sebagai berikut:

```

dalfa = 0.001;
%bobot awal
W = bobotawal(:,1:end-1);
p = bobotawal(:,end);
[baris,kolom]=size(p);
[baris1,kolom1]=size(input);
alfa = alpha;
MinAlfa = 0.000;
MaxEpoch = epoch;
DecAlfa = dalfa;
%mulai iterasi
epoh = 1;
while (epoh <= MaxEpoch) && (alfa >=
MinAlfa)
    for k = 1:baris1
        for c = 1:baris
            D(c) = 0;
            for i = 1:kolom1
                D(c) = D(c)+(W(c,i)-
input(k,i))^2;
            end
            D(c) = sqrt(D(c));
        end
        [jmin,idx]=min(D);
        if idx==target(k)
            W(idx,:) =
W(idx,:)+alfa*(input(k,:)-W(idx,:));
        else
            W(idx,:) = W(idx,:)-
alfa*(input(k,:)-W(idx,:));
        end
        alfa = alfa-DecAlfa*alfa;
        epoh = epoh+1;
    end
end

```

Hasil dari proses *training* berupa bobot berukuran jumlah *cluster* (vektor ciri) \times jumlah individu.

5.5.2 Proses *Testing*

Input yang dibutuhkan untuk proses *testing* adalah bobot akhir dari proses *training*. Hasil dari proses *testing* adalah nama individu pemilik suara. Implementasi program untuk proses *testing* adalah sebagai berikut :

```
H = [];  
[baris, kolom] = size(input);  
[baris1, kolom1] = size(bobot);  
for k = 1:baris  
    for c = 1:baris1  
        D(c)=0;  
        for i = 1:kolom1  
            D(c)=D(c)+(bobot(c,i) -  
input(k,i))^2;  
        end  
        D(c)=sqrt(D(c));  
    end  
    [jmin,idx] = min(D);  
    H = [H;idx]  
end  
end
```

BAB VI

UJI COBA DAN EVALUASI SISTEM

Pada bab ini dijelaskan tahap-tahap uji coba berdasarkan implementasi sistem yang telah dibuat beserta pengujian kinerja sistem. Pengujian dilakukan terhadap dataset sinyal suara yang disimpan di dalam direktori komputer dan akan dihitung tingkat akurasi dari sistem.

6.1 Dataset Uji Coba *Training*

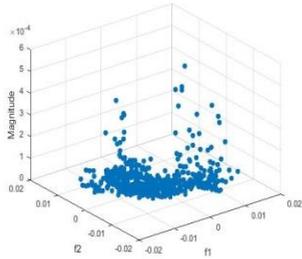
Dataset uji coba *training* merupakan data suara yang diperoleh dari 30 individu seperti yang disajikan pada Lampiran 1. Kata yang digunakan adalah 10 jenis kata. Kata tersebut akan dijadikan referensi sebanyak 1,3,5 kata untuk proses *training* menggunakan metode LVQ dengan jumlah epoch 100 dan nilai *learning rate* sebesar 0.001.

6.2 Uji Coba Estimasi Trispektrum

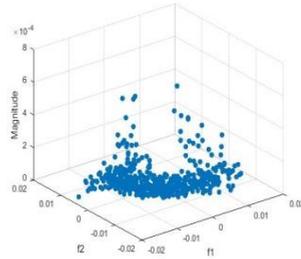
Uji coba dilakukan dengan menjalankan implementasi pengenalan individu melalui suara ucapan. Data sinyal suara akan dilakukan proses prapengolahan data pada Halaman Prapengolahan Data. Pada penelitian ini uji coba proses estimasi trispektrum menggunakan *chunk* 128.

Data yang telah diolah dengan menggunakan estimasi trispektrum akan disimpan dalam format *.xlsx. Visualisasi dari estimasi trispektrum akan memperlihatkan perbedaan ciri sinyal suara dari setiap individu. Karena besaran estimasi trispektrum merupakan ruang 4 dimensi maka salah satu nilai dari nilai f_1 atau f_2 atau $f_3 = 0$ supaya dapat divisualisasikan dengan ruang tiga dimensi.

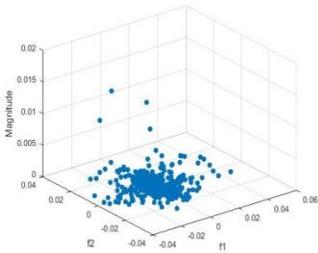
Gambar 6.1 menunjukkan hasil visualisasi dari proses estimasi trispektrum dari 4 individu dengan $f_3 = 0$.



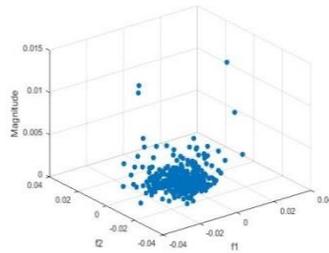
Nama Individu : Afifah
Jenis Kata : "Maju"



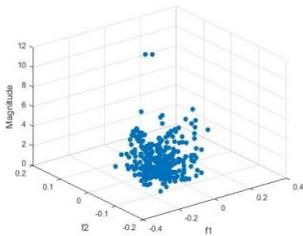
Nama Individu : Afifah
Jenis Kata : "Maju"



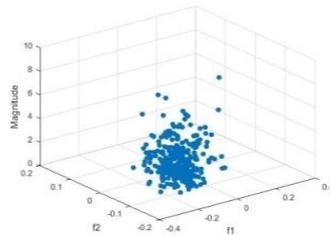
Nama Individu : Airin
Jenis Kata : "Maju"



Nama Individu : Airin
Jenis Kata : "Maju"



Nama Individu : Candra
Jenis Kata : "Maju"



Nama Individu : Candra
Jenis Kata : "Maju"

Gambar 6. 1 Visualisasi Hasil Estimasi Trispektrum dari Kata "Maju"

Pada Gambar 6.1 menunjukkan bahwa kata sama yang diucapkan oleh individu yang sama memiliki ciri yang mirip, sedangkan kata sama yang diucapkan oleh individu yang berbeda memiliki ciri yang tidak mirip.

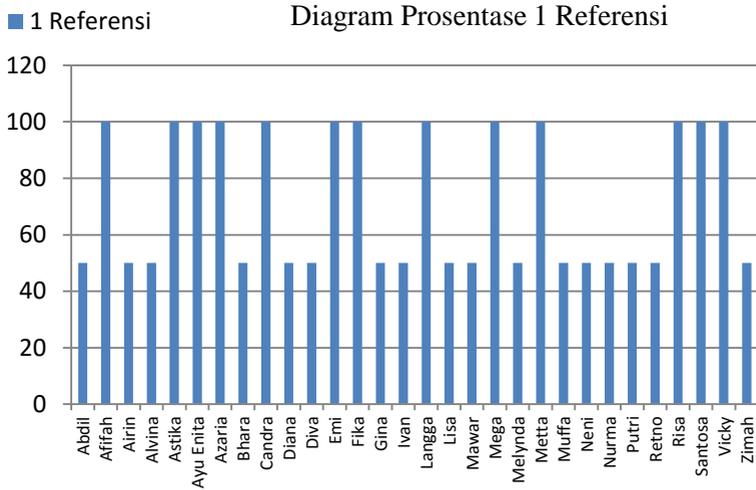
6.3 Uji Coba Proses Pengelompokan Ciri

Data dari proses estimasi trispektrum selanjutnya akan digunakan untuk proses pengelompokan ciri pada Halaman Pengelompokan Ciri. Nilai parameter yang digunakan adalah $k_p = 0.2, k_d = 0.1, s = 8, \alpha = 0.35, Tmax = 10000$. Pada tahap ini data akan dikelompokkan menjadi 8 nilai yang mewakili ciri dari satu sinyal suara. Nilai tersebut akan disimpan dalam format *.xlsx.

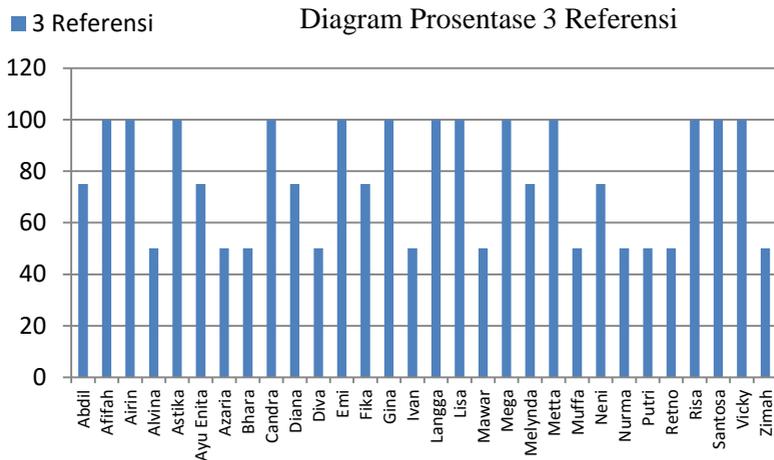
Setelah proses pengelompokan ciri selesai maka data magnitude sinyal suara akan dilabeli dan digunakan dalam inputan pada proses *training*. Hasil dari pembobotan pada proses *training* selanjutnya akan dijadikan inputan pada proses *testing*.

6.4 Uji Coba Pengenalan Individu dengan Kata Sama

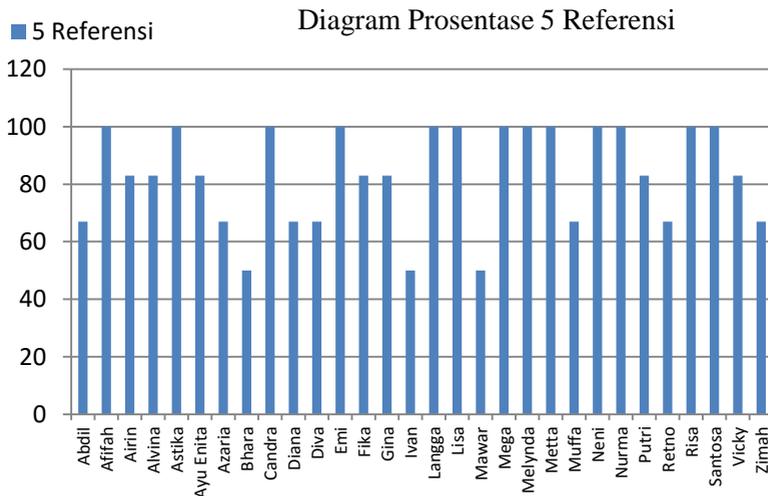
Dalam uji coba pengenalan individu dengan kata yang sama, dilakukan ujicoba terhadap satu jenis kata yang diucapkan oleh 30 individu berbeda yaitu “maju” dengan kata referensi sebanyak 1, 3, 5 kata referensi. Kata yang digunakan sebagai referensi adalah kata “maju”. Dari hasil pengujian diperoleh tingkat akurasi dari sistem pengenalan individu yang disajikan pada diagram prosentase berikut :



Gambar 6. 2 Diagram Prosentase Pengenalan Individu dengan Kata Sama (chunk = 128, vektor ciri = 8, 1 kata referensi)



Gambar 6. 3 Diagram Prosentase Pengenalan Individu dengan Kata Sama (chunk = 128, vektor ciri = 8, 3 kata referensi)

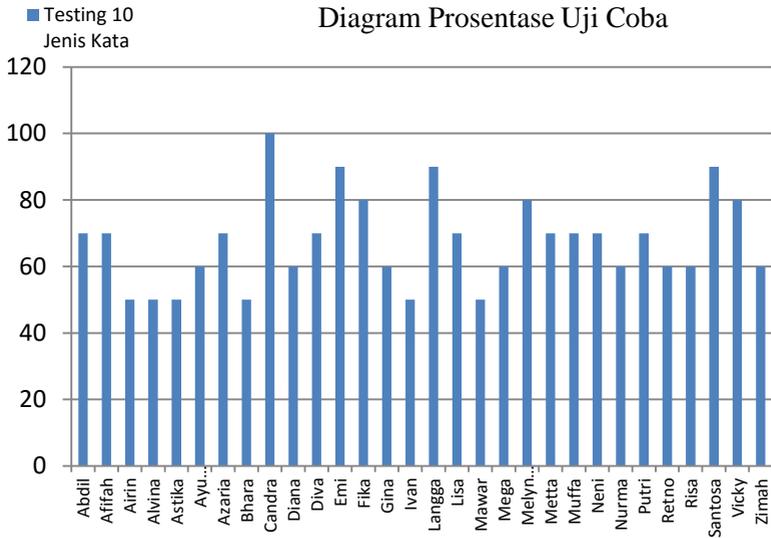


Gambar 6. 4 Diagram Prosentase Pengenalan Individu dengan Kata Sama (chunk = 128, vektor ciri = 8, 5 kata referensi)

Dari diagram diatas, didapatkan tingkat akurasi rata-rata sebesar 71,67%, 76,67%, 83.33% untuk uji coba dengan 1, 3, 5 referensi. Rincian perhitungan dari hasil pengujian sistem dapat dilihat pada Lampiran 4.

6.5 Uji Coba Pengenalan Individu dengan Kata Berbeda

Dalam uji coba pengenalan individu dengan kata yang berbeda, dilakukan ujicoba terhadap 10 jenis kata yang diucapkan oleh 30 individu berbeda dengan kata referensi sebanyak 1 kata referensi. Kata yang digunakan sebagai referensi adalah kata “maju”. Dari hasil pengujian diperoleh tingkat akurasi dari sistem pengenalan individu yang disajikan pada diagram prosentase berikut :

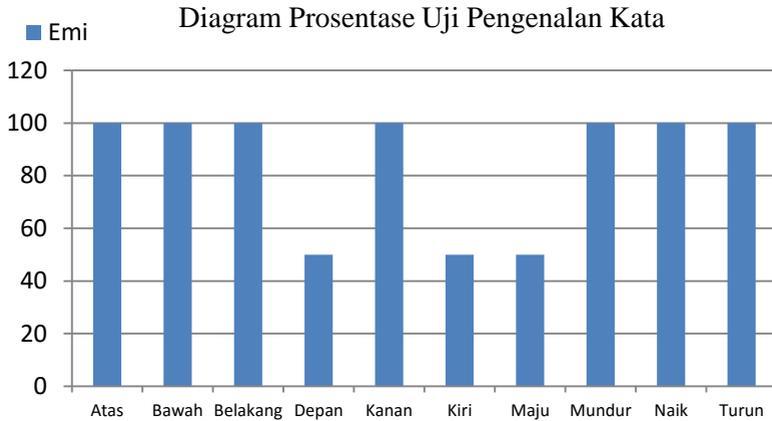


Gambar 6. 5 Diagram Prosentase Pengenalan Individu dengan Kata Berbeda

Dari diagram diatas, didapatkan tingkat akurasi rata-rata sebesar 67,33% untuk uji pengenalan individu dengan kata berbeda. Rincian perhitungan dari hasil pengujian sistem dapat dilihat pada Lampiran 5.

6.6 Uji Coba Pengenalan Kata

Uji coba pengenalan kata dilakukan terhadap individu dengan identitas “Emi”. Pada ujicoba ini kata yang digunakan untuk pengujian sebanyak 10 jenis kata yaitu “maju, mundur, kanan, kiri, atas, bawah, naik, turun, depan dan belakang”. Dari pengujian ini dapat dilihat prosentase kebenaran dari jenis kata yang diucapkan oleh individu. Hasil dari proses pengujian disajikan dalam diagram prosentase berikut :



Gambar 6. 6 Diagram Prosentase Uji Pengenalan Kata

Dari diagram diatas, didapatkan tingkat akurasi rata-rata sebesar 85% untuk uji pengenalan kata. Rincian perhitungan dari hasil pengujian sistem dapat dilihat pada Lampiran 6.

BAB VII

KESIMPULAN DAN SARAN

7.1 Kesimpulan

Dari hasil ujicoba dan evaluasi, dapat disimpulkan beberapa hal sebagai berikut :

1. Algoritma *Ants Clustering* dan jaringan LVQ dapat diimplementasikan pada sistem pengenalan individu berbasis suara ucapan
2. Implementasi sistem pengenalan individu berbasis suara ucapan dengan menggunakan algoritma *Ants Clustering* dan jaringan LVQ berhasil melakukan pengenalan individu dengan rata-rata tingkat akurasi mencapai 83,33% untuk pengujian dengan kata yang sama dengan 5 referensi dan untuk uji pengenalan kata mencapai 85% dengan nilai *chunk*=128, vektor ciri=8, epoch=100 dan *learning rate*=0.001.

7.2 Saran

Saran yang diberikan untuk perbaikan dan kelanjutan dari penelitian ini adalah :

1. Melakukan penelitian dengan metode pengelompokan ciri yang lebih cepat, dan metode klasifikasi yang lainnya.
2. Mencari data suara dengan intonasi yang jelas dan stabil.
3. Mengembangkan sistem pengenalan individu dengan mengucapkan kalimat sederhana.

DAFTAR PUSTAKA

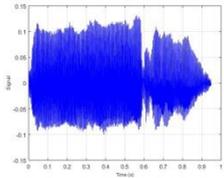
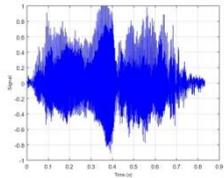
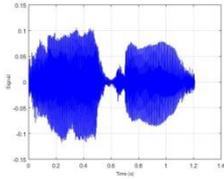
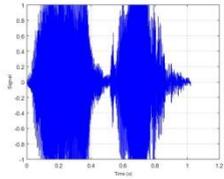
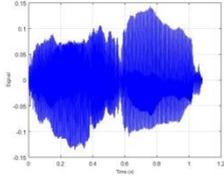
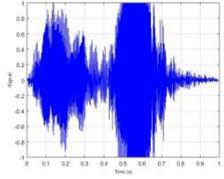
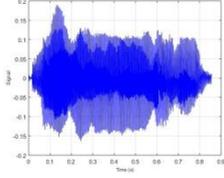
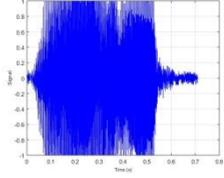
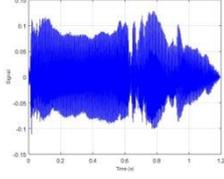
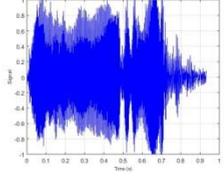
- [1] Amin, M.S. 2006. *Pengelompokan Ciri dan Klasifikasi Sinyal Suara Menggunakan Metode Kuantisasi Vektor MSA-RLS dan Jaringan Syaraf Tiruan LVQ*. Surabaya : Jurusan Matematika, Institut Teknologi Sepuluh Nopember
- [2] Riduwan, M.A. 2007. *Pembuatan Perangkat Lunak Pengestimasi Sinyal Suara Berbasis Visual*. Surabaya : Jurusan Matematika, Institut Teknologi Sepuluh Nopember
- [3] M Bagus, H.S. 2010. *Klasterisasi Sinyal Suara Menggunakan Metode Particle Swarm Optimization pada Pengembangan Sistem Pengenalan Individu Berbasis Suara Ucapan*. Surabaya: Jurusan Matematika, Institut Teknologi Sepuluh Nopember
- [4] Fausett, L. 1994. *Fundamentals of Neural Networks : Architectures, Algorithms, and Applications*. Prentice Hall International, Inc
- [5] Rajat, H. 2016. *Learning Vector Quantization (LVQ) Neural Network Approach for Multilingual Speech Recognition*. International Research Journal of Engineering Volume:03 Issue:05.
- [6] Santosa, Budi, 2007. *Segmentasi Pasar dengan Ant-Kmeans Clustering*. Jurusan Teknik Industri, Institut Teknologi Sepuluh Nopember
- [7] Evi, A. 2009. *Analisa dan Identifikasi Berbagai Penyakit Paru-paru dengan Metode Hidden Markov Model*. Depok : Jurusan Tehnik Elektro, Universitas Indonesia
- [8] Urszula, B. *Finding groups in data : Cluster analysis with ants*. Applied Soft Computing. Elsevier
- [9] Brigham, E. Oran. 1974. *The Fast Fourier Transform*. Prentice-Hall, Inc

- [10] Anusuya, A & Katti, K. 2009. *Speech Recognition by Machine*. Volume 6, No. 3. Department of Computer Science and Engineering. India
- [11] Siang, JJ., 2005, *Jaringan Syaraf Tiruan dan Pemrogramannya Menggunakan Matlab*. Yogyakarta : Andi Offset.
- [12] Kanade , Parag M. and Hall , Lawrence O., 2003. *Fuzzy Ants as a Clustering Concept*, NAFIPS Conference

LAMPIRAN

Lampiran 1 Sampel Visualisasi Input Data Suara

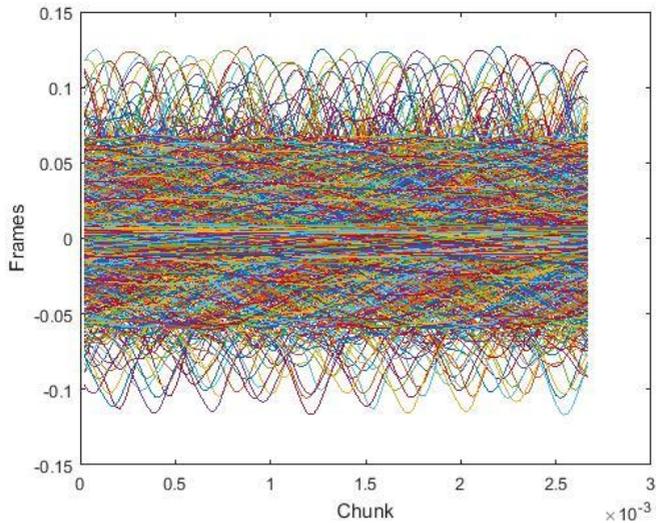
No	Jenis Kata	Nama Pemilik Suara	
		Afifah	Candra
1	Atas		
2	Bawah		
3	Depan		
4	Belakang		
5	Kanan		

6	Kiri		
7	Maju		
8	Mundur		
9	Naik		
10	Turun		

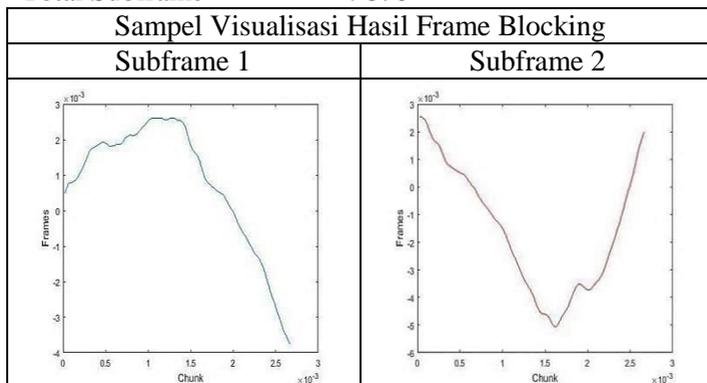
Lampiran 2 Sampel Visualisasi Prapengolahan Data

Nama Pemilik Suara : Afifah
 Nama File Suara : maju1.wav
 Panjang Sinyal : 55540

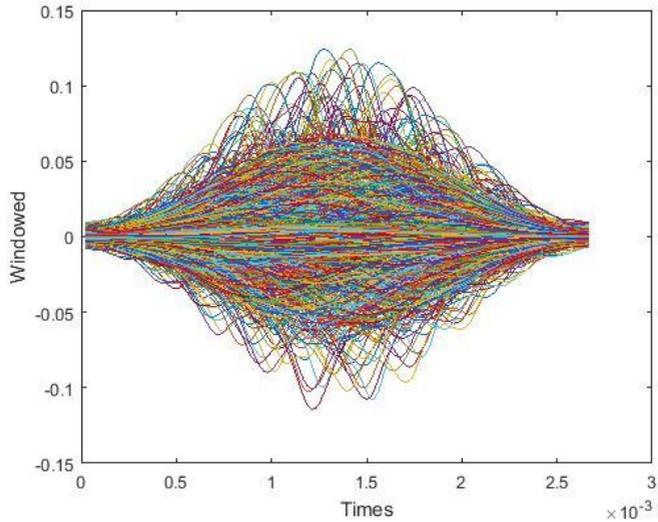
A. Visualisasi Hasil Frame Blocking



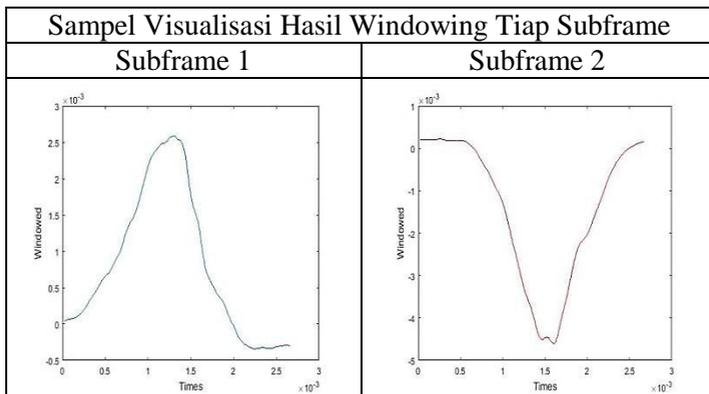
Keterangan Gambar : Hasil Frame Blocking
 Chunk : 128
 Total Subframe : 876



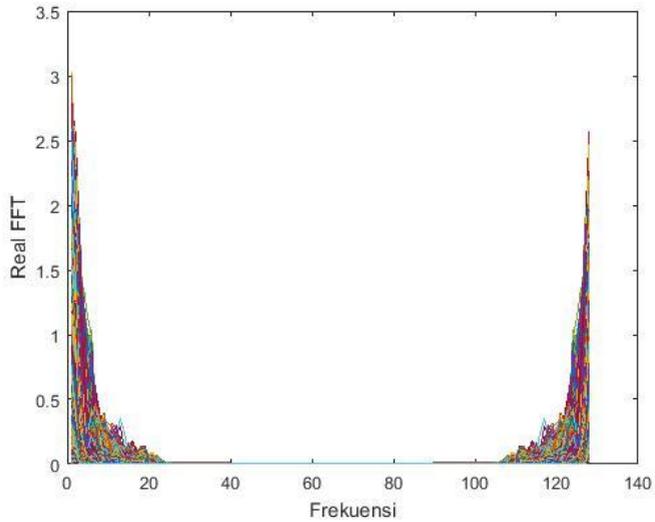
B. Visualisasi Hasil Windowing



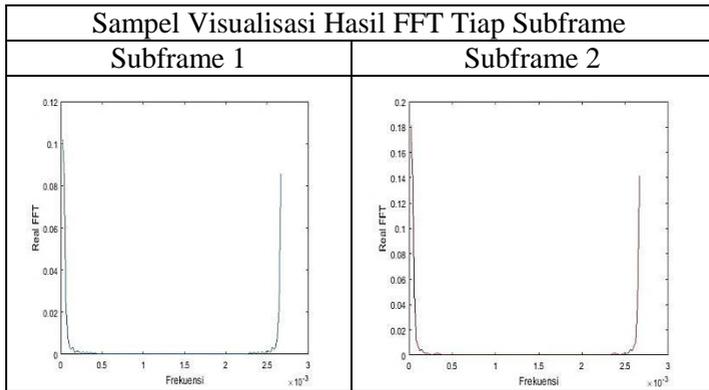
Keterangan Gambar : Hasil Windowing



C. Visualisasi Hasil FFT

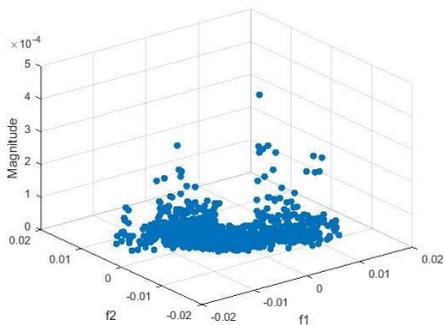


Keterangan Gambar : Hasil Fast Fourier Transform

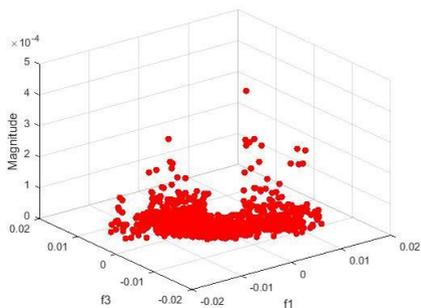


D. Visualisasi Hasil Estimasi Trispektrum

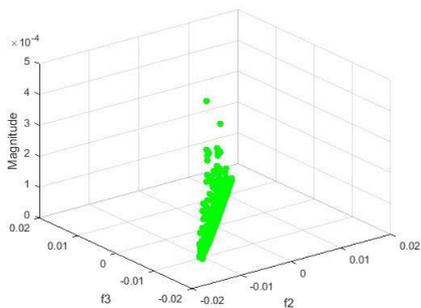
- $f_3 = 0$



- $f_2 = 0$



- $f_1 = 0$



Lampiran 3 Tabel Identitas Individu Pemilik Suara

P = Perempuan, L = Laki-laki

No	Nama Individu	Jenis Kelamin	Usia (tahun)
1	Abdil	L	18
2	Afifah	P	22
3	Airin	P	22
4	Alvina	P	22
5	Astika	P	21
6	Ayu Enita	P	22
7	Azaria	P	22
8	Bhara	L	22
9	Candra	L	16
10	Diana	P	22
11	Diva	P	21
12	Emi	P	45
13	Fika	P	22
14	Gina	P	21
15	Ivan	L	22
16	Langga	L	14
17	Lisa	P	22
18	Mawar	P	20
19	Mega	P	22
20	Melynda	P	22
21	Metta	P	22
22	Muffa	P	22
23	Neni	P	23
24	Nurma	P	22
25	Putri	P	21
26	Retno	P	21
27	Risa	P	22
28	Santosa	L	50
29	Vicky	P	22
30	Zimah	P	15

Lampiran 4 Tabel Hasil Pengujian Sistem dengan Kata Sama

Referensi Kata “Maju”

- 1 Kata Referensi

No	Nama	Data Training		Data Testing		Total
		Benar	Salah	Benar	Salah	
1	Abdil	1	0	0	1	2
2	Afifah	1	0	1	0	2
3	Airin	1	0	0	1	2
4	Alvina	1	0	0	1	2
5	Astika	1	0	1	0	2
6	Ayu Enita	1	0	1	0	2
7	Azaria	1	0	1	0	2
8	Bhara	1	0	0	1	2
9	Candra	1	0	1	0	2
10	Diana	1	0	0	1	2
11	Diva	1	0	0	1	2
12	Emi	1	0	1	0	2
13	Fika	1	0	1	0	2
14	Gina	1	0	0	1	2
15	Ivan	1	0	0	1	2
16	Langga	1	0	1	0	2
17	Lisa	1	0	0	1	2
18	Mawar	1	0	0	1	2
19	Mega	1	0	1	0	2
20	Melynda	1	0	0	1	2
21	Metta	1	0	1	0	2
22	Muffa	1	0	0	1	2
23	Neni	1	0	0	1	2
24	Nurma	1	0	0	1	2
25	Putri	1	0	0	1	2
26	Retno	1	0	0	1	2
27	Risa	1	0	1	0	2

28	Santosa	1	0	1	0	2
29	Vicky	1	0	1	0	2
30	Zimah	1	0	0	1	2
Total		30	0	13	17	60
Akurasi		100%		43.33%		
Akurasi Total		71,67%				

• **3 Kata Referensi**

No	Nama	Data Training		Data Testing		Total
		Benar	Salah	Benar	Salah	
1	Abdil	2	1	1	0	4
2	Afifah	3	0	1	0	4
3	Airin	3	0	1	0	4
4	Alvina	2	1	0	1	4
5	Astika	3	0	1	0	4
6	Ayu Enita	2	1	1	0	4
7	Azaria	2	1	0	1	4
8	Bhara	2	1	0	1	4
9	Candra	3	0	1	0	4
10	Diana	2	1	1	0	4
11	Diva	2	1	0	1	4
12	Emi	3	0	1	0	4
13	Fika	2	1	1	0	4
14	Gina	3	0	1	0	4
15	Ivan	2	1	0	1	4
16	Langga	3	0	1	0	4
17	Lisa	3	0	1	0	4
18	Mawar	2	1	0	1	4
19	Mega	3	0	1	0	4
20	Melynda	3	0	0	1	4
21	Metta	3	0	1	0	4
22	Muffa	2	1	0	1	4
23	Neni	3	0	0	1	4
24	Nurma	2	1	0	1	4

25	Putri	2	1	0	1	4
26	Retno	2	1	0	1	4
27	Risa	3	0	1	0	4
28	Santosa	3	0	1	0	4
29	Vicky	3	0	1	0	4
30	Zimah	2	1	0	1	4
Total		75	15	17	13	120
Akurasi		83.33%		56.67%		
Akurasi Total		76.67%				

• **5 Kata Referensi**

No	Nama	Data Training		Data Testing		Total
		Benar	Salah	Benar	Salah	
1	Abdil	4	1	0	1	6
2	Afifah	5	0	1	0	6
3	Airin	4	1	1	0	6
4	Alvina	4	1	1	0	6
5	Astika	5	0	1	0	6
6	Ayu Enita	4	1	1	0	6
7	Azaria	4	1	0	1	6
8	Bhara	3	2	0	1	6
9	Candra	5	0	1	0	6
10	Diana	4	1	0	1	6
11	Divia	4	1	0	1	6
12	Emi	5	0	1	0	6
13	Fika	4	1	1	0	6
14	Gina	4	1	1	0	6
15	Ivan	3	2	0	1	6
16	Langga	5	0	1	0	6
17	Lisa	5	0	1	0	6
18	Mawar	3	2	0	1	6
19	Mega	5	0	1	0	6
20	Melynda	5	0	1	0	6
21	Metta	5	0	1	0	6

22	Muffa	4	1	0	1	6
23	Neni	5	0	1	0	6
24	Nurma	5	0	1	0	6
25	Putri	4	1	1	0	6
26	Retno	3	2	1	0	6
27	Risa	5	0	1	0	6
28	Santosa	5	0	1	0	6
29	Vicky	4	1	1	0	6
30	Zimah	4	1	0	1	6
Total		129	21	21	9	180
Akurasi		86%		70%		
Akurasi Total		83.33%				

Lampiran 5 Tabel Hasil Pengujian Sistem dengan Kata Berbeda

No	Nama	Data		Total
		Benar	Salah	
1	Abdil	7	3	10
2	Afifah	7	3	10
3	Airin	5	5	10
4	Alvina	5	5	10
5	Astika	5	5	10
6	Ayu Enita	6	4	10
7	Azaria	7	3	10
8	Bhara	5	5	10
9	Candra	10	0	10
10	Diana	6	4	10
11	Diva	7	3	10
12	Emi	9	1	10
13	Fika	8	2	10
14	Gina	6	4	10
15	Ivan	5	5	10
16	Langga	9	1	10
17	Lisa	7	3	10
18	Mawar	5	5	10
19	Mega	6	4	10
20	Melynda	8	2	10
21	Metta	7	3	10
22	Muffa	7	3	10
23	Neni	7	3	10
24	Nurma	6	4	10
25	Putri	7	3	10
26	Retno	6	4	10
27	Risa	6	4	10
28	Santosa	9	1	10
29	Vicky	8	2	10

30	Zimah	6	4	10
Total		202	98	300
Akurasi		67.33 %		

Lampiran 6 Tabel Hasil Uji Pengenalan Kata

• Nama Individu : Emi

No	Jenis Kata	Data Training		Data Testing		Total
		Benar	Salah	Benar	Salah	
1	Atas	1	0	1	0	2
2	Bawah	1	0	1	0	2
3	Belakang	1	0	1	0	2
4	Depan	1	0	0	1	2
5	Kanan	1	0	1	0	2
6	Kiri	1	0	0	1	2
7	Maju	1	0	0	1	2
8	Mundur	1	0	1	0	2
9	Naik	1	0	1	0	2
10	Turun	1	0	1	0	2
Total		10	0	7	3	20
Akurasi		100%		70%		
Akurasi Total		85%				

BIODATA PENULIS



Penulis memiliki nama lengkap Metta Andriani dengan nama panggilan Metta. Lahir di Tulungagung, 02 Maret 1995 dan tinggal di Tulungagung sejak 1995. Jenjang pendidikan formal yang ditempuh yaitu SDN 1 Ngujang (2001-2007), SMPN 1 Tulungagung (2007-2010), SMAN 1 Kedungwaru (2010-2013). Pada tahun 2013 penulis diterima di Departemen Matematika ITS melalui jalur tulis untuk menempuh pendidikan S1 selama empat tahun. Di Departemen Matematika ITS penulis mengambil bidang minat ilmu komputer (Computer Science). Penulis juga aktif berorganisasi di KM ITS, yaitu sebagai staff Departemen Kesejahteraan Mahasiswa di Himpunan Mahasiswa Matematika (HIMATIKA ITS) (2014-2015), kepala biro pelayanan Departemen Kesejahteraan Mahasiswa di Himpunan Mahasiswa Matematika (HIMATIKA ITS) (2015-2016), dan penulis juga aktif dalam kepanitiaan acara tingkat Nasional yaitu Olimpiade Matematika ITS sebagai sekretariat di dalam kampus. Penulis juga melaksanakan Kerja Praktek di PT. Infoglobal Teknologi Semesta Surabaya. Jika ingin memberikan saran, kritik, dan pertanyaan mengenai Tugas Akhir ini, bisa menghubungi melalui email mettaandriani23@gmail.com.