UNIVERSITAS INDONESIA

SISTEM IDENTIFIKASI PEMBICARA BERBAHASA INDONESIA MENGGUNAKAN SUPPORT VECTOR MACHINE (SVM)

TESIS

Diajukan sebagai salah satu syarat untuk
memperoleh gelar Magister Ilmu Komputer

WORO SUDARYANTI
0706193542

FAKULTAS ILMU KOMPUTER
PROGRAM STUDI MAGISTER ILMU KOMPUTER
DEPOK
JULI 2009

UNIVERSITAS INDONESIA

SISTEM IDENTIFIKASI PEMBICARA BERBAHASA INDONESIA MENGGUNAKAN SUPPORT VECTOR MACHINE (SVM)

TESIS

WORO SUDARYANTI
0706193542

FAKULTAS ILMU KOMPUTER
PROGRAM STUDI MAGISTER ILMU KOMPUTER
DEPOK
JULI 2009

HALAMAN PERNYATAAN ORISINALITAS

Tesis ini adalah hasil harya saya sendiri,

dan semua sumber baik yang dikutip maupun dirujuk

telah saya nyatakan dengan benar

Nama : Woro Sudaryanti
NPM : 0706193542
Tanda Tangan:

Tanggal : 21 Juli 2009

HALAMAN PENGESAHAN

Tesis ini diajukan oleh :
Nama : Woro Sudaryanti
NPM : 0706193542
Program Studi : Magister Ilmu Komputer
Judul Tesis : Sistem Identifikasi Pembicara Berbahasa Indonesia Menggunakan Support Vector Machine (SVM)

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Magister Ilmu Komputer pada Program Studi Magister Ilmu Komputer Fakultas Ilmu Komputer Universitas Indonesia.

DEWAN PENGUJI

Pembimbing : Mirna Adriani, Ph.D ( )

Penguji : Hisar Maruli Manurung, Ph.D. ( )

Penguji : Dr. Achmad Nizar Hidayanto ( )

Penguji : Dr. Indra Budi ( )

Ditetapkan di : Depok

Tanggal : 21 Juli 2009

KATA PENGANTAR/UCAPAN TERIMA KASIH

Puji syukur saya panjatkan kepada Allah, SWT, karena atas berkat dan rahmat-Nya, saya dapat menyelesaikan tesis ini. Penulisan tesis ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Magister Ilmu Komputer pada Fakultas Ilmu Komputer Universitas Indonesia. Saya menyadari bahwa tanpa bantuan dan bimbingan dari berbagai pihak, dari masa perkuliahan sampai pada penyusunan tesis ini, sangatlah sulit bagi saya untuk menyelesaikan tesis ini. Oleh karena itu, saya mengucapkan terima kasih kepada:

1. Dra. Mirna Adriani, Ph.D selaku pembimbing yang telah menyediakan waktu, tenaga, dan pikiran untuk mengarahkan saya dalam penyusunan tesis ini.
2. pihak PT.Lima Titik Satu, yang telah memberikan kelonggaran waktu dan pekerjaan pada saya selama masa perkuliahan.
3. orang tua dan keluarga yang telah memberikan bantuan dukungan material dan moral
4. teman-teman Magister Ilmu Komputer Fasilkom UI angkatan 2007, semua sahabat, teman, dan pihak lain yang telah membantu saya.

Akhir kata, saya berharap Allah, SWT berkenan membalas segala kebaikan semua pihak yang telah membantu. Semoga tesis ini membawa manfaat bagi pengembangan ilmu.

Depok, 21 Juli 2009

Penulis

HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademik Universitas Indonesia, saya yang bertanda tangan di bawah ini:

Nama : Woro Sudaryanti
NPM : 0706193542
Program Studi : Magister Ilmu Komputer
Fakultas : Ilmu Komputer
Jenis Karya : Tesis

demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Indonesia **Hak Bebas Royalti Noneksklusif** (***Non-exclusive Royalti-Free Right***) atas karya ilmiah saya yang berjudul:

Sistem Identifikasi Pembicara Berbahasa Indonesia Menggunakan Support Vector Machine (SVM)

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan memublikasikan tugas akhir saya tanpa meminta izin dari saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok
Pada tanggal : 21 Juli 2009
Yang menyatakan

(Woro Sudaryanti)

ABSTRAK

Nama : Woro Sudaryanti
Program Studi : Magister Ilmu Komputer
Judul : Sistem Identifikasi Pembicara Berbahasa Indonesia Menggunakan Support Vector Machine (SVM)

Penelitian ini melakukan studi mengenai sistem identifikasi pembicara berbahasa Indonesia menggunakan SVM. Parameter sistem terdiri atas *silence removal*, PCA, nilai rata-rata dan varians MFCC. Ujicoba menggunakan data berita berbahasa Indonesia dari televisi dan radio yang disegmen dalam 5, 10, 15 detik dengan jumlah data 26 jam (715 pembicara). Hasil penelitian ini menunjukkan ketepatan pengenalan pembicara sebesar 94-98% untuk kombinasi parameter *silence removal* dan rata-rata MFCC dengan akurasi terbaik pada segmen waktu 10 detik. Namun dengan bertambahnya jumlah pembicara, ketepatan pengenalan cenderung berkurang. Penelitian ini dapat dikembangkan untuk sistem perolehan informasi data *speech* berdasarkan siapa yang berbicara dalam suatu sesi data.

Kata kunci: identifikasi pembicara, svm, mfcc

ABSTRACT

Name : Woro Sudaryanti
Program Study: Magister of Computer Science
Title : Speaker Identification System for Indonesian Speech Based on Support Vector Machine (SVM)

This research studies speaker identification system for Indonesian speech based on SVM. Parameters of this system are silence removal, PCA, average and varians values of MFCC. The experiments use 26 hours (715 speakers) Indonesian broadcast news from radio and television segmented into 5, 10, 15 seconds. The results achieve 94-98% identification accuracy for combination of parameters silence removal and average of MFCC. The best accuracy comes from 10 seconds time segment. However, the accuracy falls when the number of speakers increases. This study could be used for speech retrieval system based on who speaks in a speech session.

Keywords: speaker identification, svm, mfcc

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB 1
PENDAHULUAN

1.1 Latar Belakang

Dengan semakin berkembangnya teknologi telekomunikasi, internet menjadi sesuatu yang tidak lagi sulit dan mahal. Kemudahan ini menyebabkan internet dipenuhi berbagai macam informasi yang tidak hanya berupa teks, tetapi juga gambar, audio, dan video. Kebutuhan akan identifikasi pembicara juga semakin bertambah, sehingga perolehan informasi mengenai seseorang akan dapat menghasilkan data audio atau video dari orang tersebut. Misalnya, seorang wartawan ingin mencari seseorang dari data suara yang ada, dengan identifikasi pembicara hal itu dapat dilakukan. Data audio dan video yang diperoleh dapat memperkaya informasi mengenai orang tersebut sehingga tulisan yang akan dibuat oleh si wartawan menjadi semakin menarik.

Identifikasi pembicara (*speaker identification*) merupakan suatu proses untuk mengidentifikasi siapa yang berbicara (pembicara) pada suatu sesi data *speech*. Pada umumnya, identifikasi pembicara terbagi atas *text independent* dan *text dependent*. Untuk kategori *text independent* proses identifikasi tidak dipengaruhi oleh isi pembicaraan si pembicara. Sebaliknya, pada *text dependent*, seluruh pembicara harus menggunakan kata yang sama atau beberapa kata yang sudah ditentukan. Selain itu, berdasarkan data yang diproses, identifikasi pembicara terbagi atas dua, yaitu berbasis linguistik dan berbasis sinyal akustik. Jika berbasis linguistik, data akan berisi unsur kebahasaan, seperti kata, kalimat, dan kamus data. Sedangkan yang berbasis sinyal akustik, data akan merepresentasikan bagaimana karakteristik *speech* tersebut terdengar di telinga manusia. Sistem yang berbasis sinyal akustik tidak mempunyai batasan kebahasaan.

Aplikasi identifikasi pembicara banyak digunakan untuk keamanan, yaitu biometrik, dan perolehan informasi multimedia. Pada aplikasi biometrik, data pembicara yang digunakan biasanya tidak sebanyak pada aplikasi sistem perolehan informasi dan lebih mengarah ke verifikasi pembicara dari pada

identifikasi pembicara. Sedangkan sistem perolehan informasi berkembang karena adanya kebutuhan pencarian data tertentu pada koleksi data yang semakin banyak.

Penelitian di bidang identifikasi pembicara sudah berlangsung lama. Hampir semua metode sudah digunakan. Metode yang dijadikan acuan adalah Gaussian Mixture Models (GMM) [Reynolds dan Rose, 1995] yang menghasilkan akurasi yang sangat baik, yaitu 80.8% untuk data *speech* dengan kualitas pembicaraan lewat telpon (*telephone speech*) dan 96,8% untuk data *speech* yang jernih (*clean speech*). Metode GMM digunakan berdasarkan interpretasi bahwa komponen Gaussian dapat merepresentasikan bentuk spektral pembicara secara umum dan kemampuan Gaussian mixtures dalam memodelkan densitas data. Metode lain yang digunakan adalah Hidden Markov Model (HMM) [Buono et al, 2008]. Metode ini memodelkan tidak hanya suara, tapi juga urutan suara, dan lebih cocok untuk *text-dependent speaker identification*. Selain itu, metode Vector Quantization (VQ) [Kamruzzaman et al, 2007] juga sudah digunakan. Dengan metode ini, setiap pembicara mempunyai *codebook* berisi model spektral dari suaranya. Yang terakhir adalah Support Vector Machine (SVM) [Schmidt dan Gish, 1996]. Metode SVM digunakan karena sederhana dan lebih efektif, di mana fitur suara dari pembicara dapat digunakan secara langsung tanpa perlu estimasi densitas terlebih dahulu.

Berdasarkan data yang diproses, penelitian identifikasi pembicara terbagi atas yang berbasis linguistik dan sinyal akustik. Untuk sinyal akustik sendiri, fitur data yang digunakan antara lain berupa Linear Predictive Coding (LPC) [Wan dan Campbell, 2000], Mel-scale Frequency Cepstrum Coefficient (MFCC) [Moreno dan Ho, 2008], Mel-scale Bispectrum [Buono et al, 2008], dan wavelet [Lin et al, 2006]. MFCC banyak digunakan karena dianggap yang paling sesuai dalam memodelkan frekuensi suara manusia. Sedangkan bispektrum digunakan untuk mengatasi *noise* pada suara, namun hal ini bisa juga diatasi dengan menggunakan principal component analysis (PCA) pada saat ekstraksi fitur suara [Li et al, 2008].

Tesis ini mengusulkan sistem identifikasi pembicara dengan menggunakan metode klasifikasi SVM dan fitur audio MFCC. Data *speech* yang digunakan

adalah data berbahasa Indonesia yang berasal dari beberapa stasiun berita televisi dan radio.

1.2 Perumusan Masalah

Berdasarkan latar belakang di atas, dapat dirumuskan beberapa permasalahan sebagai berikut:

1. Mengetahui hasil identifikasi pembicara dengan metode SVM.
2. Mengetahui panjang data dan jumlah data yang cukup dari setiap pembicara untuk dapat memperoleh hasil identifikasi yang cukup akurat.
3. Mengetahui metode SVM *multi-class* yang tepat untuk klasifikasi kelas yang sangat banyak sehingga dapat diperoleh hasil identifikasi yang baik.
4. Mengetahui metode pemrosesan data suara untuk dapat menghilangkan gangguan pada suara (*noise*).

1.3 Tujuan Penelitian

Tesis ini bertujuan untuk melakukan identifikasi pembicara pada data berita berbahasa Indonesia dengan menggunakan salah satu metode *machine learning*, yaitu SVM. Fitur suara yang digunakan adalah MFCC.

1.4 Ruang Lingkup Penelitian

Ruang lingkup penelitian dalam tesis ini adalah sebagai berikut:

1. Koleksi data yang digunakan dalam tesis ini berasal dari stasiun berita televisi dan radio, di mana suara yang diperoleh masih mengandung *noise*.
2. Karena keterbatasan sumber daya komputasi, klasifikasi tidak akan menggunakan seluruh frame suara yang ada melainkan hanya rata-rata nilai frame dalam suatu waktu, yaitu 5, 10, dan 15 detik.
3. Jumlah data untuk setiap pembicara tidak sama (*unbalanced* data).

1.5 Metodologi Penelitian

Penelitian ini dilakukan dengan langkah-langkah sebagai berikut:

1. Perumusan Masalah dan Studi Literatur

   Pada tahap ini akan dilakukan analisa terhadap masalah, pengumpulan bahan-bahan dan referensi, untuk dijadikan bahan acuan dalam melakukan studi awal pemahaman konsep dan perumusan model sistem yang akan dibuat. Literatur-literatur ini diperoleh melalui penelusuran jurnal, makalah, buku, dan informasi lain yang terkait dengan tesis ini.

2. Pengumpulan Data

   Data yang digunakan berasal dari data berita berbahasa Indonesia dari beberapa stasiun radio dan televisi yang ada.

3. Perancangan Sistem

   Pada tahap ini, dilakukan perancangan sistem identifikasi pembicara yang menggunakan metode SVM dan fitur suara MFCC.

4. Implementasi Sistem

   Tahap ini merupakan implementasi dari tahap perancangan sistem. Sistem yang dibuat terbagi atas pra-pemrosesan, ekstraksi fitur *speech*, pemodelan pembicara, dan identifikasi pembicara.

5. Uji Coba Sistem

   Pada tahap ini dilakukan identifikasi dari sejumlah data *speech*. Hasil evaluasi akan dianalisa kebenarannya.

1.6 Sistematika Penulisan

Tesis ini disusun dengan sistematika sebagai berikut:

Bab 1 PENDAHULUAN: berisi penjelasan mengenai latar belakang, rumusan masalah, tujuan penelitian, ruang lingkup penelitian, metodologi, serta sistematika penyusunan laporan penelitian.

Bab 2 LANDASAN TEORI: membahas algoritma, metode, dan teori-teori yang digunakan dalam perancangan sistem identifikasi pembicara.

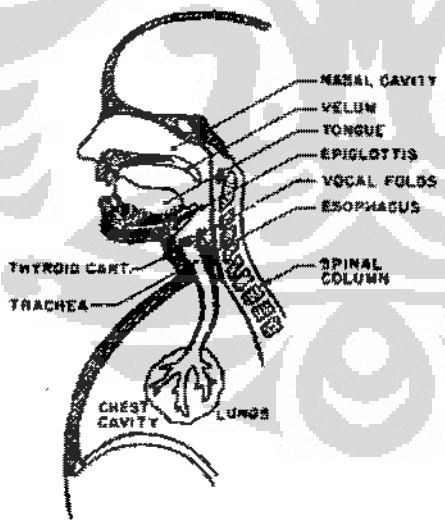
Bab 3 EKSPERIMEN: berisi penjelasan mengenai eksperimen yang dilakukan, yaitu: data yang digunakan, perancangan sistem, implementasi sistem, dan pengujian sistem.

Bab 4 HASIL UJICOBA DAN ANALISA: berisi hasil uji coba sistem yang telah dikembangkan dan analisa unjuk kerja sistem identifikasi pembicara bedasarkan hasil pengujian yang diperoleh.

Bab 5 KESIMPULAN DAN SARAN: berisi kesimpulan dari hasil penelitian ini dan saran untuk pengembangan selanjutnya.

BAB 2
LANDASAN TEORI

Sinyal *speech* adalah sinyal yang mengandung unsur kebahasaan yang dihasilkan oleh saluran suara manusia. Pada gambar 2.1 terlihat bagian dari tubuh manusia yang menghasilkan sinyal *speech*. Saluran suara (*vocal tract*) dimulai dari pita suara sampai bibir [Rabiner dan Juang, 1993], terdiri atas *pharynx*, yang menghubungkan *esophagus* dengan mulut, dan mulut (*oral cavity*). Saluran hidung (*nasal tract*) dimulai dari *velum* sampai lubang hidung (*nostril*). Ketika udara dikeluarkan dari paru-paru melalui *trachea*, aliran udara akan menyebabkan pita suara bergetar. Bila pita suara bergetar dalam keadaan menegang, maka akan dihasilkan suara *speech* yang berbunyi (*voiced speech sounds*). Dan bila pita suara bergetar dalam keadaan relaks, maka yang dihasilkan adalah suara yang tidak berbunyi (*unvoiced sounds*). Bunyi yang dihasilkan tergantung pada posisi, bentuk, dan ukuran rahang, lidah, *velum*, bibir, dan mulut.



Gambar 2.1 Skema mekanisme suara manusia

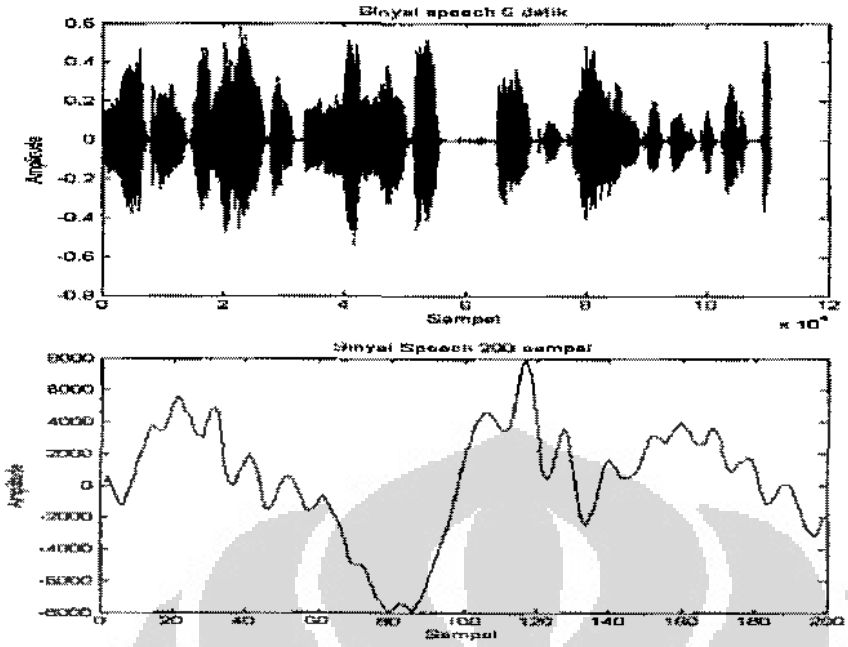
[Flanagan,1972]

Perbedaan saluran suara ini yang menyebabkan karakteristik suara manusia berbeda satu sama lainnya, sehingga dapat digunakan untuk melakukan identifikasi terhadap seseorang.

Sinyal merupakan sesuatu yang membawa informasi. Secara garis besar, sinyal dibedakan atas sinyal analog dan sinyal dijital. Sinyal analog atau sinyal dengan waktu kontinu mempunyai nilai yang tak terbatas (tak terhingga) sedangkan sinyal dijital atau sinyal dengan waktu diskrit mempunyai nilai integer yang diperoleh hanya pada saat tertentu dari sinyal analog. Sinyal dijital pada umumnya merupakan konversi dari sinyal analog. Konversi dari analog ke dijital ini disebut proses *sampling* di mana sinyal analog akan dibagi menjadi bagian yang lebih kecil yang disebut sampel sesuai kecepatan *sampling*nya. Semakin tinggi kecepatan sampling, maka semakin banyak perubahan dalam sinyal analog yang dapat ditangkap.

Gambar 2.2 Contoh sinyal analog dan dijital

Sinyal sering dibedakan berdasarkan media yang membawanya, seperti sinyal elektronik yang dibawa oleh rangkaian elektronik, sinyal akustik yang dibawa oleh bunyi, sinyal *speech* yang dibawa oleh suara manusia, dan sebagainya. *Speech* menghasilkan bunyi, maka speech termasuk ke dalam sinyal akustik. Contoh bentuk gelombang sinyal speech dapat dilihat pada gambar 2.3.



Gambar 2.3 Sinyal speech berukuran 5 detik (atas) dan 200 sampel (bawah)

Penelitian di bidang identifikasi pembicara sudah berlangsung lama. Hampir semua metode sudah digunakan. Metode yang dijadikan acuan adalah Gaussian Mixture Models (GMM) [Reynolds dan Rose, 1995] yang menghasilkan akurasi yang sangat baik, yaitu 80,8% untuk data *speech* dengan kualitas pembicaraan lewat telpon (*telephone speech*) dan 96,8% untuk data *speech* yang jernih (*clean speech*). Metode GMM digunakan berdasarkan interpretasi bahwa komponen Gaussian dapat merepresentasikan bentuk spektral pembicara secara umum dan kemampuan Gaussian mixtures dalam memodelkan densitas data. Metode lain yang digunakan adalah Hidden Markov Model (HMM) [Buono et al, 2008]. Metode ini memodelkan tidak hanya suara, tapi juga urutan suara. Selain itu, metode Vector Quantization (VQ) [Kamruzzaman et al, 2007] juga sudah digunakan, di mana setiap pembicara mempunyai *codebook* berisi model spektral dari suaranya. Yang terakhir adalah Support Vector Machine (SVM) [Schmidt dan Gish, 1996]. Metode SVM digunakan karena sederhana dan lebih efektif, di mana fitur suara dari pembicara dapat digunakan secara langsung tanpa perlu estimasi densitas terlebih dahulu.

Sistem identifikasi pembicara dalam tesis ini terbagi atas pra-pemrosesan, ekstraksi fitur sinyal *speech*, pemodelan pembicara, dan identifikasi pembicara.

Pra-pemrosesan merupakan proses untuk meningkatkan kualitas data sebelum dilakukan ekstraksi fitur. Mula-mula data suara diubah dari bentuk analog ke bentuk dijital melalui proses *sampling*. Kemudian bagian yang *silence* atau *unvoice* dari data akan dibuang dengan menggunakan *threshold* dari rata-rata energi [Chakroborty et al, 2007]. Dan untuk menaikkan rasio antara sinyal dengan *noise* (SNR), dilakukan proses *pre-emphasis* yang pada dasarnya merupakan sebuah filter *high pass* yang akan melewatkan data frekuensi tinggi dan membuang data frekuensi rendah.

Ekstraksi fitur *speech* merupakan proses untuk memperoleh karakteristik *speech* dari pembicara. Hal ini dilakukan dengan membagi data *speech* menjadi sub data yang lebih kecil (*frame*) untuk kemudian dianalisa sehingga diperoleh fitur *speech* yang diinginkan. Fitur *speech* yang digunakan pada tesis ini adalah MFCC. MFCC banyak digunakan karena dianggap yang paling sesuai dalam memodelkan frekuensi suara manusia.

Pemodelan pembicara bertujuan untuk memperoleh data model referensi dari setiap pembicara dan parameter sistem. Hal ini dilakukan dengan melatih sistem dengan menggunakan metode *machine learning*, yaitu *multi-class* Support Vector Machine (SVM) *one-vs-one* dengan kernel RBF. SVM digunakan karena sederhana dan lebih efektif, di mana fitur suara dari pembicara dapat digunakan secara langsung tanpa perlu estimasi densitas terlebih dahulu. Sedangkan metode *one-vs-one* dipilih karena menghasilkan akurasi yang lebih baik dibanding metode *one-vs-rest* [Schmidt dan Gish, 1996].

Identifikasi pembicara merupakan proses identifikasi siapa yang berbicara dalam suatu data *speech*. Pada proses ini, karakteristik data *speech* akan dibandingkan dengan referensi data pembicara yang sudah ada. Hasilnya adalah pembicara yang model datanya dianggap paling mirip dengan data *speech*.

2.1 Sampling

Sampling [Nyquist,1920] merupakan proses untuk mengubah sinyal analog menjadi sinyal dijital (diskrit). Sinyal *speech* di-*sampling* dengan

kecepatan *sampling* fs sehingga menghasilkan sampel dengan jumlah fs x t, dimana t adalah durasi waktu.

Contoh data *speech* yang sudah dalam bentuk dijital dapat dilihat pada tabel 2.1. Pada tabel ini data berukuran 9 milidetik dengan frekuensi *sampling* 22kHz. Ini berarti data tersebut mempunyai 22050 sampel setiap detiknya atau 22,05 sampel setiap milidetik. Dengan demikian dalam 9 milidetik akan terdapat kurang lebih 200 sampel.

Dua ratus sampel data *speech* pada tabel 2.1 akan menjadi contoh data pada pemrosesan data *speech* selanjutnya:

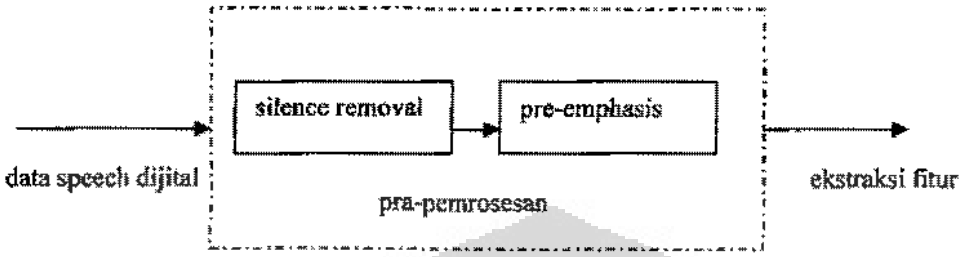
Tabel 2.1 Contoh 200 sampel data speech 22kHz 16 bit

Data Speech									
166	5614	1968	-621	-7609	2604	1676	1363	3592	365
542	5387	1470	-886	-7358	3433	622	1005	3276	-468
25	4758	1000	-1257	-7405	3952	471	535	2769	-1126
-436	4463	-66	-1694	-7536	4239	900	486	2641	-1035
-1037	3624	-1270	-2166	-7742	4597	2033	561	2654	-784
-1268	3158	-1457	-2769	-7860	4620	3004	642	2787	-541
-548	3139	-1435	-3422	-7564	4460	3598	780	3403	7
-82	3002	-743	-4311	-7120	4139	3421	1194	3651	64
426	3970	-32	-4926	-6583	3798	2313	1785	3528	62
1326	4742	341	-4972	-5884	3513	829	2470	3097	-68
1868	4926	542	-4984	-5247	3438	-744	3145	2020	-754
2767	4582	586	-5007	-4703	3692	-1849	3120	1261	-1255
3619	3049	300	-5349	-4116	4376	-2454	3073	953	-1855
3763	1430	-289	-5949	-3441	5555	-2280	2922	916	-2548
3689	413	-611	-6438	-2748	6833	-1823	2677	1268	-2902
3448	12	-1361	-6990	-1992	7776	-1096	3096	1499	-3137
3483	184	-1629	-7326	-1138	7874	-15	3350	1666	-3020
4029	878	-1417	-7645	-320	7007	698	3649	1811	-2454
4777	1195	-1311	-7913	667	5296	1460	3990	1696	-1973
5288	1509	-762	-7811	1776	3422	1660	3788	1173	-1728

2.2 Pra-pemrosesan

Pra-pemrosesan adalah proses yang dilakukan sebelum fitur audio diekstrak untuk digunakan dalam pemodelan data pembicara. Tujuannya adalah untuk memperbaiki kualitas data audio sehingga diharapkan data akan berfungsi lebih optimal pada tahap pemodelan data pembicara. Proses yang umum dilakukan adalah dengan membuang bagian yang dianggap diam (*silence*) dari

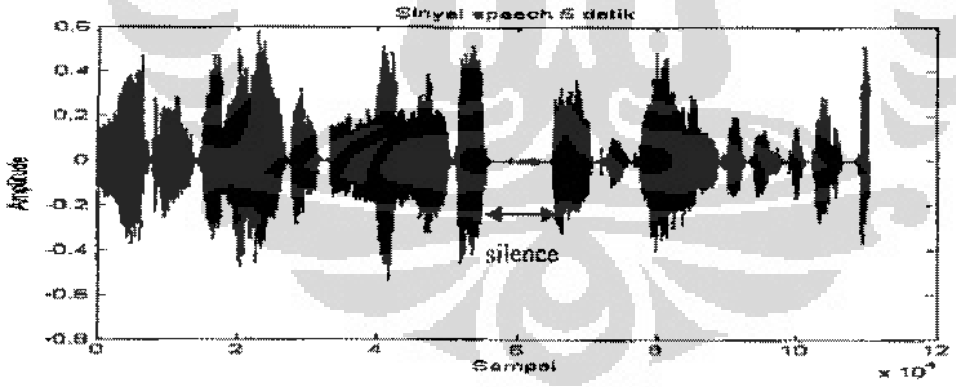
data dan memperbesar rasio antara sinyal dengan *noise* (SNR) dengan *pre-emphasis*, seperti terlihat pada gambar 2.4.



Gambar 2.4 Detail pra-pemrosesan

2.2.1 Silence Removal

Silence merupakan bagian dari data *speech* dengan tingkat energi yang lebih rendah dibanding data sekitarnya. Contoh sinyal *speech* dengan *silence* dapat dilihat pada gambar 2.5.



Gambar 2.5 Sinyal yang mengandung *silence*

Silence removal dilakukan dengan membandingkan rata-rata energi dari suatu sub data speech dengan rata-rata energi keseluruhan. Dengan membuang *silence*, diharapkan data yang akan diproses kemudian akan lebih mencerminkan karakteristik *speech* pembicara.

Rata-rata energi dihitung dengan persamaan:

$$P = \frac{1}{L}\sum_{n=i}^{L} x(n)^2 \qquad (2.1)$$

di mana L adalah panjang segmen sinyal $x(n)$, dan P adalah rata-rata energi.

Perhitungan *silence removal* adalah sebagai berikut:

1. Hitung rata-rata energi untuk seluruh data dengan persamaan (2.1), misalnya P_{all}
2. Tentukan kriteria *threshold*, misalnya 0,2 dari P_{all}, hasilnya misalnya adalah P_T
3. Bagi data menjadi sub yang lebih kecil misalnya per 80 sampel.
4. Hitung rata-rata energi pada sub data tersebut dengan persamaan (2.1), misalnya P_{sub}
5. Bandingkan P_{sub} dengan P_T. Bila $P_{sub} < P_T$, maka sub data akan dianggap sebagai *silence* dan dibuang.

Berdasarkan langkah-langkah di atas, maka perhitungan *silence removal* untuk data pada tabel 2.1 adalah sebagai berikut

1. Rata-rata energi seluruh sampel

   P_{all} = jumlah data seluruh sampel/200

   $= (166^2 + 542^2 + 25^2 + ... + (-1973)^2 + (-1728)^2) / 200$

   $= 1{,}2236 \times 10^7$

2. $P_T = 0{,}2 \times P_{all} = 0{,}2 \times 1{,}2236 \times 10^7 = 2.4472 \times 10^6$
3. Dengan membagi data menjadi sub data berukuran 80 sampel, berarti untuk 200 sampel akan ada 2 sub data dengan sisa data 40 sampel.
4. Rata-rata energi sub data 1:

$$P_{sub1} = \frac{1}{80}\sum_{n=1}^{80} x(n)^2$$

$$P_{sub1} = (166^2 + 542^2 + 25^2 + ...)/80$$

$$= 1.1923 \times 10^7$$

$$P_{sub2} = \frac{1}{80}\sum_{n=81}^{160} x(n)^2$$

$$P_{sub2} = ((-7609)^2 + (-7358)^2 + ...)) / 80 = 1.6448e \times 10^7$$

5. $P_{sub1} > P_T$ dan $P_{sub2} > P_T$. Kedua sub data tidak dianggap *silence*.

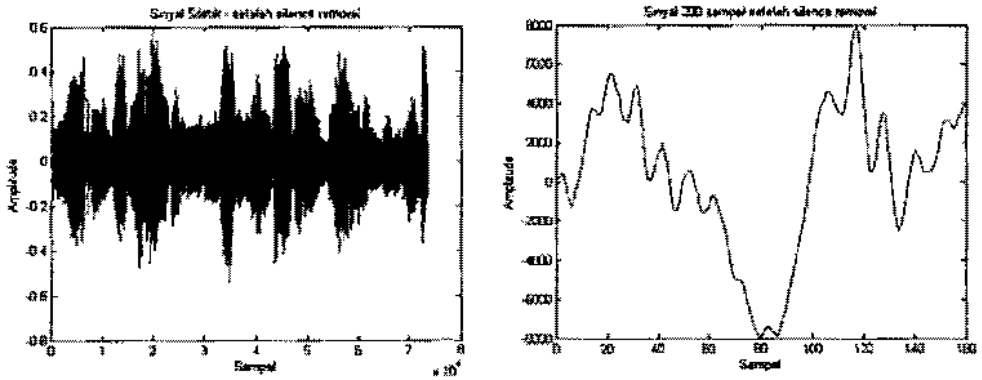
6. Sisa sampel sebanyak 40 akan dibuang, sehingga data menjadi 160 sampel.

Data pada tabel 2.1 setelah melalui proses ini akan menjadi seperti yang terlihat pada tabel 2.2

Tabel 2.2 Contoh 200 sampel data speech
setelah proses silence removal menjadi 160 sampel

Data Speech							
166	5614	1968	-621	-7609	2604	1676	1363
542	5387	1470	-886	-7358	3433	622	1005
25	4758	1000	-1257	-7405	3952	471	535
-436	4463	-66	-1694	-7536	4239	900	486
-1037	3624	-1270	-2166	-7742	4597	2033	561
-1268	3158	-1457	-2769	-7860	4620	3004	642
-548	3139	-1435	-3422	-7564	4460	3598	780
-82	3002	-743	-4311	-7120	4139	3421	1194
426	3970	-32	-4926	-6583	3798	2313	1785
1326	4742	341	-4972	-5884	3513	829	2470
1868	4926	542	-4984	-5247	3438	-744	3145
2767	4582	586	-5007	-4703	3692	-1849	3120
3619	3049	300	-5349	-4116	4376	-2454	3073
3763	1430	-289	-5949	-3441	5555	-2280	2922
3689	413	-611	-6438	-2748	6893	-1823	2677
3448	12	-1361	-6990	-1992	7776	-1096	3096
3483	184	-1629	-7326	-1138	7874	-15	3350
4029	878	-1417	-7645	-320	7007	698	3649
4777	1195	-1311	-7913	667	5296	1460	3990
5288	1509	-762	-7811	1776	3422	1660	3788

Gambar 2.6 menunjukkan gambar 2.5 yang telah melalui proses *silence removal*. Pada gambar sebelah kiri sinyal sudah tidak mengandung *silence*.



Gambar 2.6 Sinyal 5 detik (kiri) dan 200 sampel (kanan)
setelah proses *silence removal*

2.2.2 Pre-emphasis

Pre-emphasis merupakan proses untuk menaikkan rasio antara sinyal dengan *noise* (SNR). SNR dihitung dengan persamaan:

$$SNR = 10 \log_{10} \frac{P_{sinyal}}{P_{noise}} dB \tag{2.2}$$

di mana SNR adalah rasio antara sinyal dengan noise, P_{sinyal} adalah daya sinyal dan P_{noise} adalah daya *noise*.

Pada dasarnya *pre-emphasis* diimplementasikan dengan suatu filter *high pass* yang akan melewatkan frekuensi tinggi dan membuang frekuensi rendah. Fungsi transfer filter *high pass* untuk *pre-emphasis* adalah:

$$H(z) = 1 - 0.95Z^{-1} \tag{2.3}$$

dengan bentuk umum dari persamaan (2.3) adalah:

$$y(n) = x(n) - 0.95x(n-1) \tag{2.4}$$

di mana y(n) adalah output sinyal ke-n, x(n) adalah input sinyal ke-n

Contoh perhitungan *pre-emphasis* untuk data pada tabel 2.2 adalah sebagai berikut:

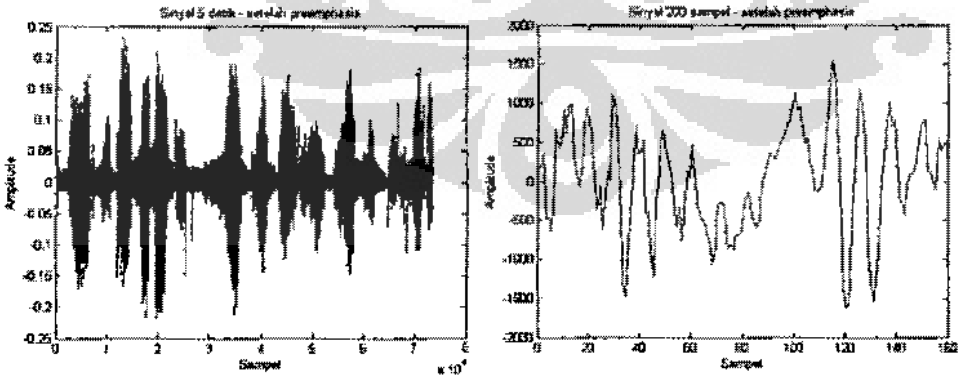
1. Untuk sampel 1, n=1, y(1) = x(1) = 166
2. Untuk sampel 2 dan seterusnya nilai y akan dipengaruhi oleh data sebelumnya. Misalnya pada n=2, y(2) = x(2) - 0,95 x (x(1)) = 542 – (0,95 x 166) = 384,3

Data pada tabel 2.2 setelah melalui proses ini akan menjadi seperti yang terlihat pada tabel 2.3

Tabel 2.3 Contoh 160 sampel data speech setelah proses *pre-emphasis*

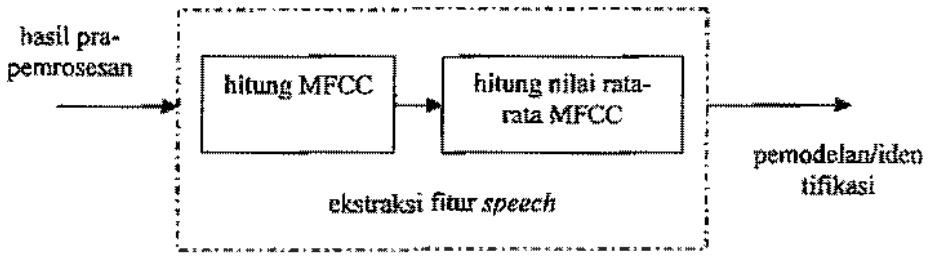
Data speech							
166	590.4	534.5	102.9	-188.6	916.8	-1574.9	-214
384.3	53.7	-399.6	-296.1	-129.5	959.2	-970.2	-289.8
-489.9	-359.6	-396.5	-415.3	-414.9	690.7	-119.9	-419.8
-459.7	-57.1	-1016	-499.9	-501.2	484.6	452.6	-22.2
-622.8	-615.8	-1207.3	-556.7	-582.8	570	1178	99.3
-282.9	-284.8	-250.5	-711.3	-505.1	252.9	1072.7	109.1
656.6	138.9	-50.9	-791.5	-97	71	744.2	170.1
438.6	20	620.2	-1060.1	65.8	-98	2.9	453
503.9	1118.1	673.9	-830.6	181	-134	-936.9	650.7
921.3	970.5	371.4	-292.3	369.8	-95.1	-1368.4	774.2
608.3	421.1	218.1	-260.6	342.8	100.7	-1531.6	798.5
992.4	-97.7	71.1	-272.2	281.6	425.9	-1142.2	132.3
990.3	-1303.9	-256.7	-592.4	351.8	868.6	-697.5	109
325	-1466.5	-574	-867.4	469.2	1397.8	51.3	2.7
114.2	-945.5	-336.4	-786.4	520.9	1555.8	343	-98.9
-56.5	-380.3	-780.6	-873.9	618.6	1284.7	635.8	552.8
207.4	172.6	-336	-685.5	754.4	486.8	1026.2	408.8
720.2	703.2	130.5	-685.3	761.1	-473.3	712.3	466.5
949.5	360.9	35.1	-650.2	971	-1360.6	796.9	523.5
749.9	373.8	483.5	-293.7	1142.4	-1609.2	273	-2.5

Dari persamaan (2.3) dan (2.4) terlihat bahwa sinyal difilter dengan mengurangi nilai (*magnitude*) dari sinyal dengan 95% *magnitude* sinyal sebelumnya. Bentuk sinyal yang telah terfilter bisa dilihat pada gambar 2.7, yang merupakan hasil filter dari gambar 2.6.



Gambar 2.7 Sinyal 5 detik (kiri) dan 200 sampel (kanan) setelah proses *pre-emphasis*

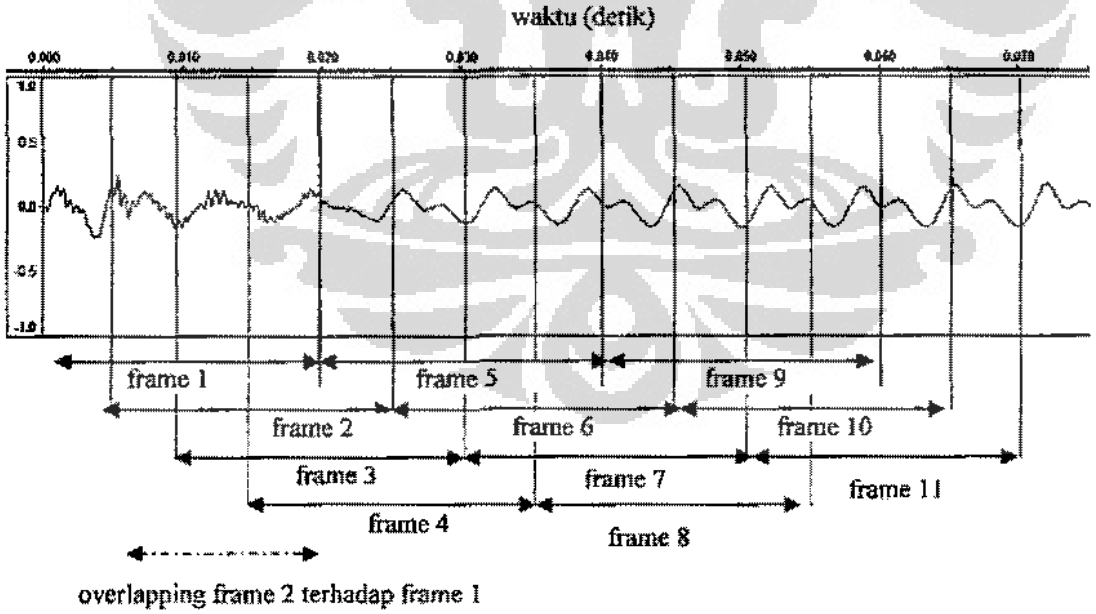
2.3 Ekstraksi Fitur Speech



Gambar 2.8 Ekstraksi fitur *speech*

Ekstraksi fitur *speech* merupakan proses untuk memperoleh karakteristik *speech* dari pembicara. Hasil pra-pemrosesan akan digunakan untuk menghitung fitur *speech*. Seperti yang terlihat pada gambar 2.8 fitur *speech* yang akan digunakan adalah MFCC.

2.3.1 Short-time Analysis



Gambar 2.9 *Short-time analysis* dengan panjang frame 20 ms dan *overlapping* frame 15 ms

Pada dasarnya, sinyal *speech* merupakan sinyal yang lamban berubah [Rabiner dan Juang, 1993] atau *quasi-stationary*. Sehingga, ketika dianalisis dalam suatu jangka waktu yang cukup pendek (20-30 milidetik) sebagai suatu frame, sinyal *speech* mempunyai karakteristik yang cukup stabil. Pada saat inilah ekstraksi fitur *speech* dilakukan. *Overlapping frame* digunakan untuk mengurangi adanya informasi yang hilang antar frame. Ukuran *overlapping frame* biasanya sebesar 30%-50% dari panjang frame.

Jika data pada tabel 2.3 dianalisis dalam suatu frame yang berukuran 20 sampel dengan overlapping 15 sampel, maka jumlah frame yang dihasilkan dari 160 sampel adalah:

$$jumlah\_frame = \frac{jumlah\_sampel - panjang\_frame + overlapping\_frame}{overlapping\_frame}$$

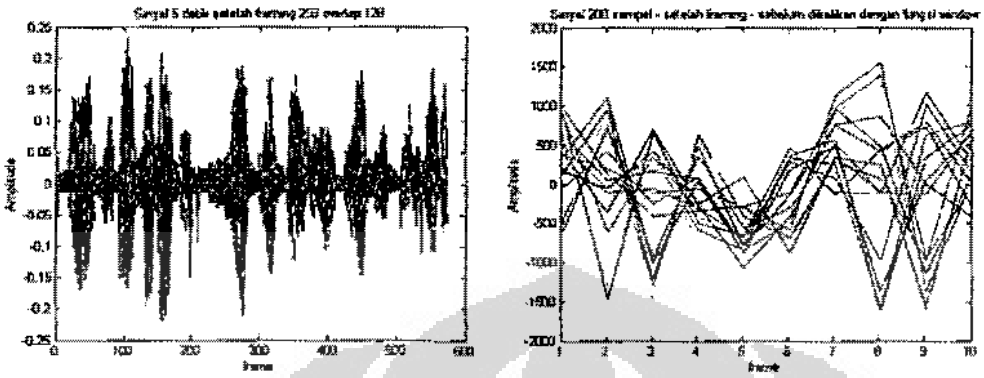
$$= (160 - 20 + 15)/15 = 10 \text{ frame}$$

Hasil selengkapnya data pada tabel 2.3 yang telah dikelompokkan dalam frame dapat dilihat pada tabel 2.4.

Tabel 2.4 Data tabel 2.3 yang sudah dikelompokkan dalam frame

Sam Pel	Frame									
	1	2	3	4	5	6	7	8	9	10
1	166	-56.5	421.1	-250.5	102.9	-873.9	342.8	252.9	-1574.9	635.8
2	384.3	207.4	-97.7	-50.9	-296.1	-685.5	281.6	71	-970.2	1026.2
3	-489.9	720.2	-1303.9	620.2	-415.3	-685.3	351.8	-98	-119.9	712.3
4	-459.7	949.5	-1466.5	673.9	-499.9	-650.2	469.2	-134	452.6	796.9
5	-622.8	749.9	-945.5	371.4	-556.7	-293.7	520.9	-95.1	1178	273
6	-282.9	590.4	-380.3	218.1	-711.3	-188.6	618.6	100.7	1072.7	-214
. . .	. . .									. . .
15	114.2	970.5	-1207.3	483.5	-786.4	369.8	570	-1609.2	343	774.2
16	-56.5	421.1	-250.5	102.9	-873.9	342.8	252.9	-1574.9	635.8	798.5
17	207.4	-97.7	-50.9	-296.1	-685.5	281.6	71	-970.2	1026.2	132.3
18	720.2	-1303.9	620.2	-415.3	-685.3	351.8	-98	-119.9	712.3	109
19	949.5	-1466.5	673.9	-499.9	-650.2	469.2	-134	452.6	796.9	2.7
20	749.9	-945.5	371.4	-556.7	-293.7	520.9	-95.1	1178	273	-98.9

Bentuk sinyal dalam frame yang dihasilkan dapat dilihat pada gambar 2.10.



Gambar 2.10 Sinyal 5 detik setelah di-framing 256 sampel dan *overlap* 128 sampel (kiri), sinyal 200 sampel setelah di-framing 20 sampel, *overlap* 15 sampel (kanan)

Untuk mencegah adanya perubahan yang drastis pada akhir frame, maka sinyal audio akan dikalikan dengan fungsi windowing, yaitu dalam hal ini adalah fungsi window Hamming,

$$w(k+1) = 0.54 - 0.46\cos\left(2\pi \frac{k}{n-1}\right) \tag{2.5}$$

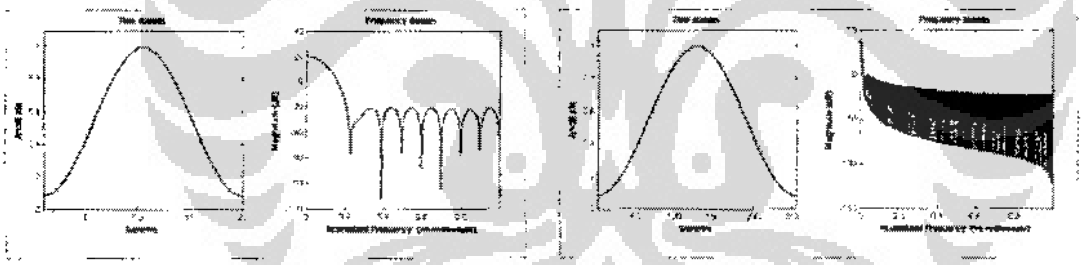
di mana n adalah jumlah sampel dalam satu frame, dan k = 0, 1, 2, ..., n-1

Selanjutnya data pada tabel 2.4 akan dikalikan dengan fungsi window hamming pada persamaan (2.5). Contoh perhitungan hamming window untuk 20 sampel per frame adalah sebagai berikut:

1. Untuk sampel pertama n = 0, w(1) = 0,54 – 0,46cos(0) = 0,08
2. Untuk sampel kedua, n = 1, w(2) = 0,54 – 0,46cos(2π) = 0,1049
3. Begitu seterusnya sampai jumlah sampel 20. Nilai windowing selengkapnya dapat dilihat pada tabel 2.5. Dan gambar 2.11 menunjukkan window yang dihasilkan.

Tabel 2.5 Hasil hamming window 20 sampel

Sampel (n)	w (n)
1	0.0800
2	0.1049
3	0.1770
4	0.2884
5	0.4271
6	0.5780
7	0.7248
8	0.8515
9	0.9446
10	0.9937
11	0.9937
12	0.9446
13	0.8515
14	0.7248
15	0.5780
16	0.4271
17	0.2884
18	0.1770
19	0.1049
20	0.0800



Gambar 2.11 Hamming window dengan 20 sampel (kiri) dengan 256 sampel (kanan)

Setiap data pada tabel 2.4 akan dikalikan dengan fungsi window sesuai urutan sampelnya. Contoh perhitungannya sebagai berikut:

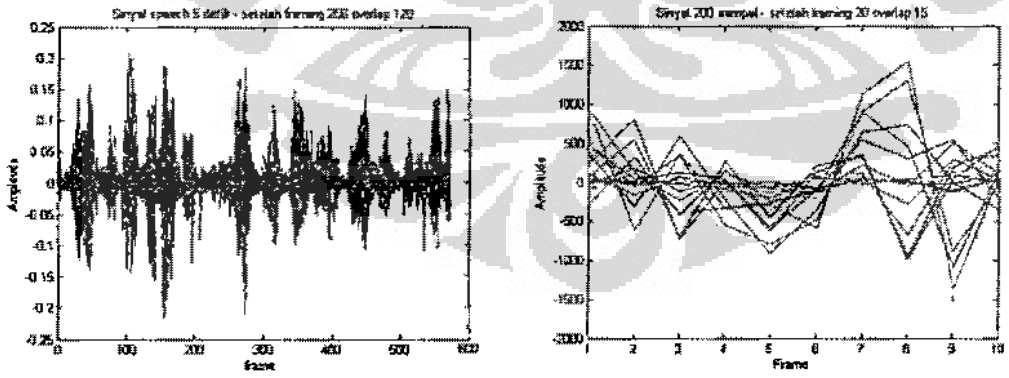
1. Frame 1, sampel 1, nilai data = 166 dikalikan dengan fungsi window sampel 1 nilai data baru = 166 x 0,08 = 13,28
2. Frame 2, sampel 1, nilai data = -56,5 dikalikan dengan fungsi window sampel 1, nilai data baru = -56,5 x 0,08 = -4,52
3. Frame 5, sampel 2, nilai data = -296,1 dikalikan dengan fungsi window sampel 2, nilai data baru = -296,1 x 0,1049 = -31,06

Hasil selengkapnya data pada tabel 2.4 yang telah dikalikan dengan fungsi window dapat dilihat pada tabel 2.6.

Tabel 2.6 Data tabel 2.4 yang telah dikalikan dengan fungsi window pada tabel 2.5

Sam	Frame									
Pel	1	2	3	4	5	6	7	8	9	10
1	13.28	-4.52	33.69	-20.04	8.23	-69.91	27.40	20.20	-126.00	50.87
2	40.32	21.76	-10.25	-5.34	-31.06	-71.93	29.60	7.40	-101.80	107.67
3	-86.71	127.46	-230.78	109.78	-73.51	-121.30	62.30	-17.30	-21.20	126.06
.	.									.
.	.									.
.	.									.
19	99.62	-153.88	70.70	-52.45	-68.23	49.23	-14.10	47.50	83.60	0.28
20	59.99	-75.64	29.71	-44.54	-23.49	41.68	-7.60	94.20	21.80	-7.91

Bentuk sinyal yang sudah dikalikan dengan fungsi window dapat dilihat pada gambar 2.12. Jika dibandingkan dengan sinyal pada gambar 2.9, terlihat bahwa dengan windowing, bagian tepi sinyal menjadi lebih halus.



Gambar 2.12 Sinyal 5 detik (kiri) dan 200 sampel (kanan) setelah dikalikan dengan fungsi window hamming

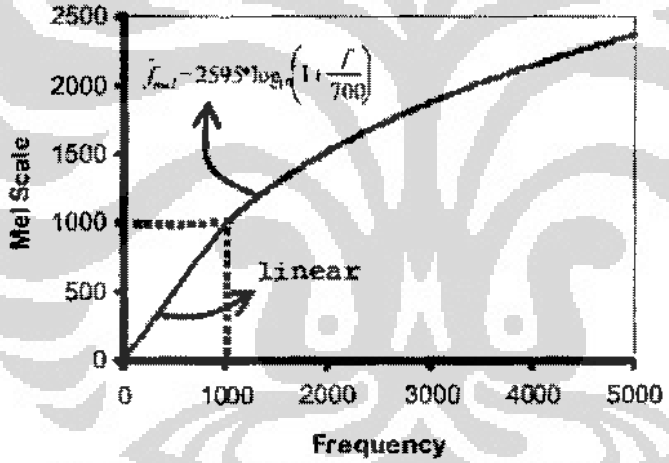
2.3.2 Mel-scale Frequency Cepstrum Coefficient (MFCC)

Mel frequency cepstrum coeffisient merupakan fitur audio yang banyak digunakan dalam pemrosesan audio. Penggunaan MFCC didasarkan pada kenyataan bahwa komponen frekuensi rendah dari sinyal audio lebih penting bagi manusia dibanding komponen frekuensi tinggi. Untuk frekuensi di bawah 1 kHz, telinga manusia akan mendengar suara dalam skala linear sedangkan di atas 1 kHz dalam skala logaritmis. Perhitungan mel untuk frekuensi f (Hz):

$$mel(f) = 2595 * \log\left(1 + \frac{f}{700}\right) \tag{2.6}$$

di mana mel(f) adalah frekuensi dalam skala mel, dan f adalah frekuensi.

Gambar 2.13 menunjukkan hubungan antara frekuensi f dengan frekuensi mel.



Gambar 2.13 Hubungan antara frekuensi audio dengan frekuensi dalam skala mel [Buono et al, 2008]

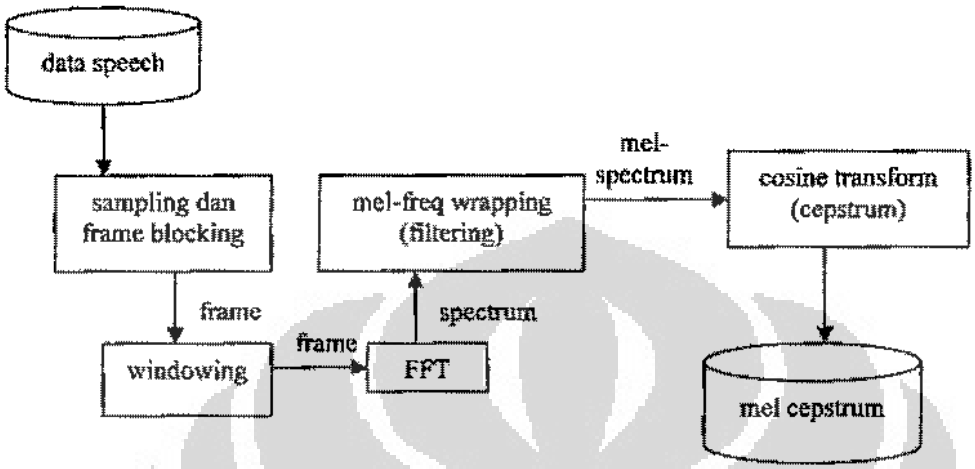
Setelah dilakukan framing, kemudian dilakukan ekstraksi fitur *speech* MFCC. Proses perhitungan MFCC seperti yang ditunjukkan oleh gambar 2.14 adalah sebagai berikut:

1. Fast Fourier Transform (FFT)

FFT [Cooley and Tukey, 1965] merupakan metode yang digunakan untuk menghitung Discrete Fourier Transform (DFT), dengan persamaan:

$$X(k) = \sum_{n=0}^{N} x(n)\exp(-j2\pi kn / N) \qquad (2.7)$$

di mana x(n) adalah sinyal dengan panjang N

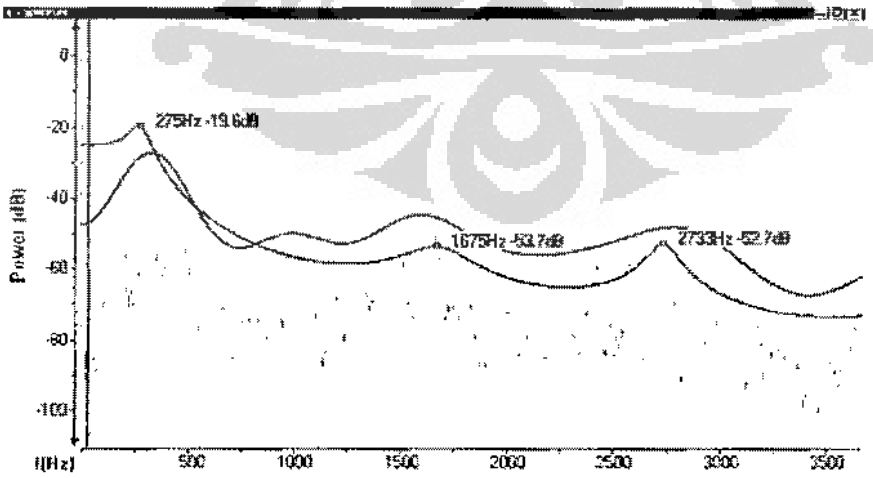


Gambar 2.14 Proses perhitungan MFCC

[Buono et al, 2008]

Dengan DFT, data pada setiap frame akan diubah dari domain waktu (*time-domain*) ke domain frekuensi (*frequency-domain*), sehingga dapat dianalisa karakteristik spektralnya. Hasilnya berupa spektrum atau *periodogram*.

Bentuk spektrum dari data 200 sampel dapat dilihat pada gambar 2.15



Gambar 2.15 Spektrum data 200 sampel

2. Mel-frequency wrapping (filtering)

Spektrum dimasukkan ke dalam beberapa mel-filter yang berbentuk segitiga (*triangular*). Mel-filter akan mengubah frekuensi suara yang lebih tinggi dari 1 kHz sesuai skala mel pada persamaan (2.6).

Mel-frequency spectrum coefficient dihitung sebagai jumlah hasil filter, yaitu:

$$X_i = \log\left(\sum_{f=0}^{N-1} abs(X(j)) * H_i(f)\right) \tag{2.8}$$

di mana i adalah jumlah filter, N adalah jumlah koefisien FFT , abs(X(j)) adalah nilai koefisien ke-j dari periodogram, dan $H_i(f)$ adalah nilai filter triangular ke-i di titik f.

Perhitungan mel-filter dipengaruhi oleh jumlah filter, jumlah window fft, dan frekuensi *sampling*.

3. Cosine Transform

Mel frequency spectrum coefficient dikembalikan ke domain waktu dengan menggunakan *discrete cosine transform* (DCT). DCT pada dasarnya hanya menghitung bagian angka real dari DFT, dengan persamaan

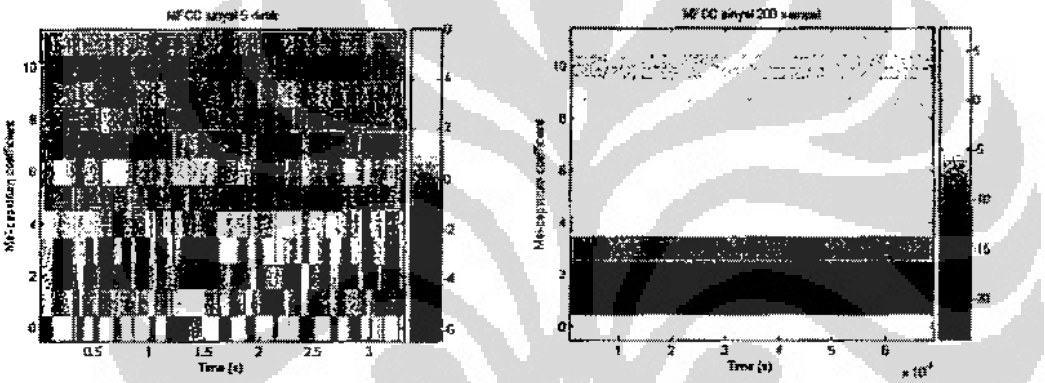
$$C_j = \sum_{i=1}^{M} X_i \cos\left(\frac{j(i-0.5)\pi}{20}\right) \tag{2.9}$$

di mana j adalah jumlah koefisien MFCC, M adalah jumlah filter segitiga, dan X_i adalah koefisien *mel-spectrum*.

Data MFCC koefisien 0-11 untuk data pada tabel 2.6 dapat dilihat pada tabel 2.7, berupa matriks berukuran 10 x 12. Dan bentuk sinyalnya dapat dilihat pada gambar 2.16

Tabel 2.7 Data fitur MFCC dari data pada tabel 2.6

Data MFCC											
Koef 0	Koef 1	Koef 2	Koef 3	Koef 4	Koef 5	Koef 6	Koef 7	Koef 8	Koef 9	Koef 10	Koef 11
7.53	-23.67	-17.16	-9.36	0.19	5.24	6.50	4.12	0.46	-3.63	-5.52	-2.64
6.87	-24.19	-17.10	-8.10	1.40	3.97	5.90	3.92	0.38	-2.93	-5.09	-3.17
7.55	-24.24	-17.53	-8.17	1.03	4.41	5.93	3.98	0.53	-3.23	-5.07	-3.21
.											.
.											.
.											.
6.15	-23.22	-17.49	-7.85	0.37	4.62	6.15	3.82	0.57	-3.43	-4.90	-3.12



Gambar 2.16. MFCC sinyal 5 detik (kiri) dan MFCC sinyal 200 sampel (kanan)

2.4 Principal Component Analysis (PCA)

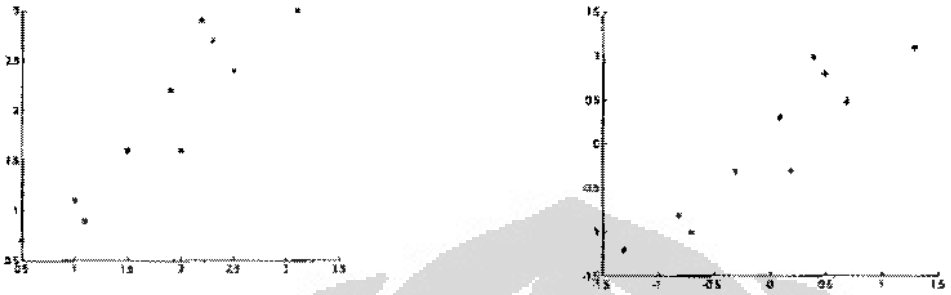
PCA [Pearson, 1901] merupakan metode yang digunakan untuk mereduksi jumlah dimensi data, dengan demikian jumlah data yang akan diproses akan berkurang. Meskipun dimensi data berkurang, tetapi informasi yang ada di dalamnya tidak banyak hilang, karena yang disimpan adalah bagian data yang penting.

Algoritma penerapan PCA pada data adalah sebagai berikut:

1. Normalisasi data dengan *mean-centering*

Data dinormalisasi dengan cara *mean-centering*, di mana setiap data dikurangi dengan rata-rata data per dimensinya. Hasil normalisasi berupa data baru dengan rata-rata nol (*zero mean*). Data baru ini lebih berkurang variasinya

dan mendekat ke pusat data. Gambar 2.17 menunjukkan plot data sebelum dan sesudah normalisasi.



Gambar 2.17 Plot data sebelum dinormalisasi (kiri), setelah dinormalisasi (kanan)

Perhitungan normalisasi data adalah sebagai berikut:

1. Hitung rata-rata (*mean*) data per dimensi. Rata-rata suatu dimensi diperoleh dengan menghitung jumlah data dimensi tersebut dibagi dengan banyak data dimensi tersebut.
2. Kurangi setiap data dengan rata-rata sesuai dimensinya

Contoh perhitungan normalisasi untuk data pada tabel 2.7 adalah sebagai berikut:

1. Rata-rata dimensi 1 =

$$\frac{7{,}5312 + 6{,}8729 + 7{,}5531 + 6{,}7675 + 6{,}9760 + 5{,}9881 + 6{,}0343 + 7{,}0092 + 7{,}6119 + 6{,}1471}{10}$$

$= 6{,}8491$

2. Kurangi data dimensi pertama dengan rata-rata dimensi pertama. Misalnya untuk data pertama dimensi pertama, nilai hasil normalisasinya adalah:

   7,5312 - 6,8491 = 0,6821.

Hasil selengkapnya data pada tabel 2.7 yang telah dinormalisasi, dapat dilihat pada tabel 2.8

Tabel 2.8 Data tabel 2.7 yang telah dinormalisasi

Data MFCC											
Koef 0	Koef 1	Koef 2	Koef 3	Koef 4	Koef 5	Koef 6	Koef 7	Koef 8	Koef 9	Koef 10	Koef 11
0.682	-0.598	0.757	-0.869	-0.354	0.686	-0.025	-0.078	0.287	-0.200	-0.503	0.554
0.024	-1.122	0.815	0.386	0.857	-0.581	-0.621	-0.274	0.207	0.494	-0.077	0.026
0.704	-1.172	0.388	0.321	0.487	-0.146	-0.588	-0.216	0.354	0.191	-0.057	-0.017
.											.
.											.
.											.
-0.702	-0.150	0.427	0.642	-0.179	0.065	-0.370	-0.377	0.393	-0.003	0.115	0.071

2. Hitung matriks kovarians (*covariance matrix*)

Dari nilai kovarians, dapat terlihat tingkat hubungan antara dua dimensi data, karena kovarians mengukur varians antara dua dimensi data. Nilai kovarians positif, menunjukkan korelasi positif antara kedua dimensi data sedangkan bila negatif, maka korelasi kedua dimensi data adalah negatif.

Matriks kovarians dihitung dengan persamaan:

$$\mathrm{cov}(x) = \frac{1}{m-1} X^T X \qquad (2.10)$$

di mana m adalah jumlah baris dari matrik X.

Matriks kovarians untuk data pada tabel 2.8 adalah matriks bujur sangkar berukuran 12 x 12, beberapa datanya dapat dilihat pada tabel 2.9.

Tabel 2.9 Nilai kovarians untuk data pada tabel 2.8

Data MFCC											
Koef 0	Koef 1	Koef 2	Koef 3	Koef 4	Koef 5	Koef 6	Koef 7	Koef 8	Koef 9	Koef 10	Koef 11
0.388	-0.185	-0.177	0.010	0.108	0.002	-0.059	-0.003	0.069	-0.002	0.388	-0.026
-0.185	0.804	-0.705	-0.046	-0.198	0.060	0.313	0.153	-0.163	-0.145	-0.185	-0.097
-0.177	-0.705	1.147	-0.214	0.027	0.042	-0.219	-0.134	0.091	0.108	-0.177	0.224
.											
.											
.											
-0.026	-0.097	0.224	-0.116	-0.022	0.059	-0.027	-0.021	0.031	-0.005	-0.050	0.080

3. Hitung vektor eigen (*eigenvector*) dan nilai eigen (*eigenvalue*) dari matriks *covariance*

Eigenvector [Brauer dan Weyl, 1935] hanya dimiliki oleh matriks bujur sangkar, namun tidak semua matriks bujur sangkar mempunyai *eigenvector*. Jumlah *eigenvector* yang dimiliki oleh matriks NxN adalah sebanyak N. *Eigenvector* pada suatu matriks saling tegak lurus satu sama lain.

Eigenvalue [Eddington, 1927] untuk data pada tabel 2.9 dapat dilihat pada tabel 2.10. Pada tabel terlihat bahwa hanya 7 nilai eigen pertama yang mempunyai arti, sisanya kosong. Vektor eigen untuk data pada tabel 2.9 merupakan matriks berukuran 12x12 yang dapat dilihat pada tabel 2.11

Tabel 2.10 Nilai eigen untuk data pada tabel 2.9

Nilai Eigen
1.9239
0.8586
0.4528
0.2735
0.1113
0.0127
0.0053
0.0000
0.0000
-0.0000
-0.0000
-0.0000

Tabel 2.11 Vektor eigen dari data pada tabel 2.6

Vektor eigen											
Koef 0	Koef 1	Koef 2	Koef 3	Koef 4	Koef 5	Koef 6	Koef 7	Koef 8	Koef 9	Koef 10	Koef 11
0.011	-0.439	0.664	0.241	-0.239	-0.014	0.024	-0.185	0.011	0.162	0.338	-0.275
-0.597	0.331	-0.086	-0.244	0.215	0.002	-0.013	-0.153	0.029	0.247	0.488	-0.313
0.718	0.411	-0.132	-0.011	-0.153	0.092	0.000	-0.072	0.178	0.147	0.427	-0.162
.											.
.											.
.											.
0.129	0.171	0.147	-0.004	0.314	-0.363	0.199	-0.742	0.063	-0.176	-0.265	-0.080

4. Pilih komponen yang diinginkan

Urutkan nilai eigen dari besar ke kecil dan susun vektor eigen berdasarkan urutan nilai eigen. Vektor eigen dengan nilai eigen yang kecil dapat diabaikan sehingga dimensi data set yang baru akan berkurang sebanyak jumlah nilai eigen yang diabaikan.

Nilai eigen yang dipilih dari data pada tabel 2.10 adalah sebanyak 7 buah, karena nilai eigen yang ke-8 sampai 12 adalah 0, sehingga dapat diabaikan. Ketujuh nilai eigen yang dipilih dapat dilihat pada tabel 2.12

Tabel 2.12 Nilai eigen yang dipilih dari data tabel 2.10

Nilai Eigen
1.9239
0.8586
0.4528
0.2735
0.1113
0.0127
0.0053

Berdasarkan nilai eigen yang dipilih pada tabel 2.12, maka vektor eigen yang dipilih mempunyai ukuran 12x7, seperti terlihat pada tabel 2.13.

Tabel 2.13 Vektor eigen yang dipilih dari data tabel 2.11

Vektor Eigen						
Koef 0	Koef 1	Koef 2	Koef 3	Koef 4	Koef 5	Koef 6
0.011	-0.439	0.664	0.241	-0.239	-0.014	0.024
-0.597	0.331	-0.086	-0.244	0.215	0.002	-0.013
0.718	0.411	-0.132	-0.011	-0.153	0.092	0.000
.						.
.						.
.						.
0.129	0.171	0.147	-0.004	0.314	-0.363	0.199

5. Bentuk data baru

Data set baru diperoleh dengan mengalikan komponen vektor eigen yang dipilih dengan data awal yang sudah dinormalisasi.

Data baru dari data tabel 2.7 merupakan matriks berukuran 10 x 7, dapat dilihat pada tabel 2.14. Jumlah dimensi data baru sama dengan jumlah dimensi nilai eigen yang dipilih.

Tabel 2.14 Data baru dari data tabel 2.7 yang telah direduksi dimensinya dengan PCA

Data MFCC						
Koef 0	**Koef 1**	**Koef 2**	**Koef 3**	**Koef 4**	**Koef 5**	**Koef 6**
1.047	0.571	1.323	0.426	0.258	-0.118	0.023
1.594	-0.951	-0.470	-0.455	-0.025	-0.096	0.119
1.243	-1.142	0.241	0.161	0.033	0.046	-0.054
.						.
.						.
.						.
0.494	0.076	-0.927	0.352	0.587	0.139	-0.007

2.5 Support Vector Machine (SVM)

Support Vector Machine (SVM) [Vapnik, 1979] merupakan metode yang menggunakan konsep *hyperplane* untuk memisahkan kelas data yang satu dengan kelas data lainnya. *Support vector* merupakan data yang posisinya paling dekat dengan *hyperplane*.

Pada kasus di mana data sulit untuk dipisahkan, maka data akan diubah ke dimensi yang lebih tinggi dengan menggunakan fungsi kernel. Beberapa data juga akan diperbolehkan untuk *misclassified*, namun dengan pinalti (ξ) yang semakin besar sesuai dengan semakin jauh jaraknya dari *hyperplane*. Hal ini dilakukan untuk memperoleh margin yang optimal.

Optimalisasi *hyperplane* dapat diperoleh dengan meminimalisasi

$$C\sum_{n=1}^{N}\xi_n + \frac{1}{2}\| w \|^2 \qquad (2.11)$$

di mana C = koefisien regulasi, ξ_n = error pelatihan, dan w =bobot

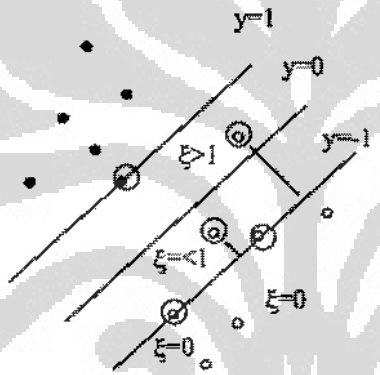
dengan batasan

$$t_n y(x_n) \geq 1 - \xi_n \tag{2.12}$$

di mana t_n = target dari data ke-n, $y(x_n)$ = hasil keluaran, n= data 1, ..., N

$$\xi_n \geq 0 \tag{2.13}$$

Data dengan $\xi_n = 0$ berada di margin atau di dalam margin kelas yang benar, dan data dengan $0 < \xi_n \leq 1$ berada di dalam margin dengan kelas yang benar, sedangkan $\xi_n > 0$ berada di sisi yang salah dari margin (*misclassified*), seperti pada gambar 2.18.



Gambar 2.18 Posisi data dan hubungannya dengan error pelatihan (ξ)
Data dalam lingkaran adalah support vector.

Parameter C > 0 merupakan *regulator* yang mengontrol keseimbangan antara *training error* (ξ) dengan margin (kompleksitas model).

Representasi dual Lagrangian untuk optimalisasi adalah:

$$\overline{L}(\mathrm{a}) = \sum_{n=1}^{N} a_n - \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} a_n a_m t_n t_m k(x_n, x_m) \tag{2.14}$$

di mana a_n, a_m adalah pengali lagrange; t_n, t_m adalah target; $k(x_n, x_m)$ adalah fungsi kernel; n dan m adalah data

dengan batasan

$$0 \leq a_n \leq C \tag{2.15}$$

$$\sum_{n=1}^{N} a_n t_n = 0 \tag{2.16}$$

Fungsi kernel didefinisikan sebagai berikut:

$$k(x,x') = \varphi(x)^T \varphi(x') \tag{2.17}$$

di mana φ(x) adalah *basis function*.

Klasifikasi data baru dalam parameter a_n dan fungsi kernel adalah:

$$y(x) = \sum_{n=1}^{N} a_n t_n k(x,x_n) + b \tag{2.18}$$

di mana a_n adalah pengali lagrange, t_n adalah target, $k(x,x_n)$ adalah fungsi kernel, b adalah bias, dan n adalah data

2.5.1 Kernel Radial Basis Function (RBF)

Kernel yang umum digunakan adalah RBF atau kernel Gaussian.

$$k(x,x') = \exp\left(-\frac{\|x-x'\|^2}{2\sigma^2}\right) = \exp\left(-\gamma \|x-x'\|^2\right) \tag{2.19}$$

di mana x adalah support vector, x' adalah data testing, σ adalah *radius*, dan $\gamma = 1/2\sigma^2$

Dari persamaan di atas, terlihat bahwa hasil keluaran dari kernel tergantung pada jarak Euclidean dari x dan x' (*support vector* dan data testing). *Support vector* akan menjadi pusat RBF dan σ akan menentukan daerah pengaruh *support vector* terhadap data. Semakin besar σ maka semakin besar daerah pengaruhnya, semakin sedikit jumlah *support vector*.

2.5.2 Grid Search

Parameter C > 0 merupakan pengontrol keseimbangan antara *training error* (ξ) dengan margin (kompleksitas model), sedangkan γ (1/2σ) merupakan parameter kernel RBF. Untuk memperoleh kombinasi parameter C dan γ yang akan menghasilkan akurasi pelatihan dan identifikasi yang optimal, maka digunakan metode *grid search* dan *cross validation*. *Grid search* pada dasarnya hanya mencoba kombinasi C dan γ dalam pelatihan sistem, pada interval C dan γ tertentu yang diperkirakan akan menghasilkan akurasi yang optimal. *Cross validation* digunakan untuk mencegah terjadinya *overfitting*, di mana akurasi

pelatihan sangat baik, namun akurasi identifikasi tidak terlalu baik. *Overfitting* terjadi ketika sistem terlalu *fit* (pas) dengan data pelatihan sehingga akurasi pelatihannya sangat baik, namun tidak bisa mengenali banyak data lainnya sehingga akurasi identifikasinya kurang baik. *Cross validation* mengatasi hal ini dengan membagi data menjadi n (jumlah *cross validation*) bagian, di mana dalam setiap giliran, 1 bagian akan menjadi data testing sedangkan (n-1) bagian akan menjadi data pelatihan. Setiap giliran akan mempunyai akurasi, yang kemudian akan dirata-rata untuk memperoleh akurasi *cross validation*. Dengan demikian, jika suatu kombinasi C dan γ mempunyai akurasi *cross validation* yang cukup baik, diharapkan akan diperoleh akurasi identifikasi yang sama baiknya.

Proses *grid search* dengan *cross validation* ini yang paling banyak memerlukan sumber daya komputasi dan waktu sehingga data SVM perlu disederhanakan sebelum memasuki proses ini.

Penggunaan *grid search* dalam penentuan parameter kernel C dan γ adalah sebagai berikut:

1. Tentukan interval C dan γ di mana kemungkinan akan diperoleh akurasi terbesar.
2. Untuk setiap kombinasi (C, γ) akan dilakukan pelatihan sistem.
3. Kombinasi (C, γ) dengan akurasi *cross validation* yang paling besar yang akan dipilih.

Penentuan akurasi *cross validation* adalah sebagai berikut:

1. Tentukan jumlah *cross validation*, misalnya n dan umumnya 5.
2. Data pelatihan akan dibagi menjadi n bagian. Kemudian secara bergiliran satu bagian akan menjadi *validator* (data testing) dan yang lain ((n-1) bagian) akan digunakan dalam pelatihan. Setiap giliran akan mempunyai akurasi validasi.
3. Rata-rata dari n akurasi validasi akan menjadi akurasi *cross validation*.

Parameter C dan γ yang dihasilkan oleh *grid search* dengan *cross validation* kemudian akan digunakan dalam pemodelan pembicara.

2.5.3 Pemodelan Pembicara

Metode yang digunakan dalam pemodelan pembicara adalah SVM *multi-class* yang pada dasarnya merupakan kombinasi dari beberapa SVM dua kelas. Secara umum ada dua macam metode SVM *multi-class*, yaitu *one-vs-one* dan *one-vs-rest*.

Pada SVM *multi-class one-vs-one* [Knerr et al., 1990], setiap kelas akan dibandingkan dengan setiap kelas lainnya, yaitu setiap dua kelas .yang berbeda akan menghasilkan 1 model data. Dengan demikian, jumlah model yang dihasilkan dalam satu kali pelatihan dengan K kelas data adalah K(K-1)/2. Jika terdapat 20 kelas data, maka dalam satu kali pelatihan sistem, akan diperoleh 20(19)/2 = 190 model. Metode ini menghasilkan akurasi yang lebih baik dibanding metode *one-vs-rest* [Schmidt dan Gish, 1996] meskipun jumlah model yang dihasilkan lebih banyak.

Sedangkan pada SVM *multi-class one-vs-rest* [Vapnik, 1998], setiap kelas akan dibandingkan dengan seluruh kelas lainnya, sehingga jumlah model data yang dihasilkan adalah sebanyak jumlah kelasnya. Untuk penggunaan data dengan kelas yang sangat banyak sehingga jumlah data menjadi sangat besar, metode ini kurang menguntungkan, karena setiap kali pembuatan model digunakan seluruh data, yang berarti membutuhkan sumber daya komputasi yang tidak sedikit.

Pada tesis ini, metode SVM *multi-class* yang digunakan adalah *one-vs-one*. Misalnya pada sistem dengan 3 pembicara, masing-masing mempunyai satu data pelatihan. Setelah dilakukan pelatihan sistem, diperoleh parameter SVM dan model data dari ketiga pembicara.

Tabel 2.15 Parameter SVM contoh pemodelan

Total SV	γ	Kelas	b	Label	Jumlah sv	Koefisien sv
3	0,8333	1	0	1	1	[1 1 1]
		2	0	2	1	[-1 1 1]
		3	0	3	1	[-1 -1 1]

Parameter SVM untuk contoh 3 pembicara dengan masing-masing satu data pelatihan ditunjukkan pada tabel 2.15, yaitu berupa jumlah *support vector* (sv), parameter kernel γ, nilai bias b, jumlah *support vector* untuk masing-masing kelas, dan koefisien *support vector*. Selain itu, juga diperoleh support vector dari masing-masing kelas, seperti yang ditunjukkan oleh tabel 2.16.

Tabel 2.16 Support vector per kelas

Dimensi	Support Vector Kelas		
	1	2	3
1	-0.4325	0.26954	0.200371
2	-0.0378	0.059777	-0.06737
3	-0.2772	-0.22018	-0.53681
4	-0.31847	-0.2164	-0.31605
5	-0.73252	0.133674	0.178344
6	0.911854	0.071199	0.090145
7	0.733544	0.136396	0.546666
8	0.103156	0.445278	0.439891
9	0.119475	-0.5417	-0.43006
10	-0.91211	-0.04098	-0.20524
11	-0.15764	0.220912	-0.28576
12	0.273128	-0.43828	-0.50909

Pemodelan pembicara pada tesis ini dilakukan dengan menggunakan software SVM, yaitu LibSVM [Chang dan Lin, 2001].

2.5.4 Identifikasi Pembicara

Karena metode SVM *multi-class* yang digunakan adalah *one-vs-one*, maka identifikasi kelas untuk input baru dilakukan dengan cara *majority voting*, yaitu kelas mana yang paling banyak kesesuaiannya (menang).

Contoh identifikasi pembicara: jika ada dua input data pembicara yang tidak diketahui siapa pembicaranya, maka dengan menggunakan parameter SVM dari 3 pembicara sebelumnya (tabel 2.15 dan 2.16) dapat diketahui pembicara

pada kedua input tersebut. Data yang akan diidentifikasi dapat dilihat pada tabel 2.17.

Tabel 2.17 Dua data input contoh

Data Input		
Dimensi	Data 1	Data 2
1	0.050374	0.234384
2	-0.06152	-0.16138
3	-0.32142	-0.1974
4	-0.31337	-0.56989
5	-0.38604	0.379574
6	0.569763	0.114835
7	0.611197	0.409962
8	0.433974	0.601415
9	-0.51572	-0.44296
10	-0.41975	-0.28311
11	-0.26885	-0.35019
12	0.00091	-0.15012

Berdasarkan fungsi (2.18), perhitungan identifikasi pembicara dari data input 1 adalah sebagai berikut:

1. Hitung kuadrat jarak antara data dengan setiap model data.

$$d_k = \sum_{n=1}^{N} (x_n - y_n)^2$$

k = kelas data, N = dimensi data

Jarak input 1 dimensi 1 dengan SV kelas 1 dimensi 1 :

$d_{11} = (0{,}050374 - -0{,}432502)^2 = 0{,}23317$

Jarak input 1 dimensi 2 dengan SV kelas 1 dimensi 2 :

$d_{12} = (-0{,}062 - -0{,}038)^2 = 0{,}001$

Total jarak input 1 dengan kelas 1 :

$d_1 = d_{11} + d_{12} + d_{13} + d_{14} + d_{15} + d_{16} + d_{17} + d_{18} + d_{19} +$

$d_{110} + d_{111} + d_{112}$

$= 1.329559$

2. Hitung kernel $k(n) = e^{-\gamma d}$ untuk setiap kelas n

   Kelas 1 = $k(1) = e^{-\gamma d_1} = e^{-(0,08333)(1,329559)} = 0.895121$

3. Dengan menggunakan koefisien *support vector* (sv), pada tabel 2.15, bandingkan data input dengan *support vector* data kelas. Bila hasilnya > 0, maka yang terpilih adalah kelas 1, dan jika hasilnya < 0 maka yang terpilih adalah kelas ke-2 Yang paling banyak terpilih adalah yang menjadi kelas data tersebut.

   Koefisien sv kelas 1 = c1 = [1 1 1]

   Koefisien sv kelas 2 = c2 = [-1 1 1]

   Koefisien sv kelas 3 = c3 = [-1 -1 1]

   Bandingkan kelas pembicara 1 dengan kelas pembicara 2 :

   $y_{12} = (k(1) \times c1[1] + k(2) \times c2[1]) + b_1 = ((0.895)(1) + (0.890)(-1)) - 0 = 0.005$

   $y_{12} > 0$ => maka kelas pembicara 1 yang terpilih

   Bandingkan kelas pembicara 1 dengan kelas pembicara 3 :

   $y_{13} = (k(1) \times c1[1] + k(3) \times c3[1]) + b_2 = ((0.895)(1) + (0.925)(-1)) - 0 = -0.03$

   $y_{13} < 0$ => maka kelas pembicara 3 yang terpilih

   Bandingkan kelas pembicara 2 dengan kelas pembicara 3 :

   $y_{23} = (k(2) \times c2[2] + k(3) \times c3[2]) + b_3 = ((0.890)(1) + (0.925)(-1)) - 0 = -0.035$

   $y_{23} < 0$ => maka kelas pembicara 3 yang terpilih

Karena kelas pembicara 3 yang paling banyak terpilih, maka hasil identifikasi input 1 adalah kelas pembicara 3.

Hasil perhitungan selengkapnya untuk input 1 dapat dilihat pada tabel 2.18

Tabel 2.18 Contoh perhitungan identifikasi

Dimensi	Input 1 (A)	SV Kelas1 (B)	SV Kelas 2 (C)	SV Kelas 3 (D)	(Input 1-SVKelas 1)^2 $(A-B)^2$	(Input 1-SVKelas 2)^2 $(A-C)^2$	(Input 1-SVKelas 3)^2 $(A-D)^2$
1	0.050	-0.433	0.270	0.200	0.233	0.048	0.022
2	-0.062	-0.038	0.060	-0.067	0.001	0.015	0.000
3	-0.321	-0.277	-0.220	-0.537	0.002	0.010	0.046
4	-0.313	-0.318	-0.216	-0.316	0.000	0.009	0.000
5	-0.386	-0.733	0.134	0.178	0.120	0.270	0.319
6	0.570	0.912	0.071	0.090	0.117	0.249	0.230
7	0.611	0.734	0.136	0.547	0.015	0.225	0.004
8	0.434	0.103	0.445	0.440	0.109	0.000	0.000
9	-0.516	0.119	-0.542	-0.430	0.403	0.001	0.007
10	-0.420	-0.912	-0.041	-0.205	0.242	0.143	0.046
11	-0.269	-0.158	0.221	-0.286	0.012	0.240	0.000
12	0.001	0.273	-0.438	-0.509	0.074	0.193	0.260
Jumlah (s)					1.330	1.404	0.935
e^{-sx}					0.895	0.890	0.925

2.5.5 Metode Dekomposisi

Metode dekomposisi [Osuna et al, 1997] merupakan metode lain untuk *multi-class* SVM. Pada dasarnya metode ini merupakan metode *one-vs-one* yang menggunakan proses dekomposisi dengan formula *bound-constrained*. Tujuan dari metode ini adalah untuk solusi masalah klasifikasi yang besar.

Pada metode dekomposisi, dalam pembuatan model pembicara, digunakan proses iteratif di mana dalam setiap iterasi, data dibagi menjadi dua, yaitu B dan N di mana B merupakan *working set*. Variabel yang berhubungan dengan N akan tetap sedangkan *sub-problem* dari variabel yang berhubungan dengan B diminimalkan.

Grid search, pemodelan, dan identifikasi pembicara dengan metode dekomposisi menggunakan software BSVM [Hsu dan Lin, 2002]. Cara

pemodelan dan identifikasinya sama dengan metode *one-vs-one* tanpa dekomposisi.

2.6 Tinjauan Pustaka Penelitian Identifikasi Pembicara

Moreno dan Ho (2008) menggunakan Support Vector Machine (SVM) dengan kernel *probabilistic distance*. Metode ini diujicoba dengan korpus HUB4-96 yang berisi data berita *broadcast* dan korpus KING versi *Narrowband*. Kedua korpus ini digunakan untuk membandingkan unjuk kerja data kualitas *broadcast* (16kHz) dengan kualitas pembicaraan telepon (8kHz). Pada ujicoba, digunakan 50 pembicara dengan fitur suara MFCC 39 dimensi, yaitu terdiri atas 13 koefisien standar beserta turunan pertama dan kedua. Metode SVM yang digunakan adalah *one-vs-rest*, dengan 2 kernel *probabilistic distance*, yaitu Gaussian Mixture Model/Kullback-Leibler (GMM/KL) *divergence* dan full-covariance/Arithmetic Harmonic Sphericity (AHS) distance. Hasil akurasi identifikasi untuk korpus HUB4 adalah 83,8% dengan SVM GMM/KL dan 84.7% dengan SVM AHS. Dan akurasi identifikasi untuk korpus KING adalah 72,7% dengan SVM GMM/KL dan 79,7% dengan SVM AHS.

Yan Wang, Xueyan Liu, Yujuan Xing, dan Ming Li (2008) menggunakan metode *multi-reduced* SVM untuk mengurangi *noise* dan jumlah data. Metode reduksi yang digunakan adalah Principal Component Analysis (PCA) untuk mengurangi dimensi data dan kernel-based fuzzy clustering untuk mengurangi jumlah data dari setiap pembicara. Ujicoba dilakukan dengan menggunakan 30 pembicara (23 laki-laki dan 7 perempuan) dengan data *clean-speech* 11kHz, 16 bit yang berukuran 10-15 detik untuk pelatihan sistem dan 5 detik untuk testing. Sebelum diproses, bagian *silence* dibuang terlebih dahulu. Fitur yang digunakan adalah 12 koefisien MFCC dan turunan pertamanya, sehingga berjumlah 24 dimensi, yang diekstraksi dengan frame 30ms dan *overlap* 15ms yang kemudian dikalikan dengan hamming window untuk mencegah perubahan drastis pada akhir frame. Dengan PCA, dimensi MFCC berkurang menjadi 13. Parameter SVM yang digunakan adalah kernel RBF dengan metode *multi-class one-vs-one*. Hasil akurasi yang diperoleh dengan jumlah data untuk setiap pembicara sebanyak 25

adalah 98,5% untuk 10 pembicara, 97,6% untuk 20 pembicara, dan 98.1% untuk 30 pembicara.

Wan dan Campbell (2000) menggunakan SVM dengan kernel polynomial yang dinormalisasi. Ujicoba dilakukan dengan menggunakan YOHO database yang terdiri atas 138 pembicara, 69 di antaranya digunakan untuk pelatihan, dan 138 untuk testing. Pada saat pelatihan, data setiap pembicara dikuantisasi menjadi 100 *centroid* dengan k-means *clustering*. Fitur *speech* yang digunakan adalah 12 koefisien Linear Predictive Coding (LPC) beserta turunannya, sehingga berjumlah 24. Hasil kesalahan identifikasi yang diperoleh adalah 23,5% untuk polynomial orde 2; 7,3% untuk polynomial orde 4; 5,2% untuk polynomial orde 6; 4,4% untuk polynomial orde 8; dan 4,5% untuk polynomial orde 10.

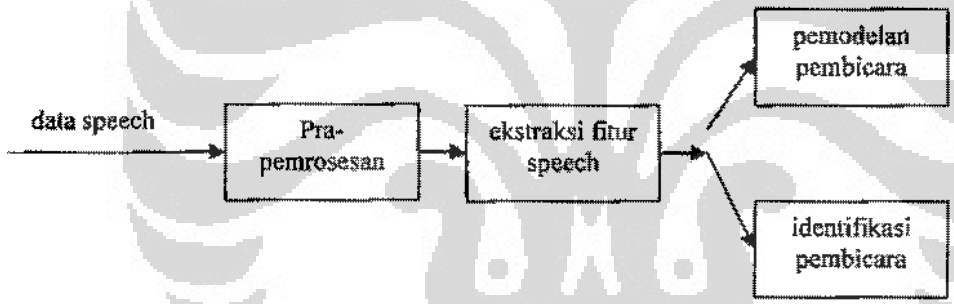
Agus Buono, Wisnu Jatmiko, dan Benyamin Kusumodiputro (2008) menggunakan HMM dengan 2D-MFCC untuk memproses data bispektrum. Bispektrum digunakan karena dianggap lebih tahan terhadap *noise* dibanding *power spectrum*. Ujicoba dilakukan dengan menggunakan 10 pembicara. Setiap pembicara mempunyai 80 data masing-masing berukuran 1,28 detik, 11kHz. Fitur data *speech* yang digunakan adalah 13 koefisien 2D-MFCC yang diekstraksi dengan 512 sampel per frame dan *overlap* 256 sampel. Hasil akurasi yang diperoleh dengan 13 koefisien 2D-MFCC adalah 98% untuk data tanpa *noise* dan 47% untuk data dengan penambahan *noise* 20dB. Sedangkan hasil akurasi yang diperoleh dengan 12 koefisien 2D-MFCC (tanpa koefisien pertama) adalah 99,4% untuk data tanpa *noise* dan 74,8% untuk data dengan *noise* 20dB.

BAB 3
EKSPERIMEN

Berikut adalah penjelasan mengenai rancangan sistem yang diteliti, korpus *speech*, peralatan yang digunakan, dan rancangan ujicoba yang dilakukan.

3.1 Rancangan Sistem

Rancangan sistem yang diusulkan pada penelitian ini terdiri atas, pra-pemrosesan, ekstraksi fitur *speech*, pemodelan pembicara, dan identifikasi pembicara, seperti yang terlihat pada gambar 3.1. Pra-pemrosesan dan ekstraksi fitur *speech* akan digunakan dalam pemodelan pembicara dan identifikasi pembicara.



Gambar 3.1 Garis besar rancangan sistem

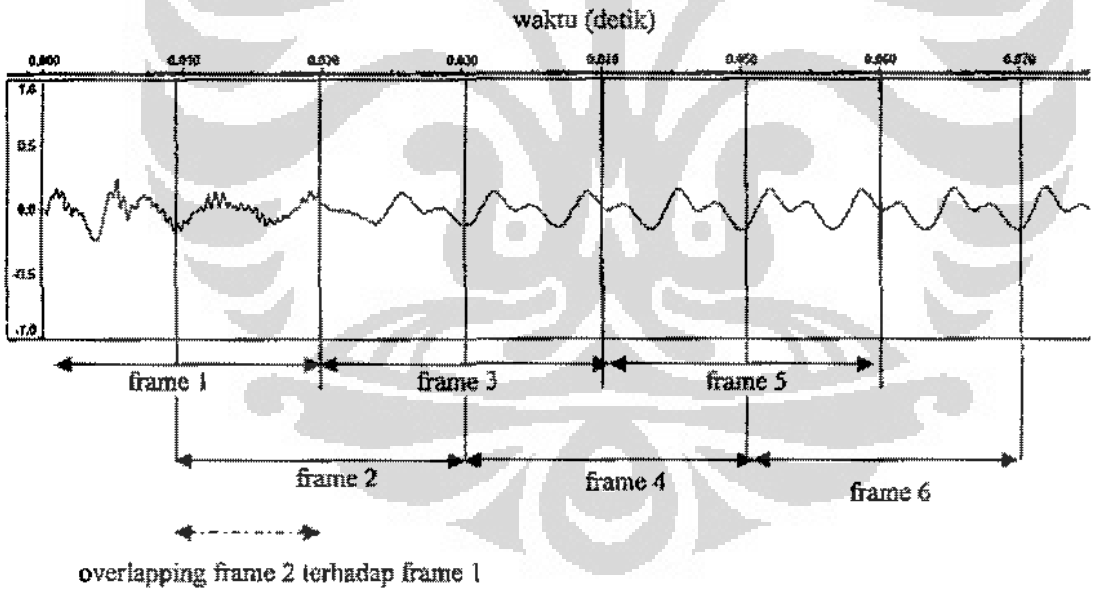
3.1.1 Pra-pemrosesan

Pra-pemrosesan yang digunakan dalam penelitian ini adalah *silence removal* dengan menggunakan *energy threshold* sebesar 0,2 dan *pre-emphasis* dengan faktor 0,95. Ukuran *energy threshold* dan faktor *pre-emphasis* ini merupakan ukuran yang umum digunakan. Pra-pemrosesan diimplementasikan dengan menggunakan matlab.

3.1.2 Ekstraksi Fitur Speech

Fitur *speech* yang digunakan adalah nilai rata-rata frame dari MFCC koefisien 0-11 pada suatu segmen waktu. Nilai rata-rata digunakan untuk

mengurangi jumlah data sehingga akan menghemat waktu pelatihan. Untuk mendapatkan karakteristik data yang cukup stabil, ekstraksi fitur *speech* dilakukan setelah data dibagi menjadi sub-data yang lebih kecil (frame), seperti terlihat pada gambar 3.2. Panjang frame umumnya adalah 20 milidetik dengan *overlapping* 30-50%, namun pada tesis ini panjang frame yang digunakan adalah 256 sampel atau 11 milidetik karena data yang akan digunakan adalah nilai rata-rata frame pada suatu segmen waktu dan bukan nilai per-framenya. *Overlapping* frame sebesar 128 sampel digunakan untuk meminimalisasi adanya informasi yang hilang antar frame. Dan untuk mencegah adanya perubahan drastis pada akhir frame, maka sinyal *speech* akan dikalikan dengan fungsi window, yaitu dalam hal ini fungsi window hamming. Fitur MFCC akan diperoleh setelah sinyal *speech* difilter dengan menggunakan mel filter. Jumlah mel filter yang digunakan pada tesis ini adalah sebanyak jumlah koefisien yang akan digunakan, yaitu 12.

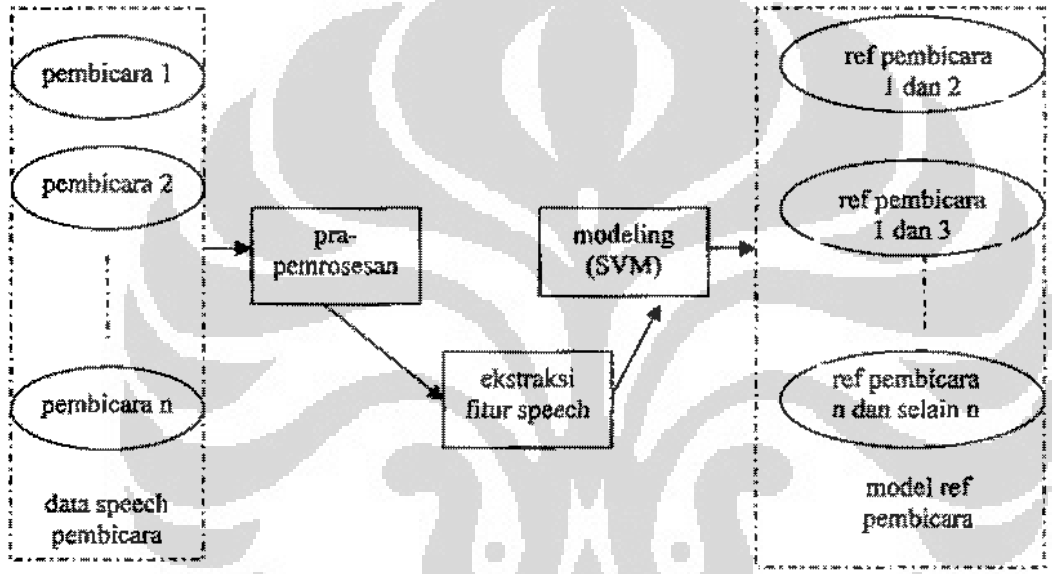


Gambar 3.2 Short-time analysis dengan panjang frame 256 sampel dan overlapping frame 128 sampel

3.1.3 Pemodelan Pembicara

Pemodelan pembicara dilakukan melalui pelatihan sistem. Seperti terlihat pada gambar 3.3, data setiap pembicara akan diproses dahulu dengan membuang

silence dan memperbaiki rasio antara sinyal dengan *noise*. Selanjutnya dilakukan ekstraksi fitur suara. Hasil ekstraksi fitur suara, akan digunakan untuk pemodelan pembicara (pelatihan sistem). Metode klasifikasi yang digunakan adalah SVM *multi-class one-vs-one* dengan kernel RBF. SVM *multi-class one-vs-one* digunakan karena memberikan hasil akurasi yang lebih baik daripada *one-vs-rest* [Schmidt dan Gish, 1996; Hsu dan Lin, 2002]. Sedangkan kernel RBF merupakan kernel yang umum digunakan dalam SVM.



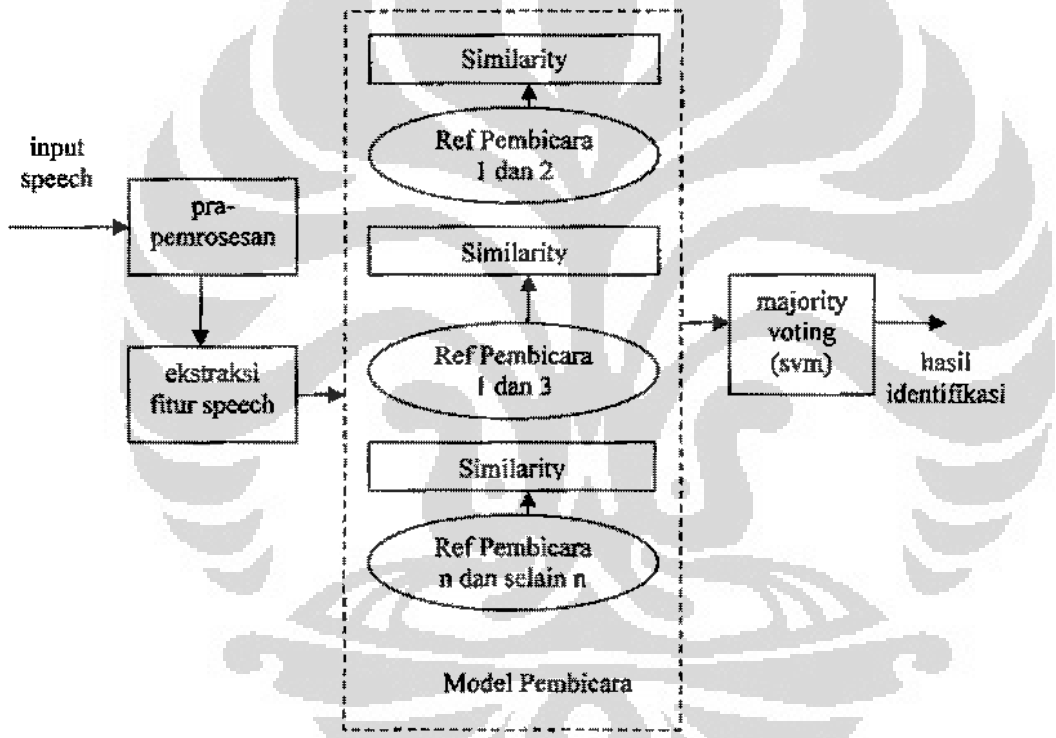
Gambar 3.3 Skema pemodelan pembicara

Pemodelan pembicara dengan menggunakan metode SVM *multi-class one-vs-one* dilakukan dengan cara membandingkan setiap kelas dengan setiap kelas lainnya. Setiap kelas akan mempunyai (n-1) model data atau secara keseluruhan akan diperoleh n*(n-1)/2 model untuk satu kali pelatihan sistem dengan n kelas data.

Parameter SVM C dan γ diperoleh melalui metode *grid search* dengan interval pencarian C adalah $2^{-3} - 2^{13}$ dan interval pencarian γ adalah $2^{-3} - 2^{3}$, karena berdasarkan beberapa ujicoba yang telah dilakukan sebelumnya, interval ini memberi hasil akurasi yang cukup baik.

3.1.4 Identifikasi Pembicara

Seperti terlihat pada gambar 3.4, data *speech* akan melalui pra-pemrosesan terlebih dahulu kemudian akan dilakukan ekstraksi fitur speech. Fitur speech yang diperoleh akan dibandingkan dengan data model pembicara yang dihasilkan oleh pemodelan pembicara. Karena setiap model pembicara merepresentasikan dua kelas, maka ketika data input dibandingkan dengan satu model, hasilnya adalah data input lebih mirip kelas pertama atau kelas kedua. Setelah semua model dibandingkan dengan data input, hasil identifikasi adalah kelas yang paling banyak kemiripannya dengan data input (*majority voting*).



Gambar 3.4 Skema identifikasi pembicara

3.2 Korpus Speech

Data *speech* yang digunakan dalam sistem ini adalah data berbahasa Indonesia yang diambil dari beberapa stasiun televisi dan radio, yaitu Radio

Republik Indonesia (RRI)[1], Televisi Republik Indonesia (TVRI)[2], TVOne[3], Surya Citra Televisi (SCTV)[4], MetroTV[5], British Broadcasting Corporation (BBC) Indonesia[6], dan Voice of America (VOA) Indonesia[7].

Data yang diperoleh berasal dari sesi berita dan wawancara. Cara pembuatan data pembicara untuk pemodelan data adalah sebagai berikut:

1. Satu sesi berita atau wawancara direkam secara manual atau diunduh melalui situs stasiun radio atau televisi.
2. Setiap sesi berita akan dipotong/disegmen berdasarkan pembicara pada segmen berita tersebut. Setiap segmen data hanya akan berisi data *speech* dari satu pembicara.
3. Setiap segmen *speech* hasil langkah 2 akan diberi label atau nama sesuai nama pembicara, identitas pembicara, asal data, dan tanggal kemunculan data.

Pemberian label pada data sangat penting dan harus sesingkat mungkin. Yang perlu diperhatikan pada saat pembuatan label adalah:

1. Cukup mewakili identitas pembicara dan asal data, sehingga ketika perlu dicari data aslinya dapat dengan mudah ditemukan.
2. Adanya kemungkinan data pembicara dengan sumber yang berbeda.

Berdasarkan aturan penamaan di atas, maka penamaan dilakukan dengan cara:

1. Setiap jenis label akan dipisahkan dengan tanda garis bawah (_).
2. Label jenis pertama adalah nama dari si pembicara.
3. Label jenis kedua adalah profesi atau identitas unik lainnya dari si pembicara
4. Label jenis ketiga adalah asal data si pembicara.
5. Label jenis keempat adalah tanggal kemunculan data tersebut.

Contoh label data dari seseorang dengan inisial xy, identitas anggota dpr, muncul di rri pada tanggal 1 april 2008 adalah xy_dpr_rri_010408.

[1] www.pro3rri.com
[2] www.tvri.co.id
[3] www.tvone.co.id
[4] www.liputan6.com
[5] www.metrotvnews.com
[6] www.bbc.co.uk/indonesian/
[7] www.voanews.com/indonesian/

Sistem hanya akan mengenali dua jenis label yang pertama, sehingga ketika ada pembicara yang memiliki data dengan sumber yang berbeda masih akan dikenali sebagai satu pembicara.

Jumlah data yang digunakan dalam tesis ini adalah 4,08 GB atau setara dengan 26 jam, yang terdiri atas 715 pembicara (549 laki-laki dan 166 perempuan). Detail distribusi data dapat dilihat pada tabel 3.1

Tabel 3.1 Detail data pembicara berdasarkan sumber dan gender

	Jumlah per sumber pembicara						
Gender	**RRI**	**TVRI**	**SCTV**	**MetroTV**	**TVOne**	**BBC Indonesia**	**VOA Indonesia**
L	429		7	62	15	48	2
P	132	3	6	12	1	11	2
Total	561	3	13	74	16	59	4

Setiap pembicara mempunyai jumlah data yang bervariasi, dengan durasi minimal untuk setiap pembicara adalah 40 detik dan maksimal 300 detik. Data ini berasal dari satu sesi hingga sepuluh sesi berita atau wawancara. Sumber data setiap pembicara juga bervariasi, tidak selalu berasal dari stasiun radio atau televisi yang sama. Misalnya pembicara pertama mempunyai data yang berasal dari RRI dan SCTV dengan durasi masing-masing 1 menit, data pembicara kedua berasal dari RRI dan MetroTV dengan durasi masing-masing 30 detik dan 45 detik, sedangkan data pembicara ketiga hanya berasal dari TVRI saja dengan durasi 40 detik.

Semua data yang digunakan merupakan data audio wav yang di-*sampling* dengan frekuensi 22kHz, 16 bit.

3.3 Peralatan

Peralatan yang digunakan dalam implementasi tesis ini adalah:

1. SIL Speech Analyzer untuk merekam dan mengedit data *speech*.
2. FLV Audio Extractor untuk memisahkan data audio dalam suatu data video flv.

3. Voicebox, matlab toolbox untuk pemrosesan *speech*, dari Department of Electrical and Electronic Engineering of Imperial College, UK [Brooks, 1997]
4. LibSVM, software SVM untuk pemodelan dan identifikasi pembicara [Chang dan Lin, 2001]
5. BSVM, software SVM untuk pemodelan dan identifikasi pembicara dengan metode dekomposisi [Hsu dan Lin, 2002]
6. Matlab, untuk pra-pemrosesan data *speech*
7. Komputer portabel dengan prosesor 1.83GHz, memory 2 GB, dan sistem operasi windows.

3.4 Ujicoba

Ujicoba dilakukan dengan menggunakan korpus *speech* seperti yang telah disebutkan pada bagian 3.2. Seluruh data yang digunakan diacak kemudian dibagi dua dengan perbandingan 70% untuk pemodelan pembicara (pelatihan sistem) dan 30% untuk identifikasi pembicara (testing).

Tabel 3.2 Pembagian data untuk ujicoba

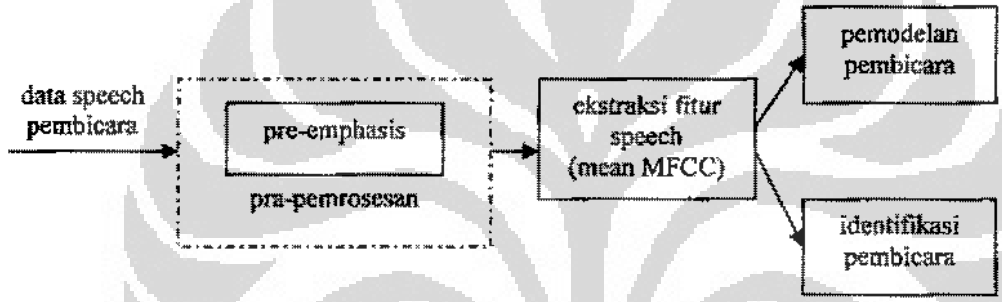
Segmen (detik)	250 pembicara			500 pembicara			715 pembicara		
	Total data	Data training (70%)	Data testing (30%)	Total data	Data training (70%)	Data testing (30%)	Total data	Data training (70%)	Data testing (30%)
5	7085	4960	2125	13461	9423	4038	19187	13431	5756
10	3412	2389	1023	6485	4540	1945	9244	6471	2773
15	2186	1531	655	4154	2908	1246	5935	4155	1780

Setiap data pembicara disegmen menjadi 5, 10, dan 15 detik untuk mengetahui seberapa pendek informasi data *speech* dapat digunakan untuk memperoleh hasil identifikasi yang baik. Karena untuk suatu data berita, biasanya setiap orang tidak berbicara lama, hanya sekitar 5 sampai 15 detik. Penggunaan jumlah pembicara juga dilakukan secara bertahap, mulai dari 250, 500, kemudian 715 pembicara. Hal ini dilakukan untuk mengetahui pengaruh penambahan

jumlah pembicara terhadap akurasi identifikasi. Jumlah data yang diperoleh dengan segmentasi 5, 10, dan 15 detik dan jumlah pembicara 250, 500, dan 715 yang akan digunakan dalam ujicoba dapat dilihat pada tabel 3.2.

Ujicoba dilakukan dengan mengubah beberapa parameter sistem. Sistem yang diusulkan pada penelitian ini akan diujicoba pada ujicoba 2. Berikut rancangan sistem pada ujicoba yang dilakukan. Parameter sistem yang tidak disebutkan, berarti sama dengan rancangan sistem pada bagian 3.1.

3.4.1 Ujicoba 1



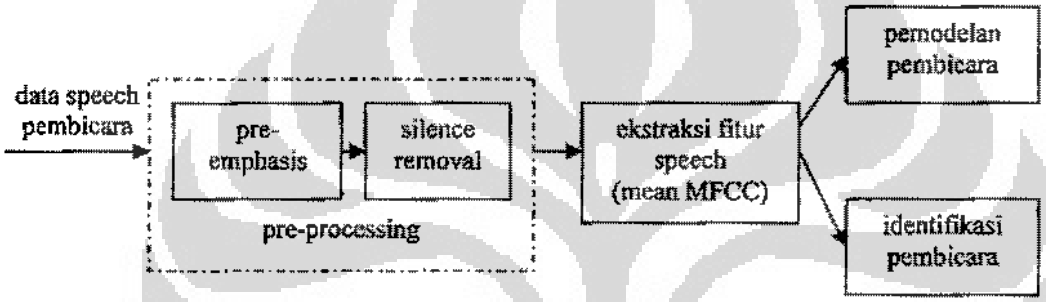
Gambar 3.5 Sistem pada ujicoba 1

Seperti terlihat pada gambar 3.5, ujicoba 1 hanya menggunakan satu metode pra-pemrosesan, yaitu *pre-emphasis*. *Silence removal* tidak digunakan, sehingga data yang digunakan masih mengandung *silence*. Ujicoba ini dilakukan untuk melihat pengaruh *silence* terhadap akurasi identifikasi. Fitur *speech* yang digunakan adalah nilai rata-rata frame dari MFCC koefisien 0-11 pada suatu segmen waktu. Nilai rata-rata frame dihitung dengan cara menghitung MFCC koefisien 0-11 pada suatu segmen waktu, kemudian masing-masing dimensi (koefisien) akan dicari nilai rata-ratanya. Dengan demikian, setiap segmen waktu hanya akan mempunyai satu data MFCC koefisien 0-11.

3.4.2 Ujicoba 2

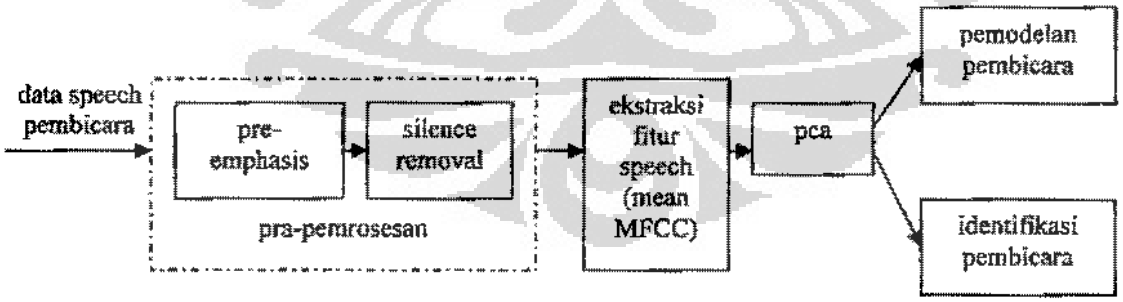
Pada ujicoba 2, kedua metode pra-pemrosesan, yaitu *pre-emphasis* dan *silence removal* digunakan, seperti terlihat pada gambar 3.6. Sistem ujicoba kedua

ini sama dengan sistem yang diusulkan pada tesis ini. Perbedaan dengan sistem pada ujicoba 1 adalah adanya penambahan *silence removal*, sehingga bagian *silence* dari data akan dibuang. Fitur *speech* yang digunakan adalah nilai rata-rata frame dari MFCC koefisien 0-11 pada suatu segmen waktu. Nilai rata-rata frame dihitung dengan cara menghitung MFCC koefisien 0-11 pada suatu segmen waktu, kemudian masing-masing dimensi (koefisien) akan dicari nilai rata-ratanya. Dengan demikian, setiap segmen waktu hanya akan mempunyai satu data MFCC koefisien 0-11.



Gambar 3.6 Sistem pada ujicoba 2

3.4.3 Ujicoba 3

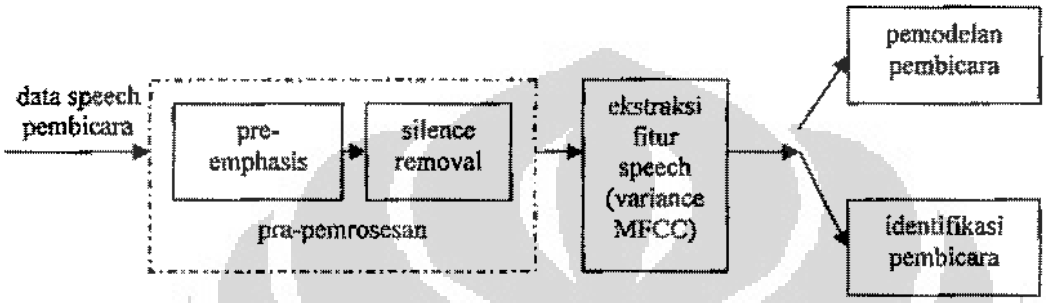


Gambar 3.7 Sistem pada ujicoba 3

Seperti terlihat pada gambar 3.7, sistem pada ujicoba 3 menggunakan kedua metode pra-pemrosesan, yaitu *pre-emphasis* dan *silence removal*. PCA juga digunakan untuk melihat apakah dimensi data bisa dikurangi tanpa memperburuk

akurasi identifikasi yang dihasilkan. Dengan pengurangan dimensi data, maka akan mengurangi waktu pelatihan sistem. Seperti halnya ujicoba 1 dan 2, fitur *speech* yang digunakan adalah nilai rata-rata frame dari MFCC koefisien 0-11.

3.4.4 Ujicoba 4

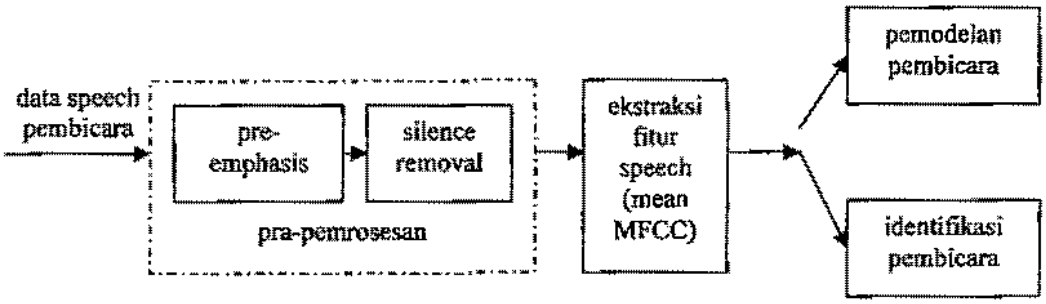


Gambar 3.8 Sistem pada ujicoba 4

Seperti terlihat pada gambar 3.8, pada dasarnya sistem yang digunakan pada ujicoba 4 sama dengan sistem pada ujicoba 2. Sistem menggunakan kedua metode pra-pemrosesan, yaitu *pre-emphasis* dan *silence removal*. Namun, fitur audio yang digunakan bukan nilai rata-rata frame dari MFCC pada suatu segmen waktu, melainkan nilai varians frame dari MFCC pada suatu segmen waktu. Hal ini dilakukan untuk melihat apakah nilai varians juga dapat menghasilkan akurasi yang sama baik seperti nilai rata-rata. Nilai varians frame dihitung dengan cara menghitung MFCC koefisien 0-11 pada suatu segmen waktu, kemudian masing-masing dimensi (koefisien) akan dicari nilai variansnya. Dengan demikian, setiap segmen waktu hanya akan mempunyai satu data MFCC koefisien 0-11.

3.4.5 Ujicoba 5

Pada ujicoba 5, sistem yang digunakan adalah sama dengan sistem yang diusulkan (gambar 3.9, ujicoba 2). Sistem menggunakan kedua metode pra-pemrosesan, yaitu *pre-emphasis* dan *silence removal*. Fitur *speech* yang digunakan adalah nilai rata-rata frame dari MFCC koefisien 0-11. Namun metode klasifikasi dan identifikasi yang digunakan adalah svm *multi-class one-vs-one* dengan metode dekomposisi.



Gambar 3.9 Sistem pada ujicoba 5

BAB 4
HASIL UJICOBA DAN ANALISA

Sistem identifikasi pembicara yang diusulkan pada tesis ini terdiri atas pra-pemrosesan, ekstraksi fitur *speech*, pemodelan pembicara, dan identifikasi pembicara. Pra-pemrosesan terdiri atas dua proses, yaitu *pre-emphasis* dan *silence removal*. Fitur *speech* yang digunakan adalah rata-rata nilai frame MFCC koefisien 0-11 dalam suatu segmen waktu. Pemodelan dan identifikasi pembicara menggunakan metode SVM *multi-class one-vs-one* dengan kernel RBF.

Ujicoba dilakukan dengan menggunakan 5 skenario ujicoba seperti yang telah dijelaskan pada bagian 3.4. Ujicoba pertama hanya menggunakan satu metode pra-pemrosesan, yaitu *pre-emphasis*. Ujicoba kedua hampir sama dengan ujicoba pertama, yaitu dengan menambahkan *silence removal* pada saat pra-pemrosesan. Ujicoba ketiga menggunakan PCA untuk mengurangi jumlah dimensi dari data. Ujicoba keempat menggunakan nilai varians frame dari MFCC pada suatu segmen waktu sebagai pengganti dari nilai rata-rata frame dari MFCC pada suatu segmen waktu. Ujicoba kelima menggunakan SVM *multi-class* metode dekomposisi sebagai pengganti metode SVM *multi-class one-vs-one* tanpa metode dekomposisi. Hasil ujicoba 1, 3, 4, dan 5 akan dibandingkan dengan hasil sistem identifikasi yang diusulkan (hasil ujicoba 2).

Berikut hasil ujicoba yang telah dilakukan dan analisanya.

4.1 Hasil Ujicoba

Hasil ujicoba pada dasarnya terdiri atas parameter SVM, sebagai hasil dari *grid search* yang akan digunakan dalam pemodelan pembicara, dan identifikasi pembicara. Perbandingan data antara pelatihan sistem dan identifikasi dalam seluruh ujicoba adalah 70%:30%.

4.1.1 Ujicoba 1

Ujicoba 1 hanya menggunakan satu metode pra-pemrosesan, yaitu *pre-emphasis*. Dengan demikian, data yang digunakan dalam pemodelan pembicara merupakan gabungan dari *speech* dan *silence*.

Tabel 4.1 Parameter SVM ujicoba 1

Segmen (detik)	Parameter SVM								
	250 pembicara			500 pembicara			715 pembicara		
	C	γ	Jumlah support vector	C	γ	Jumlah support vector	C	γ	Jumlah support vector
5	8	2	4069	16	2	8222	8	4	11633
10	8	2	2169	16	2	4252	16	4	6013
15	16	2	1450	8	4	2757	32	1	4087

Parameter SVM pada tabel 4.1 diperoleh melalui *grid search*. Pada tabel tersebut terlihat bahwa dengan nilai C dan γ yang relatif kecil, sudah dapat menghasilkan akurasi identifikasi yang cukup baik. Hasil akurasi dari ujicoba 1 dapat dilihat pada tabel 4.2.

Tabel 4.2 Hasil akurasi identifikasi ujicoba 1

Segmen (detik)	Akurasi Identifikasi (%)		
	250 pembicara	500 pembicara	715 pembicara
5	96.52	95.72	94.87
10	96.87	96.81	95.53
15	95.57	94.94	93.88

Akurasi yang paling baik diperoleh pada saat penggunaan 250 pembicara dengan segmen waktu 10 detik, yaitu 96,87% yang diikuti oleh penggunaan 500 pembicara dengan segmen waktu 10 detik, yaitu 96,81%. Secara keseluruhan nilai akurasi yang diperoleh sudah cukup baik. Pada penggunaan segmen waktu yang tetap dengan jumlah pembicara yang berbeda, nilai akurasi yang diperoleh cenderung turun seiring dengan bertambahnya jumlah pembicara, meskipun tidak terlalu berbeda. Begitu pula dengan penggunaan segmen waktu yang semakin panjang dengan jumlah pembicara tetap, akurasi akan bertambah sampai dengan segmen waktu 10 detik, namun kemudian berkurang pada segmen waktu 15 detik.

4.1.2 Ujicoba 2

Pada sistem ujicoba 2, *silence* sudah dibuang dari data, sehingga data hanya berisi *speech* dari pembicara. Hasil parameter SVM yang diperoleh melalui *grid search* dapat dilihat pada tabel 4.3.

Tabel 4.3 Parameter SVM ujicoba 2

Segmen (detik)	Parameter SVM								
	250 pembicara			500 pembicara			715 pembicara		
	C	γ	Jumlah support vector	C	γ	Jumlah support vector	C	γ	Jumlah support vector
5	16	1	4277	8	2	8190	16	2	12023
10	8	2	2137	16	2	4243	8	4	5995
15	256	1	1461	32	2	2782	64	1	4085

Pada tabel 4.3 terlihat bahwa nilai C dan γ cukup kecil, namun akurasi yang dihasilkan sudah cukup baik. Hasil akurasi identifikasi yang diperoleh dapat dilihat pada tabel 4.4.

Tabel 4.4. Hasil akurasi identifikasi ujicoba 2

Segmen (detik)	Akurasi Identifikasi (%)		
	250 pembicara	500 pembicara	715 pembicara
5	95.53	95.52	94.09
10	98.14	97.58	95.89
15	95.88	94.94	95.56

Secara keseluruhan hasil yang diperoleh cukup baik. Hasil yang paling baik diperoleh pada penggunaan 250 pembicara dengan segmen waktu 10 detik, yaitu sebesar 98,14% dan diikuti oleh penggunaan 500 pembicara dengan segmen waktu 10 detik, yaitu sebesar 97,58%. Pada penggunaan segmen waktu yang tetap

dengan jumlah pembicara yang berbeda, nilai akurasi yang diperoleh cenderung turun seiring dengan bertambahnya jumlah pembicara. Begitu pula dengan penggunaan segmen waktu yang semakin panjang dengan jumlah pembicara tetap, akurasi akan bertambah sampai dengan segmen waktu 10 detik, namun kemudian berkurang pada segmen waktu 15 detik.

4.1.3 Ujicoba 3

Pada ujicoba 3, dilakukan pengurangan jumlah dimensi data MFCC dengan menggunakan PCA, sebelum digunakan untuk pemodelan dan identifikasi pembicara. Nilai eigen yang diperoleh dapat dilihat pada tabel 4.5. Pada tabel tersebut terlihat bahwa semua dimensi mempunyai nilai eigen masing-masing. Nilai eigen terbesar dimiliki oleh dimensi pertama (koefisien 0 MFCC), yaitu sekitar 8. Nilai eigen yang diperoleh semakin berkurang sesuai urutan dimensi data, sehingga nilai eigen paling kecil dimiliki oleh dimensi data ke-12 (koefisien 11 MFCC). Nilai eigen terkecil adalah kurang lebih 0,01.

Pada tabel 4.5 juga terlihat bahwa secara keseluruhan, nilai eigen setiap dimensi untuk 250 pembicara hampir sama dengan 500 pembicara dan 715 pembicara, sehingga ketika dilakukan reduksi jumlah dimensi, maka jumlah nilai eigen yang dapat diabaikan adalah sama, yang berarti jumlah dimensi yang dikurangi juga sama.

Pengurangan jumlah dimensi dalam PCA dapat dilakukan dengan beberapa cara. Menurut kriteria Kaiser [Kaiser, 1960], dimensi yang akan digunakan adalah yang mempunyai nilai eigen lebih besar dari satu, sedangkan yang kurang dari satu dapat diabaikan. Kriteria lainnya adalah *scree* test [Cattel, 1966], di mana nilai eigen yang diabaikan adalah nilai yang penurunannya sudah tidak terlalu tajam atau hampir sama.

Jika kriteria Kaiser yang digunakan, maka untuk nilai eigen pada tabel 4.5, hanya dua nilai eigen pertama yang digunakan, yaitu 8 dan 1,6, sisanya diabaikan. Sedangkan jika menggunakan *scree* test, yang akan digunakan adalah 3 nilai eigen pertama karena nilai eigen selanjutnya turun tidak terlalu jauh satu sama lain. Pada kedua kriteria tersebut, dimensi data yang digunakan terlalu sedikit, sehingga tidak digunakan.

Tabel 4.5 Nilai eigen data ujicoba 3, yang ditebalkan dan dimiringkan adalah posisi nilai eigen ke-10 dan 12.

Segmen (detik)	Nilai eigen		
	250 pembicara	500 pembicara	715 pembicara
5	8.2861	8.2330	8.6490
	1.6774	1.6103	1.5748
	0.9706	0.9322	0.8849
	0.6463	0.6412	0.6479
	0.4244	0.4469	0.4414
	0.1485	0.1488	0.1564
	0.0980	0.1072	0.1118
	0.0808	0.0791	0.0795
	0.0465	0.0517	0.0501
	0.0272	***0.0303***	***0.0306***
	0.0196	0.0193	0.0191
	0.0099	***0.0099***	***0.0100***
10	8.0735	8.1343	8.5540
	1.6299	1.5650	1.5313
	0.9628	0.9221	0.8713
	0.6130	0.6175	0.6343
	0.4128	0.4369	0.4289
	0.1397	0.1408	0.1493
	0.0902	0.0982	0.1033
	0.0749	0.0736	0.0744
	0.0433	0.0478	0.0470
	0.0248	***0.0281***	***0.0286***
	0.0173	0.0172	0.0173
	0.0088	***0.0086***	***0.0088***
15	8.0751	8.0839	8.3653
	1.5931	1.5536	1.5117
	0.9556	0.9101	0.8695
	0.5984	0.6268	0.6120
	0.4044	0.4339	0.4211
	0.1362	0.1373	0.1456
	0.0858	0.0949	0.1003
	0.0723	0.0706	0.0703
	0.0426	0.0462	0.0456
	0.0236	***0.0282***	***0.0279***
	0.0171	0.0164	0.0164
	0.0080	***0.0084***	***0.0084***

Jumlah dimensi dikurangi dengan mencari nilai eigen yang cukup kecil untuk dapat diabaikan. Pada dimensi ke-10, nilai eigen pada posisi tersebut dianggap cukup kecil sehingga mungkin bisa diabaikan, maka dilakukan ujicoba menggunakan 9 dimensi data pertama, sedangkan dimensi 10-12 diabaikan.

Parameter SVM hasil *grid search* untuk data dengan 9 dimensi dapat dilihat pada tabel 4.6. Pada tabel tersebut terlihat bahwa parameter yang diperoleh untuk C dan γ cukup kecil, namun akurasi yang dihasilkan sudah cukup baik.

Tabel 4.6 Parameter SVM ujicoba 3 dengan 9 dimensi data

Segmen (detik)	Parameter SVM								
	250 pembicara			500 pembicara			715 pembicara		
	C	γ	Jumlah support vector	C	γ	Jumlah support vector	C	γ	Jumlah support vector
5	16	2	4161	64	1	8475	8	4	11893
10	8	2	2161	8	4	4152	32	2	6137
15	128	1	1467	8	4	2756	16	4	3996

Hasil akurasi identifikasi untuk penggunaan 9 dimensi data dapat dilihat pada tabel 4.7. Secara keseluruhan hasil yang diperoleh cukup baik. Hasil yang paling baik diperoleh pada penggunaan 250 pembicara dengan segmen waktu 10 detik, yaitu sebesar 93,84 % dan diikuti oleh penggunaan 250 pembicara dengan segmen waktu 15 detik, yaitu sebesar 91,91%.

Tabel 4.7 Hasil akurasi identifikasi ujicoba 3 dengan 9 dimensi data

Segmen (detik)	Akurasi Identifikasi (%)		
	250 pembicara	500 pembicara	715 pembicara
5	90.96	89.33	86.14
10	93.84	90.08	89.47
15	91.91	89.33	89.38

Hasil akurasi identifikasi dengan 9 dimensi hanya 86-94% cukup jauh di bawah hasil akurasi ujicoba 2, yaitu 94-98%, sehingga belum bisa digunakan.

Oleh karena itu, dicoba dengan 11 dimensi data untuk mengetahui akurasi identifikasinya.

Parameter SVM hasil *grid search* untuk data dengan 11 dimensi dapat dilihat pada tabel 4.8. Pada tabel tersebut terlihat bahwa parameter yang diperoleh untuk C dan γ cukup kecil, namun akurasi yang dihasilkan sudah cukup baik.

Tabel 4.8 Parameter SVM ujicoba 3 dengan 11 dimensi data

Segmen (detik)	Parameter SVM								
	250 pembicara			500 pembicara			715 pembicara		
	C	γ	Jumlah support vector	C	γ	Jumlah support vector	C	γ	Jumlah support vector
5	32	1	4287	8	4	8108	16	1	12429
10	8	2	2174	256	1	4342	16	2	6184
15	32	1	1472	64	0,25	2861	16	2	4051

Hasil akurasi identifikasi untuk 11 dimensi dapat dilihat pada tabel 4.9. Secara keseluruhan hasil yang diperoleh cukup baik. Hasil yang paling baik diperoleh pada penggunaan 250 pembicara dengan segmen waktu 10 detik, yaitu sebesar 95,99 % dan diikuti oleh penggunaan 500 pembicara dengan segmen waktu 10 detik, yaitu sebesar 94.55%.

Tabel 4.9 Hasil akurasi identifikasi ujicoba 3 dengan 11 dimensi data

Segmen (detik)	Akurasi Identifikasi (%)		
	250 pembicara	500 pembicara	715 pembicara
5	94.02	92.97	90.90
10	95.99	94.55	93.65
15	93.89	91.33	93.76

Hasil akurasi identifikasi untuk 9 dimensi data dan 11 dimensi data pada tabel 4.7 dan 4.9 sama-sama menunjukkan bahwa pada penggunaan segmen waktu yang tetap dengan jumlah pembicara yang berbeda, nilai akurasi yang diperoleh cenderung turun seiring dengan bertambahnya jumlah pembicara. Begitu pula dengan penggunaan segmen waktu yang semakin panjang dengan jumlah pembicara tetap, akurasi akan bertambah sampai dengan segmen waktu 10 detik, namun kemudian berkurang pada segmen waktu 15 detik.

4.1.4 Ujicoba 4

Pada ujicoba 4, sistem tidak menggunakan nilai rata-rata frame dari MFCC pada suatu segmen waktu, melainkan nilai variansnya. Parameter SVM yang diperoleh melalui *grid search* dapat dilihat pada tabel 4.10.

Tabel 4.10 Parameter svm ujicoba 4

Segmen (detik)	Parameter SVM								
	250 pembicara			500 pembicara			715 pembicara		
	C	γ	Jumlah support vector	C	γ	Jumlah support vector	C	γ	Jumlah support vector
5	4	2	4775	4	2	9234	4	2	13209
10	8	2	2330	16	2	4464	16	1	6407
15	32	1	1508	128	1	2888	16	1	4130

Dengan interval pencarian (*grid search*) C sebesar $2^{-3} - 2^{13}$ dan interval pencarian γ sebesar $2^{-3} - 2^{3}$, hasil akurasi identifikasi paling baik diperoleh pada parameter SVM C dan γ yang relatif kecil. Namun hasil identifikasi yang diperoleh kurang memuaskan, cukup jauh di bawah hasil identifikasi ujicoba 2. Hasil identifikasi keseluruhan untuk ujicoba 3 dapat dilihat pada tabel 4.11.

Tabel 4.11 Hasil akurasi identifikasi ujicoba 4

Segmen (detik)	Akurasi Identifikasi (%)		
	250 pembicara	500 pembicara	715 pembicara
5	67.95	58.62	54.66
10	79.67	74.04	70.36
15	82.60	77.37	74.72

Hasil yang paling baik diperoleh pada penggunaan 250 pembicara dengan segmen waktu 15 detik, yaitu sebesar 82,60% dan diikuti oleh penggunaan 250 pembicara dengan segmen waktu 10 detik, yaitu sebesar 79,67%. Pada penggunaan segmen waktu yang tetap dengan jumlah pembicara yang berbeda, nilai akurasi yang diperoleh cenderung turun seiring dengan bertambahnya jumlah pembicara. Sedangkan pada penggunaan segmen waktu yang semakin panjang dengan jumlah pembicara tetap, akurasi akan semakin bertambah.

4.1.5 Ujicoba 5

Pada ujicoba 5 pemodelan dan identifikasi pembicara dilakukan dengan menggunakan SVM metode dekomposisi. Parameter SVM hasil *grid search* dapat dilihat pada tabel 4.12.

Tabel 4.12 Parameter SVM ujicoba 5

Segmen (detik)	Parameter SVM								
	250 pembicara			500 pembicara			715 pembicara		
	C	γ	Jumlah support vector	C	γ	Jumlah support vector	C	γ	Jumlah support vector
5	8	2	4085	16	1	8477	8	4	11618
10	8	2	2126	32	1	4336	32	1	6255
15	16	1	1462	8	2	2776	64	1	4085

Pada tabel tersebut terlihat bahwa nilai C dan γ yang diperoleh relatif kecil. Hasil akurasi identifikasi yang diperoleh secara keseluruhan cukup baik, seperti terlihat pada tabel 4.13.

Tabel 4.13 Hasil akurasi identifikasi ujicoba 5

Segmen (detik)	Akurasi Identifikasi (%)		
	250 pembicara	500 pembicara	715 pembicara
5	95.67	95.37	94.25
10	98.14	97.17	95.24
15	88.18	94.70	95.62

Akurasi identifikasi paling baik diperoleh pada penggunaan 250 pembicara dan 500 pembicara dengan segmen waktu 10 detik, yaitu masing-masing sebesar 98,14% dan 97,17%. Pada penggunaan segmen waktu yang tetap dengan jumlah pembicara yang berbeda, nilai akurasi yang diperoleh cenderung turun seiring dengan bertambahnya jumlah pembicara. Begitu pula dengan penggunaan segmen waktu yang semakin panjang dengan jumlah pembicara tetap, akurasi akan bertambah sampai dengan segmen waktu 10 detik, namun kemudian berkurang pada segmen waktu 15 detik.

4.1.6 Waktu Eksekusi Grid Search

Pada ujicoba 1, 2, 3, 4,dan 5, *grid search* digunakan untuk mencari parameter SVM C dan γ yang terbaik. Interval C yang digunakan adalah 2^{-3} - 2^{13} dan interval γ adalah 2^{-3} - 2^{3}. Untuk mempercepat waktu komputasi, *grid search* dilakukan dengan menggunakan 2 *thread*.

Detail waktu eksekusi *grid search* uji coba 1, 2, 4, dan 5 dapat dilihat pada tabel 4.14. Pada tabel tersebut terlihat bahwa waktu yang diperlukan grid search paling besar adalah pada penggunaan 715 pembicara dengan segmen waktu 5 detik, yaitu selama 13 jam dengan 2 *thread* atau 26 jam dengan 1 thread. Pada sesi tersebut data yang digunakan adalah yang paling besar, yaitu 13431 data dengan 715 pembicara. Seiring dengan berkurangnya jumlah data dan jumlah pembicara

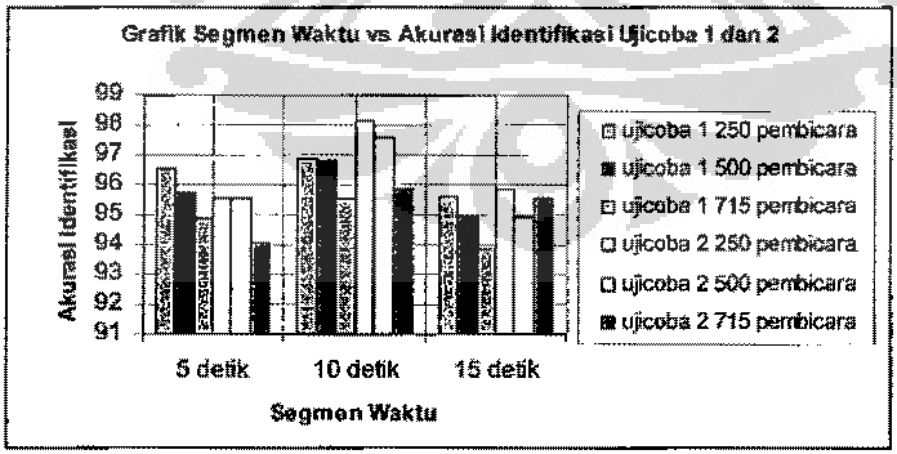
yang akan diklasifikasi, maka waktu eksekusi *grid search* semakin pendek. Dengan demikian waktu terpendek adalah pada penggunaan 250 pembicara dengan segmen waktu 15 detik, yaitu sebesar 1 jam dengan 2 *thread* atau 2 jam dengan 1 *thread*, di mana data yang digunakan adalah yang paling sedikit, yaitu 1531 data.

Tabel 4.14 Waktu eksekusi grid search

Segmen (detik)	Waktu eksekusi (jam) dengan 2 thread		
	250 pembicara	500 pembicara	715 pembicara
5	2,5	9	13
7	2	6	9
10	1,5	4	6
15	1	3	4

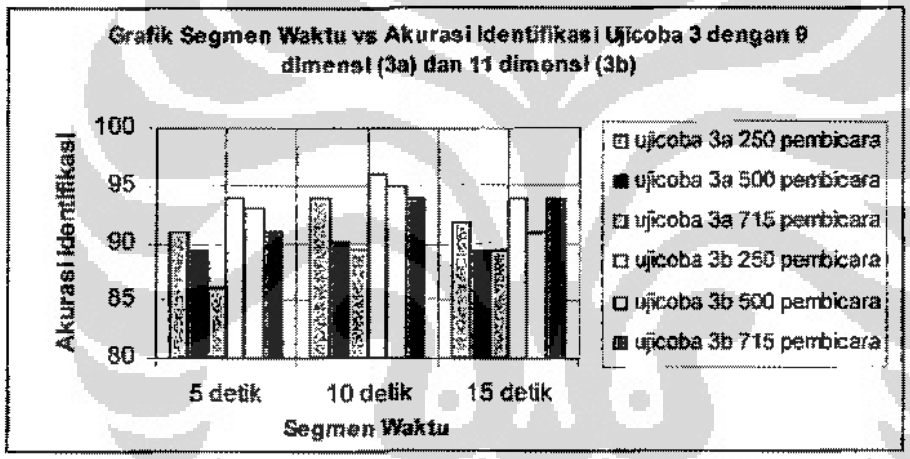
4.2.1 Analisa Hasil Ujicoba

Hasil akurasi identifikasi dari ujicoba 1, 2, 3, 4, dan 5 ditampilkan dalam bentuk grafik terhadap segmen waktu dapat dilihat pada gambar 4.1, 4.2, dan 4.3.



Gambar 4.1 Grafik segmen waktu vs akurasi identifikasi ujicoba 1 dan 2

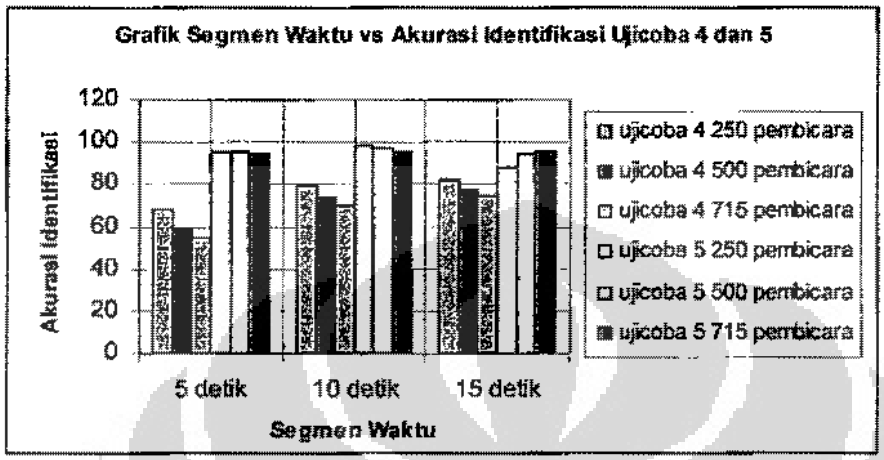
Hasil akurasi identifikasi dari ujicoba 1 dan 2 terhadap segmen waktu dapat dilihat pada gambar 4.1. Pada gambar tersebut terlihat bahwa untuk ujicoba 1, akurasi identifikasi yang diperoleh adalah 93,87-96.87%, di mana akurasi paling baik diperoleh pada segmen waktu 10 detik dengan 250 pembicara, yaitu sebesar 96,87%, dan yang terendah pada penggunaan segmen waktu 15 detik dengan 715 pembicara, yaitu 93,87%. Sedangkan pada ujicoba 2, akurasi identifikasi yang diperoleh adalah 94,1-98,14%, di mana akurasi tebesar diperoleh pada penggunaan segmen waktu 10 detik dengan 250 pembicara, yaitu sebesar 98,14%, dan yang terkecil diperoleh pada penggunaan segmen waktu 5 detik dengan 715 pembicara, yaitu sebesar 94,1%.



Gambar 4.2 Grafik segmen waktu vs akurasi identifikasi ujicoba 3

Akurasi identifikasi yang diperoleh pada ujicoba 3 terhadap segmen waktu dapat dilihat pada gambar 4.2. Pada gambar tersebut terlihat bahwa untuk ujicoba dengan 9 dimensi data, akurasi identifikasi yang diperoleh adalah 86,14-93.84%, di mana akurasi paling baik diperoleh pada segmen waktu 10 detik dengan 250 pembicara, yaitu sebesar 93,84%, dan yang terendah pada penggunaan segmen waktu 5 detik dengan 715 pembicara, yaitu 86,14%. Sedangkan pada ujicoba dengan 11 dimensi data, akurasi identifikasi yang diperoleh adalah 90,9-95,99%, di mana akurasi tebesar diperoleh pada penggunaan segmen waktu 10 detik

dengan 250 pembicara, yaitu sebesar 95,99%, dan yang terkecil diperoleh pada penggunaan segmen waktu 5 detik dengan 715 pembicara, yaitu sebesar 90,9%.



Gambar 4.3 Grafik segmen waktu vs akurasi identifikasi ujicoba 4 dan 5

Hasil akurasi identifikasi dari ujicoba 4 dan 5 terhadap segmen waktu dapat dilihat pada gambar 4.3. Pada gambar tersebut terlihat bahwa untuk ujicoba 4, akurasi identifikasi yang diperoleh adalah 54,66-82,6%, di mana akurasi paling baik diperoleh pada segmen waktu 15 detik dengan 250 pembicara, yaitu sebesar 82,6%, dan yang terendah pada penggunaan segmen waktu 5 detik dengan 715 pembicara, yaitu 54,66%. Sedangkan pada ujicoba 5, akurasi identifikasi yang diperoleh adalah 88,18-98,14%, di mana akurasi tebesar diperoleh pada penggunaan segmen waktu 10 detik dengan 250 pembicara, yaitu sebesar 98,14%, dan yang terkecil diperoleh pada penggunaan segmen waktu 15 detik dengan 250 pembicara, yaitu sebesar 88,18%.

Hasil akurasi ujicoba 1, 2, 3, dan 5 (Gambar 4.1, 4.2, dan sebagian 4.3) menunjukkan bahwa dengan jumlah pembicara yang sama, memperpanjang segmen waktu sampai 10 detik dapat meningkatkan akurasi identifikasi sampai dengan 3%. Jika diperpanjang sampai 15 detik, akurasi identifikasi akan berkurang sampai dengan 10%.

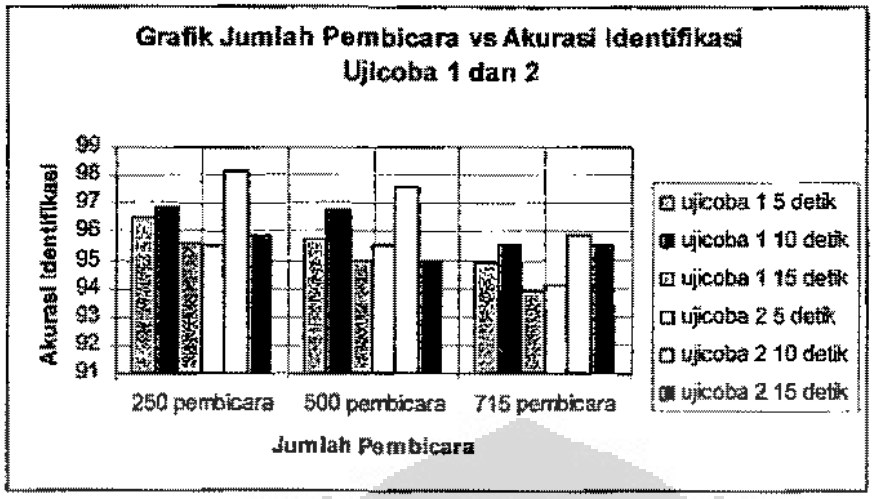
Pada dasarnya dengan semakin panjang segmen waktu dari suatu data, maka akan semakin banyak pula informasi yang diperoleh. Namun, yang

digunakan dalam tesis ini adalah nilai rata-rata fitur *speech* dalam suatu segmen waktu. Dengan demikian, jika semakin panjang segmen yang digunakan pada tiap data, justru akan menghilangkan variasi data dalam pembicara itu sendiri. Karena mungkin dalam jangka waktu tersebut model suara pembicara sudah berubah beberapa kali, sedangkan yang digunakan sebagai data hanya nilai rata-ratanya saja. Sehingga data tidak akan mencerminkan model suara dari pembicara. Oleh karena itu, yang perlu dicari adalah seberapa panjang model suara pembicara masih tidak berubah sehingga ketika diambil nilai rata-ratanya tidak akan menghilangkan variasi data pembicara tersebut.

Selain itu, penggunaan segmen waktu yang semakin panjang, akan menyulitkan dalam pengumpulan data untuk pembuatan model data pembicara. Karena dalam suatu data *speech*, misalnya data berita, seseorang hanya berbicara sebentar, kira-kira 5-10 detik, kemudian ganti orang lain yang berbicara, dan seterusnya. Kemungkinan orang tersebut akan berbicara lagi pada kesempatan lain belum tentu ada. Dengan demikian, semakin panjang segmen waktu yang digunakan akan semakin sedikit pembicara yang bisa dibuat model suaranya, karena seseorang yang berbicara dengan durasi kurang dari segmen tersebut tidak akan bisa dibuat model data suaranya. Penambahan data untuk setiap pembicara juga lebih sulit, karena sesi bicara yang diperlukan juga semakin panjang.

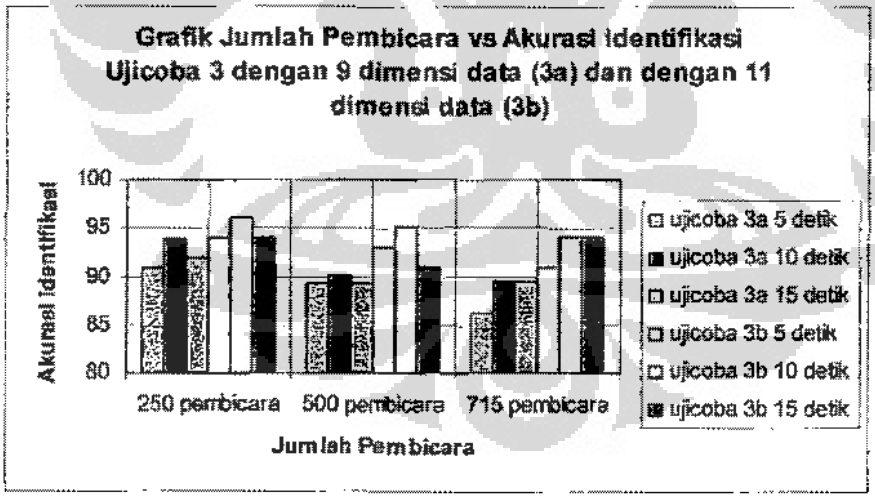
Hasil akurasi identifikasi pada ujicoba 1, 2, 3, dan 5 yang paling baik diperoleh pada penggunaan segmen waktu 10 detik dengan pembicara 250 orang. Sehingga secara umum, pada ujicoba 1,2, dan 3, penggunaan segmen waktu 10 detik menghasilkan akurasi yang paling baik. Jadi bisa dikatakan bahwa dalam tesis ini penggunaan segmen waktu 10 detik cukup mewakili data pembicara pada suatu sesi.

Hasil akurasi identifikasi dari ujicoba 1, 2, 3, 4, dan 5 ditampilkan dalam bentuk grafik terhadap jumlah pembicara ditunjukkan oleh gambar 4.4, 4.5, dan 4.6.



Gambar 4.4 Grafik jumlah pembicara vs akurasi identifikasi ujicoba 1 dan 2

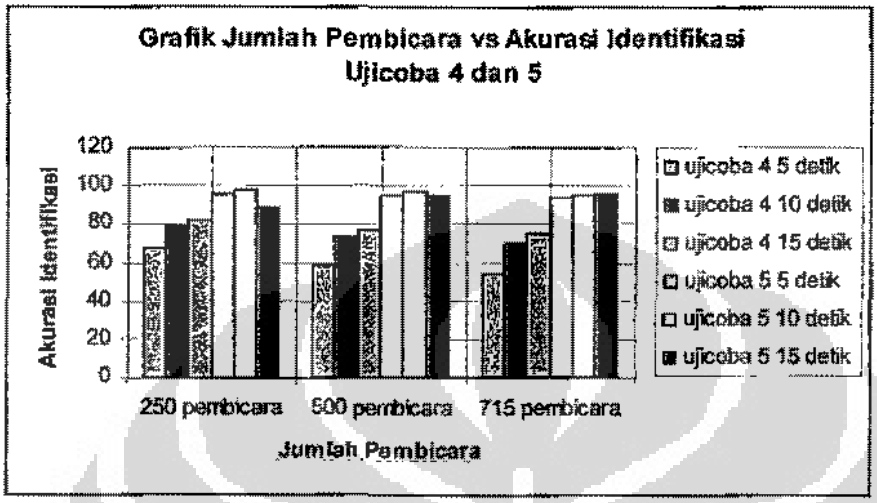
Pada gambar 4.4 terlihat bahwa untuk ujicoba 1 akurasi identifikasi cenderung menurun seiring dengan bertambahnya jumlah pembicara dalam segmen waktu yang tetap. Hal yang sama juga terjadi pada ujicoba 2.



Gambar 4.5 Grafik jumlah pembicara vs akurasi identifikasi ujicoba 3

Begitu pula pada ujicoba 3, baik yang menggunakan 9 dimensi data maupun 11 dimensi data sama-sama mempunyai kecenderungan menurunnya

akurasi identifikasi seiring dengan bertambahnya jumlah pembicara pada segmen waktu yang sama, seperti ditunjukkan oleh gambar 4.5.



Gambar 4.6 Grafik jumlah pembicara vs akurasi identifikasi ujicoba 4 dan 5

Pada tabel 4.6 terlihat bahwa untuk unjicoba 4 dan 5 juga terdapat kecenderungan berkurangnya akurasi identifikasi seiring dengan bertambahnya jumlah pembicara dalm segmen waktu yang sama.

Dengan demikian, secara umum, gambar 4.4, 4.5, dan 4.6 menunjukkan bahwa dalam suatu segmen waktu yang sama, penambahan jumlah pembicara akan mengurangi akurasi identifikasi. Hal ini disebabkan oleh jumlah data yang tidak sama dari setiap pembicara. Dengan bertambahnya jumlah pembicara, jumlah kelas yang harus dipisahkan juga semakin banyak, sedangkan beberapa pembicara tidak mempunyai cukup banyak data untuk memodelkan kelasnya. Ada kemungkinan dengan bertambahnya jumlah pembicara, jumlah pembicara dengan data yang sedikit juga semakin banyak.

Hasil akurasi identifikasi ujicoba 2 sedikit lebih baik dibanding ujicoba 1, yaitu sebesar 1-2% lebih baik. Ujicoba 1 menghasilkan akurasi sebesar 93-96% sedangkan ujicoba 2 menghasilkan akurasi 94-98%. Ini berarti penggunaan *silence removal* sedikit memperbaiki performa sistem. Hal ini tidaklah mengherankan karena dengan membuang bagian yang diam dari data, berarti data

yang diproses hanya akan berisi informasi data pembicara saja. Namun, *silence removal* mempunyai beberapa metode tersendiri karena bukan hal mudah untuk membuang *silence* tanpa mempengaruhi data. Metode *silence removal* yang digunakan pada tesis ini adalah metode standar yang menggunakan nilai batas (*threshold*) tertentu, di mana nilai yang berada di bawah nilai batas akan dibuang. Dengan demikian, jika ada informasi yang ada di bawah nilai batas, maka informasi tersebut akan hilang.

Pada ujicoba 3, penggunaan PCA untuk mengurangi dimensi data tanpa memperburuk hasil identifikasi kurang berhasil. Pada penggunaan 9 dimensi data dan 11 dimensi data, hasil akurasi yang diperoleh masih di bawah akurasi ujicoba 2, yaitu 86-94% untuk penggunaan 9 dimensi data dan 91-96% untuk penggunaan 11 dimensi data sedangkan akurasi ujicoba 2 adalah 94-98%. Dengan demikian, jumlah dimensi data tidak bisa dikurangi tanpa mengurangi nilai akurasi identifikasi.

Hasil akurasi identifikasi ujicoba 2 jauh lebih baik dibanding ujicoba 4. Ujicoba 2 menghasilkan akurasi 94-98% sedangkan ujicoba 4 hanya menghasilkan akurasi 54-82%. Ini berarti nilai rata-rata frame dari MFCC pada suatu segmen waktu lebih dapat mewakili daripada nilai varians frame dari MFCC pada suatu segmen waktu.

Jika dibandingkan antara sistem pada ujicoba 5 dengan sistem pada ujicoba 2, hasil akurasi keduanya hampir sebanding. Pada penggunaan jumlah pembicara dan segmen waktu yang sama, hasil akurasi yang diperoleh hampir sama, hanya berbeda pada saat penggunaan 250 pembicara dengan segmen waktu 15 detik. Misalnya pada penggunaan 250 pembicara dengan segmen waktu 10 detik, ujicoba 2 dan ujicoba 5 sama-sama menghasilkan akurasi 98%. Begitu pula pada penggunaan 500 pembicara dengan segmen waktu 10 detik, kedua ujicoba ini sama-sama menghasilkan akurasi 97%. Hal ini tidaklah mengherankan karena ujicoba 2 dan ujicoba 5 sama-sama menggunakan metode SVM *multi-class one-vs-one*, hanya pemrosesan data dalam SVM yang berbeda, di mana ujicoba 5 menggunakan metode dekomposisi yang pada dasarnya membagi data menjadi beberapa bagian pada saat klasifikasinya. Metode dekomposisi dimaksudkan untuk mengatasi masalah klasifikasi dalam jumlah besar. Dan karena sistem

identifikasi pembicara merupakan klasifikasi dalam jumlah yang semakin besar, baik kelas maupun datanya, maka metode dekomposisi diujicoba untuk melihat hasil akurasinya. Dengan hasil yang sebanding ini, maka metode dekomposisi juga dapat dipertimbangkan dalam pengembangan selanjutnya.

Seluruh ujicoba menggunakan campuran antara data yang jernih (*clean speech*) dengan yang mengandung *noise*. Hal ini disebabkan karena sebagian besar data berita baik radio maupun televisi masih mengandung noise walaupun sedikit. Pada saat pra-pemrosesan sudah diupayakan untuk mengurangi *noise* pada data, dengan asumsi noise berada pada frekuensi rendah, yaitu dengan *silence removal*, di mana data yang energinya berada di bawah ambang tertentu akan dibuang; dan dengan pre-emphasis, yang merupakan filter *high pass* yang akan melewatkan frekuensi tinggi dan menolak frekuensi rendah tertentu. Dengan perolehan akurasi identifikasi yang cukup baik menunjukkan sistem yang ada sudah cukup baik dalam mengetasi *noise*. Selain itu, jika dilihat dari hasil identifikasi, pembicara dengan data yang jernih pun dapat mengalami kesalahan identifikasi.

Tabel 4.15 Kesalahan identifikasi berdasarkan gender pada ujicoba 2

Jumlah Pembicara	Kesalahan identifikasi berdasarkan gender (%)					
	5 detik		10 detik		15 detik	
	L	P	L	P	L	P
250	21,89	20	5,97	2	7,5	2
500	22	25	8	4	9	7,9
715	28,6	27	12,6	9	7	9

Jika dilihat dari jenis kelamin pembicara, seperti terlihat pada tabel 4.15, pada segmen waktu 5 detik, perbandingan kesalahan identifikasi untuk pria dan wanita hampir sama, meskipun jumlah pembicara pria jauh lebih besar dibanding jumlah pembicara wanita. Untuk penggunaan 250 pembicara terdapat 201 pembicara pria dan 49 wanita. Pada penggunaan 500 pembicara, terdapat 386 pria dan 114 wanita. Sedangkan pada penggunaan 715 pembicara terdapat 549 pria

dan 166 wanita. Pada tabel terlihat bahwa pada segmen waktu 10 detik akurasi identifikasi pembicara wanita lebih baik dibanding pembicara pria, dan pada segmen waktu 15 detik, akurasi identifikasi pembicara pria juga cenderung lebih buruk dibanding pembicara wanita.

Pada hasil identifikasi ujicoba 2, pembicara yang mempunyai 1 data pelatihan dan 1 data testing pada penggunaan segmen waktu 15 detik, selalu menghasilkan akurasi identifikasi 100%. Hal ini mungkin disebabkan karena data pelatihan dan data testing berasal dari satu sesi berita atau wawancara (minimum data adalah 40 detik) sehingga sangat mirip satu dengan lainnya. Kesalahan identifikasi dimulai pada data dengan jumlah data pelatihan sebesar 2 dan data testing sebanyak 1. Pada penggunaan segmen waktu 10 detik, satu pembicara mempunyai minimal 4 data, 2 data untuk pelatihan dan 2 data untuk testing. Sedangkan pada penggunaan segmen waktu 5 detik, satu pembicara minimal mempunyai 8 data, 4 data untuk pelatihan dan 4 data untuk testing.

Grid search digunakan untuk memperoleh parameter SVM (C dan γ) yang optimal. Pada seluruh ujicoba, interval C yang digunakan adalah 2^{-3} - 2^{13} sedangkan interval γ adalah 2^{-3} - 2^{3}. Dengan demikian terdapat 17 x 7 = 119 kombinasi (C, γ). Bila *cross validation* adalah 5, maka setiap kombinasi (C, γ) akan digunakan 5 kali, kemudian akan dihitung rata-rata akurasi validasinya untuk memperoleh akurasi *cross validation*, di mana nilai akurasi yang terbesar akan digunakan untuk pelatihan sistem. Jumlah eksekusi pelatihan sistem dalam suatu *grid search* adalah 119 x 5 = 595. Jika dalam eksekusinya, *grid search* menggunakan 2 buah *thread*, maka kompleksitasnya akan dibagi 2, masing-masing thread melakukan 119/2 x 5 = 300 dan 295 kali pelatihan sistem.

Waktu yang diperlukan *grid search* tergantung pada jumlah data, jumlah kelas data, sumber daya komputasi yang digunakan, dan kompleksitas data itu sendiri. Jika dilihat dari data yang digunakan, ujicoba 1, 2, 4, dan 5 mempunyai jumlah data dan jumlah kelas data yang sama, namun karena datanya berbeda maka kompleksitas datanya tidak sama. Dengan demikian waktu yang diperlukan hampir sama, seperti pada tabel 4.14. Data pada ujicoba 3 sama dengan pada ujicoba 2, hanya berbeda jumlah dimensi datanya. Secara matematis, jumlah data keseluruhan ujicoba 3 lebih kecil dari ujicoba 2, kompleksitasnya pun lebih

sedikit karena dimensinya lebih sedikit, sehingga waktu yang diperlukan seharusnya lebih sedikit dari yang tersebut pada tabel 4.14.

Cara lain untuk mengurangi waktu komputasi *grid search* adalah dengan meningkatkan sumber daya komputasi sehingga komputasi dapat dilakukan secara paralel.

Berdasarkan hasil akurasi yang diperoleh pada kelima ujicoba dapat dikatakan bahwa sistem identifikasi pembicara pada ujicoba 2 (sistem yang diusulkan) memperoleh hasil yang cukup baik. Jika dibandingkan dengan ujicoba 1 yang hanya menggunakan satu metode pra-pemrosesan (tanpa *silence removal*), ujicoba 2 yang menggunakan 2 metode pra-pemrosesan (dengan *silence removal*) memperoleh akurasi identifikasi yang sedikit lebih baik, yaitu 1-2%. Kemudian jika dibandingkan dengan ujicoba 3 yang menggunakan 9 dan 11 dimensi data, ujicoba 2 yang menggunakan 12 dimensi data menghasilkan akurasi identifikasi yang lebih baik, yaitu sebesar 3% jika dibandingkan dengan penggunaan 11 dimensi data, dan 4-8% dibandingkan dengan penggunaan 9 dimensi data. Jika dibandingkan dengan ujicoba 4 yang menggunakan nilai varians data MFCC pada suatu waktu, ujicoba 2 yang menggunakan nilai rata-rata frame dari MFCC pada suatu waktu menghasilkan akurasi identifikasi yang jauh lebih baik, yaitu 94-98% untuk ujicoba 2 dan 54-82% untuk ujicoba 4. Lalu, jika dibandingkan dengan ujicoba 5 yang menggunakan metode SVM *one-vs-one* dekomposisi, ujicoba 2 yang menggunakan metode SVM *one-vs-one* tanpa dekomposisi memperoleh hasil akurasi yang hampir sama baiknya, yaitu, 88-98% untuk ujicoba 5 dan 94-98% untuk ujicoba 2. Selain itu, pada ujicoba 2 terdapat kecenderungan akurasi akan meningkat pada penggunaan segmen waktu yang semakin panjang sampai dengan 10 detik dan kemudian akurasi akan berkurang pada 15 detik. Akurasi identifikasi juga menurun seiring dengan bertambahnya jumlah pembicara. Secara umum, akurasi yang paling baik diperoleh pada penggunaan segmen waktu 10 detik. Dan jumlah minimum data pelatihan yang diperlukan adalah 1 data. Sedangkan jika dilihat dari sisi gender, pembicara wanita mempunyai akurasi lebih baik dibanding pembicara pria.

BAB 5
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Pada penelitian ini telah diteliti sistem identifikasi pembicara dengan menggunakan metode Support Vector Machine (SVM). Fitur speech yang digunakan adalah nilai rata-rata frame dari MFCC koefisien 0-11 dalam suatu segmen waktu dan metode SVM yang digunakan adalah SVM *multi-class one-vs-one* dengan kernel RBF. Sistem ini telah diujicoba dengan menggunakan data berita berbahasa Indonesia dari radio dan televisi yang disegmen dalam 5, 10, dan 15 detik. Hasil akurasi identifikasi yang diperoleh, yaitu 94-98%, dengan akurasi terbaik diperoleh pada segmen waktu 10 detik.

Beberapa ujicoba telah dilakukan untuk membandingkan akurasi identifikasi sistem pembicara. Pada ujicoba tanpa *silence removal*, akurasi yang diperoleh adalah 93-96%, lebih buruk 1-2% dari sistem yang menggunakan *silence removal*. Pada ujicoba dengan menggunakan nilai varians sebagai pengganti nilai rata-rata, diperoleh akurasi identifikasi sebesar 54%-82%, lebih buruk dari penggunaan nilai rata-rata yang menghasilkan akurasi 94-98%. Ujicoba lain dengan mengurangi jumlah dimensi data menjadi 11 hanya menghasilkan akurasi sebesar 91-95%, masih di bawah akurasi sistem yang menggunakan 12 dimensi data yang menghasilkan 94-98%, sehingga dimensi data tidak bisa dikurangi tanpa mengurangi akurasi sistem. Sedangkan pada ujicoba dengan metode SVM *multi-class* lain, yaitu metode dekomposisi, menghasilkan akurasi yang hampir sama dengan sistem yang tanpa metode dekomposisi, yaitu 88-98% untuk metode dekomposisi dan 94-98% untuk sistem tanpa metode dekomposisi.

Panjang data 10 detik cukup mewakili satu data pembicara, karena segmen waktu 10 detik menghasilkan akurasi yang paling baik. Jumlah data minimum yang diperlukan adalah satu data. Metode SVM multi-class *one-vs-one* (dengan atau tanpa metode dekomposisi) yang digunakan pada semua ujicoba sudah dapat menghasilkan akurasi identifikasi yang baik. Dan *noise* pada suara dapat dikurangi dengan menggunakan *pre-emphasis* dan *silence removal*. Sedangkan

jika dilihat dari sisi gender, pembicara wanita mempunyai akurasi lebih baik dibanding pembicara pria.

Sistem identifikasi pembicara mempunyai kecenderungan berkurangnya akurasi identifikasi seiring dengan penambahan jumlah pembicara, sehingga perlu diujicoba dengan menggunakan data pembicara yang lebih banyak lagi.

5.2 Saran

Beberapa masalah yang dapat diperbaiki adalah:

1. Penggunaan nilai rata-rata dapat digantikan oleh fitur data lainnya, misalnya *centroid* pada data yang di-*cluster* terlebih dahulu.
2. Mengubah metode pra-pemrosesan, antara lain dengan menggunakan metode lain dalam *silence removal*. Penentuan *energy threshold* dalam tesis ini bisa jadi kurang akurat, sehingga kemungkinan ada informasi di bawah *threshold* yang hilang.
3. Perhitungan PCA dilakukan pada data MFCC yang belum dirata-rata (data MFCC asli).
4. Untuk mempersingkat waktu eksekusi grid search, maka interval C dapat diperpendek menjadi 2^3-2^8 dan interval γ menjadi 2^{-2}-2^2.

DAFTAR PUSTAKA

[1] Bishop, Christopher M., "*Pattern Recognition and Machine Learning*", Springer, New York, 2006

[2] Brooks, Mike, "VOICEBOX: Speech Processing Toolbox for MATLAB", Department of Electrical and Electronic Engineering, Imperial College, 1997

[3] Buono, Agus, Wisnu Jatmiko, Benyamin Kusumodiputro, "Genetics Algorithm for 2D-MFCC Filter Development in Speaker Identification System Using HMM", *Proceedings of International Conference on Advanced Computational Intelligence and Its Application*, 2008

[4] Buono, Agus, Wisnu Jatmiko, Benyamin Kusumodiputro, "Development of 2D Mel-Frequency Cepstrum Coefficients Method for Processing Bispectrum Data as Feature Extraction Technique in Speaker Identification System", *Proceedings of International Conference on Advanced Computational Intelligence and Its Application*, 2008

[5] Burges, Christopher J.C., "A Tutorial on Support Vector Machines for Pattern Recognition", *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp.1-47, 1998

[6] Campbell, Joseph P. Jr, "Speaker Recognition: A Tutorial", *Proceedings of The IEEE*, Vol. 85, No. 9, September 1997

[7] Chakroborty, Sandipan, Anindya Roy, Sourav Majumdar, Goutam Saha, "Capturing Complementary Information via Reversed Filter Bank and Parallel Implementation with MFCC for Improved Text-Independent Speaker Identification", *Proceedings of the International Conference on Computing: Theory and Applications (ICCTA)*, 2007

[8] Chang, Chih-Chung dan Chih-Jen Lin, "LIBSVM : A Library for Support Vector Machines", http://www.csie.ntu.edu.tw/~cjlin/libsvm, 2001

[9] Dagtas, Serhan, Mustafa Sarimollaoglu, Kamran Iqbal, "A Multimodal Virtual Environment with Text-Independent Real-Time Speaker Identification", *Proceedings of the IEEE Sixth International Symposium on Multimedia Software Engineering (ISMSE)*, 2004

[10] Fine, Shai, Jiří Navrátil, Ramesh A. Gopinath, "A Hybrid GMM/SVM Approach to Speaker Identification", *Proceedings of the IEEE International Conference on Acoustic, Speech, and Signal Processing (ICASSP)*, 2001

[11] Flanagan, J.L., "*Speech Analysis, Synthesis, and Perception*", 2nd ed., Springer-Verlag, New York, 1972

[12] Hsu, C.W. dan C.J Lin, "A Comparison of Methods for Multi-class Support Vector Machines", *IEEE Transactions on Neural Networks*, No.13, pp. 415-425, 2002

[13] Hsu, C.W. dan C.J Lin, "A Simple Decomposition Method for Support Vector Machines", *Machine Learning*, No.46, pp. 291-314, 2002

[14] Kamruzzaman, S.M., A.N.M. Rezaul Karim, Saiful Islam, dan Emdadul Haque, "Speaker Identification Using MFCC-Domain Support Vector Machine", *International Journal of Electrical and Power Engineering* 1 (3), pp.274-278, 2007

[15] Karpov, Evgeny, "Real-Time Speaker Identification", Master's of Thesis, Department of Computer Science University of Joensuu, 15 January 2003

[16] Knerr, S., L.Personnaz, dan G.Dreyfus, "Single-layer Learning Revisited: A Stepwise Procedure for Building and Training a Neural Network", in: *Neurocomputing: Algorithms, Architectures and Applications*, NATO ASI Series, Springer, Berlin, 1990.

[17] Lee, Yuh-Jye, Olvi L. Mangasarian, "RSVM: Reduced Support Vector Machines", *First SIAM International Conference on Data Mining, Data Mining Institute Technical Report*, 00-07, July 2000

[18] Li, Ming, Ruiling Luo, Yu-Juan Xing, "A Novel Multi-Reduced SVM Approach for Speaker Recognition", *Fifth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, Vol. 4, pp.462-466, 2008

[19] Li, Ming, Xue-Yan Liu, Yu-Juan Xing, "A Novel Hierarchical Speaker Identification Method", *Congress on Image and Signal Processing (CISP)*, Vol. 4, pp.511-515, 2008

[20] Li, Ming, Xueyan Liu, Fuwen Wu, "Speaker Identification based on Multi-Reduced SVM", *Fourth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, Vol. 3, pp.371-375, 2007

[21] Lin, Chien-Chang, Shi-Huang Chen, Tsung-Ching Lin, dan T.K. Truong, "Feature Comparisons among Various Wavelets in Speaker Recognition using Support Vector Machine", *Proceedings of the Eighth IEEE International Symposium on Multimedia (ISM)*, 2006.

[22] Moreno, Pedro J., Purdy P. Ho, "A New SVM Approach to Speaker Identification and Verification Using Probabilistic Distance Kernels", *Eurospeech*, September, 2003.

[23] Osuna, E., R.Freund, dan F.Girosi, "Support Vector Machines: Training and Application", Technical Report, AIM-1602, MIT, 1997

[24] Rabiner, Lawrence, Biing-Hwang Juang, "*Fundamental of Speech Recognition*", Prentice-Hall, Englewood Cliffs, N.J., 1993

[25] Raynolds, Dauglas A., Richard C. Rose, "Robust Text-Independent Speaker Identification Using Gaussian Mixture Models", *IEEE Transactions on Speech and Audio Processing*, Vol.3, No.1, January 1995

[26] Schmidt, M., H. Gish, "Speaker Identification via Support Vector Classifiers", *Proceedings of the IEEE International Conference Acoustic, Speech, and Signal Processing (ICASSP)*, Vol.1, pp.105-108, 1996

[27] Smith, Julius O. III, *"Introduction to Digital Filters with Audio Applications"*, Center for Computer Research in Music and Acoustic (CCRMA), Department of Music, Stanford University, Stanford, California, USA, 2008

[28] Smith, Lindsay I., "A Tutorial on Principal Component Analysis", http://www.cs.otago.ac.nz/cosc453/student_tutorials/principal_components.pdf, 26 Februari 2002.

[29] V, Moonasar., Venayagamoorty G.K, "Speaker Identification Using a Combination of Different Parameters as Feature Inputs to an Artificial Neural Network Classifier", *IEEE APRICON*, Vol. 1, pp.189-194, 28 Sept.-1 Oct, 1999

[30] Vapnik, Vladimir N., "*The Nature of Statistical Learning Theory*", Springer, New York, 1995

[31] Vapnik, Vladimir N., "*Statistical Learning Theory*", Wiley, New York, 1998

[32] Wan, Vincent, William M.Campbell, "Support Vector Machines for Speaker Verification and Identification", *Proceedings Neural Networks for Signal Processing X*, pp.775-784, 1999

[33] Wang,Yan, Xueyan Liu, Yujuan Xing, Ming Li, "A Novel Reduction Method for Text-Independent Speaker Identification", *Fourth International Conference on Natural Computation (ICNC)*, Vol. 4, pp. 66-70, 2008