



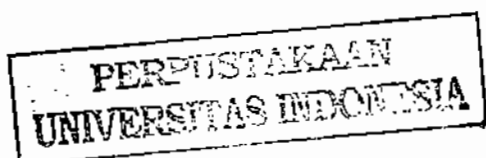
UNIVERSITAS INDONESIA

**REPRESENTASI NILAI HOS DAN MODEL MFCC SEBAGAI
EKSTRAKSI CIRI PADA SISTEM IDENTIFIKASI
PEMBICARA DI LINGKUNGAN BER-*NOISE*
MENGUNAKAN HMM**

DISERTASI

**AGUS BUONO
9205000026**

**FAKULTAS ILMU KOMPUTER
PROGRAM STUDI ILMU KOMPUTER
DEPOK
APRIL, 2009**





UNIVERSITAS INDONESIA

**REPRESENTASI NILAI HOS DAN MODEL MFCC SEBAGAI
EKSTRAKSI CIRI PADA SISTEM IDENTIFIKASI
PEMBICARA DI LINGKUNGAN BER-*NOISE*
MENGUNAKAN HMM**

DISERTASI


**Diajukan sebagai salah satu syarat untuk memperoleh gelar Doktor Ilmu
Komputer pada Program Studi Ilmu Komputer Fakultas Ilmu Komputer
Universitas Indonesia**

**AGUS BUONO
9205000026**

**FAKULTAS ILMU KOMPUTER
PROGRAM STUDI ILMU KOMPUTER
DEPOK
APRIL, 2009**

**Disertasi ini adalah hasil karya saya sendiri di
bawah pembimbingan, dan semua sumber baik
yang dikutip maupun dirujuk telah saya nyatakan
dengan benar.**

**Nama : Agus Buono
NPM : 9205000026**

Tanda Tangan : 

Tanggal : 29 April 2009

Disertasi ini diajukan oleh :

NAMA : AGUS BUONO
NPM : 9205000026
Program Studi : ILMU KOMPUTER
Judul Disertasi : REPRESENTASI NILAI HOS DAN MODEL MFCC
SEBAGAI EKSTRAKSI CIRI PADA SISTEM
IDENTIFIKASI PEMBICARA DI LINGKUNGAN BER-
NOISE MENGGUNAKAN HMM

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

DEWAN PENGUJI

Promotor : Prof. Benyamin Kusumoputro, Dr. Eng. (.....)

Kopromotor : Wisnu Jatmiko, Dr. Eng. (.....)

Tim Penguji : Prof. Dr. Aniati Murni (Ketua) (.....)

: Prof. Dr. Toemin A. Masoem (Anggota) (.....)

: Lim Yohanes Stefanus, Ph.D. (Anggota) (.....)

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

: Prof. Dr. Marimin, M.Sc. (Anggota) (.....)

Ditetapkan di : Depok

Tanggal : 29 April 2009

KATA PENGANTAR/UCAPAN TERIMA KASIH

Puji syukur Penulis panjatkan kepada Allah SWT. Karena atas rahmat dan karunia-Nya maka disertasi yang berjudul “ **Representasi Nilai HOS dan Model MFCC sebagai Ekstraksi Ciri pada Sistem Identifikasi Pembicara di Lingkungan Ber-noise Menggunakan HMM**”, dapat diselesaikan. Penulisan ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Doktor Ilmu Komputer pada Fakultas Ilmu Komputer Universitas Indonesia. Saya menyadari bahwa tanpa bantuan dan bimbingan dari berbagai pihak, dari masa perkuliahan sampai pada penyusunan disertasi ini, sangatlah sulit bagi saya untuk menyelesaikannya. Oleh karena itu, perkenankan saya mengucapkan terima kasih yang setinggi-tingginya kepada Prof. Benyamin Kusumoputro, Dr. Eng., sebagai promotor yang terus menerus memberikan bimbingan, kritik, saran, dan arahan yang sangat berharga baik saat persiapan, pelaksanaan hingga penyelesaian studi Program Doktor. Kepada Wisnu Jatmiko, Dr. Eng. sebagai ko-promotor yang telah memberikan berbagai saran untuk kesempurnaan disertasi ini, saya mengucapkan terima kasih yang sebesar-besarnya. Kepada Prof. Dr. Aniati Murni, Prof. Dr. Toemin A. Masoem, Lym Yohanes Stefanus, Ph.D., Hisar Maruli Manurung, Ph.D., dan Prof. Dr. Marimin, M.Sc. atas kesediaannya menjadi Penguji serta saran-saran yang telah diberikan. Pada kesempatan ini saya juga menyampaikan ucapan terima kasih kepada :

1. Pimpinan Institut Pertanian Bogor yang telah memberikan ijin untuk melanjutkan studi di Program Doktor Program Studi Ilmu Komputer Fakultas Ilmu Komputer Universitas Indonesia
2. Segenap Pimpinan Program Pascasarjana, Dekan Fakultas Ilmu Komputer, Ketua Program Studi Ilmu Komputer Fakultas Ilmu komputer Universitas Indonesia serta jajaran dosen dan administrasinya yang telah memberikan layanan pengajaran serta administrasi dengan baik.

3. Ketua Departemen Ilmu Komputer Fakultas Matematika Institut Pertanian Bogor yang telah menyediakan laboratorium serta perangkat pendukung lainnya untuk keperluan pelaksanaan penelitian
4. Direktorat Jenderal Pendidikan Tinggi atas bantuan dana berupa beasiswa BPPS selama menempuh pendidikan
5. Teman-teman mahasiswa di Laboratorium Kecerdasan Komputasional, juga teman-teman sesama Program Doktor atas diskusinya selama penelitian.
6. Teman-teman dosen di Departemen Ilmu Komputer Fakultas Matematika Institut Pertanian Bogor atas diskusi selama penyelesaian penelitian.
7. Semua pihak yang telah membantu yang tidak dapat kami sebutkan satu per satu.

Menyadari bahwa semua keberhasilan ini tidak lepas dari peran orang tua yang telah memberi didikan dari kecil juga keikhlasan doa yang senantiasa dipanjatkan kepada Allah SWT bagi anak-anaknya, maka pada kesempatan ini Penulis mengucapkan terima kasih yang sebesar-besarnya kepada Ibunda Sridadi. Ucapan terima kasih juga disampaikan kepada Ayah Penulis, Bapak Mardi yang telah memberi didikan sejak kecil. Mengingat betapa besar perhatian istri Penulis, Ir. Endang Sri Suswati yang setia menemani, terus-menerus memberi semangat, dan dorongan untuk selesainya disertasi ini maka pada kesempatan ini Penulis mengucapkan terima kasih atas semuanya. Juga kepada anak-anak yang baik, Putri, Dewi, Aisha, dan Imam yang harus sabar dan merelakan waktu kebersamaan dengan bapaknya yang sebagian tersita karena penyelesaian disertasi ini, maka Bapak mengucapkan terima kasih untuk kalian semua.

Penulis menyadari sepenuhnya, bahwa disertasi ini masih jauh dari sempurna, oleh karena itu kritik dan saran akan sangat Penulis hargai sebagai masukan yang baik untuk melengkapi segala kekurangan. Penulis berharap, karya yang kecil ini kiranya dapat menjadi satu masukan yang berarti dalam rangka pengembangan ilmu pengetahuan dan bermanfaat bagi pihak-pihak yang membutuhkannya.

Depok, 29 April 2009

Penulis

PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIK

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

NAMA : AGUS BUONO
NPM : 9205000026
Program Studi : ILMU KOMPUTER
Fakultas : FAKULTAS ILMU KOMPUTER
Jenis Karya : DISERTASI

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

Representasi Nilai HOS dan Model MFCC sebagai Ekstraksi Ciri pada Sistem Identifikasi Pembicara di Lingkungan Ber-noise Menggunakan HMM

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 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 : 29 April 2009

Yang menyatakan



(Agus Buono)

ABSTRAK

Nama : Agus Buono
Program Studi : Ilmu Komputer
Judul Disertasi : Representasi Nilai HOS dan Model MFCC sebagai Ekstraksi Ciri pada Sistem Identifikasi Pembicara di Lingkungan Ber-noise Menggunakan HMM

Suara merupakan suatu besaran yang memenuhi syarat sebagai ciri biometrik yang efektif dan efisien. Namun demikian, suara adalah fenomena yang merupakan perpaduan multidimensi serta dipengaruhi berbagai aspek, seperti karakteristik pembicara (dimensi titik artikularis, emosi, kesehatan, umur, jenis kelamin, dialek), bahasa, dan lingkungan (*background* dan media transmisi), sehingga sistem yang telah dikembangkan hingga sekarang belum bisa bekerja dengan baik pada situasi real. Hal inilah yang melatarbelakangi penelitian ini dilakukan.

Pada penelitian ini dilakukan kajian terhadap teknik *higher order statistics* (HOS) dan model *Mel-Frequency Cepstrum Coefficients* (MFCC) sebagai ekstraksi ciri yang diintegrasikan dengan *Hidden Markov Model* (HMM) sebagai pengenalan pola untuk menghasilkan sistem identifikasi pembicara yang lebih *robust* terhadap *noise*, khususnya *Gaussian Noise*. Penelitian yang dilakukan lebih difokuskan pada bagian ekstraksi ciri dari sistem identifikasi pembicara. Sementara itu, bagian pengenalan pola menggunakan teknik yang telah banyak dikaji pada berbagai riset pemrosesan suara dan memberikan hasil yang baik, yaitu HMM. Strategi yang dilakukan adalah melalui pendekatan empiris untuk menunjukkan kegagalan teknik ekstraksi ciri konvensional, yaitu 1D-MFCC yang berbasis power spektrum, pada lingkungan ber-*noise*, dilanjutkan dengan mengkaji permasalahannya, dan diusulkan teknik ekstraksi berbasis HOS untuk mengatasi permasalahan tersebut. Berikutnya adalah melakukan serangkaian percobaan untuk menunjukkan efektifitas teknik yang diusulkan, studi komparasi dan mengajukan suatu usulan rancangan sistem.

Berdasar bukti empiris, terlihat bahwa permasalahan 1D-MFCC adalah pada inputnya, yaitu power spektrum yang bersifat tidak stabil terhadap *noise*. Pada penelitian ini diusulkan untuk mengganti power spektrum dengan bispektrum yang secara teori lebih *robust* terhadap *noise*. Teknik yang diusulkan adalah suatu metodologi untuk mengekstrak nilai bispektrum sinyal suara dengan MFCC dan diintegrasikan dengan HMM untuk membentuk sistem identifikasi pembicara. Oleh karena itu, pada penelitian ini dilakukan perluasan teknik 1D-MFCC menjadi 2D-MFCC. Untuk meningkatkan efektifitas sistem, diusulkan teknik kuantisasi sebagai cara merepresentasikan nilai bispektrum sehingga distribusi spasialnya terakomodasi, dan dilanjutkan dengan transformasi wrapping dan kosinus seperti pada MFCC.

Hasil percobaan menunjukkan bahwa teknik konvensional yang berbasis pada power spektrum dapat menangkap ciri suara tanpa penambahan *noise* dengan baik dan jika dipadukan dengan *Mel-Frequency Cepstrum Coefficients* (MFCC) sebagai

ekstraksi ciri dan HMM sebagai pengenalan pola, maka akan menghasilkan sistem dengan akurasi di atas 98.8%. Namun demikian, dengan penambahan *noise* 20 dB, nilai power spektrum mengalami perubahan secara nyata, sehingga akurasi sistem jatuh hingga level di bawah 50%. Teknik penghapusan *noise* secara *adaptive* mampu meningkatkan akurasi menjadi 77.7%, namun dengan *noise* yang lebih besar, teknik ini gagal bekerja dengan baik.

Sistem yang dikembangkan dengan menggunakan bispektrum sebagai penentu ciri dipadukan dengan MFCC yang diperluas ke dua dimensi berhasil memberikan akurasi 99.9% untuk sinyal suara asli. Namun untuk sinyal dengan *noise* 20 dB, akurasi sistem menjadi sekitar 70%. Optimasi pada bentuk filter pada proses MFCC dengan algoritma genetika mampu meningkatkan akurasi menjadi 88.8%. Akan tetapi dengan *noise* yang lebih tinggi, sistem gagal bekerja dengan baik.

Teknik kuantisasi skalar terhadap nilai bispektrum yang dilanjutkan dengan proses *wrapping* dan transformasi kosinus seperti yang dilakukan pada MFCC mampu meningkatkan *robustness* sistem terhadap *noise* dengan akurasi 99.5% dan 83% masing-masing untuk sinyal asli dan sinyal dengan penambahan *noise* 20 dB. Namun untuk *noise* 10 dB, teknik ini gagal bekerja dengan baik. Dari percobaan dengan teknik kuantisasi vektor, terlihat bahwa rata-rata nilai bispektrum di atas kuartil tiga adalah penduga terbaik bagi nilai bispektrum setiap channel dengan jumlah 400 channel. Selain itu nilai parameter yang optimum pada proses ekstraksi ciri dengan kuantisasi vektor dilanjutkan dengan proses *wrapping* dan transformasi kosinus pada sinyal dengan penambahan *noise* adalah jarak filter linear 75, jarak filter logaritma 1.06 dan proporsi filter linear dan logaritma 30:20. Kinerja sistem menunjukkan peningkatan yang berarti dengan akurasi 88% dan 75.5% masing-masing untuk sinyal dengan penambahan *noise* 20 dB dan 10 dB. Namun demikian untuk sinyal asli justru lebih rendah, yaitu dengan akurasi maksimum hanya 94.5%. Hal ini berarti bahwa teknik ekstraksi ciri yang efektif tergantung dari kualitas sinyal masukan. Oleh karena itu sistem yang dikembangkan sebaiknya dilengkapi di bagian awalnya dengan kemampuan untuk menduga kualitas sinyal masukan.

Dari studi eksploratif terhadap nilai autokorelasi dan ragam sinyal suara, diperoleh bahwa kualitas sinyal dapat diidentifikasi dengan besaran yang dirumuskan sebagai negatif dari logaritma perkalian nilai absolut autokorelasi dari lag 1 hingga lag 21. Nilai ambang untuk membedakan sinyal sesuai kualitasnya dengan besaran tersebut adalah di antara 7 hingga 15. Jika nilai besaran tersebut kecil, maka teknik 1D-MFCC lebih sesuai untuk diterapkan. Sedangkan untuk hal lainnya, disarankan menggunakan teknik kuantisasi vektor terhadap nilai bispektrum sebagai pengekstraksi ciri. Berdasar nilai ambang inilah disusun prototipe sistem identifikasi pembicara menggunakan software Matlab.

Kata Kunci :

Gaussian Noise, *Mel-Frequency Cepstrum Coefficients* (MFCC), *Hidden Markov Model* (HMM), *Higher Order Statistic* (HOS), power spektrum, Bispektrum, dan Kuantisasi.

ABSTRACT

Name : Agus Buono
Study Program : Computer Science
Title : HOS Representation and MFCC Model as Speech Signal Feature Extraction for Speaker Identification System in the Noisy Environment Using HMM

This research is addressed to investigate the *higher order statistics* (HOS) and *Mel-Frequency Cepstrum Coefficients* (MFCC) as speech signal feature extraction technique and integrated with *Hidden Markov Model* (HMM) as classifier to form a speaker identification system that more robust to Gaussian Noise. The experiments is focused on the subsystem of feature extraction, whereas in the subsystem of classifier, we use the HMM. In this research, we show the ineffectiveness of 1D-MFCC as feature extraction in the noisy environment empirically, analysis the problem and propose some techniques for feature extraction to handle the problem. Next, we conduct a series of experiments to show the effectiveness of the propose methods. Finally, we make a comparison among methods to capture the characteristics of each and propose a prototype of speaker identification system.

According to the result, the main problem with 1D-MFCC is in the aspect of its input, i.e. power spectrum. This quantity is not stable enough with existing noise. In this research we replace the power spectrum by bispectrum that more robust to noise. Then, the propose methods is focused on how to extract the bispectrum value and integrate with HMM to form the speaker identification system. Firstly, 1D-MFCC extended into 2D-MFCC, so the technique workable for bispectrum value as the input. In order to improve the system performance, we use scalar and vector quantization for bispectrum value representation and continue with wrapping and cosines transform prior to classifier process.

The experiments show that the conventional method based on power spectrum (1D-MFCC) gives a good result for signal without addition by Gaussian noise, with 98.8% of accuracy. Nevertheless, with noise only 20 dB, the system performance drop significantly with accuracy below 50%. The noise canceling technique can improve the accuracy up to 77.7%, but fails for noise more than 20 dB.

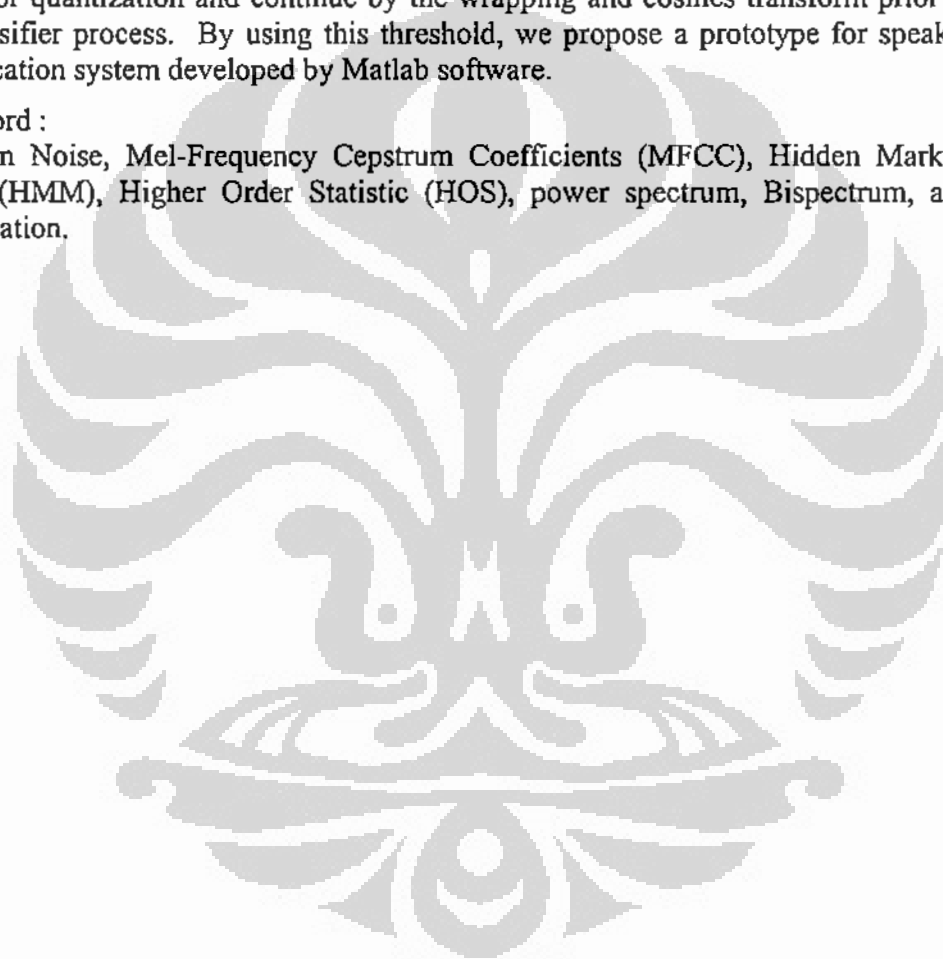
The 2D-MFCC that developed using bispectrum as speech signal feature gives 99.9% of accuracy for original signal and 88.8% for signal corrupted by 20 dB of noise. Compare with 1D-MFCC, this system performance is higher. Nevertheless, for noise more than 20 dB, the system fails. In order to improve the system performance, we propose scalar and vector quantization for representation the bispectrum value, and continue with wrapping and cosines transform prior to classifier process. The vector quantization technique yield the system more stable with noise, and gives the highest recognition compare with others, especially for signal corrupted by noise. The accuracy for signal with addition by 20 dB and 10 dB

of noise are 89% and 75.5%, respectively. But, for original signal, the accuracy is only around 90%. It means the effective technique for feature extraction depend on the quality of input signal.

According to the exploration of autocorrelation of speech signal, it is shown that the signal quality can be divided by the negative value of multiplication of absolute value of its autocorrelation from lag 1 until lag 21. The threshold lies between 7 and 15. If the value is small enough, it is better for use the 1D-MFCC technique. Otherwise, we advise to use the system based on bispectrum represented by vector quantization and continue by the wrapping and cosines transform prior to the classifier process. By using this threshold, we propose a prototype for speaker identification system developed by Matlab software.

Key Word :

Gaussian Noise, Mel-Frequency Cepstrum Coefficients (MFCC), Hidden Markov Model (HMM), Higher Order Statistic (HOS), power spectrum, Bispectrum, and Quantization.



DAFTAR ISI

	Halaman
HALAMAN JUDUL	i
HALAMAN PERNYATAAN ORISINALITAS	ii
HALAMAN PENGESAHAN	iii
KATA PENGANTAR	iv
HALAMAN PERSETUJUAN PUBLIKASI KARYA ILMIAH	vi
ABSTRAK	vii
DAFTAR ISI	xi
DAFTAR GAMBAR	xiv
DAFTAR TABEL	xxi
DAFTAR LAMPIRAN	xxii
BAB 1. PENDAHULUAN	
1.1. Latar Belakang	1
1.2. Motivasi	4
1.3. Tujuan	5
1.4. Permasalahan dan Ruang Lingkup Penelitian	5
1.5. Kontribusi Penelitian	6
1.6. Data yang Digunakan	7
1.7. Struktur Penulisan	9
BAB 2. METODOLOGI UMUM DAN TRANSFORMASI SINYAL MENJADI INFORMASI	
2.1. Metodologi Umum	13
2.2. Transformasi Sinyal Menjadi Informasi	17
2.2.1. Sinyal	18
2.2.2. Sampling	23
2.2.3. Kuantisasi dan Coding	25
2.2.4. Pembacaan Sinyal	26
2.2.5. Ekstraksi Ciri	29
2.2.6. Hidden Markov Model Sebagai Pengenal Pola	38
BAB 3. SISTEM IDENTIFIKASI PEMBICARA	
3.1. State of the Art Pemrosesan Suara dan Bahasa	47
3.2. Ranah Pemrosesan Suara	51
3.3. Sistem Identifikasi Pembicara	55

3.4. Perkembangan Penelitian Bidang Identifikasi Pembicara	61
3.5. Pertanyaan Penelitian	65
3.6. Beberapa Temuan dan Tahapan selanjutnya	67
BAB 4. SISTEM IDENTIFIKASI PEMBICARA BERBASIS POWER SPEKTRUM DALAM DOMAIN FREKUENSI	
4.1. Sistem Identifikasi Pembicara Menggunakan <i>Codebook</i>	70
4.1.1. Kerangka Pikir	70
4.1.2. Self organizing Feature Map (SOFM)	72
4.1.3. Hasil Percobaan	74
4.2. Sistem Identifikasi Pembicara Menggunakan HMM	75
4.2.1. <i>Adaptive Noise Canceling</i>	77
4.2.2. Seting Percobaan dan Hasil	79
4.3. Beberapa Temuan dan Tahapan Selanjutnya	85
BAB 5. SISTEM IDENTIFIKASI PEMBICARA MENGGUNAKAN 2D-MFCC SEBAGAI EKSTRAKSI CIRI DAN HMM SEBAGAI CLASSIFIER	
5.1. <i>Higher Order Statistics</i> , HOS	89
5.1.1. Sinyal Stokastik	90
5.1.2. Pendugaan Bispektrum	95
5.2. Perluasan 1D-MFCC ke 2D-MFCC	100
5.3. Algoritma Genetika untuk Membentuk Filter Dua Dimensi	105
5.3.1. Algoritma Genetika	105
5.3.2. Ide Pembentukan Filter Dua Dimensi dengan GA	106
5.4. Rancangan Percobaan dan Hasil	110
5.4.1. Rancangan Percobaan	110
5.4.2. Hasil	112
5.5. Beberapa Temuan dan Tahapan Selanjutnya	114
BAB 6 TEKNIK KUANTISASI UNTUK REPRESENTASI NILAI BISPEKTRUM SEBAGAI PENCIRI SINYAL PADA SIP DENGAN HMM SEBAGAI CLASSIFIER	
6.1. Kuantisasi Skalar	117
6.1.1. Reduksi Dimensi	120
6.1.2. Hasil Percobaan	124
6.1.3. Beberapa Temuan dan Tahapan selanjutnya	128
6.2. Kuantisasi Vektor	129

6.2.1. Tahapan Kuantisasi Vektor	129
6.2.2. Hasil Percobaan	133
6.2.3. Beberapa Temuan dan Tahapan selanjutnya	139
6.3. Kaji Lanjut Parameter pada Teknik Kuantisasi Vektor ...	140
6.3.1. Parameter Transformasi WC	141
6.3.2. Hasil Percobaan	146
6.3.3. Beberapa Temuan dan Tahapan selanjutnya	153
BAB 7 ANALISIS PERBANDINGAN KINERJA DAN RANCANGAN SISTEM IDENTIFIKASI PEMBICARA	
7.1. Sistem Berbasis Power Spektrum vs Bispektrum	154
7.2. 2D-MFCC vs Kuantisasi Skalar dan Vektor	156
7.3. HMM vs PNN	160
7.4. Rancangan Sistem Identifikasi Pembicara	161
7.5. Beberapa Temuan	166
BAB 8. KESIMPULAN DAN SARAN	169
DAFTAR REFERENSI	172
LAMPIRAN	176

DAFTAR GAMBAR

Nomor	Teks	Halaman
Gambar 1.1.	Beberapa Sinyal Sesuai Cara Pengucapan dari Seorang Pembicara	8
Gambar 1.2.	Ilustrasi Penghapusan Bagian <i>Silence</i>	8
Gambar 1.3.	Diagram Alur Penulisan Disertasi dan Keterkaitan antar Bab	9
Gambar 2.1.	Kerangka Berpikir Penyelesaian Studi	13
Gambar 2.2.	Alur Proses Transformasi Sinyal Suara Analog Menjadi Informasi	18
Gambar 2.3.	Sinyal Waktu Diskret	21
Gambar 2.4.	Proses <i>Frame Blocking</i> dan <i>Windowing</i>	27
Gambar 2.5.	Diagram Alur Teknik MFCC untuk Mengekstrak Sinyal	30
Gambar 2.6.	Beberapa Contoh Tipe Sinyal	31
Gambar 2.7.	Ilustrasi Transformasi DFT	32
Gambar 2.8.	Grafik Hubungan Frekuensi dengan Skala Mel	35
Gambar 2.9.	Beberapa Filter yang Dikenal	36
Gambar 2.10.	Ilustrasi Pembuatan Filter ke 13 dari 20 Filter	37
Gambar 2.11.	Ekstraksi Ciri Teknik MFCC dengan Panjang Frame 256	38
Gambar 2.12.	Ilustrasi X_u , X_t , dan X_s suatu Rantai Markov	39
Gambar 2.13.	Ilustrasi Rantai Markov Orde Satu	40
Gambar 2.14.	Ilustrasi Rantai Markov Orde Dua	40
Gambar 2.15.	Contoh HMM dengan Tiga Hidden State dan Distribusi Emitten <i>Gaussian</i> , (a) <i>Ergodic</i> , (b) <i>Left-Right</i> HMM	42
Gambar 3.1.	Contoh Protokol Aplikasi Berbasis Web	50
Gambar 3.2.	Cakupan Kajian dalam Pemrosesan Suara	51
Gambar 3.3.	Perbedaan Kajian pada Ranah <i>Recognition</i>	52
Gambar 3.4.	Perkembangan Penelitian Bidang Pengenalan Suara	52
Gambar 3.5.	Sistem Pengenalan Pembicara Berdasar Jumlah Kelas Output	53
Gambar 3.6.	Kajian-Kajian pada Ranah Pengenalan Pembicara	54
Gambar 3.7.	Blok Diagram Sistem Identifikasi Pembicara	55
Gambar 3.8.	Klasifikasi Pemodelan Temporal Berbasis Probabilistik	56
Gambar 3.9.	Hubungan <i>Input</i> , <i>Hidden State</i> dan <i>Emitten</i> pada IO-HMM	58
Gambar 3.10.	Hubungan antara Berbagai Ukuran Ketidakpastian	59

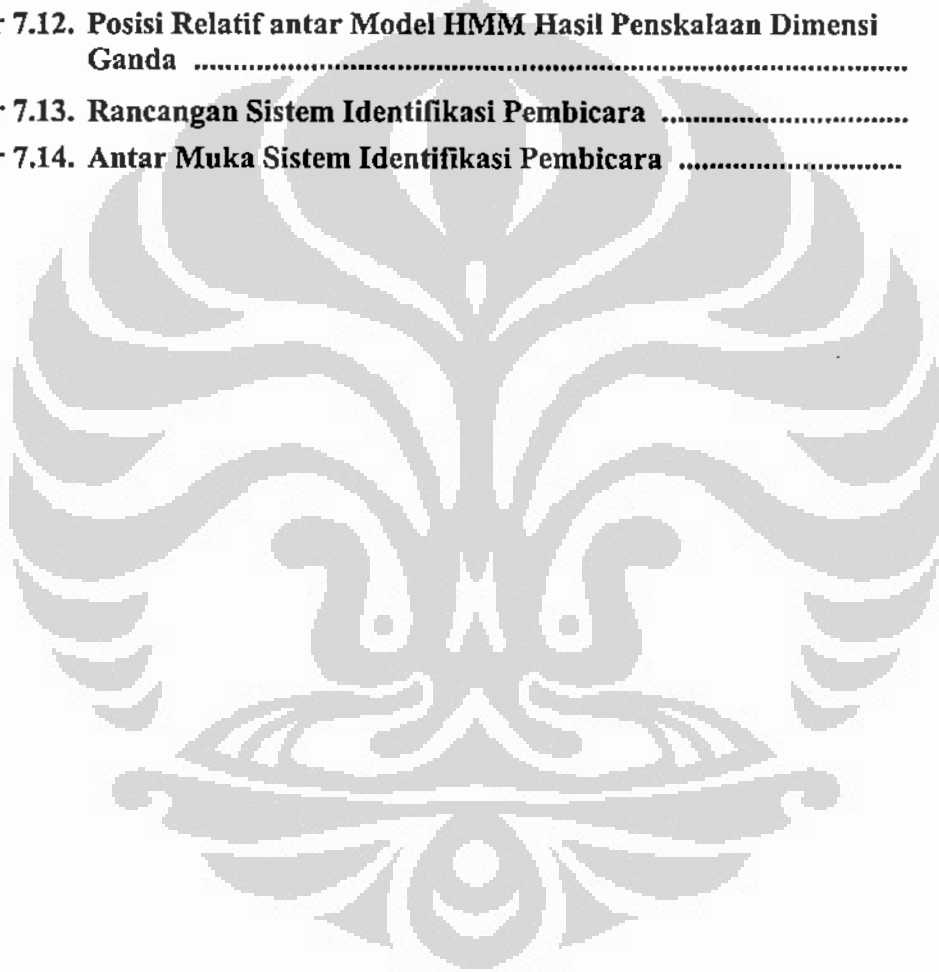
Gambar 3.11. Latice untuk Variasi Model HMM dengan Panjang Tujuh, (a) C-HMM, (b) MM-HMM	60
Gambar 3.12. Contoh Transisi Vertikal (a) dan Horizontal (b) ke State $S_{3,3}$ ($h_{i,j;x,y}$ adalah peluang transisi dari state $S_{i,j}$ ke state $S_{x,y}$)	60
Gambar 3.13. Klasifikasi Metode <i>Classifier</i> pada SIP	62
Gambar 3.14. Perbandingan <i>Gaussian Noise</i> dengan <i>Noise</i> Mesin Mobil	63
Gambar 3.15. Perbandingan Pengaruh Jenis <i>Noise</i> Terhadap Spektrum Sinyal; (a) Spektrum Sinyal Asli; (b), (c), (d), dan (e) adalah Spektrum Sinyal yang sudah Ditambah <i>Noise</i>	64
Gambar 3.16. Perbandingan Sinyal Asli dengan yang Diberi <i>Gaussian Noise</i>	66
Gambar 3.17. Distribusi <i>Noise</i> yang Ditambahkan	67
Gambar 4.1. Tebaran Suara di Sekitar <i>Codebook</i> dari Dua Pembicara	70
Gambar 4.2. Ilustrasi Penghitungan Jarak Input Suara (terdiri 10 frame) dengan <i>Codebook</i> (7 <i>codeword</i> berdimensi dua)	72
Gambar 4.3. Struktur Neuron Output Topologi Bujursangkar 7x7	73
Gambar 4.4. Arsitektur SOFM (Dimensi Input n dan Jumlah Kluster m)	73
Gambar 4.5. Alur Proses Identifikasi Pembicara dengan <i>Codebook</i>	74
Gambar 4.6. Hasil Akurasi Berbasis <i>Codebook</i> (data suara asli)	75
Gambar 4.7. Model <i>Left-Right</i> HMM dengan Tiga State yang Dipergunakan pada Penelitian	76
Gambar 4.8. Perbandingan Autokorelasi antara Sinyal Asli, Sinyal <i>Noise</i> dan Sinyal Asli Terkontaminasi <i>Noise</i>	77
Gambar 4.9. Blok Diagram <i>Adaptive Noise Canceling</i>	78
Gambar 4.10. Perbandingan Sinyal antara Sebelum dengan Setelah ANC	79
Gambar 4.11. Alur Proses Pengenalan Menggunakan MFCC-HMM	80
Gambar 4.12. Akurasi Sistem (%) Menggunakan Teknik MFCC+HMM untuk Berbagai Data Uji dengan Berbagai <i>Noise</i>	81
Gambar 4.13. Perbandingan Pola Power Spektrum antara Sinyal Asli dengan Sinyal asli yang Ditambah <i>Noise</i>	82
Gambar 4.14. Perbandingan Akurasi pada Berbagai Jumlah Hidden State	83
Gambar 4.15. Perbandingan Akurasi pada Empat Jenis Data Training	84
Gambar 4.16. Perbandingan Akurasi untuk Sistem dengan dan tanpa NC pada Berbagai <i>Noise</i>	84
Gambar 4.17. Autokorelasi Sinyal Suara antara Sebelum dan Setelah ANC	85
Gambar 5.1. Kerangka Pikir Pengembangan 1D-MFCC menjadi 2D-MFCC	88
Gambar 5.2. Klasifikasi Penggunaan HOS	89

Gambar 5.3.	Sifat Simetrik <i>Moment</i> (Contoh orde 3, dengan $\tau_1=3, \tau_2=2,$) ...	91
Gambar 5.4.	Daerah <i>Support</i> (a) dan Definisi Bispektrum (b)	95
Gambar 5.5.	Klasifikasi Metode Pendugaan Spektrum Orde Tinggi	95
Gambar 5.6.	Hubungan antara Domain Waktu, Fourier, <i>Moment</i> dan Spektrum <i>Moment</i> , [NP93]	96
Gambar 5.7.	Perbandingan Pola Spektrum (1D dengan 2D) dari Sinyal Asli dengan Sinyal yang Ditambah <i>Noise</i>	99
Gambar 5.8.	Filter untuk Proses <i>Wrapping</i> , (a) 1D-MFCC, (b) 2D-MFCC pada pada Ruang (F1,F2)	101
Gambar 5.9.	Konstruksi Filter H_{ij} pada Ruang 2D dengan Pusat ($f_{1i}f_{2j}$)	102
Gambar 5.10.	Ilustrasi Proses Ekstraksi Ciri dengan 2D-MFCC	104
Gambar 5.11.	Ilustrasi Pengkodean Kromosom dari Satu Set Filter	107
Gambar 5.12.	Alur Proses Penghitungan Fitness	109
Gambar 5.13.	Proses Mutasi Kromosome X menjadi X'	110
Gambar 5.14.	Nilai Total Fitness (a) dan Bentuk Filter Segitiga (b dan c)	110
Gambar 5.15.	Alur Percobaan dengan 2D-MFCC sebagai Ekstraksi Ciri	111
Gambar 5.16.	Perbandingan Akurasi Sistem (%) untuk Data Tanpa Penambahan <i>Noise</i> dari Tiga Metode Ekstraksi Ciri dengan HMM sebagai <i>Classifier</i>	112
Gambar 5.17.	Perbandingan Akurasi Sistem untuk Data Asli dan Data dengan Penambahan <i>Noise</i> 20 dB	112
Gambar 5.18.	Pola Vektor Ciri dari 13 Koefisien, (a) 1D-MFCC, (b) 2D-MFCC tanpa GA, dan (c) 2D-MFCC dengan GA	113
Gambar 5.19.	Perbandingan Akurasi (%) dari Tiga Ekstraksi Ciri untuk Data ber- <i>noise</i> 20 dB dengan jumlah Koefisien 13, 12 dan 11	113
Gambar 5.20.	Perbandingan Akurasi Sistem antara Teknik 2D-MFCC tanpa dan dengan GA pada Jumlah Koefisien 12	114
Gambar 6.1.	Alur Proses Percobaan Identifikasi Pembicara	117
Gambar 6.2.	Kuantisasi Skalar dengan Jumlah Channel 50 untuk Bispektrum dengan $f_{max}=128$ (jumlah indeks per kelompok $[8256/50]=165$)	118
Gambar 6.3.	Distribusi Bispektrum Sebelum dan Setelah Persentil 75%	119
Gambar 6.4.	Empat Puluh Filter pada Proses <i>Wrapping</i>	123
Gambar 6.5.	Alur Proses Ekstraksi Ciri Menggunakan Teknik Skalar Kuantisasi – <i>Wrapping</i> dan Transformasi Kosinus (WC)	124

Gambar 6.6.	Perbandingan Akurasi antara Teknik WC dengan PCA untuk Data Tanpa Penambahan <i>Noise</i>	124
Gambar 6.7.	Perbandingan Akurasi antar Berbagai Jumlah Channel dengan Kuantisasi Skalar-WC untuk Sinyal Suara yang Ditambah <i>Noise</i> 20 dB	125
Gambar 6.8.	Perbandingan Akurasi antara Statistik Rataan, Median dan Rataan BSP di atas Persentil 75% pada Berbagai Channel untuk Sinyal Asli	125
Gambar 6.9.	Perbandingan Akurasi antara Statistik Rataan, Median dan Rataan BSP di atas Persentil 75% pada Berbagai Channel untuk Sinyal Asli dengan Penambahan <i>Noise</i> 20 dB	126
Gambar 6.10.	Rasio Nilai Statistik pada Berbagai Penambahan <i>Noise</i> terhadap Nilainya pada Kondisi Tanpa Penambahan <i>Noise</i> untuk Ketiga Cara Penghitungan	127
Gambar 6.11.	Perbandingan Akurasi antar Channel untuk Berbagai <i>Noise</i>	127
Gambar 6.12.	Alur Proses Kuantisasi Vektor	129
Gambar 6.13.	Alur Proses Pembentukan Bispektrum Gabungan	130
Gambar 6.14.	Blok Diagram Sampling Titik-Titik Domain Bispektrum	131
Gambar 6.15.	Ilustrasi Tahapan Pembentukan Pusat Channel	132
Gambar 6.16.	Ilustrasi Penentuan Nilai Channel ke j yang Mempunyai Empat Titik Berlabel j dengan Statistik Rataan	133
Gambar 6.17.	Perbandingan Akurasi SIP dengan Kuantisasi Vektor dan Transformasi <i>Wrapping</i> dan Kosinus	134
Gambar 6.18.	Perbandingan Akurasi SIP antara Kuantisasi Vektor dengan Kuantisasi Skalar	134
Gambar 6.19.	Perbandingan Akurasi antar Berbagai Jenis Statistik pada Sistem dengan Kuantisasi Vektor dan Transformasi WC	135
Gambar 6.20.	Perbandingan Akurasi SIP antara Berbagai Jenis Statistik dan Jumlah Channel pada Sistem dengan Kuantisasi Vektor dan Transformasi WC	136
Gambar 6.21.	Perbandingan Efektifitas Transformasi <i>Wrapping</i> -Kosinus dengan Kosinus dalam Menangani Masalah <i>Noise</i> pada SIP	136
Gambar 6.22.	Akurasi SIP Berbasis Data Bispektrum dan Kuantisasi Vektor dengan Transformasi <i>Wrapping</i> -Kosinus sebagai Pereduksi Dimensi dan PNN sebagai <i>Classifier</i>	137
Gambar 6.23.	Perbandingan Akurasi SIP Berbasis Data Bispektrum dengan Beberapa Metode yang telah Dikembangkan	138

Gambar 6.24. Filter pada Proses <i>Wrapping</i> Beserta Parameternya	141
Gambar 6.25. Ilustrasi Perbedaan Peubah yang Dikenakan Proses <i>Wrapping</i> , (a) 5 titik channel, (b) <i>wrapping</i> pada posisi channel, w_1 , (c) 5 titik channel beserta jaraknya terhadap pusat sumbu, dan (d) <i>wrapping</i> pada jarak channel ke pusat sumbu, w_2	142
Gambar 6.26. Perbedaan Posisi Titik Peubah yang Dikenakan <i>Wrapping</i> pada Sumbu frekuensi pada Kuantisasi Vektor dengan 400 Channel, (a) 400 titik channel, (b) posisi titik yang di <i>wrapping</i> sesuai urutan channel dan (c) posisi titik yang di <i>wrapping</i> sesuai jarak	143
Gambar 6.27. Contoh Lima Filter Linear dengan Lebar 50	144
Gambar 6.28. Ilustrasi Penentuan Lebar Filter Logaritma	145
Gambar 6.29. Blok Diagram Rancangan Percobaan untuk Identifikasi Pembicara	146
Gambar 6.30. Rata-rata Akurasi Sinyal Asli pada Berbagai Nilai Peubah R	148
Gambar 6.31. Rata-rata Akurasi untuk Sinyal Asli pada Berbagai Nilai Peubah R untuk Kondisi W_1 dan W_2	149
Gambar 6.32. Perbandingan Akurasi pada Sinyal Asli untuk Berbagai Kondisi R, W, D dan G	150
Gambar 6.33. Perbandingan Akurasi antara w_1 dan w_2 pada Sinyal Asli dan Berbagai <i>Noise</i>	151
Gambar 6.34. Perbandingan Akurasi Terbaik antara Kondisi W_1 dengan W_2 pada Sinyal Asli dan yang Telah Memperoleh Penambahan <i>Noise</i>	152
Gambar 7.1. Perbandingan Akurasi Sistem yang Berbasis Power Spektrum dengan yang Berbasis Bispektrum	154
Gambar 7.2. Pola 13 Koefisien Vektor Ciri (1D-MFCC dan 2D-MFCC)	155
Gambar 7.3. Perbandingan Akurasi Sistem dengan Teknik 2D-MFCC pada Berbagai Jumlah Koefisien	155
Gambar 7.4. Perbandingan Akurasi Sistem dengan Teknik 2D-MFCC (tanpa dan dengan GA) serta 1D-MFCC untuk Jumlah Koefisien 12	156
Gambar 7.5. Perbandingan Akurasi Sistem Berbasis Bispektrum dengan Berbagai Teknik Ekstraksi Ciri pada Berbagai Level <i>Noise</i>	157
Gambar 7.6. Pola Data Hasil Pengolahan Sinyal Menjadi Vektor Ciri	158
Gambar 7.7. Perbandingan Akurasi antara 1D-MFCC dengan NC dengan Kuantisasi Vektor dengan dan tanpa NC	160

Gambar 7.8.	Perbandingan Akurasi antara Teknik HMM dengan PNN sebagai Pengenal Pola	161
Gambar 7.9.	Pola Autokorelasi dari Sinyal Asli dan Sinyal Asli yang telah Mengalami Penambahan <i>Noise</i>	162
Gambar 7.10.	Tahapan Proses Penentuan <i>Cut of Point</i>	162
Gambar 7.11.	Hasil Klasifikasi per Kombinasi COP-Sinyal Input dari 10 Pembicara (masing-masing 20 suara)	164
Gambar 7.12.	Posisi Relatif antar Model HMM Hasil Penskalaan Dimensi Ganda	165
Gambar 7.13.	Rancangan Sistem Identifikasi Pembicara	165
Gambar 7.14.	Antar Muka Sistem Identifikasi Pembicara	166



DAFTAR TABEL

Nomor	Teks	Halaman
Tabel 1.1.	Daftar 10 Pembicara yang Digunakan dalam Penelitian	7
Tabel 2.1.	Klasifikasi Filtering	26
Tabel 2.2.	Daftar Window Simetrik dan Ekspresi Matematikanya	28
Tabel 2.3.	Perbandingan Ukuran J, V dan MSE untuk Beberapa Window	28
Tabel 4.1.	Hasil Akurasi (%) dengan Teknik MFCC+HMM pada Berbagai <i>Noise</i>	81
Tabel 6.1.	Perbandingan Akurasi Sistem antara Metode Kuantisasi Skalar dan Vektor dengan <i>Classifier</i> HMM dan PNN	138
Tabel 6.2.	Tabel Analisis Ragam untuk Data Sinyal Asli	147
Tabel 6.3.	Rata-Rata Akurasi untuk Berbagai Nilai R	148
Tabel 6.4.	Hasil Akurasi untuk Sinyal Ber- <i>noise</i> 20 dB dan 10 dB	151
Tabel 7.1.	Perbandingan Akurasi Sistem untuk Berbagai Nilai COP	163
Tabel 7.2.	Jarak Kullback-Leibler antar 10 Model HMM	163

DAFTAR LAMPIRAN

Nomor	Teks	Halaman
Lampiran 1.	Algoritma Forward [Rab89]	L-1
Lampiran 2.	Algoritma Backward [Rab89]	L-1
Lampiran 3.	Algoritma Viterbi [Rab89]	L-2
Lampiran 4.	Algoritma K-Means [DD99]	L-3
Lampiran 5.	Formula Baum-Welch reestimation [Rab89]	L-4
Lampiran 6.	Tahapan Proses Kuantisasi Vektor	L-6
Lampiran 7a.	Hasil Akurasi Sistem dengan Kuantisasi Vektor dengan Wrapping pada Posisi Urutan Channel dan Transformasi Kosinus	L-7
Lampiran 7b.	Hasil Akurasi Sistem dengan Kuantisasi Vektor dan Wrapping pada Jarak Channel dan Transformasi Kosinus	L-8
Lampiran 8a.	Analisis Ragam untuk Akurasi Sistem dengan Teknik Kuantisasi Vektor pada Data Sinyal Asli + Noise 20 dB	L-9
Lampiran 8b.	Analisis Ragam untuk Akurasi Sistem dengan Teknik Kuantisasi Vektor pada Data Sinyal Asli + Noise 10 dB	L-10
Lampiran 9.	Daftar Paper yang Dipublikasikan	L-11

BAB 1 PENDAHULUAN

1.1 Latar Belakang

Seiring dengan perkembangan teknologi informasi, maka tuntutan manusia untuk memanfaatkannya guna mempermudah kehidupan sehari-hari juga makin bervariasi. Salah satu hal yang sudah dipikirkan sejak lama adalah keinginan untuk membuat komputer mampu berkomunikasi secara alami dengan manusia. Satu sistem cerdas yang pertama kali dikembangkan adalah ELIZA pada tahun 1966, yaitu suatu *artificial agent* yang mampu bercakap-cakap secara terbatas dengan user [JM00]. Ilmu yang membahas bidang ini dikenal dengan nama Pemrosesan Suara dan Bahasa Alami (*Speech and Natural Language Processing*).

Secara umum ada dua paradigma dalam bidang tersebut, yaitu paradigma stokastik dan paradigma logika formal [JM00]. Paradigma logika formal difokuskan pada pengembangan grammar (*metamorphosis grammar, definite clause grammars, functional grammars*) juga struktur unifikasi. Sedangkan Paradigma stokastik pada umumnya pada pemrosesan data suara, baik pada tahap praproses maupun pada pengenalan pola. Pada perkembangannya sekarang ini, bidang tersebut sudah meluas, sehingga tidak hanya masalah pemrosesan (*processing*), tetapi juga mencakup *speech understanding and generation*. Hal ini memunculkan bidang baru yang dikenal dengan Komputasi Linguistik dan Kecerdasan Buatan (*Computational Linguistics and Artificial Intelligence*), [Den08]. Oleh karena itu aplikasi dari bidang *Speech Processing and Understanding* menjadi luas mulai dari *document summarization (coding), transmission, text parsing (analysis), spelling/grammar correction (enhancement), natural language generation (synthesis), natural language understanding (understanding), web search (retrieval/mining)*, serta banyak lagi pada aspek *recognition* seperti mesin pendikte, identifikasi maupun verifikasi pembicara, mesin penjawab otomatis, serta interaksi manusia dengan komputer lainnya melalui suara. Selain dari aspek terapan yang begitu luas, investasi yang diperlukan lebih pada aspek *software* (bukan *hardware*), sehingga biaya yang diperlukan untuk

pengembangan produk lebih murah. Hal-hal tersebut yang menjadi alasan mengapa penelitian di bidang *speech and language processing* menarik untuk dilakukan.

Suara merupakan satu fenomena sebagai perpaduan multidimensi, mulai dari dimensi linguistik, semantik, artikularis dan akustik [Cam97]. Dimensi linguistik dan semantik bersifat *linguistic dependent*, sedangkan artikularis dan akustik bersifat *linguistic independent*. Dimensi artikularis secara detail menjadi kajian dari bidang fonologi, yang mempelajari bagaimana suara dan jenis-jenisnya dihasilkan. Sedangkan akustik terbagi menjadi dua bagian, yaitu yang mengkaji sinyal suara secara fisik dan bagian lainnya yang melakukan analisis terhadap sinyal suara. Oleh karena suara merupakan fungsi yang kompleks dari beberapa aspek, seperti karakteristik pembicara (dimensi titik artikularis, emosi, kesehatan, umur, jenis kelamin, dialek), bahasa, dan lingkungan (*background* dan media transmisi), maka pemodelan sinyal bukanlah hal yang mudah dan masih menantang untuk dilakukan.

Dengan latar belakang tersebut maka penelitian yang dilakukan pada disertasi ini adalah dalam hal pemrosesan sinyal suara untuk keperluan pengenalan dan difokuskan pada sistem identifikasi pembicara. Sistem identifikasi pembicara mengenali pembicara berdasarkan suara, yang merupakan ciri biometrik seseorang yang bersifat lebih dinamis dibanding ciri biometrik lainnya, misalkan sidik jari dan tanda retina. Sifat dinamis ini disebabkan oleh beberapa hal, seperti umur, kesehatan, emosi, cara pengucapan akan menyebabkan adanya *intraspeaker variability* (variasi pada seorang pembicara). Selain masalah *intraspeaker variability*, juga adanya *noise* yang disebabkan oleh lingkungan, dan distorsi karena alat akan menjadi sumber error, yang pada akhirnya menurunkan akurasi sistem. Oleh karena itu, meskipun beberapa hasil penelitian telah menunjukkan akurasi yang tinggi (>95%), hal ini masih terbatas pada sinyal suara yang dikondisikan, sehingga akurasi sistem akan menurun secara nyata saat diujicobakan dalam *real life situation* [Car95]. Dalam kondisi real, adanya noise dan variasi internal pembicara adalah fakta yang tidak bisa dihindari, sehingga penelitian di bidang ini masih diperlukan dan layak untuk dilakukan guna memperoleh hasil yang lebih baik.

Seperti disebutkan dalam [Rey02] bahwa persyaratan ciri biometrik sebagai pengenalan seseorang, adalah bersifat alami, mudah diukur, tidak terlalu berubah dari waktu ke waktu, tidak mudah ditiru, tidak dipengaruhi kondisi fisik, serta tidak terlalu terganggu dengan adanya gangguan lingkungan. Suara adalah besaran yang hampir memenuhi semua kriteria tersebut, kecuali dua sifat terakhir, yaitu persyaratan tidak dipengaruhi kondisi fisik, serta tidak terlalu terganggu dengan adanya gangguan lingkungan. Oleh karena itu, dalam penelitian ini akan dilakukan serangkaian percobaan untuk mendapatkan teknik yang mampu mengatasi masalah gangguan dikarenakan *noise* pada sinyal suara .

Satu permasalahan pada pengenalan suara dan hal ini juga umum terjadi pada bidang terapan lainnya adalah pada tahap ekstraksi ciri dari data masukan menjadi vektor ciri. Jika proses ekstraksi ciri dapat menghasilkan vektor ciri yang efektif mampu mencirikan obyek masukan tanpa terpengaruhi oleh adanya gangguan, maka proses pengenalan menjadi jauh lebih mudah. Satu teknik ekstraksi ciri sinyal suara yang populer saat ini adalah *Mel-Frequency Cepstrum Coeficients* (MFCC). Teknik ini telah terbukti mampu menghasilkan vektor ciri yang efektif, sehingga akurasi sistem yang diperoleh mencapai di atas 99% untuk sinyal tanpa penambahan noise, [BB08]. Namun demikian, untuk sinyal suara dengan penambahan noise yang hanya 20 dB, sistem yang dihasilkan gagal dan akurasi yang dicapai kurang dari 50%. Tahap pertama dari proses ekstraksi ciri menggunakan MFCC adalah menghitung nilai power spektrum dari sinyal input, dan diikuti dengan tahapan lanjutnya, yaitu proses wrapping, dan transformasi kosinus, yang akhirnya menghasilkan vektor ciri dari setiap frame sinyal yang dibaca. Dari studi eksploratif yang sudah dilakukan, [BJB08]^a, terlihat bahwa nilai power spektrum sebagai masukan dari MFCC sangat sensitif terhadap adanya gangguan *noise*. Hal inilah yang menyebabkan kegagalan sistem bekerja pada situasi bernois.

Sedangkan pada tahapan pengenalan pola, teknik *Hidden Markov Model* (HMM) telah umum dipakai sebagai *classifier* pada pemrosesan suara. Hal ini sesuai dengan literatur yang ada bahwa pemodelan statistik menjadi trend bagi

pengembangan sistem pemrosesan sinyal. Juga dari studi empiris yang sudah dilakukan [BJB08]^b, teknik HMM menunjukkan hasil yang lebih baik dibanding teknik berbasis jaringan syaraf tiruan. Hal ini dikarenakan HMM merupakan model stokastik yang menjelaskan hubungan antara kemunculan peubah teramati (*observable state*) sebagai outcome dari peubah tak teramati (*hidden state*) dari waktu ke waktu. Oleh karena itu model ini memiliki analogi dengan proses alamiah suatu ujaran dihasilkan. Dalam hal ini *hidden state* dapat dipandang sebagai representasi konfigurasi artikularis dan *observable state* mencerminkan sinyal suara yang dihasilkan. Oleh karena itu model HMM mampu mengakomodasi variasi antar unit waktu yang terjadi pada sinyal suara. Sementara itu, pada jaringan syaraf tiruan, informasi yang diperoleh merupakan rata-rata dari seluruh unit waktu, sehingga variasi antar unit waktu menjadi kurang terlihat.

Dari dua fakta di atas, dapat disebutkan bahwa bahwa tahapan ekstraksi ciri menjadi hal yang kritis pada proses pengenalan suara. Oleh karena itu, pada penelitian ini, untuk mengatasi gangguan *noise* yang muncul pada sistem pengenalan pembicara difokuskan pada bagian ekstraksi ciri. Dalam hal ini menggunakan teknik yang dilakukan pada metode MFCC, namun komponen input yang dipakai adalah suatu besaran yang mampu menangkap karakteristik sinyal suara dan bersifat *robust* terhadap *noise*. Output dari ekstraksi ciri ini selanjutnya menjadi masukan bagi tahap pengenalan menggunakan HMM.

1.2 Motivasi

Pada [NP93] disebutkan bahwa nilai spektrum orde tinggi (*Higher Order statistics*, HOS) untuk orde 3 (bispektrum) dari sinyal yang berdistribusi *Gaussian* adalah nol. Fakta ini diharapkan mampu menekan pengaruh *Gaussian noise* pada sinyal, yang pada akhirnya sistem yang dikembangkan lebih *robust* terhadap *noise*. Kalau besaran power spektrum mendeskripsikan sinyal sebagai proses *Gaussian* dan menggali informasi dari frekuensi dalam domain waktu tanpa memperhatikan informasi tersembunyi dari fase. Sedangkan spektrum orde tinggi menggali informasi dari magnitudo dan fase sinyal dalam domain frekuensi, [RN85].

Analisis spektrum orde tinggi ini mempunyai kemampuan untuk, [NM87]: 1. mengekstrak informasi berkaitan dengan penyimpangan sifat *Gaussian*, sehingga *Gaussian noise* yang bersifat aditif bisa diminimalkan, 2. untuk mendeteksi dan mengkarakterisasi sifat-sifat *nonlinearitas*. Bukti empiris untuk hal ini telah ditunjukkan, [BJB08]^a, bahwa secara visual terlihat nilai spektrum orde tiga (bispektrum) bersifat tegar terhadap pengaruh *noise* dibandingkan dengan power spectrum. Alberta et. al., 2005 menunjukkan kemampuan Statistik orde 4 untuk *blind source separation*, dan memberikan hasil yang baik untuk sinyal yang ditambah *Gaussian Noise* hingga 15 dB, [AFCC05]. Pada [FB98], [HB99], dan [TB00], diperlihatkan bahwa sistem identifikasi pembicara yang dibangun dengan berbasis pada HOS menghasilkan sistem yang *robust* terhadap *noise*. Namun demikian metode yang dihasilkan menggabungkan seluruh frame yang dibaca dari satu ujaran menjadi satu frame gabungan, sehingga teknik tersebut kurang fleksibel untuk dikembangkan pada berbagai bidang terapan, seperti pada konversi suara ke teks, mesin translasi, serta kurang mengakomodasi variasi antar frame. Fakta-fakta tersebut memberikan motivasi untuk mengeksplotasi kemampuan HOS sebagai pencari sinyal untuk pengembangan sistem identifikasi pembicara.

1.3 Tujuan

Tujuan penelitian ini adalah :

1. Mengintegrasikan teknik HOS dengan HMM pada sistem identifikasi pembicara
2. Mengkaji teknik HOS sebagai penentu ciri sinyal suara
3. Menemukan teknik representasi nilai HOS yang optimum
4. Membangun prototipe model sistem identifikasi pembicara yang bersifat lebih *robust* terhadap *noise*.

1.4 Permasalahan dan Ruang lingkup Penelitian

Dengan menggunakan HMM sebagai pengenalan pola, maka skor sinyal suara dihitung dengan membaca sinyal tersebut frame demi frame dan berikutnya setiap frame masuk ke algoritme evaluasi pada HMM. Dengan lebar frame berkisar 30 ms dan overlap antar frame 40%, dari satu suara dengan durasi 1 detik akan diperoleh

sekitar 70 frame. Dengan HOS sebagai penentu ciri dan disampling pada frekuensi maksimum 128, maka dari setiap frame akan diperoleh vektor berdimensi 8256. Dengan dimensi yang begitu besar ini, maka praktis HMM tidak dapat bekerja. Selain masalah kompleksitas penghitungan HOS, dimensi HOS yang tinggi inilah yang menjadi masalah utama dari integrasi HOS dengan HMM. Oleh karena itu diperlukan pemrosesan lanjut terhadap nilai HOS setiap frame sehingga diperoleh vektor ciri dengan dimensi yang jauh lebih kecil, sehingga HMM dapat bekerja dengan baik. Permasalahan inilah yang akan dicari solusinya dalam penelitian ini.

Penelitian yang dilakukan difokuskan sesuai ruang lingkup berikut :

1. Fokus bidang kajian adalah pada sistem identifikasi pembicara yang bersifat *text-dependent*.
2. Penelitian ini dibatasi pada penanganan gangguan eksternal yang berupa *noise*, dan difokuskan pada *Gaussian Noise* yang berkisar dari 20 dB hingga 0 dB.
3. Penentu ciri yang dikaji adalah *Higher Order Statistic Orde 3* (Bispektrum).
4. Metode *classifier* yang dipergunakan adalah model markov tersembunyi, (*Hidden Markov Model*, HMM).
5. Implementasi sistem identifikasi pembicara menggunakan software Matlab.

1.5 Kontribusi Penelitian

Kontribusi penelitian ini adalah sebagai berikut :

1. Model integrasi Bispektrum dengan HMM yang bersifat lebih *robust* terhadap gangguan *noise*.
2. Perluasan MFCC satu dimensi ke dua dimensi dan formulasi pembentukan filternya dengan algoritme genetika.
3. Penentuan statistik HOS yang sesuai sebagai penentu ciri sinyal suara.

4. Arsitektur model komputasi untuk pemrosesan suara dalam lingkungan *noisy*.

1.6 Data yang Digunakan

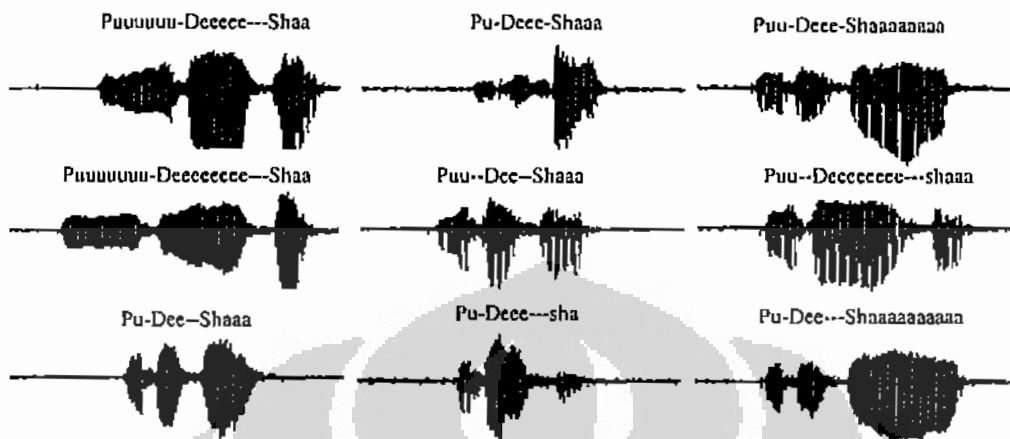
Semua studi empiris yang dilakukan pada penelitian ini menggunakan data yang berasal dari 10 pembicara yang mengucapkan kata "PUDESHA" sebanyak 80 kali tanpa dikontrol cara pengucapannya. Hal ini dengan maksud untuk menunjukkan bahwa model yang dihasilkan mampu bekerja pada suara dengan berbagai cara pengucapan (keras-lemah, panjang-pendek, maupun tinggi-rendah). Karakteristik ke sepuluh pembicara tersebut adalah seperti disajikan pada Tabel 1.1.

Tabel 1.1. Daftar 10 Pembicara yang Digunakan dalam Penelitian

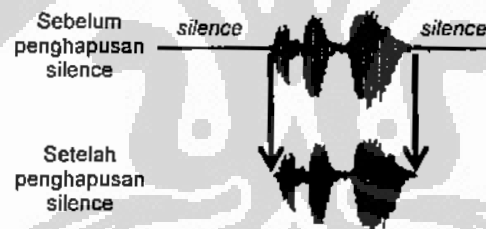
Pembicara	Jenis Kelamin	Umur (th)	Pekerjaan	Suku
1	Pria	40	PNS	Jawa
2	Pria	32	Karyawan	Sunda
3	Pria	23	Mhsw	Jawa
4	Pria	37	PNS	Batak
5	Pria	37	PNS	Jawa
6	Pria	30	PNS	Sunda
7	Pria	23	MHSW	Jawa
8	Wanita	9	Pelajar	Jawa-Minang
9	Wanita	6	Pelajar	Jawa-Minang
10	Pria	23	MHSW	Jawa-Sunda

Setiap suara disampling dengan frekuensi 1.1 kHz dan durasi 1.28 detik (sehingga diperoleh sinyal digital dengan panjang 14080) yang dilakukan dalam ruang kerja yang tertutup dan diambil dengan berbagai cara pengucapan namun belum mewakili variasi kondisi fisik (kesehatan) serta emosi pembicara. Gambar 1.1. adalah contoh variasi cara pengucapan dari seorang pembicara tertentu yang digunakan dalam model. Pada Gambar 1.1. terlihat bahwa dari satu durasi perekaman, ada bagian atau segmen yang tidak ada suara (*silence*), yang umumnya terjadi dibagian depan dan belakang. Untuk meningkatkan akurasi sistem, maka sebelum memasuki tahap ekstraksi, dilakukan proses penghapusan bagian *silence* dengan menggunakan batas sebesar rata-rata absolut bagian *silence* ditambah 1,96

dikali standar deviasinya. **Gambar 1.2.** memberikan input dan output proses penghapusan bagian *silence*.



Gambar 1.1. Beberapa Sinyal Sesuai Cara Pengucapan dari Seorang Pembicara Setelah dilakukan penghapusan bagian *silence*, dilakukan standarisasi sinyal suara dengan membagi sinyal dengan nilai mutlak simpangan maksimum, sehingga akan diperoleh sinyal dengan simpangan maksimum 1 atau -1.



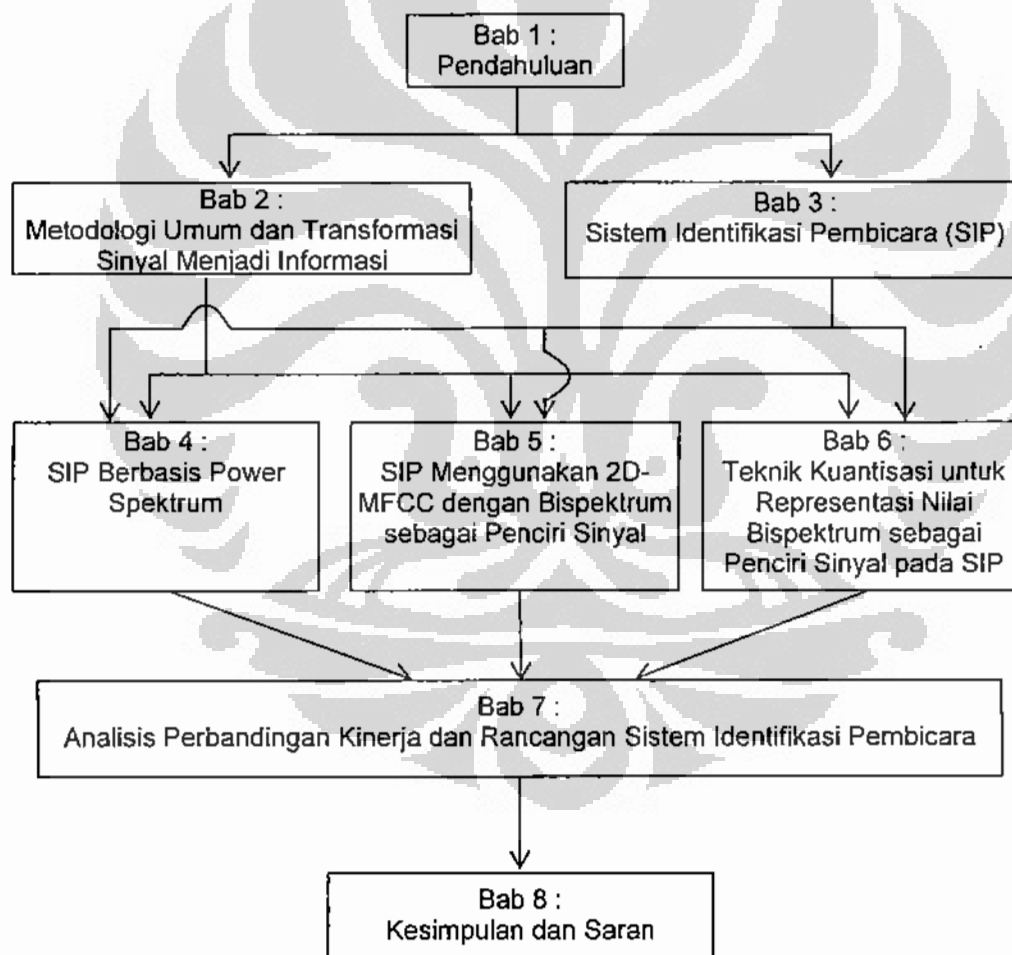
Gambar 1.2. Ilustrasi Penghapusan Bagian *Silence*

Sesuai dengan fokus dari penelitian ini yaitu untuk membangun model yang lebih bersifat *robust* terhadap *noise*, maka diperlukan sinyal *noise* yang akan mengkontaminasi sinyal asli secara aditif. Sinyal *noise* yang ditambahkan bersifat *Gaussian* dengan level 20 dB, 10 dB, 5 dB dan 0 dB. Pemilihan ini didasarkan pada fakta empiris yang telah dilakukan bahwa *noise* 20 dB mulai terasa pengaruhnya terhadap sinyal suara. Berikutnya adalah penambahan sinyal *noise* sesuai level di atas dengan menggunakan paket Matlab melalui instruksi `AWGN(sinyal asli, level noise)` terhadap semua data sinyal yang telah dihapuskan bagian *silence* tersebut.

Dengan demikian, pada akhirnya diperoleh lima set sinyal yang terdiri 800 data sinyal per set, masing-masing untuk sinyal asli, sinyal asli ditambah noise 20 dB, sinyal asli ditambah noise 10 dB, sinyal asli ditambah noise 5 dB, dan sinyal asli ditambah noise 0 dB.

1.7 Struktur Penulisan

Penulisan disertasi ini disusun mengikuti alur seperti yang disajikan pada diagram keterkaitan antar bab pada **Gambar 1.3**. Hal-hal yang melatarbelakangi dan motivasi mengapa riset ini dilakukan, tujuan, ruang lingkup, kontribusi, serta data yang digunakan disajikan pada Bab 1.



Gambar 1.3. Diagram Alur Penulisan Disertasi dan Keterkaitan antar Bab

Gambaran menyeluruh mengenai disertasi serta pendekatan atau metode umum yang dilakukan dituangkan pada Bab 2. Pada pemrosesan suara, terjadi transformasi data dari sinyal analog yang dihasilkan oleh getaran udara yang berasal dari paru-paru melalui saluran pernafasan dengan berbagai bentuk konfigurasi artikulator yang senantiasa berubah, hingga menjadi butir-butir informasi sesuai yang diinginkan. Untuk memberikan pemahaman ini lebih rinci, maka pada Bab 2 juga menyajikan tahapan serta konsep teori dari transformasi tersebut, mulai dari pengertian sinyal, digitalisasi, bagaimana sinyal dibaca, diekstraks dan dilakukan pengenalan.

Untuk memberikan pengertian yang lebih jelas mengenai sistem identifikasi pembicara serta mengetahui posisinya dalam ranah pemrosesan sinyal, maka pada Bab 3 disajikan bahasan mengenai lingkup kajian pemrosesan sinyal. Bahasan pada Bab 3 dimulai dengan *state of the art* dari pemrosesan sinyal dan bahasa, dilanjutkan dengan menyajikan komponen-komponen dari Sistem Identifikasi Pembicara (SIP), serta perkembangan penelitian yang telah ada. Bagian akhir bab ini akan menyajikan beberapa pertanyaan penelitian yang mengarahkan tahapan selanjutnya.

Setelah memahami alur transformasi sinyal menjadi informasi dan konsep teori yang melandasinya serta pertanyaan penelitian yang akan dijawab, maka Bab 4 sebagai bagian dari strategi pendekatan yang dilakukan akan menyajikan percobaan identifikasi pembicara menggunakan teknik standar yang sudah ada. Bab yang akan memberikan bukti empiris kegagalan teknik standar ini berisi dua bagian, yaitu identifikasi dengan berbasis template menggunakan *codebook*, dan yang berbasis model statistik menggunakan HMM dengan teknik ekstraksi ciri yang sudah umum digunakan, yaitu MFCC. Maksud dari Bab ini adalah untuk menunjukkan bahwa model baku dapat bekerja dengan baik untuk sinyal tanpa penambahan *noise*, namun gagal bekerja untuk sinyal dengan penambahan *noise*, bahkan dengan teknik penghapusan *noise* (*noise cancelling*) sekalipun. Dari Bab 4 ini diperlihatkan bahwa permasalahannya adalah pada input dari proses MFCC yang bersifat sensitif terhadap *noise*. Permasalahan ini dicoba untuk diatasi dengan menggunakan nilai statistik orde tiga (Bispektrum) yang bersifat lebih *robust* terhadap *noise* sebagai masukan dari

proses ekstraksi ciri MFCC. Oleh karena domain bispektrum adalah pada bidang (f_1, f_2) , maka teknik MFCC yang telah ada perlu diperluas ke dua dimensi, yang secara detail disajikan pada Bab 5.

Bab 5 menyajikan proses perluasan teknik satu dimensi MFCC (1D-MFCC) menjadi dua dimensi (2D-MFCC). Perluasan tersebut pada intinya adalah pada perubahan filter yang semula satu dimensi menjadi dua dimensi. Dalam hal ini dicobakan dua pendekatan untuk pembuatan filter, yaitu filter standard seperti yang diperoleh pada 1D-MFCC dan pembentukan filter dengan algoritme genetika. Pada Bab 5 ini diperlihatkan bahwa teknik yang dikembangkan memberikan hasil yang lebih baik dibanding 1D-MFCC, namun gagal bekerja untuk level *noise* yang lebih besar berat dari 20 dB. Satu kemungkinan penyebab kegagalan teknik ini adalah karena bispektrum dibaca langsung per titik (f_1, f_2) , maka variasi acak nilai bispektrum cukup tinggi. Hal ini berakibat distribusi spasial bispektrum kurang terlihat. Oleh karena itu dilakukan kuantisasi sebagai representasi nilai bispektrum untuk memperkecil pengaruh variasi acak ini, yang disajikan pada Bab 6.

Bab 6 menyajikan hasil percobaan identifikasi pembicara dengan nilai bispektrum yang sudah dikuantisasi, baik dengan kuantisasi skalar maupun kuantisasi vektor, dan dilanjutkan dengan transformasi wrapping dan kosinus seperti pada proses MFCC. Pada Bab 6 ini diperlihatkan bahwa teknik kuantisasi skalar memberikan hasil yang lebih baik, namun gagal untuk *noise* yang lebih besar dari 20 dB seperti pada 2D-MFCC pada Bab 5. Satu kemungkinan penyebabnya adalah bahwa kuantisasi skalar mampu menekan variasi acak nilai bispektrum, namun tetap masih belum mencerminkan distribusi spasial dari nilai bispektrum tersebut. Oleh karena itu percobaan dilanjutkan dengan melakukan teknik kuantisasi vektor terhadap nilai bispektrum. Pada kuantisasi vektor pembentukan titik channel diarahkan mengikuti distribusi spasial nilai bispektrum. Titik channel tersebut dibentuk melalui analisis kluster terhadap sejumlah titik yang diperoleh melalui sampling acak terhadap titik-titik pada domain dengan peluang terpilihnya mengikuti nilai relatif bispektrum di titik tersebut. Hasil percobaan menunjukkan bahwa sistem yang dihasilkan bersifat lebih *robust* terhadap *noise*. Juga ditunjukkan bahwa teknik HMM sebagai pengenalan

pola memiliki kinerja yang lebih bagus dibanding dengan teknik PNN (*Probabilistics Neural Network*). Untuk mengetahui parameter yang optimum pada teknik ekstraksi ciri dengan kuantisasi vektor ini, maka Bab 6 juga menyajikan hasil percobaan faktorial dengan empat faktor pada kuantisasi vektor yang diulang tiga kali.

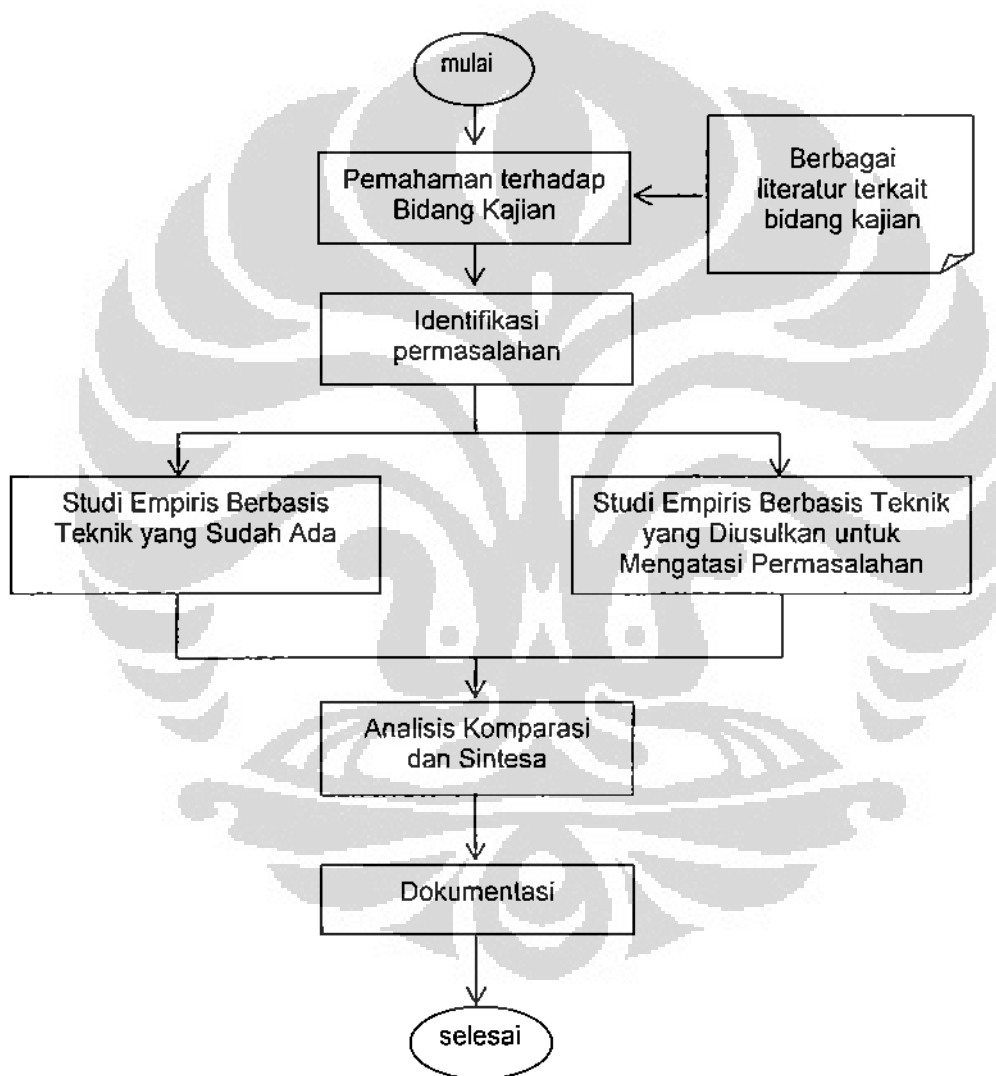
Bab 7 menyajikan perbandingan kinerja dari semua teknik yang sudah dilakukan, dan memberikan hasil bahwa teknik ekstraksi ciri yang efektif tergantung dari kualitas sinyal masukan. Oleh karena itu pada bab ini dilakukan studi eksploratif untuk mendapatkan besaran yang mampu mengenali sinyal sesuai kualitasnya. Dari sini diusulkan satu rancangan sistem identifikasi pembicara dan diimplementasikan dengan software Matlab. Sebagai penutup, kesimpulan akhir beserta saran untuk penelitian berikutnya disajikan pada Bab 8.



BAB 2 METODOLOGI UMUM DAN TRANSFORMASI SINYAL MENJADI INFORMASI

2.1. Metodologi Umum

Pendekatan yang dilakukan pada penelitian ini untuk mencapai tujuan serta mengatasi permasalahan seperti yang dijelaskan pada Bab 1 adalah melalui studi empiris dengan kerangka berpikir seperti disajikan pada **Gambar 2.1**.



Gambar 2.1. Kerangka Berpikir Penyelesaian Studi

Setelah memahami bidang kajian melalui studi literatur maka dilakukan identifikasi permasalahan utama yang muncul. Dengan memfokuskan pada satu permasalahan utama, dilakukan percobaan empiris menggunakan metode yang umum dipakai. Hal ini dilakukan selain untuk menunjukkan kegagalan metode tersebut dalam menangani permasalahan utama yang menjadi fokus juga untuk mengidentifikasi bagian yang bersifat kritis sebagai penyebab kegagalannya.

Pemahaman terhadap bidang kajian disajikan pada dua bagian, yaitu Bab 2 dan Bab 3. Bab 2 difokuskan untuk memberikan deskripsi mengenai proses transformasi sinyal suara menjadi informasi. Proses transformasi ini secara garis besar terdiri dari tiga bagian, yaitu kuantisasi gelombang suara, ekstraksi atau parameterisasi suara dan proses pengenalan suara. Sementara itu, Bab 3 difokuskan pada deskripsi mengenai ranah kajian pemrosesan sinyal. Pembahasan dimulai dengan bidang-bidang utama pengenalan suara hingga ke identifikasi pembicara. Selain penyajian aspek teknis yang dinyatakan secara skematis, juga perkembangan metodologi yang ada serta capaian hasilnya. Hal ini dimaksudkan untuk mengetahui sejauh mana perkembangan riset yang ada serta identifikasi permasalahan yang muncul.

Studi empiris berbasis teknik yang sudah umum dipergunakan pada pemrosesan sinyal disajikan pada satu bab, yaitu Bab 4. Bagian ini terdiri dari dua bagian utama yaitu identifikasi menggunakan *Mel-Frequency Cepstrum Coefficients* (MFCC) sebagai ekstraksi ciri dan template atau *codebook* sebagai pengenalan pola. Teknik dengan *codebook* ini merupakan metode yang pertama-tama dipergunakan pada pemrosesan sinyal. Oleh karena metode ini sangat sederhana, meskipun digunakan data sinyal yang asli tanpa penambahan *noise* hasilnya tidaklah memuaskan. Bagian kedua adalah identifikasi menggunakan teknik terakhir yang ada, yaitu *Mel-Frequency Cepstrum Coefficients* (MFCC) sebagai ekstraksi ciri dan Hidden Markov model (HMM) sebagai pengenalan pola. Teknik ini melandasi pada sistem modern dari pemrosesan sinyal, sehingga akan memberikan hasil yang pengenalan yang sangat baik. Namun demikian, teknik ekstraksi ciri yang

pergunakan masih berbasis pada nilai power spektrum. Nilai power spektrum ini sangat sensitif terhadap *noise*, sehingga model yang dipergunakan akan gagal bekerja dengan baik untuk sinyal yang telah mengalami penambahan *noise*. Oleh karena itu, pada Bab 4 akan ditunjukkan kegagalan teknik yang umum dipergunakan untuk dapat bekerja pada situasi bernois.

Bertolak dari studi literatur serta studi empiris dengan metode yang sudah ada diusulkan satu pemikiran atau ide untuk menangani permasalahan. Dari ide yang diusulkan ini diformulasikan beberapa teknik dengan tujuan agar ide tersebut dapat diimplementasikan dengan baik. Berikutnya dilakukan serangkaian percobaan menggunakan teknik-teknik yang diusulkan dengan tujuan untuk memberikan bukti empiris dari efektifitas ide serta teknik tersebut, yang disajikan pada dua bab, yaitu Bab 5 dan Bab 6. Bab 5 difokuskan pada perluasan teknik MFCC dari satu dimensi menjadi dua dimensi, sehingga bisa bekerja untuk data bispektrum. Perluasan teknik MFCC ini pada dasarnya adalah proses pembentukan filter dari satu dimensi menjadi dua dimensi. Pada Bab 5 tersebut disajikan dua cara pembentukan filter, yaitu menggunakan filter standar yang terdiri 20 filter (10 filter linear dan 10 filter lagartmik) dengan tinggi satu. Cara kedua adalah dengan algoritme genetika untuk menentukan lebar filternya, namun dengan tinggi sama, yaitu satu. Oleh karena teknik yang dikembangkan bekerja dengan nilai bispektrum, maka diharapkan akan memberikan hasil yang lebih baik, khususnya untuk sinyal bernois. Namun demikian, nilai bispektrum yang dipergunakan masih per titik domain (f_1, f_2) . Hal ini berakibat kurang terakomodasinya distribusi spasial bispektrum, sehingga untuk *noise* yang lebih besar lagi sistem yang disajikan pada Bab 5 gagal bekerja dengan baik.

Kelemahan teknik pada Bab 5 diperbaiki dengan teknik kuantisasi sebagai representasi nilai bispektrum yang disajikan pada Bab 6. Pembahasan pada Bab 6 secara umum terdiri dari tiga bagian, yaitu kuantisasi skalar, kuantisasi vektor dan optimisasi parameter pada kuantisasi vektor. Pada kuantisasi skalar, pembacaan bispektrum dilakukan pada setiap set yang terdiri dari sejumlah titik domain tertentu dengan jumlah yang tetap. Ada beberapa cara untuk menghitung nilai bispektrum

pada setiap set tersebut, yaitu dengan mengambil nilai rata-ratanya, median serta rata-rata nilai bispektrum di atas kuartil ke tiga. Secara empiris, teknik kuantisasi skalar ini memberikan hasil yang lebih baik dibanding dengan menggunakan nilai bispektrum pada setiap titik domain. Namun demikian, oleh karena pemilihan set domain tidak memperhatikan bentuk distribusi spasial nilai bispektrum, maka sistem yang dihasilkan juga kurang stabil terhadap *noise*. Oleh karena itu, teknik kuantisasi skalar ini diperbaiki dengan kuantisasi vektor yang juga disajikan pada Bab 6.

Pada kuantisasi vektor pemilihan set pada domain bispektrum dilakukan dengan mempertimbangkan distribusi spasial nilai bispektrum. Secara umum ada dua tahapan, yaitu pemilihan titik-titik domain yang akan akan dipergunakan untuk menentukan pusat-pusat set nilai bispektrum. Pemilihan dilakukan dengan memilih secara acak titik (f_1, f_2) pada domain dan peluang terpilih atau tidak setara dengan nilai bispektrum di titik tersebut. Setelah diperoleh sejumlah titik, maka tahap berikutnya adalah mengelompokkan titik tersebut menjadi beberapa set, dengan pusat set sebagai pusat channel. Dengan cara seperti ini, maka pemilihan titik diharapkan telah mengakomodasi distribusi spasial nilai bispektrum. Nilai bispektrum pada setiap set dihitung seperti pada kuantisasi skalar, yaitu rata-rata, median dan rata-rata nilai bispektrum di atas kuartil 3. Dengan teknik kuantisasi vektor ini, akurasi sistem yang diperoleh akan lebih baik dibanding dengan teknik-teknik sebelumnya.

Setelah nilai bispektrum dikuantisasi (skalar maupun vektor), tahap berikutnya adalah mereduksi dimensi channel dengan transformasi wrapping dan dilanjutkan dengan kosinus. Untuk mengetahui parameter optimal dari transformasi wrapping dan kosinus tersebut, bagian akhir Bab 6 menyajikan rancangan percobaan faktorial yang melibatkan empat parameter pada transformasi tersebut.

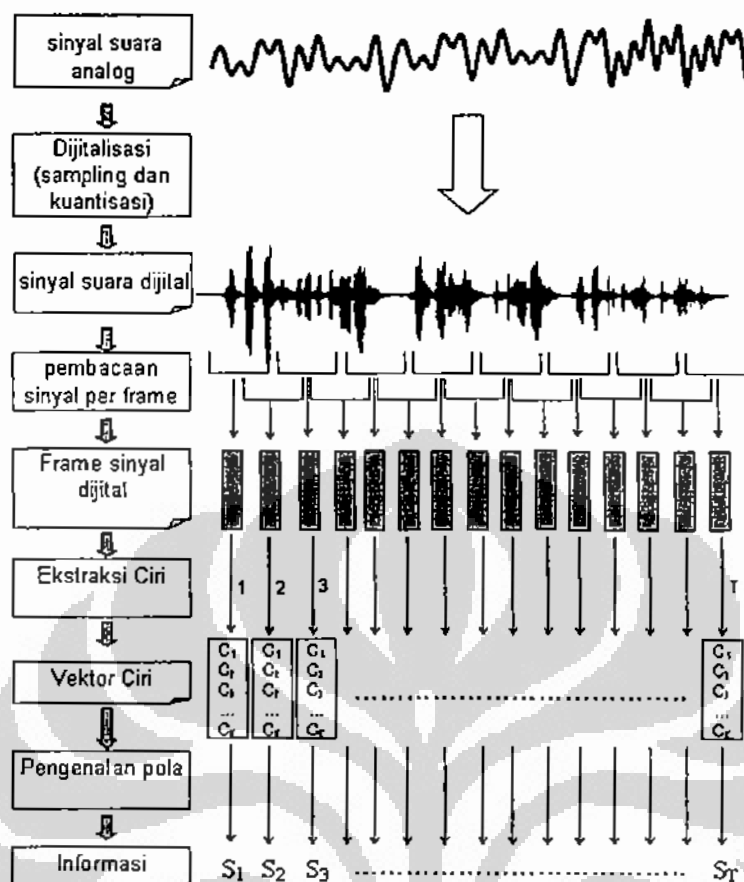
Setelah melakukan studi empiris baik dengan metode yang sudah ada maupun yang diusulkan, berikutnya adalah melakukan komparasi dan sintesa dari semua teknik yang telah digunakan untuk melihat karakteristiknya. Dari sini diusulkan suatu prototipe sistem identifikasi pembicara yang bersifat lebih *robust* terhadap *noise*.

Bagian akhir dari kerangka pikir penyelesaian studi ini adalah dokumentasi yang berupa dokumen disertasi yang berisi materi mulai dari hal-hal yang melatarbelakangi, tujuan, batasan, landasan teori, hasil-hasil percobaan baik dengan teknik yang sudah ada maupun yang diusulkan, serta hal-hal yang bisa ditarik sebagai kesimpulan dari penelitian ini serta saran untuk penelitian selanjutnya.

2.2. Transformasi Sinyal Menjadi Informasi

Dalam [Aka07] disebutkan bahwa sinyal suara merupakan gelombang longitudinal yang tercipta dari tekanan udara yang berasal dari paru-paru yang berjalan melewati lintasan suara menuju mulut dan rongga hidung dengan bentuk artikulator yang senantiasa berubah. Pemrosesan suara merupakan teknik mentransformasi gelombang longitudinal tersebut menjadi informasi yang berarti sesuai yang diinginkan. Secara umum proses transformasi tersebut terdiri dari digitalisasi sinyal analog, ekstraksi ciri dan diakhiri dengan pengenalan pola untuk klasifikasi, seperti diilustrasikan pada Gambar 2.2.

Sesuai dengan Gambar 2.2. tersebut, maka proses transformasi sinyal suara menjadi informasi yang akan dijelaskan pada Sub Bab ini disajikan dalam tiga konsep, yaitu mengenai sinyal, ekstraksi ciri, dan pengenalan pola. Pada bagian sinyal, pembahasan dimulai dengan terminologi sinyal, ukuran kualitas sinyal, sampling dan kuantisasi, serta pembacaan sinyal untuk pemrosesan. Untuk ekstraksi ciri, pembahasan difokuskan pada teknik *mel-frequency cepstrum coefficients* (MFCC). Hal ini dengan pertimbangan teknik tersebut relatif lebih baik dibanding teknik lain yang sudah ada. Untuk pengenalan pola, akan difokuskan pada model *hidden markov model*, (HMM). Hal ini didasarkan fakta bahwa HMM merupakan model yang menjadi *trend* serta paling banyak dikaji pada riset terbaru mengenai pemrosesan sinyal. Algoritme detail mengenai HMM (algoritme *forward*, *backward*, *viterbi*, *k-means* serta algoritme *Baum-Welch*) disajikan secara terpisah pada lampiran disertasi ini. Algoritme *forward* dan *backward* dipergunakan untuk menghitung peluang barisan observasi, algoritme *viterbi* untuk menduga barisan *hidden state* yang optimum dan algoritme *k-means* dan *Baum-Welch* untuk menduga parameter HMM.



Gambar 2.2. Alur Proses Transformasi Sinyal Suara Analog Menjadi Informasi

2.2.1. Sinyal

Sinyal adalah suatu kuantitas fisik yang bervariasi dengan waktu, ruang, maupun sembarang satu atau lebih peubah bebas lainnya, [PM96]. Secara matematika suatu sinyal dirumuskan sebagai fungsi dari satu atau lebih peubah bebas. Sebagai contoh, berikut adalah dua sinyal, yang pertama sebagai fungsi dari waktu, dan yang kedua sebagai fungsi dari bidang spasial.

$$s_1(t)=20t^2 \quad t \text{ adalah waktu}$$

$$s_2(x,y)=3x+2xy+10y^2, \quad x \text{ dan } y \text{ menunjukkan posisi koordinat}$$

Dalam banyak kasus, fungsi yang menghubungkan antara waktu dengan kuantitas sinyal tidak diketahui dan sangat kompleks, sehingga tidak praktis dalam penerapannya, seperti pada sinyal suara. Suatu segmen dari suara sinyal suara

direpresentasikan dengan akurasi yang tinggi yang merupakan jumlah dari beberapa fungsi sinus yang berbeda amplitudo dan frekuensinya, dan ditulis sebagai :

$$\sum_{i=1}^N A_i(t) \sin[2\pi F_i(t) + \theta_i(t)] \quad (2.1)$$

Dengan $\{A_i(t)\}$, $\{F_i(t)\}$ dan $\{\theta_i(t)\}$ adalah himpunan berbagai kemungkinan amplitudo, frekuensi dan fase dari gelombang sinus untuk setiap waktu t .

Untuk menangkap pesan dari satu segmen sinyal suara, yang dilakukan adalah dengan mengukur amplitudo, frekuensi dan fase pada segmen tersebut. Beberapa sinyal lain adalah *electrocardiogram* (ECG) untuk mendeteksi kondisi hati, *electroencephalogram* (EEG) sinyal untuk melihat aktifitas otak. Suara, ECG, dan EEG adalah contoh sinyal sebagai fungsi dari satu peubah bebas, yaitu waktu. Sedangkan sinyal *image* adalah contoh sinyal sebagai fungsi dari dua peubah bebas yaitu koordinat spasial.

Sinyal dibangkitkan oleh suatu sistem yang merupakan respon dari suatu stimulus atau gaya. Misalkan sistem sinyal suara terdiri dari *vocal cords* dan *vocal tract*, yang disebut sebagai *vocal cavity*. Sedangkan sebagai stimulus adalah udara yang dihembuskan dari paru-paru melalui batang tenggorokan. Stimulus dan sistem secara bersama-sama disebut sebagai sumber sinyal (*signal source*).

Terminologi

Secara umum sinyal dapat dikategorikan sesuai dengan sumber, jenis peubah bebas, skala nilainya, serta sifat kemunculannya, seperti disajikan berikut ini, [PM96] :

a. Sinyal *Multichannel* dan Sinyal *Multidimensional*

Sinyal seringkali dibangkitkan oleh beberapa sumber (*multiple sensor*). Sebagai contoh pada ECG dipasang tiga sensor untuk menangkap kondisi hati dari tiga sudut pandang yang berbeda, dan merupakan fungsi dari waktu, t . Oleh karena itu sinyal yang ditangkap sebagai sinyal *multichannel*, dan direpresentasikan dalam vektor 3 elemen :

$$S(t) = \begin{pmatrix} s_1(t) \\ s_2(t) \\ s_3(t) \end{pmatrix}$$

Pada contoh di atas, hanya ada satu peubah bebas, yaitu t . Jika peubah bebas lebih dari satu, misal ditambah dengan x dan y yang menunjukkan koordinat, maka sinyal tersebut menjadi *multidimensional* (dalam contoh ini sekaligus juga *multichannel*).

$$S(x, y, t) = \begin{pmatrix} s_1(x, y, t) \\ s_2(x, y, t) \\ s_3(x, y, t) \end{pmatrix}$$

Nilai pada setiap komponen dapat berupa skalar bilangan real (*real-valued scalar*) ataupun skalar bilangan kompleks (*complex-valued scalar*).

b. Sinyal Waktu Kontinu dan Sinyal Waktu Diskret

Berdasar peubah bebas waktu, t , sinyal dapat dibedakan menjadi dua, yaitu :

Sinyal waktu kontinu (sinyal analog) : kuantitas sinyal terdefinisi pada setiap waktu dalam selang kontinu (a, b) , misalkan

$$x(t) = \cos(\pi t) \quad 0 < t < \infty \quad (2.2)$$

Sinyal waktu diskret : kuantitas sinyal terdefinisi pada suatu waktu diskret tertentu, yang dalam hal ini jarak antar waktu tidak harus sama. Namun untuk kemudahan penurunan sifat matematikanya, maka biasanya jarak antar waktu adalah sama, misalkan

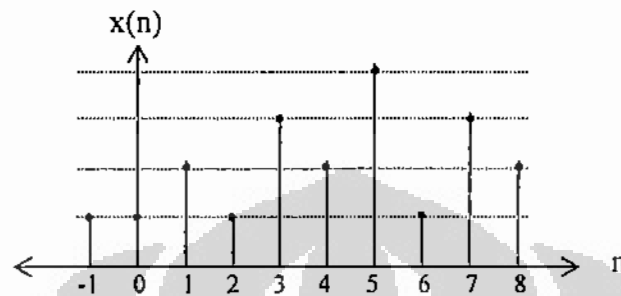
$$x(t_n) = \cos(\pi t_n) \quad n=0, \pm 1, \pm 2, \pm 3, \dots \quad (2.3)$$

Sinyal diskret ini bisa dibangkitkan dengan cara sampling ataupun dengan cara mengkumulatikan dalam suatu selang waktu.

c. Sinyal *Continuous-Valued* dan *Discrete-Valued*

Jika nilai dari sinyal waktu kontinu maupun waktu diskret adalah sembarang nilai dari suatu selang (baik terbatas maupun tidak) maka ini disebut sinyal nilai

kontinyu (*continuous-valued*). Sedangkan jika nilai sinyal tersebut adalah satu dari beberapa kemungkinan nilai yang ada dalam gugus terbatas, maka ini disebut sebagai sinyal nilai diskret (*discrete-valued*). Sinyal waktu diskret yang mempunyai nilai bersifat diskret disebut sebagai sinyal digital (*digital signal*).



Gambar 2.3. Sinyal Waktu Diskret

Untuk tujuan pemrosesan sinyal secara digital, maka harus dalam bentuk diskret dalam waktu dan nilai kuantitas sinyal adalah diskret (sinyal digital), seperti ditunjukkan pada **Gambar 2.3**. Oleh karena itu, sinyal analog disampling dan dikuantisasi sehingga diperoleh nilai diskret.

d. Sinyal *Deterministic* dan Sinyal *Random*

Jika nilai sinyal dari waktu ke waktu dapat dideskripsikan secara langsung menggunakan suatu fungsi (atau tabel) secara tepat, tanpa ada ketidakpastian (*uncertainty*), maka sinyal seperti ini disebut sebagai sinyal deterministik. Persamaan matematika yang menghubungkan peubah bebas dengan kuantitas sinyal disebut sebagai model sinyal. Sedangkan jika nilai sinyal pada setiap periode waktu tidak dapat diprediksi sebelumnya dengan tepat tanpa kesalahan, maka sinyal seperti ini dikenal dengan sinyal *random*.

Ukuran Kualitas Sinyal

Kualitas sinyal merujuk pada sejauh mana sinyal asli telah terdistorsi oleh suatu sebab tertentu. Makin banyak distorsi yang terjadi maka kualitas sinyal makin jelek. Ada beberapa ukuran kualitas, yaitu : SNR (*Signal-to-Noise Rasio*), *Segmental-mean* SNR, dan *Segmental-median* SNR, [NE02]. Andaikan $s(n)$ menotasikan sinyal bebas

noise (*noise-free signal*) pada periode n dan $\hat{s}(n)$ adalah sinyal $s(n)$ yang telah terdistorsi, maka sinyal error dapat dituliskan sebagai :

$$\varepsilon(n) = s(n) - \hat{s}(n) \quad (2.4)$$

Oleh karena itu energi sinyal error ini adalah :

$$E_\varepsilon = \sum_{n=-\infty}^{\infty} \varepsilon^2(n) \quad (2.5)$$

Sedangkan energi sinyal asli yang tidak terdistorsi adalah :

$$E_s = \sum_{n=-\infty}^{\infty} s^2(n) \quad (2.6)$$

Signal-to-Noise Rasio (SNR) dirumuskan sebagai :

$$SNR = 10 \log_{10} \frac{E_s}{E_\varepsilon} = 10 \log_{10} \frac{\sum_n s^2(n)}{\sum_n [s(n) - \hat{s}(n)]^2} \quad (2.7)$$

Sedangkan ukuran lainnya adalah :

$$SNR_{seg-mean} = \frac{1}{M} \sum_{j=0}^{M-1} 10 \log_{10} \left[\frac{\sum_{n=m_j-N+1}^{m_j} s^2(n)}{\sum_{n=m_j-N+1}^{m_j} [s(n) - \hat{s}(n)]^2} \right] \quad (2.8)$$

$$SNR_{seg-median} = median \left\{ 10 \log_{10} \left[\frac{\sum_{n=m_j-N+1}^{m_j} s^2(n)}{\sum_{n=m_j-N+1}^{m_j} [s(n) - \hat{s}(n)]^2} \right] \right\} \quad (2.9)$$

Dalam hal ini $m_0, m_1, m_3, \dots, m_{M-1}$ adalah periode akhir dari frame ke 1, frame ke 2, dan seterusnya hingga frame ke M , dengan panjang setiap frame adalah N .

2.2.2. Sampling

Sampling merupakan pengamatan nilai sinyal waktu kontinyu (sinyal analog) pada suatu waktu tertentu, sehingga diperoleh sinyal waktu diskret. Banyak cara untuk melakukan sampling pada sinyal analog. Salah satu yang sering digunakan adalah *periodic* atau *uniform sampling*. Dalam hal ini sampling dilakukan pada setiap selang waktu yang tetap, yaitu pada setiap selang waktu T . Hubungan antara sinyal waktu diskret hasil sampling dengan sinyal analog adalah sebagai berikut :

$$x(n) = x_a(nT) \quad \text{dengan } -\infty < n < \infty \quad (2.10)$$

$x(n)$ adalah sinyal waktu diskret yang diperoleh dari sampling terhadap sinyal waktu nyata $x_a(t)$ setiap T detik. Dalam hal ini T sebagai periode sampling, dan $F_s = 1/T$ adalah sampling rate (Hertz). Sebagai ilustrasi, misalkan sinyal analog disampling dengan sampling rate 1.1 kHz, ini berarti setiap detik disampling sebanyak 11000 kali. Dengan kata lain, setiap detik dicatat nilai simpangan sinyal sebanyak 11000 data. Atau dengan kata lain lagi, jarak antara satu data simpangan dengan data berikutnya adalah $1/11000$ detik. Permasalahan dalam sampling adalah bagaimana menentukan T atau juga F_s dari suatu sinyal analog dengan frekuensi F .

Sinyal sinusoid analog dengan frekuensi F untuk setiap waktu t dirumuskan sebagai :

$$x_a(t) = A \cos(2\pi Ft + \theta)$$

Oleh karena itu :

$$x_a(nT) = A \cos(2\pi FnT + \theta) = A \cos\left(\frac{2\pi nF}{F_s} + \theta\right)$$

Sementara itu sinyal diskret sinusoid, $x(n)$, dapat dirumuskan sebagai :

$$x(n) = A \cos(2\pi fn + \theta)$$

Dari tiga persamaan terakhir, terlihat bahwa $f = \frac{F}{F_s}$, yang dapat diartikan sebagai jumlah gelombang per sample. Misalkan $F=10\text{Hz}$ (yang berarti 10 gelombang per

detik) dan $F_s=5$ sample per detik, maka $f=10/5=2$ gelombang per sample. Berdasar persamaan inilah, nilai F_s akan ditentukan sehingga semua komponen frekuensi dalam sinyal dapat direpresentasikan secara khas. Sesuai sifat kosinus, maka persamaan ketiga di atas juga bisa ditulis sebagai :

$$x(n) = A \cos(2\pi f n + \theta) = A \cos(2\pi f n + \theta) = A \cos(2\pi f n + 2k\pi + \theta)$$

untuk $k=0,1,2,3,\dots$. Hal ini berarti bahwa sinyal diskret dengan kecepatan sudut $\omega_k = 2\pi f + 2k\pi$ tidak dapat dibedakan dengan sinyal diskret dengan kecepatan sudut $\omega_0 = 2\pi f$. Sesuai sifat kosinus, sinyal diskret yang dapat dibedakan hanya pada rentang kecepatan sudut $-\pi < \omega_0 = 2\pi f < \pi$ atau dengan kata lain $-1/2 < f < 1/2$. Oleh karena karena $f = \frac{F}{F_s}$, maka sampling dengan frekuensi sampling F_s hanya mampu memberikan hasil yang berbeda untuk sinyal-sinyal kontinyu dengan frekuensi $F < 0.5F_s$. Sebagai gambaran, misalkan ada dua sinyal dengan frekuensi $F_1=10$ Hz ($x_1(t) = \cos 2\pi 10t$) dan $F_2=50$ Hz ($x_2(t) = \cos 2\pi 50t$), maka kalau masing-masing disampling dengan $F_s=40$ sample per detik akan dihasilkan sinyal diskret :

$$x_1(n) = \cos 2\pi \frac{10}{40} n = \cos \frac{\pi}{2} n$$

$$x_2(n) = \cos 2\pi \frac{50}{40} n = \cos \frac{5\pi}{2} n = \cos(2\pi + \frac{\pi}{2} n) = \cos \frac{\pi}{2} n$$

Terlihat sampling dari dua sinyal analog tersebut dengan $F_s=40$ Hz akan menghasilkan dua sinyal diskret yang sama persis. Hal ini disebut bahwa sinyal dengan frekuensi 50 Hz adalah alias dari sinyal dengan dengan frekuensi $F=10$ Hz dengan sampling rate 40 sample per detik.

Oleh karena sinyal analog dapat direpresentasikan sebagai penjumlahan dari gelombang sinus dengan amplitudo, frekuensi dan fase yang berbeda, yang dalam hal ini sebanyak N komponen sesuai formula berikut :

$$x_a(t) = \sum_{i=1}^N A_i(t) \sin[2\pi F_i(t) + \theta_i(t)] \quad (2.11)$$

maka nilai sampling rate yang dapat menangkap semua komponen sinyal haruslah minimal dua kali frekuensi maksimum yang ada dalam sinyal. Nilai sampling rate sebesar $F_s = 2F_{\max}$ disebut sebagai *Nyquist rate*.

2.2.3. Kuantisasi dan Coding

Kuantisasi merupakan proses mengkonversi nilai amplitudo yang bersifat kontinyu pada suatu titik waktu tertentu menjadi sinyal digital dengan mengekspresikannya menggunakan sejumlah digit tertentu. Konversi nilai kontinyu menggunakan sejumlah digit ini akan menghasilkan *error* yang disebut *quantization error* atau *quantization noise*.

Secara umum, proses kuantisasi dilakukan dengan pembulatan ke nilai terdekat (*rounding*), atau bisa juga dengan pemotongan bagian sisa (*truncating*). *Error* karena kuantisasi dengan metode pembulatan, $e_q(n)$, adalah pada selang :

$$-\frac{\Delta}{2} \leq e_q(n) \leq \frac{\Delta}{2}, \text{ dengan } \Delta = \frac{x_{\max} - x_{\min}}{L-1}$$

L adalah banyaknya level kuantisasi, x_{\max} dan x_{\min} adalah nilai maksimum dan minimum yang akan dikuantisasi. Dalam hal ini *error* kuantisasi merupakan selisih antara nilai sinyal analog dengan nilai hasil kuantisasinya, yaitu :

$$e_q(t) = x_a(t) - x_q(t) \quad (2.12)$$

Ukuran kualitas output dari suatu mesin konversi analog ke digital (*A/D converter*) biasanya diukur dengan *signal-to-quantization noise ratio (SQNR)* yang dinyatakan sebagai rasio energi signal terhadap energi *noise*, yaitu [PM96] :

$$SQNR = \frac{P_x}{P_q} = \frac{3}{2} 2^{2b} \quad (2.13)$$

Dengan satuan *decibel*, *dB*, maka SQNR dirumuskan sebagai :

$$SQNR(\text{dB}) = 10 \log_{10}(SQNR) = 1.76 + 6.02b \quad (2.14)$$

Ini berarti setiap penambahan 1 bit pada representasi digital, akan meningkatkan SQNR sekitar 6 dB. Sebagai contoh pada *compact disc player*, menggunakan representasi 16 bit, sehingga nilai SQNR adalah lebih dari 96 dB.

Coding merupakan pemberian bilangan biner pada setiap level kuantisasi. Jika kuantisasi yang diterapkan mempunyai level sebanyak L , maka setidaknya harus tersedia L bilangan biner yang berbeda. Sedangkan kode biner dengan panjang b akan dapat menghasilkan kode berbeda sebanyak 2^b . Oleh karena itu untuk kuantisasi dengan L level diperlukan bilangan biner dengan panjang $b \geq \log_2 L$.

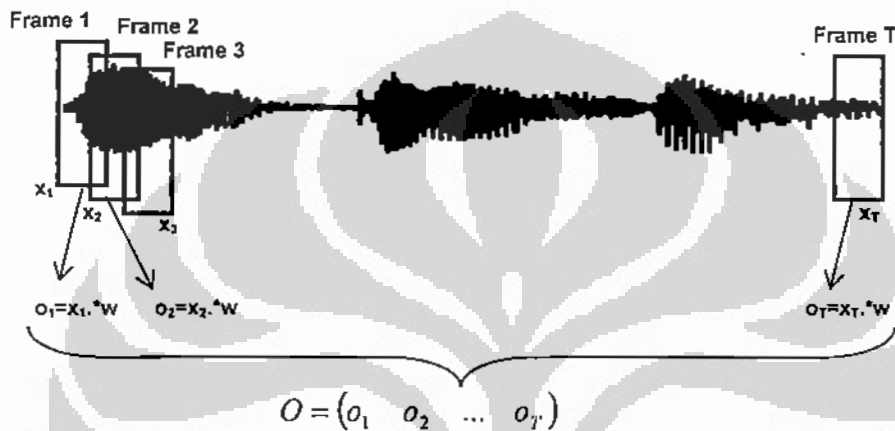
2.2.4. Pembacaan Sinyal

Untuk keperluan pemrosesan, sinyal analog yang sudah didigitalkan (dengan sampling dan kuantisasi) dibaca dari frame demi frame dengan lebar tertentu yang saling tumpang tindih. Panjang frame ini biasanya 5 hingga 100 *milisecond* dengan overlap antar frame yang berurutan adalah 0, 25, 50 atau 75%. Proses ini dikenal dengan *frame blocking*. Satu frame tersebut sebagai satu unit terkecil yang mengandung satu unit informasi, sehingga barisan frame akan menyimpan suatu informasi yang lengkap dari sebuah sinyal suara. Untuk itu, distorsi antar frame harus diperkecil atau diminimalisasi. Satu teknik untuk meminimalkan distorsi antar frame adalah dengan melakukan proses filtering pada setiap frame. Secara umum fungsi filtering ada dua, yaitu untuk memisahkan sinyal dari berbagai sumber lain yang “mengotori” serta untuk “menjernihkan” sinyal dari adanya distorsi. Secara umum dikenal enam jenis filter seperti disajikan pada **Tabel 2.1**.

Tabel 2.1. Klasifikasi Filtering

Domain Penerapan	Metode Penerapan	
	Konvolusi (FIR)	Rekursif (IIR)
Domain waktu (smoothing, DC removal)	Moving Average	Single pole
Domain Frekuensi (memisahkan frekuensi)	Windowing	Chebyshev
Kustomisasi (dekonvolusi)	FIR custom	Iterative design

Oleh karena dalam penelitian ini pemrosesan terhadap sinyal dilakukan dalam domain frekuensi, maka proses filtering yang dipilih akan menggunakan *windowing*. Proses *windowing* dilakukan pada setiap frame, dalam hal ini sinyal digital tersebut dikalikan dengan fungsi window tertentu yang berukuran sama dengan ukuran frame. Jika sinyal digital frame ke i adalah x_i dan fungsi window yang digunakan adalah w , maka output *windowing* frame ke i adalah $y_i = x_i \cdot w$, yaitu perkalian setiap komponen yang seletak dari vektor x_i dengan vektor w , seperti diilustrasikan pada **Gambar 2.4**.



Gambar 2.4. Proses *Frame Blocking* dan *Windowing*

Terlihat bahwa sinyal analog yang sudah diubah menjadi digital dibaca frame demi frame dan pada setiap frame dilakukan *windowing* dengan fungsi window tertentu. **Tabel 2.2.** menyajikan jenis-jenis window yang bisa dipergunakan, (Rao dan Gabr, 1984; Sasaki, et al., 1975; dalam [NP93]). Beberapa ukuran untuk membandingkan kinerja window adalah bias (J), Variance (V), mean-squared error (MSE) dari nilai spektrum dugaan terhadap nilai sebenarnya. Fungsi window dengan MSE yang optimal diberikan oleh formula, (Rao dan Gabr, 1984; dalam [NP93]) :

$$w(u_1, u_2) = \frac{8}{7\pi^3} \{g(u_1, u_2) + g(-u_1, u_2 - u_1) + g(u_1 - u_2, -u_2)\} \quad (2.15)$$

Dengan :

$$g(u_1, u_2) = \frac{2u_1^2 + 2u_2^2 + u_1 u_2}{\pi u_1^3 u_2^3} \cos[(u_2 - u_1)\pi] - \frac{u_2 - u_1}{u_1^2 u_2^2} \cos[(u_2 - u_1)\pi] \quad (2.16)$$

Tabel 2.2. Daftar Window Simetrik dan Ekspresi Matematikanya

Nama window	Ekspresi Matematika
Daniell	$d(u) = \begin{cases} \frac{\sin(\pi u)}{\pi u} & u \leq 1 \\ 0 & \text{lainnya} \end{cases}$
Hamming	$d(u) = \begin{cases} 0.54 + 0.46 \cos(\pi u) & u \leq 1 \\ 0 & \text{lainnya} \end{cases}$
Parzen	$d(u) = \begin{cases} 1 - 6u^2 + 6 u ^3 & u \leq 0.5 \\ 2(1 - u)^3 & 0.5 \leq u \leq 1 \\ 0 & \text{lainnya} \end{cases}$
Priestly	$d(u) = \begin{cases} \frac{3}{(\pi u)^2} \left(\frac{\sin(\pi u)}{\pi u} - \cos(\pi u) \right) & u \leq 1 \\ 0 & \text{lainnya} \end{cases}$
Sasaki	$d(u) = \begin{cases} \frac{1}{\pi} \sin(\pi u) + (1 - u) \cos(\pi u) & u \leq 1 \\ 0 & \text{lainnya} \end{cases}$

$$u = \tau_i \Delta_n, \Delta_n = \frac{1}{L_n}, \text{ dan } \tau_i \text{ diambil bilangan bulat.}$$

Tabel 2.3. menyajikan nilai-nilai ukuran kinerja window, [NP93]. Terlihat bahwa window MSE optimal relatif lebih baik dibanding window lainnya. Namun formula yang diberikan cukup kompleks. Terlihat bahwa window Hamming memiliki J dan MSE yang moderat. Ditambah dengan pertimbangan kesederhanaan formula, maka penggunaan window Hamming cukup beralasan.

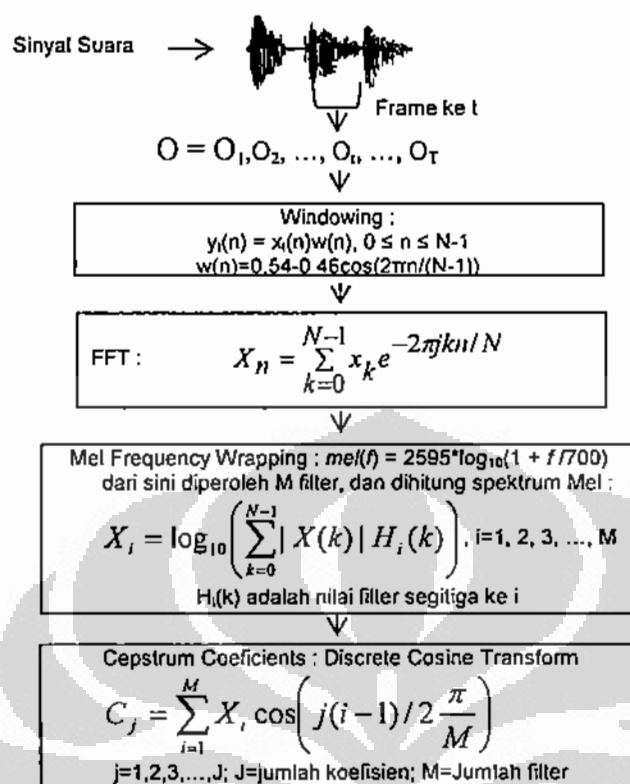
Tabel 2.3. Perbandingan Ukuran J, V dan MSE untuk Beberapa Window

Window	J	V	MSE
Daniell	99.468,50	0,1199	1.078,50
Hamming	60.664,80	0,9067	567,76
Parzen	8.392,43	0,0409	54,20
Priestly	288.002,12	0,2032	2.216,61
Sasaki	1.315,74	0,0486	97.29
MSE optimal	2220.74	0.0691	31.68

2.2.5. Ekstraksi Ciri

Ekstraksi ciri merupakan proses untuk menentukan satu nilai atau vektor yang dapat dipergunakan sebagai penciri obyek atau individu. Di dalam pemrosesan suara, ciri yang biasa dipergunakan adalah nilai koefisien cepstral dari sebuah frame. Satu teknik ekstraksi ciri sinyal suara yang umum dan menunjukkan kinerja yang baik adalah teknik *Mel-Frequency Cepstrum Coefficient*, (MFCC) yang menghitung koefisien cepstral dengan mempertimbangkan persepsi sistem pendengaran manusia terhadap frekuensi suara. Dibandingkan dengan metode ekstraksi ciri lainnya, Davis dan Mermelstein memperlihatkan bahwa MFCC sebagai teknik ekstraksi ciri memberikan hasil pengenalan yang tinggi, [Gan05]. Setelah diperkenalkannya teknik ini, berbagai variasi telah dikembangkan, terutama dalam hal jumlah, bentuk, dan lebar filter serta cara membentuk intervalnya. *Mel-Frequency Cepstrum Coefficient*, (MFCC) sebagai pengekstraksi ciri dan teknik untuk parameterisasi sinyal suara telah banyak digunakan pada berbagai bidang area pemrosesan suara, terutama pada sistem identifikasi pembicara. Diagram alur teknik MFCC dalam mengekstrak sinyal suara adalah seperti pada **Gambar 2.5.**, [Gan05].

Dari **Gambar 2.5.** terlihat bahwa sinyal dibaca frame demi frame, dan dilakukan windowing untuk setiap frame untuk berikutnya dilakukan transformasi Fourier. Dari nilai hasil transformasi Fourier ini selanjutnya dihitung spektrum mel menggunakan sejumlah filter yang dibentuk sedemikian sehingga jarak antar pusat filter adalah konstan pada ruang frekuensi mel. Dari literatur yang ada, skala mel ini dibentuk untuk mengikuti persepsi sistem pendengaran manusia yang bersifat linear untuk frekuensi rendah dan logaritmik untuk frekuensi tinggi, dengan batas pada nilai frekuensi akustik sebesar 1000 Hz. Proses ini dikenal dengan nama *Mel-Frequency Wrapping*. Koefisien MFCC merupakan hasil transformasi *Cosinus* dari spektrum mel tersebut, dan dipilih K koefisien. Transformasi kosinus berfungsi untuk mengembalikan domain, dari frekuensi ke domain waktu. Berikut disajikan detail proses setiap tahapan serta teori yang melandasinya, yang dimulai dengan transformasi Fourier.



Gambar 2.5. Diagram Alur Teknik MFCC untuk Mengekstrak Sinyal

Transformasi Fourier

Analisis Fourier muncul dari paper yang ditulis oleh Jean Baptiste Joseph Fourier (1768-1830) yang di-review oleh Joseph Louis Lagrange (1736-1813) dan Pierre Simon de Laplace (1749-1827). Analisis ini merupakan suatu teknik matematika untuk mendekomposisi sinyal menjadi sinyal-sinyal sinusoidal, dan terdiri dari dua versi, yaitu versi real dan versi imajiner. Sesuai dengan tipe sinyal yang dianalisis, analisis Fourier dapat dibagi menjadi empat kategori, yaitu :

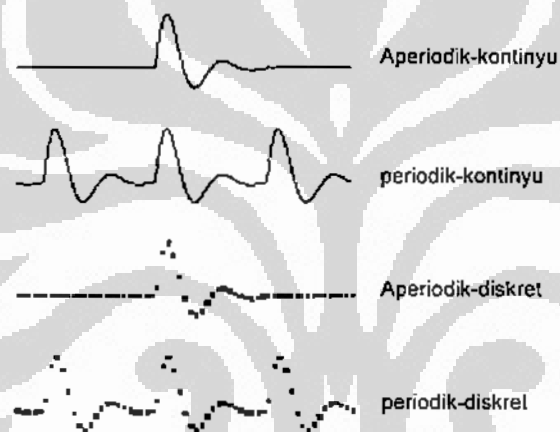
Transformasi Fourier (*Fourier Transform*) : analisis ini berkaitan dengan sinyal aperiodik-kontinyu, yaitu sinyal kontinyu yang dapat diperlebar tanpa batas ke kiri dan ke kanan, tanpa adanya pattern yang berulang.

Barisan Fourier (*Fourier Series*) : analisis ini berkaitan dengan sinyal periodik-kontinyu, yaitu sinyal kontinyu dengan pattern berulang ke kiri maupun ke kanan tanpa batas (misal gelombang sinus).

Transformasi Fourier Waktu Diskret (*Discrete Time Fourier Transform, DTFT*) : analisis ini berkaitan dengan sinyal aperiodik-diskret, yaitu sinyal diskret yang didefinisikan pada beberapa titik waktu saja, dan tidak ada pattern yang berulang.

Transformasi Fourier Diskret (*Diskret Fourier Transform, DFT*) : analisis ini berkaitan dengan sinyal periodik-diskret, yaitu sinyal diskret yang dapat diperlebar tanpa batas ke kiri dan ke kanan, dengan pattern yang berulang. DFT kadang-kadang dikenal dengan sebutan *Discrette Fourier series*.

Gambar 2.6. berikut adalah contoh-contoh sinyal di atas :

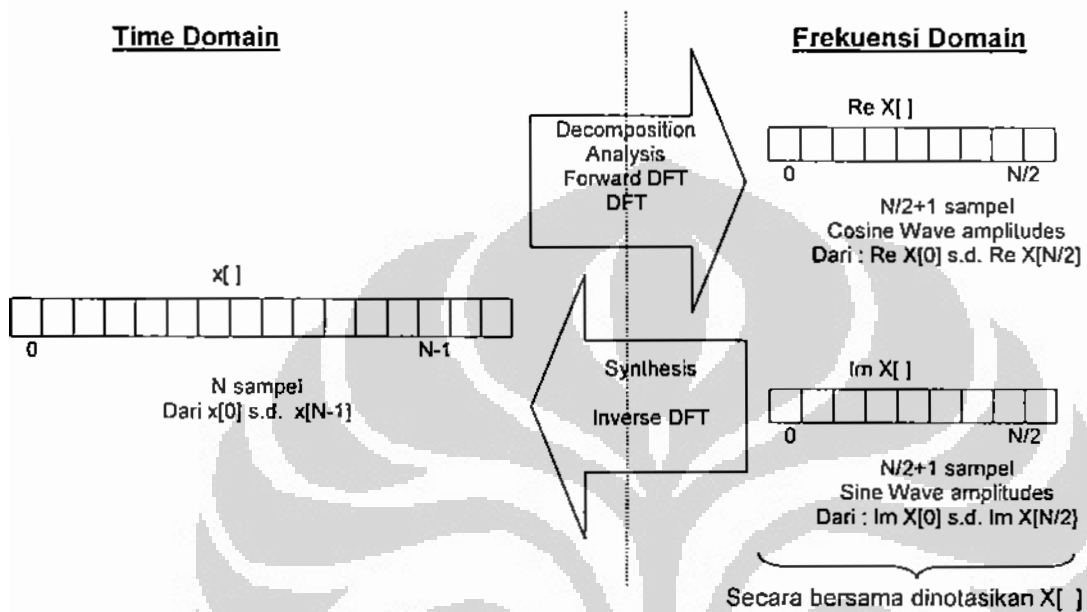


Gambar 2.6. Beberapa Contoh Tipe Sinyal

Pada analisis sinyal digital, dilakukan sampling sinyal sebanyak N . Anggap bahwa sampel ini sebagai satu periode dan dapat diduplikasi terus-menerus ke kiri dan ke kanan, sehingga dapat dianalisis dengan menggunakan *Discrete Fourier transform* (DFT). DFT mentransformasi N titik sinyal sebagai input menjadi $N/2+1$ titik sinyal output, seperti diilustrasikan pada **Gambar 2.7**. Notasi untuk domain frekuensi adalah $\text{Re } X[k]$ dan $\text{Im } X[k]$ untuk $k \in [0, N/2]$ atau $\text{Re } X[f]$ dan $\text{Im } X[f]$ untuk $f = k/N$ atau $f \in [0, 0.5]$. Hal ini dikarenakan sinyal diskret hanya mengandung frekuensi antara 0 dan 0.5 dari sampling rate. Atau bisa juga $\text{Re } X[\omega]$ dan $\text{Im } X[\omega]$ untuk

$\omega=2\pi k/N \in [0, \pi]$. Oleh karena itu gelombang kosinus pada domain frekuensi tersebut bisa dituliskan sebagai :

$$c[n]=\cos(2\pi kn/N)=\cos(2\pi fn)=\cos(\omega n) \quad (2.17)$$



Gambar 2.7. Ilustrasi Transformasi DFT

Fungsi basis (*basis function*) dalam DFT adalah gelombang sinus dan kosinus dengan amplitudo satu. Fungsi basis ini dituliskan sebagai :

$$c_k[i]=\cos(2\pi ki/N) \text{ dan } s_k[i]=\sin(2\pi ki/N) \quad (2.18)$$

untuk $i=0, 1, 2, \dots, N-1$, dan $k=0, 1, 2, \dots, N/2$ (dalam hal ini k adalah banyaknya gelombang pada N sampel). Sebagai contoh untuk 32 titik sampel, DFT dapat direpresentasikan dengan basis gelombang sinus dan kosinus masing-masing sebanyak 17 buah, yaitu untuk $k=0,1,2,\dots,16$. Dalam hal ini sinyal dalam domain waktu dapat dirumuskan sebagai penjumlahan terboboti dari fungsi basis dengan formula :

$$x[i]=\sum_{k=0}^{N/2} \text{Re } \bar{X}[k] \cos(2\pi ki/N) + \sum_{k=0}^{N/2} \text{Im } \bar{X}[k] \sin(2\pi ki/N) \quad (2.19)$$

dengan $i=0,1,2,\dots,N-1$, yang disebut persamaan sintesa. Sinyal sampel sebanyak N titik dibentuk oleh $N/2+1$ gelombang kosinus dan $N/2+1$ gelombang sinus dengan amplitudo untuk kosinus dan sinus masing-masing adalah array $\text{Re } \bar{X}[k]$ dan $\text{Im } \bar{X}[k]$, yang dalam hal ini :

$$\text{Re } \bar{X}[k] = \frac{\text{Re } X[k]}{N/2} \text{ dan } \text{Im } \bar{X}[k] = -\frac{\text{Im } X[k]}{N/2} \quad (2.20)$$

Sedangkan untuk $k=0$ dan $k=N/2$, bagian real adalah :

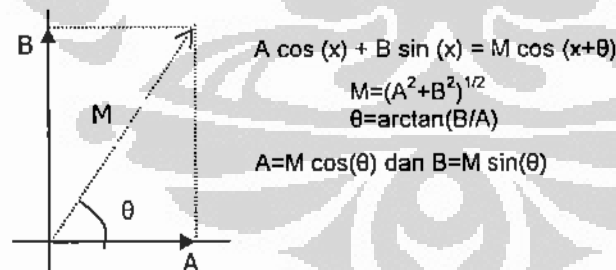
$$\text{Re } \bar{X}[0] = \frac{\text{Re } X[0]}{N} \text{ dan } \text{Re } \bar{X}[N/2] = \frac{\text{Re } X[N/2]}{N} \quad (2.21)$$

Nilai-nilai DFT untuk $k=0,1,2,\dots,N/2$ dihitung dari sinyal input dengan rumus sebagai berikut :

$$\text{Re } X[k] = \sum_{i=0}^{N-1} x[i] \cos(2\pi ki / N) \quad (2.22)$$

$$\text{Im } X[k] = -\sum_{i=0}^{N-1} x[i] \sin(2\pi ki / N) \quad (2.23)$$

yang disebut sebagai persamaan analisis. Ada kalanya, *array* dalam domain frekuensi direpresentasikan dengan koordinat polar. Dalam koordinat polar pasangan $\text{Re } X[k]$ dan $\text{Im } X[k]$ digantikan oleh pasangan magnitudo, $\text{Mag } X[k]$, dan fase, $\text{Phase } X[k]$, seperti gambar berikut :



Oleh karena itu, dalam koordinat polar, domain frekuensi tersebut dapat ditulis :

$$\text{Mag } X[k] = (\text{Re } X[k]^2 + \text{Im } X[k]^2)^{1/2} \text{ dalam amplitudo}$$

$$\text{Phase } X[k] = \arctan\left(\frac{\text{Im } X[k]}{\text{Re } X[k]}\right) \text{ dalam radian}$$

Juga sebaliknya :

$$\text{Re } X[k] = \text{Mag } X[k] \cos(\text{Phase } X[k])$$

$$\text{Im } X[k] = \text{Mag } X[k] \sin(\text{Phase } X[k])$$

Oleh karena itu, sinyal input dalam domain waktu dapat ditransformasi menjadi domain frekuensi melalui analisis DFT, baik dalam bentuk komponen real dan imajiner, maupun dalam bentuk koordinat polar (magnitodu dan fase), tanpa mengubah informasi yang terkandung dalam sinyal. Secara umum ada tiga cara untuk menghitung DFT, yaitu melalui persamaan simultan, korelasi dan algoritme FFT (*Fast Fourier Transform*). Dengan algoritme FFT, kompleksitas menjadi rendah dari n^2 menjadi $n \cdot \lg_2(n)$. Dengan alasan inilah maka pada penelitian ini transformasi Fourier menggunakan algoritme FFT.

Mel-Frequency Wrapping dan Transformasi Cosinus

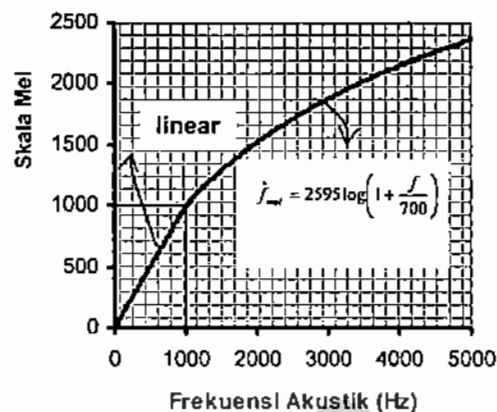
Untuk proses wrapping diperlukan beberapa filter yang saling overlap dalam domain frekuensi. Filter yang digunakan adalah berbentuk segitiga dengan tinggi satu dan rentang filter segitiga tersebut ditentukan berdasar hasil studi psikologi mengenai persepsi manusia dalam menerima frekuensi bunyi. Dari studi psikologi, telinga manusia mempunyai persepsi terhadap frekuensi suara secara tidak linear pada frekuensi di atas 1000 Hz. Ukuran persepsi ini dinyatakan dalam skala mel (*Melody*), dan hubungan skala mel dengan frekuensi dirumuskan sebagai, [NE02] :

$$F_{mel} = \begin{cases} 2595 * \log_{10} \left(1 + \frac{F_{Hz}}{700} \right) & \text{jika } F_{Hz} > 1000 \\ F_{Hz} & \text{jika } F_{Hz} \leq 1000 \end{cases} \quad (2.24)$$

dan dilukiskan seperti pada **Gambar 2.8**. Dari formula di atas, maka nilai frekuensi, F_{Hz} , sebagai fungsi dari skala Mel adalah :

$$F_{Hz} = 700 \left(10^{\frac{F_{mel}}{2595}} - 1 \right) \quad (2.25)$$

Terlihat bahwa untuk frekuensi rendah, filter yang digunakan menggunakan skala linear, sehingga lebarnya konstan. Sedangkan untuk frekuensi tinggi (>1000 Hz), filter dibentuk dengan skala logaritma.



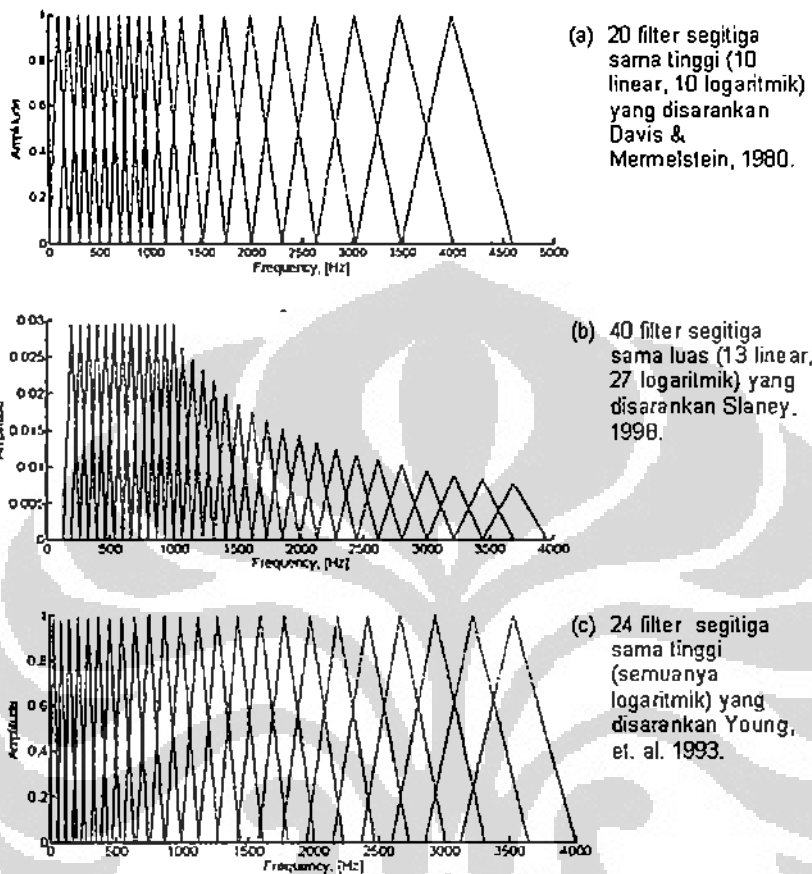
Gambar 2.8. Grafik Hubungan Frekuensi dengan Skala Mel

Sesuai dengan perkembangannya, teknik MFCC mengalami beberapa variasi terutama dalam hal bentuk filter dan jumlahnya. Bentuk filter mengalami perubahan dalam hal tinggi, sedangkan bentuk yang digunakan tetap segitiga, dengan jumlah 20, 24 atau 40. **Gambar 2.9.** menyajikan tiga contoh filter yang dikenal, [Gan05]. Selain tiga filter tersebut, dikenal juga filter dengan 29 segitiga sama tinggi yang dikembangkan oleh Skowronski dan Harris, 2004, yang dikenal dengan nama *Human Factor Cepstral coefficients*, HFCC. Pada HFCC ini, lebar filternya ditentukan berdasarkan konsep *Equivalent Rectangular bandwidth* (ERB) yang dirumuskan oleh Moore dan Glasberg, 1983.

Untuk memberikan gambaran yang lebih detail mengenai pembentukan filter tersebut, maka berikut akan diuraikan tahapan pembentukan filter yang pertama kali diperkenalkan oleh Davis dan Mermelstein pada tahun 1980. Pada penjelasan ini dimisalkan frekuensi suara yang akan dicakup adalah pada rentang 0 hingga 5000 Hz. Secara umum ada tiga tahapan, yaitu :

- Gambarkan fungsi F_{Mel} mulai frekuensi terendah hingga $F_{Hz}=5000$ Hz dengan menggunakan persamaan (2.24), dan tentukan nilai mel untuk frekuensi akustik 5000 Hz (misal M_2) dan nilai mel untuk frekuensi akustik 1000 Hz (misal M_1).
- Sumbu Mel dari 0 hingga M_1 dan M_1 ke M_2 disekat masing-masing menjadi $M/2$ sekatan yang sama lebarnya.

- c. Dari setiap batas sekatan ditarik garis ke kurva fungsi dan diproyeksikan ke sumbu F_{Hz} sehingga diperoleh titik tengah filter pada sumbu F_{Hz} .



Gambar 2.9. Beberapa Filter yang Dikenal

Algoritme detail untuk membentuk M filter pada teknik MFCC yang disarankan oleh Davis dan Mermelstein tersebut disajikan pada Algoritme 1.

Algoritme 1 : Mengkonstruksi M filter

- a. Pilih jumlah filter yang akan dibuat (M)
- b. Pilih frekuensi terbesar (f_{high}). Dari nilai ini, maka nilai tertinggi dari \hat{f}_{mel} adalah :

$$\hat{f}_{mel}^{high} = 2595 * \log_{10} \left(1 + \frac{f_{high}}{700} \right)$$

- c. Pusat filter ke i adalah f_i :

- c.1. $f_i = \frac{1000}{0.5 * M} * i$ untuk $i=1, 2, 3, \dots, M/2$

- c.2. untuk $i=M/2, M/2+1, \dots, M$, maka f_i dihitung dengan prosedur berikut:

1. skala mel disekat dengan lebar yang sama, yaitu sebesar Δ , dengan:

$$\Delta = \frac{\hat{f}_{mel}^{high} - 1000}{0.5 * M}$$

Dari point (b), maka nilai Δ dapat ditulis sebagai :

$$\Delta = \frac{5190}{M} \log\left(\frac{700 + f_{high}}{1700}\right)$$

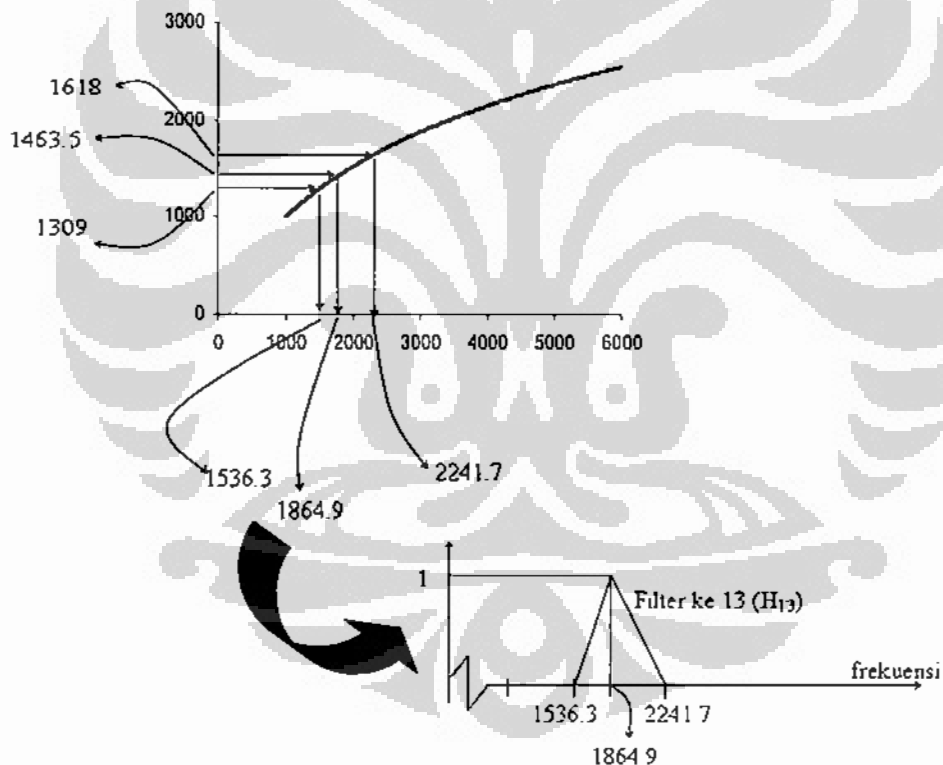
2. nilai mel untuk pusat filter ke i adalah:

$$a = 1000 + (i - 0.5 * M) * \Delta$$

3. pusat dari filter ke i adalah:

$$f_i = 700 * (10^{a/2595} - 1)$$

Sebagai ilustrasi, **Gambar 2.10.** berikut menyajikan contoh pembuatan filter ke 13 dari 20 filter yang akan dibuat.



Gambar 2.10. Ilustrasi Pembuatan Filter ke 13 dari 20 Filter

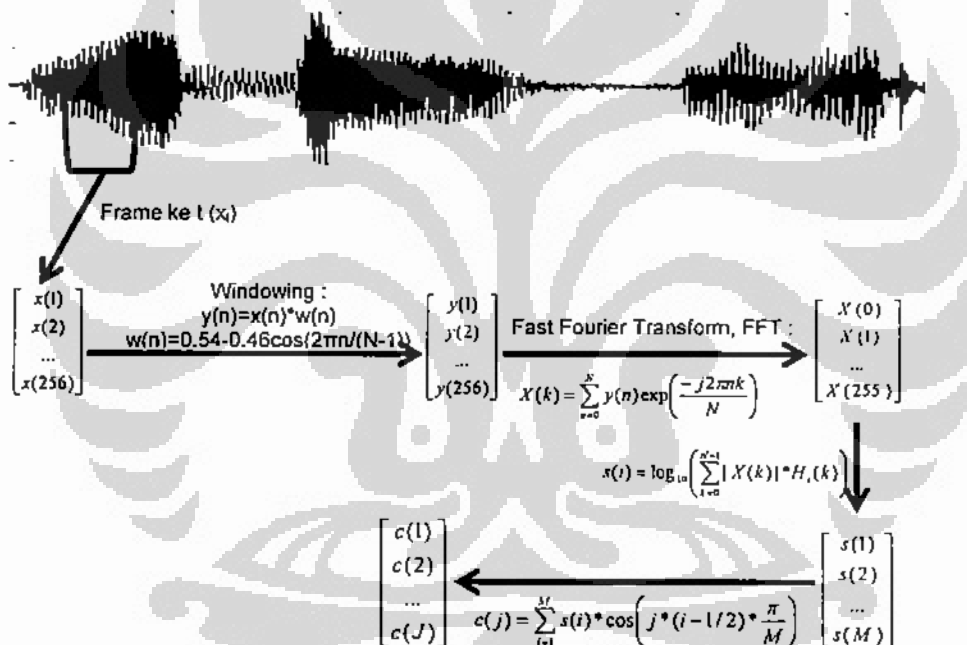
Dari M filter yang sudah dibentuk, maka dilakukan wrapping terhadap sinyal dalam domain frekuensi dan menghasilkan satu komponen untuk setiap filter dengan formula berikut :

$$X_i = \log_{10} \left(\sum_{k=0}^{N-1} |X(k)| H_i(k) \right) \quad (2.26)$$

Dalam hal ini $i=1, 2, 3, \dots, M$ (M adalah jumlah filter segitiga) dan $H_i(k)$ adalah nilai filter segitiga ke i untuk frekuensi akustik sebesar k . Nilai koefisien MFCC ke j akhirnya diperoleh menggunakan transformasi kosinus sesuai formula berikut :

$$C_j = \sum_{i=1}^M X_i \cos \left(j(i-1) / 2 \frac{\pi}{M} \right) \quad (2.27)$$

dengan $j=1,2,3,\dots,K$, K adalah jumlah koefisien MFCC yang diinginkan dan M adalah jumlah filter. **Gambar 2.11.** memberikan ilustrasi ekstraksi ciri menggunakan teknik MFCC dari satu ujaran.



Gambar 2.11. Ekstraksi Ciri Teknik MFCC dengan Panjang Frame 256

2.2.6. Hidden Markov Model Sebagai Pengenal Pola

Fenomena alam merupakan suatu sistem yang di dalamnya terdapat sejumlah entitas ataupun subsistem yang saling berinteraksi. Untuk mempelajari sistem fenomena alam tersebut, seringkali digunakan model. Secara umum ada tiga model, yaitu model fisik (miniatur sistem), model verbal, maupun model matematika. Di

dalam model matematika, interaksi antar entitas atau subsistem diwujudkan secara kuantitatif menggunakan formula matematika, yang bersifat deterministik, probabilistik, ataupun campuran keduanya. Hubungan dinamis pada kasus fenomena alam yang berupa kejadian acak (*random event*) dari waktu ke waktu dapat dengan baik dimodelkan secara probabilistik, yang dikenal dengan pemodelan stokastik.

Proses Stokastik

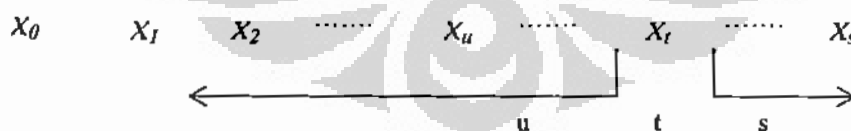
Proses stokastik merupakan suatu keluarga peubah acak X_t dengan t adalah suatu parameter *running over* pada suatu himpunan indeks T , [TK84]. Dalam hal ini t merepresentasikan waktu, frekuensi, atau jarak, dan biasanya dinotasikan dengan $T = \{0, 1, 2, \dots\}$ untuk diskret, atau $T = [0, \infty)$ untuk kontinyu. Suatu proses stokastik dispesifikasikan oleh :

- Ruang State (*state space*), yaitu semua kemungkinan nilai peubah X_t
- Himpunan Index (*index set*), T
- Keterkaitan antar peubah X_t .

Jika gugus T adalah diskret, maka disebut sebagai *discrete-time process*. Sedangkan jika kontinyu, maka disebut *continuous-time process*. Satu hal khusus dari proses stokastik adalah Rantai Markov (*Markov Chain*), $\{X_t\}$, atau sering juga disebut sebagai Proses Markov adalah suatu proses stokastik dengan sifat [TK84] :

“Jika diketahui nilai X_t , maka X_s dengan $s > t$, tidak tergantung dari X_u untuk $u < t$ ”

Sebagai ilustrasi perhatikan Gambar 2.12. berikut :



Gambar 2.12. Ilustrasi X_u , X_t , dan X_s suatu Rantai Markov

Suatu Rantai Markov dikatakan diskret (*Discrete Time Markov Chain*) kalau ruang space dari Proses Markov tersebut adalah himpunan yang terbatas (*finite*) atau tercacah (*countable*), dengan himpunan indeks adalah $T = \{0, 1, 2, \dots\}$. Jika nilai

suatu state pada periode tertentu hanya tergantung pada state satu periode sebelumnya, maka disebut sebagai Rantai Markov Orde Satu (*First Orde Markov Chain*). Rantai Markov orde satu ini secara matematika dirumuskan sebagai :

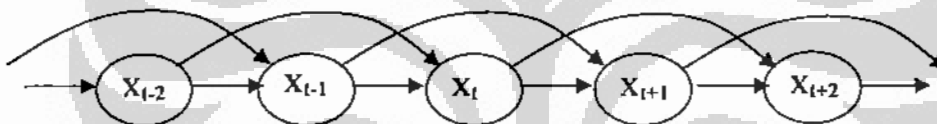
$$P\{X_{n+1}=j|X_0=i_0, \dots, X_{n-1}=i_{n-1}, X_n=i\} = P\{X_{n+1}=j|X_n=i\} \quad (2.28)$$

untuk semua waktu n dan state $i_0, \dots, i_{n-1}, i, j$. Ilustrasi rantai markov orde satu disajikan pada **Gambar 2.13**.



Gambar 2.13. Ilustrasi Rantai Markov Orde Satu

Sedangkan **Gambar 2.14**, berikut adalah ilustrasi untuk rantai markov orde dua, yaitu nilai X_t dipengaruhi oleh dua periode sebelumnya (X_{t-1} dan X_{t-2}) :



Gambar 2.14. Ilustrasi Rantai Markov Orde Dua

Peluang bahwa X_{n+1} berada pada state j kalau X_n berada dalam state i disebut sebagai peluang transisi satu step (*one-step transition probability*).

$$P_{ij}^{n,n+1} = P\{X_{n+1} = j | X_n = i\} \quad (2.29)$$

Jika peluang transisi ini bebas dari indeksnya, maka disebut proses markov dengan peluang transisi stasioner, sehingga peluang transisi tersebut dapat ditulis :

$$P_{ij}^{n,n+1} = P_{ij} \quad (2.30)$$

Peluang transisi ini disusun dalam bentuk matriks, yang disebut matriks peluang transisi, P , yang setiap unturnya adalah P_{ij} , yaitu $P(X_{t+1}=j|X_t=i)$. Matriks P ini berdimensi $s \times s$, dengan s adalah banyaknya kemungkinan nilai state, dan ditulis sebagai :

$$P_{sxs} = \begin{pmatrix} p_{11} & p_{12} & \dots & p_{1s} \\ p_{21} & p_{22} & \dots & p_{2s} \\ \dots & \dots & \dots & \dots \\ p_{s1} & p_{s2} & \dots & p_{ss} \end{pmatrix}$$

Suatu proses markov telah didefinisikan dengan lengkap kalau telah ditentukan matriks peluang transisi dan peluang setiap state pada tahap awal, yaitu :

$$P\{X_0=i\}=p_i, \text{ untuk } i=1, 2, 3, \dots, s.$$

Dengan spesifikasi ini maka nilai $P\{X_0=i_0, X_1=i_1, X_2=i_2, \dots, X_{n-1}=i_{n-1}, X_n=i_n\}$ dapat dihitung secara rekursif dengan rumus berikut :

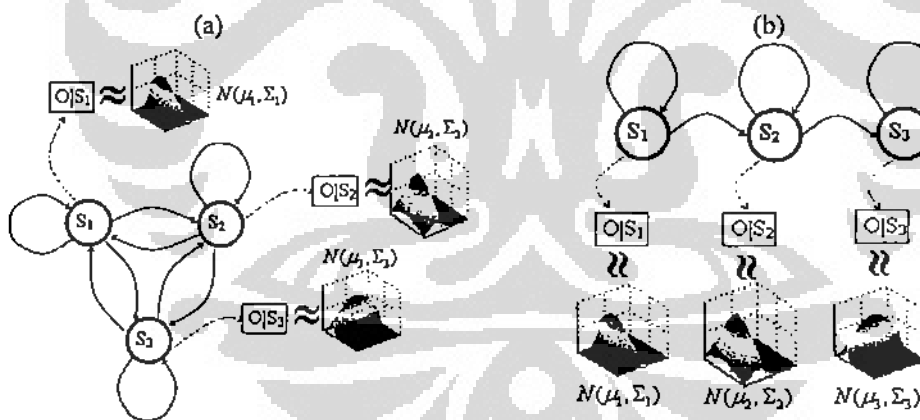
$$\begin{aligned} P\{X_0 = i_0, X_1 = i_1, X_2 = i_2, \dots, X_{n-1} = i_{n-1}, X_n = i_n\} \\ = p_{i_0} P_{i_0 i_1} \dots P_{i_{n-2} i_{n-1}} P_{i_{n-1} i_n} \end{aligned} \quad (2.31)$$

Pada suatu kondisi tertentu, nilai-nilai peubah X pada proses Markov di atas tidak teramati secara langsung (disebut sebagai *unobservable variable* atau *unobservable state* atau *hidden state*), namun dapat dievaluasi dari peubah lain yang dapat diamati secara langsung (disebut *observable variable* atau peubah *emitten*) dan merupakan efek dari peubah tak teramati X tersebut. Sebagai contoh pada bidang kesehatan, kondisi jantung seorang pasien (tidak teramati) dapat dievaluasi berdasar tekanan darah, suhu ataupun peubah *emitten* lain yang dapat diobservasi langsung. Untuk situasi seperti ini, model Markov tersebut diperluas dan dikenal dengan nama Model Markov Tersembunyi (*Hidden Markov Model*, HMM). Oleh karena itu, selain parameter peluang transisi antar state, juga diperlukan distribusi untuk peubah *emitten* (*observable state*) yang dalam hal ini merupakan peluang bersyarat (*conditional probability*).

Perumusan *Hidden Markov Model*

Hidden Markov Model (HMM) merupakan perluasan dari model markov orde satu maupun orde dua, yang dalam hal ini kondisi state merupakan peubah yang tidak teramati secara langsung, dan disebut sebagai *hidden state* (S). Setiap *hidden state*

dapat menghasilkan suatu *outcome* yang teramati pada setiap periode t , yaitu O_t . *Outcome* dari *hidden state* ini disebut sebagai *observable state* atau *observation state* atau *emitten*. Oleh karena itu, dari periode $t=1$ hingga $t=T$ diperoleh barisan peubah teramati (*observation state*) $O=O_1, O_2, O_3, \dots, O_T$, yang merupakan *outcome* dari barisan peubah tak teramati $Q=q_1, q_2, q_3, \dots, q_T$. *Observation state* ini dapat berupa skalar atau vektor, diskret ataupun kontinu dengan distribusi peluang mengikuti *hidden state* sebagai sumbernya. Secara visual, model ini dapat digambarkan menggunakan suatu *finite state automata* dengan banyaknya *state* adalah sesuai dengan banyaknya kemungkinan nilai peubah *hidden*, dan *link* yang ada menunjukkan transisi antar *state*. Berdasar hubungan antar *state*, dikenal dua jenis HMM, yaitu *ergodic* dan *left-right* HMM. Pada *Ergodic* HMM, antar dua *state* selalu ada *link*, sehingga disebut juga sebagai *fully connected* HMM. Sedangkan pada *left-right* HMM, *state* dapat disusun dari kiri ke kanan sesuai dengan *link*-nya. **Gambar 2.15.** memberikan contoh *finite state automata* untuk *ergodic* dan *left-right* HMM dengan tiga *hidden state* dengan distribusi peubah *emitten*-nya adalah *Gaussian*.



Gambar 2.15. Contoh HMM dengan Tiga Hidden State dan Distribusi Emitten *Gaussian*, (a) *Ergodic*, (b) *Left-Right* HMM

Untuk mempermudah perumusan matematika, berikut disajikan notasi-notasi mengenai HMM [DD96].

T : Panjang barisan observasi atau panjang periode pengamatan.

N : Banyaknya kemungkinan nilai *hidden state*.

- S : Himpunan nilai-nilai *state* yang mungkin, $S = \{S_1, S_2, S_3, \dots, S_N\}$
- Q : $(q_1, q_2, q_3, \dots, q_T)$ adalah barisan *state* dari periode ke 1 hingga T, q_t adalah *state* yang dikunjungi pada periode t
- M : Banyaknya kemungkinan kemunculan peubah teramati
- V : Himpunan kemungkinan observasi, $V = \{v_1, v_2, v_3, \dots, v_M\}$
- Π : Adalah himpunan $\{\pi_i\}$, dengan $\pi_i = P(q_1=i)$, yaitu peluang pada tahap awal berada pada *state* i. Dalam hal ini berlaku $\sum_{i=1}^N \pi_i = 1$
- A : Adalah himpunan $\{a_{ij}\}$ dengan $a_{ij} = P(q_{t+1}=S_j | q_t=S_i)$, yaitu peluang berada di *state* S_j pada waktu t+1, kalau pada waktu t berada di *state* S_i . Dalam hal ini diasumsikan a_{ij} bebas dari waktu.
- B : Adalah himpunan $\{b_j(k)\}$, dengan $b_j(k) = P(v_k \text{ pada waktu } t | q_t=S_j)$, yaitu peluang peubah teramati yang muncul adalah simbol v_k kalau *state* yang terjadi adalah S_j .
- O : $(O_1, O_2, O_3, \dots, O_T)$ adalah barisan observasi, dengan O_t sebagai nilai atau vektor yang teramati (*observable symbol*) pada waktu t.

Dengan notasi-notasi seperti di atas, maka suatu *Hidden Markov Model* (HMM) dilambangkan dengan :

$$\lambda = (A, B, \Pi)$$

Dari sebuah model HMM, dikenal tiga algoritme sesuai problem yang akan dijawab, yaitu algoritme evaluasi yang dipergunakan untuk menduga peluang kemunculan sebuah barisan observasi, algoritme pelatihan untuk menduga nilai-nilai parameter HMM dan algoritme *decode* untuk menduga kemungkinan barisan *state*. Berikut disajikan perumusan ke tiga algoritme tersebut.

1. **Problem 1 (Evaluation)** : Untuk suatu $\lambda = (A, B, \Pi)$ tertentu, ingin diketahui $P(O|\lambda)$, yaitu peluang munculnya barisan $O = O_1, O_2, O_3, \dots, O_T$.

Solusi :

Barisan $O = O_1, O_2, O_3, \dots, O_T$ adalah nilai teramati yang merupakan refleksi atau *emitten* dari barisan *hidden state* $Q = q_1, q_2, \dots, q_T$. Untuk suatu barisan *hidden state* tertentu, $Q = q_1, q_2, \dots, q_T$, nilai $P(O|\lambda)$ dapat dihitung dengan penurunan berikut :

$$P(O|Q, \lambda) = \prod_{t=1}^T P(O_t | q_t, \lambda) \quad (2.32)$$

Dengan asumsi kebebasan setiap antar observasi, maka nilai tersebut dapat dirumuskan menjadi :

$$P(O|Q, \lambda) = b_{q_1}(O_1) \cdot b_{q_2}(O_2) \cdots b_{q_T}(O_T) \quad (2.33)$$

Sedangkan peluang kemunculan barisan $Q = q_1, q_2, \dots, q_T$ tertentu adalah :

$$P(Q|\lambda) = \pi_{q_1} a_{q_1 q_2} a_{q_2 q_3} \cdots a_{q_{T-1} q_T} \quad (2.34)$$

Distribusi bersama O dengan Q diperoleh dengan mengalikan keduanya :

$$P(O, Q|\lambda) = P(O|Q, \lambda) \cdot P(Q|\lambda) \quad (2.35)$$

Oleh karena itu nilai $P(O|\lambda)$ diperoleh dengan menjumlahkan formula di atas untuk semua kombinasi barisan *hidden state* yang mungkin.

$$\begin{aligned} P(O|\lambda) &= \sum_{\forall Q} P(O|Q, \lambda) P(Q|\lambda) \\ &= \sum_{\forall Q} \pi_{q_1} b_{q_1}(O_1) a_{q_1 q_2} b_{q_2}(O_2) \cdots a_{q_{T-1} q_T} b_{q_T}(O_T) \\ &= \sum_{\forall Q} \pi_{q_1} b_{q_1}(O_1) \prod_{t=2}^T a_{q_{t-1} q_t} b_{q_t}(O_t) \end{aligned} \quad (2.36)$$

dengan $Q = q_1, q_2, q_3, \dots, q_T$. Terlihat orde perkalian tersebut adalah $(2TN^T)$. Sebagai ilustrasi, dari sebuah sinyal suara dengan durasi 1 detik yang disampling dengan frekuensi 1,28 kHz dan dibaca per frame 30 ms dengan overlap antar frame 40% akan diperoleh T sebesar 71. Hal ini berarti jumlah komputasi untuk menghitung peluang observasi dari sebuah suara yang hanya 1.28 detik dengan model HMM dengan tiga *hidden state*

adalah kurang lebih sebesar $2 \cdot 71 \cdot 3^{71}$, sehingga diperlukan algoritme yang efisien. Ada dua algoritme yang bisa diterapkan, yaitu algoritme *Forward* dan *Backward*, [Rab89], dengan kompleksitas $O(2N^2T)$. Untuk kasus di atas, jumlah komputasi hanya sekitar $2 \cdot 3^2 \cdot 71$ atau 1278. Secara lengkap, algoritme *forward* disajikan pada Lampiran 1 dan *Backward* pada Lampiran 2.

2. **Problem 2 (Decoding)** : Proses *decoding* dari model $\lambda = (A, B, \Pi)$ adalah memilih barisan state $Q = q_1, q_2, \dots, q_T$ yang 'optimal', yaitu yang paling besar kemungkinannya menghasilkan observasi $O = O_1, O_2, O_3, \dots, O_T$.

Solusi :

Pada problem 1, solusi diperoleh melalui penjumlahan peluang observasi pada semua kemungkinan barisan *state* yang bisa terjadi, sehingga solusi yang diberikan bersifat pasti. Sedangkan pada problem 2, solusi tergantung dari kriteria optimum yang dipakai. Ada beberapa kriteria optimum, yaitu :

- a. Memaksimumkan banyaknya *hidden state* yang sesuai.

Besaran untuk optimisasi ini adalah $\gamma_t(i) = P(q_t = S_i | O, \lambda)$ yaitu peluang pada periode t , *state* yang muncul adalah S_i kalau diketahui observasinya adalah O , dan dirumuskan sebagai :

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{P(O | \lambda)} = \frac{\alpha_t(i)\beta_t(i)}{\sum_{i=1}^N \alpha_t(i)\beta_t(i)} \quad (2.38)$$

Hidden state yang paling mungkin untuk setiap periode t adalah :

$$q_t = \arg \max_{1 \leq i \leq N} \gamma_t(i) \quad \text{untuk } 1 \leq t \leq T \quad (2.39)$$

Kelemahan algoritme ini adalah tidak memperhatikan adanya transisi *state* yang tidak mungkin, sehingga tidak menutup kemungkinan munculnya barisan *state* yang 'janggal'.

- b. Modifikasi kriteria point a, yaitu dengan memaksimalkan banyaknya segmen *hidden state* yang benar, yaitu dua (q_t, q_{t+1}), tiga (q_t, q_{t+1}, q_{t+2}), atau lebih segmen *hidden state* yang berurutan.
 - c. Menemukan satu barisan *hidden state (path)* yang paling sesuai. Solusi dengan kriteria ini diperoleh dengan memaksimalkan peluang kemunculan barisan *state*, Q , untuk O dan λ tertentu, $P(Q|O, \lambda)$, yang setara dengan memaksimalkan $P(O, Q|\lambda)$. Pencarian path ini dilakukan dengan konsep pemrograman dinamik (*dynamic programming*) dan dikenal dengan algoritme *viterbi* yang disajikan pada Lampiran 3.
3. **Problem 3 (Learning)** : Problem 3 ini adalah berkaitan dengan pembelajaran model HMM dengan menggunakan data yang ada, yang pada dasarnya adalah melakukan pendugaan terhadap parameter model HMM, yaitu A , B dan J . Seperti pada proses pendugaan parameter pada umumnya, hasil pendugaan tergantung kriteria optimum yang dipakai.

Solusi :

Yang menjadi tujuan dari pembelajaran adalah menentukan parameter model HMM dari suatu set data, sedemikian sehingga model mampu mengenali obyek baru yang mempunyai karakteristik "mirip" dengan data yang dipergunakan untuk training tersebut. Secara umum sudah dikenal dua jenis pendugaan parameter HMM, yaitu teknik Segmental K-Means yang menggunakan kriteria memaksimalkan $P(O, Q|\lambda)$. Jadi dalam hal ini, dari observasi O , parameter HMM diarahkan sehingga peluang kemunculan observasi O dan barisan *state* Q tersebut maksimum. Teknik yang kedua adalah Baum-Welch yang kriterianya adalah memaksimalkan peluang kemunculan barisan observasi O , yaitu $P(O|\lambda)$, [DHS01]. Algoritme detail dari Segmental K-means disajikan pada Lampiran 4, dan Baum-Welch pada Lampiran 5.

BAB 3 SISTEM IDENTIFIKASI PEMBICARA

Bagian ini menyajikan materi tentang sistem identifikasi pembicara (SIP), dengan tujuan untuk memberikan gambaran mengenai posisinya pada ranah pemrosesan suara, komponen SIP serta riset-riset yang sudah ada. Agar diperoleh gambaran yang lebih menyeluruh maka pembahasan diawali dengan menyajikan sekilas mengenai *state of the art* dari bidang pemrosesan suara dan bahasa, dilanjutkan dengan area dari pemrosesan suara, komponen-komponen SIP, dan diakhiri dengan menyajikan perkembangan riset yang ada.

3.1. *State of the Art* Pemrosesan Suara dan bahasa

Kajian mengenai pemrosesan suara dan bahasa (*Speech and Language Processing*) dilakukan diberbagai bidang, seperti Ilmu Komputer (Pemrosesan Bahasa Alami, *Natural Language Processing, NLP*), Ilmu Bahasa (Komputasi Linguistik, *Computational Linguistics*), Elektro (*Speech Recognition*), Psikologi (Komputasi Psikolinguistik, *Computational Psycholinguistics*). Sejarah perkembangan pemrosesan suara dan bahasa sejalan dengan perkembangan teknologi komputer itu sendiri, dan dibagi dalam beberapa tahap.

Tahap 1940 – 1960 : Pada tahap ini ada dua paradigma, yaitu automata (yang melandasi teori bahasa formal, *formal language theory*) dan probabilistik (yang melandasi model teori informasi) untuk pemrosesan suara. Model komputasi Turing mendasari munculnya automaton dan berkembang ke *finite state automata* dan ekspresi regular (Kleene, 1951 dan 1956). Shannon (1948) mengembangkan model probabilistik (Proses Markov Diskret) untuk pemrosesan bahasa. Hal ini diikuti oleh Chomsky (1956) yang mengembangkan *finite state grammar* (context-free grammar) untuk bahasa alami.

Paradigma kedua adalah model komputasi probabilistik untuk pemrosesan suara dan bahasa. Pada tahap ini dikembangkan suatu metaphor untuk *noisy*

channel dan *decoding* untuk transmisi bahasa melalui media komunikasi oleh Shannon. Shannon juga memperkenalkan konsep entropi dari teori termodinamika sebagai ukuran kapasitas informasi suatu channel, kandungan informasi suatu bahasa, dan pertama kali dikembangkan ukuran entropi untuk model probabilistik bahasa Inggris. Pada tahap ini pertama kali dikembangkan spektograph (Koenig et. Al., 1946) yang memunculkan penelitian dalam bidang *fonetic* yang merupakan dasar dari *speech recognition*, dan dari sinilah ditemukan mesin *speech recognizers* yang pertama (1950). Pada tahun 1952 peneliti dari Bell Labs (Davis et al.) mengembangkan *speaker-dependent recognizer* dengan model statistik yang mampu mengenali 10 digit yang merepresentasikan dua formant pertama untuk vokal. Mesin yang dilatih dengan 10 *speaker* ini mampu mengenali 10 digit dari sembarang *speaker* dengan akurasi 97-99% yang berbasis template berdasar korelasi antara pattern dengan input.

Tahap 1957 – 1970 : Penelitian bidang *speech recognition* berada pada dua paradigma, yaitu *symbolic* dan *stochastic*. Penelitian pada bidang *symbolic* banyak dilakukan oleh ahli di bidang komputer ataupun linguistik. Sedangkan penelitian pada bidang *stochastic* banyak dilakukan di departemen statistika ataupun *electrical engineering*.

Jalur simbolik mengikuti penelitian dari Chomsky yang mengembangkan berbagai penelitian seperti : algoritme parsing, juga berbagai algoritme pada *artificial intelligent* (John McCarthy, Marvin Minsky, Claude Shannon, dan Nathaniel Rochester). Pada tahap ini mulai dikembangkan *natural language understanding* sederhana yang mampu melakukan *reasoning* untuk memberikan jawaban pertanyaan.

Pada pendekatan *stochastic*, mulai dikembangkan berbagai sistem seperti : *optical character recognition* dan *text-recognition* (Bledsoe dan Browning), yang menerapkan metode Bayes. Pada tahap ini mulai dikembangkan korpus

yang memuat satu juta kata yang diambil dari 500 teks dari berbagai sumber (surat kabar, novel, nonfiksi, akademik, dsb.).

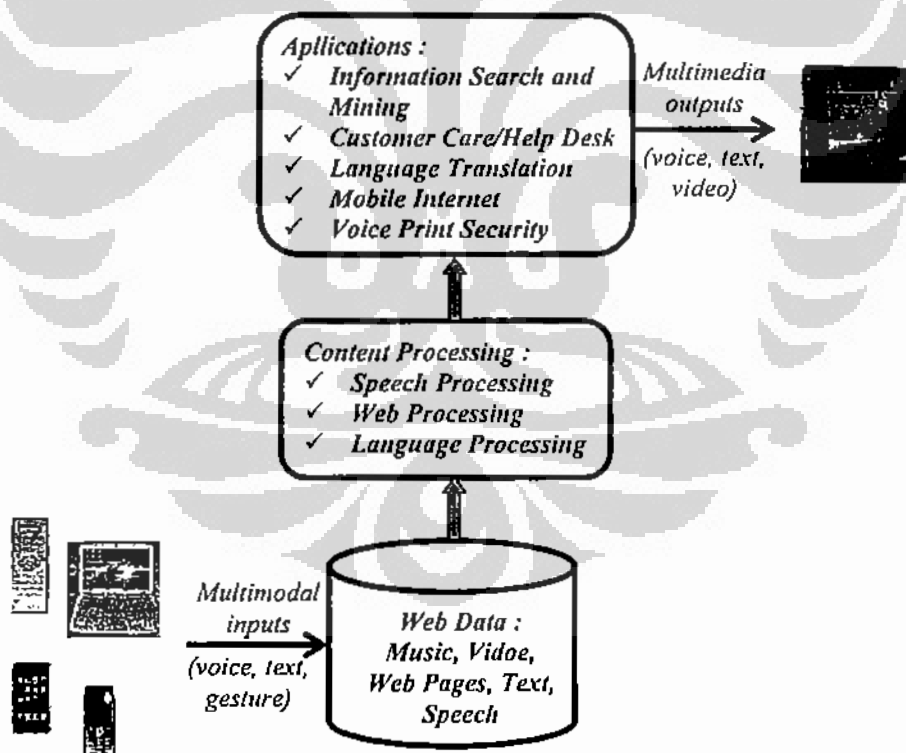
Tahap 1970 – 1983 : Pada tahap ini banyak sekali dilakukan penelitian mengenai *speech and language processing*, baik dengan paradigma *stochastic* maupun paradigma *logic*. Pada paradigma *stochastic*, terdapat beberapa penelitian seperti penerapan *Hidden Markov Model* (HMM) untuk algoritme pengenalan suara, juga metaphor dari *noisy channel* dan *decoding*, yang dilakukan secara terpisah oleh Jelinek, Bahl, Mercer, para ahli dari IBM dan dari Carneige Mellon University (CMU). Rabiner dan Juang (AT&T's Bell Labs) melakukan penelitian dalam bidang *speech recognition* dan *synthesis*. Pada paradigma logika (*logic based*), penelitian terutama pada pengembangan *grammar* (*metamorphosis grammar*, *definite clause grammars*, *functional grammars*) juga struktur unifikasi.

Pada tahap ini dikembangkan *natural language understanding* yang berupa robot yang dapat menerima perintah berupa teks berbahasa alami. Penelitian pada bidang *natural language understanding* ini menerapkan model-model *grammar*, *parsing*, *semantik* dan model *discourse*.

Tahap 1983 – 1993 : Pada tahap ini ada dua trend yang berkembang. Pertama adalah penelitian *finite-state model*, seperti *finite state* untuk fonologi dan morfologi, juga *finite state* untuk sintaks. Kedua adalah penelitian empiris mengenai *speech recognition* menggunakan model-model probabilistik. Penelitian yang banyak dipelopori oleh ahli dari IBM dengan pendekatan *data driven* ini menfokuskan pada *part-of-tagging*, *parsing* dan *attachment ambiguities*, dan *conectionist speech recognition* hingga analisis semantik.

Tahap 1994 – 1999 : Pada akhir milenium ke 20, model-model probabilistik dan *data driven* menjadi standar pengembangan sistem pemrosesan bahasa alami. Algoritme-algoritme untuk *parsing*, *part-of-speech tagging*, *reference resolution* dan pemrosesan wacana (*discourse processing*) didasarkan pada

konsep peluang dan diterapkan pada *speech recognition* maupun *information retrieval*. Dengan berkembangnya kecepatan dan *memory* komputer mendorong munculnya berbagai produk mengenai *speech and language processing*, terutama *speech recognition*, *spelling* dan *grammar checking*. Pada tahap ini juga mulai muncul pemikiran mengenai *information retrieval* dan *extraction* melalui Web yang didasarkan pada bahasa (*language-based information retrieval and information extraction*). Pada dekade akhir abad 20 dan awal abad 21 sekarang ini, trend komunikasi pada berbagai bidang mengarah pada era digital melalui internet, maka teknologi pemrosesan suara dan bahasa menjadi mesin pendorong terjadinya perubahan cara seseorang berkomunikasi dan mengakses informasi, baik yang berupa teks, suara, video, grafik, maupun audio. **Gambar 3.1.** menyajikan protokol sederhana dari aplikasi berbasis web yang menggunakan berbagai alat akses, [GF08].

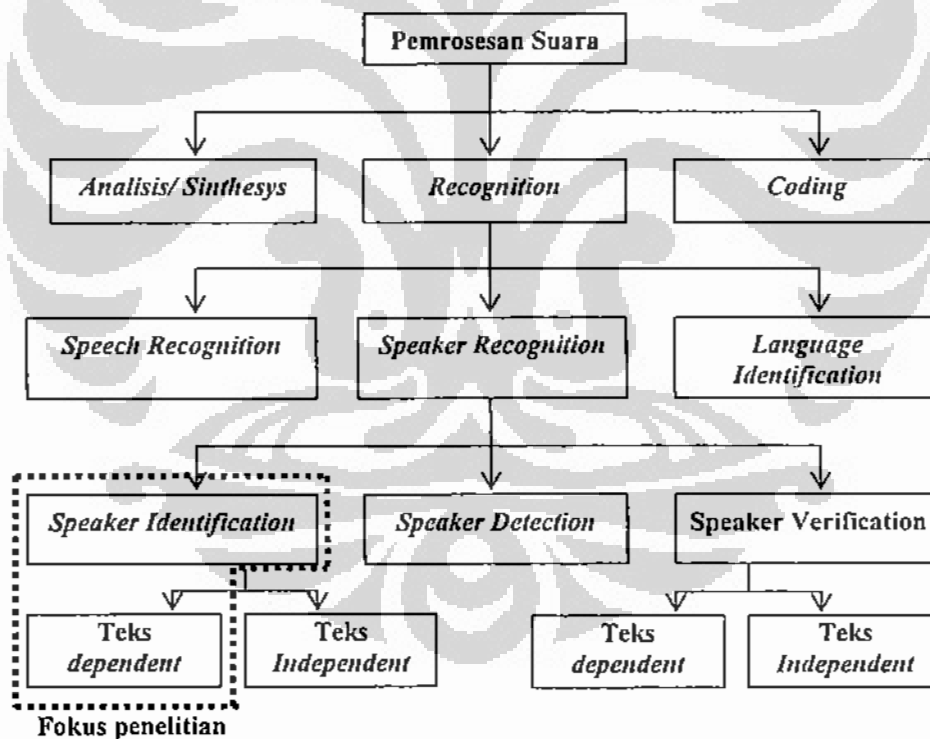


Gambar 3.1. Contoh Protokol Aplikasi Berbasis Web

Beberapa aplikasi yang menantang, seperti *voice search* yang mengabungkan *automatic speech recognition* dengan *document search*, *spoken document retrieval* (SDR), dan *spoken language understanding and translation* (*bilingual* dan *multilingual*) belum menemukan solusi yang optimun, sehingga bidang ini masih terbuka lebar untuk pengembangan riset.

3.2. Ranah Pemrosesan Suara

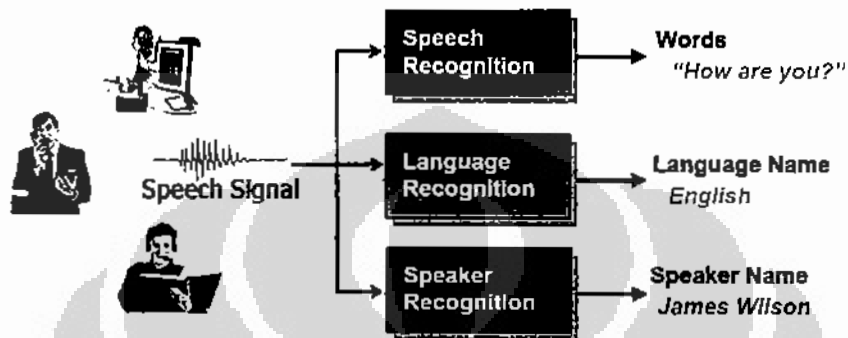
Kalau pada bagian sebelumnya telah disajikan paradigma dari perkembangan riset bidang pemrosesan suara dan bahasa, serta tantangan aplikasi komersial yang ada, maka pada bagian ini ditunjukkan cakupan riset tentang pemrosesan sinyal suara dari sudut pandang ranah kajian. Campbell, 1997, menyebutkan bahwa pemrosesan suara mempunyai tiga ranah kajian, yaitu masalah *synthesis*, *recognition* dan *coding*, seperti yang sajikan pada **Gambar 3.2.**, [Cam97].



Gambar 3.2. Cakupan Kajian dalam Pemrosesan Suara

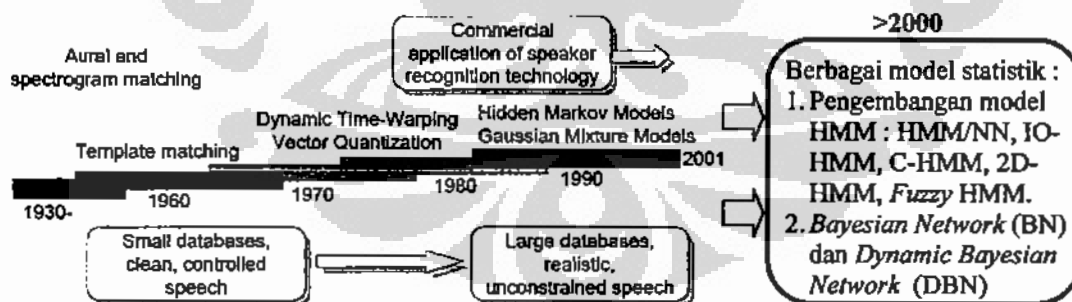
Kajian bidang *Recognition* diarahkan pada pemrosesan suara untuk klasifikasi yang secara umum terdiri dari dua sub sistem, yaitu ekstraksi ciri dan pengenal pola.

Sesuai keluaran sistem, bidang *recognition* dapat dipisahkan menjadi tiga ranah kajian, yaitu pengenalan suara (*speech recognition*), pengenalan pembicara (*speaker recognition*), dan identifikasi bahasa (*language identification*). Perbedaan dari ketiga area tersebut diperlihatkan pada **Gambar 3.3.**, [Rey02].



Gambar 3.3. Perbedaan Kajian pada Ranah *Recognition*

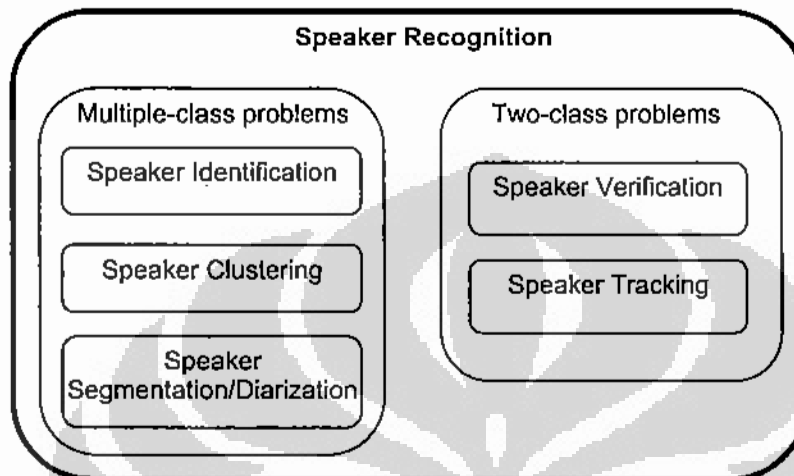
Hal yang dilakukan pada area pengenalan (*recognition*) adalah untuk mengekstrak informasi yang terkandung di dalam sinyal suara, [Rey02]. Penelitian di bidang pengenalan ini telah terjadi sejak tahun 1960 mulai dari pemodelan yang bersifat deterministik hingga probabilistik seperti disajikan pada **Gambar 3.4.**, (dimodifikasi dari [Rey02]).



Gambar 3.4. Perkembangan Penelitian Bidang Pengenalan Suara

Pengenalan Pembicara (*Speaker Recognition*) merupakan suatu proses yang secara otomatis mengenali siapa pembicara (*who is speaking*) menggunakan informasi spesifik yang terkandung pada sinyal suara, [Fur97]. Berdasar keluaran

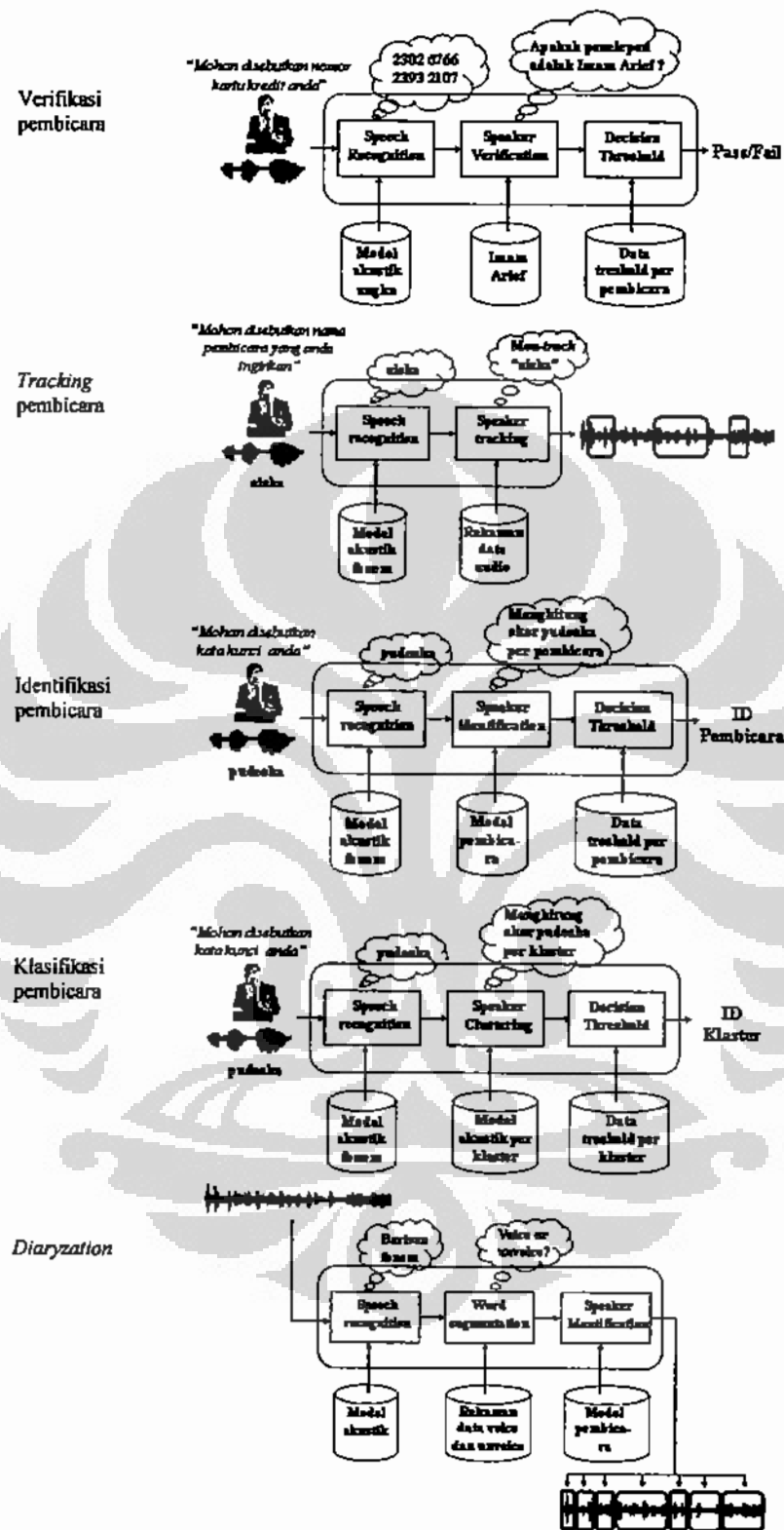
sistem, Gancev membagi sistem pengenalan pembicara menjadi dua, yaitu *multiple-class problem* dan *two-class problem* [Gan05], seperti disajikan pada Gambar 3.5.



Gambar 3.5. Sistem Pengenalan Pembicara Berdasar Jumlah Kelas Output

Gambar 3.6. menyajikan alur yang terdiri dari input-proses-output dari ke lima kajian pada *speaker recognition* di atas. *Speaker Verification* adalah suatu permasalahan dua kelas, yaitu sistem akan menolak atau menerima suatu klaim mengenai identitas seorang pembicara dengan berdasarkan data suara yang diberikan. Sementara itu, pada *speaker tracking /detection*, sistem akan mendeteksi bagian atau segmen waktu yang merupakan suara seorang pembicara tertentu dari sebuah segmen sinyal suara dengan durasi tertentu.

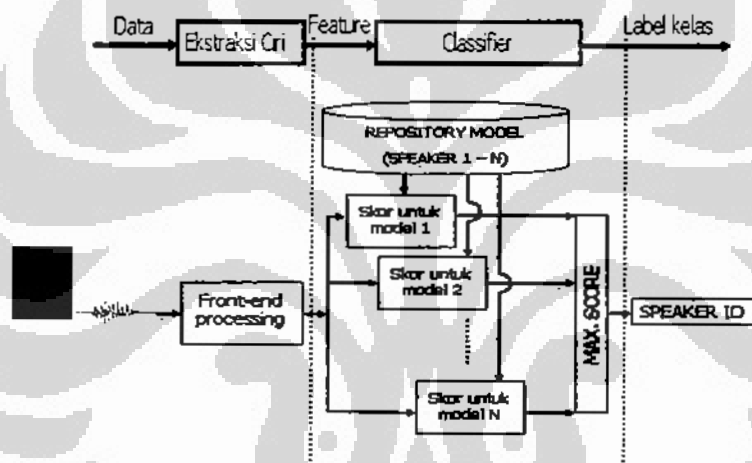
Speaker segmentation akan memberikan label pada setiap segmen tertentu dari sinyal input sesuai kode pembicara yang paling sesuai. *Speaker clustering* melakukan pengelompokkan sehingga sinyal suara yang mirip ada dalam satu kelompok yang merupakan milik seorang pembicara atau kelas pembicara ke dalam database. Sedangkan *speaker identification* adalah menentukan pembicara yang paling mungkin dari sinyal suara yang diberikan. Kajian yang akan dilakukan pada penelitian ini adalah pada bidang *speaker recognition* dan difokuskan pada *speaker identification* atau identifikasi pembicara.



Gambar 3.6. Kajian-Kajian pada Ranah Pengenalan Pembicara

3.3. Sistem Identifikasi Pembicara

Identifikasi pembicara merupakan proses untuk menentukan secara otomatis siapa pemilik dari suara yang diberikan ke dalam sistem. Blok diagram dari sistem identifikasi pembicara adalah seperti disajikan pada Gambar 3.7. Pada sistem tersebut seorang pembicara yang akan diidentifikasi berdasarkan suaranya mengucapkan suatu kata atau frase tertentu. Berikutnya pada bagian ekstraksi ciri dihitung nilai ciri (*feature*) dari sinyal suara masukan. Nilai ciri inilah yang diproses di bagian pengenalan (*classifier*) untuk diberikan skor sesuai kelas yang telah ada dalam sistem. Sistem akan memberikan label kelas dari sinyal suara masukan tersebut sesuai skor tertinggi.

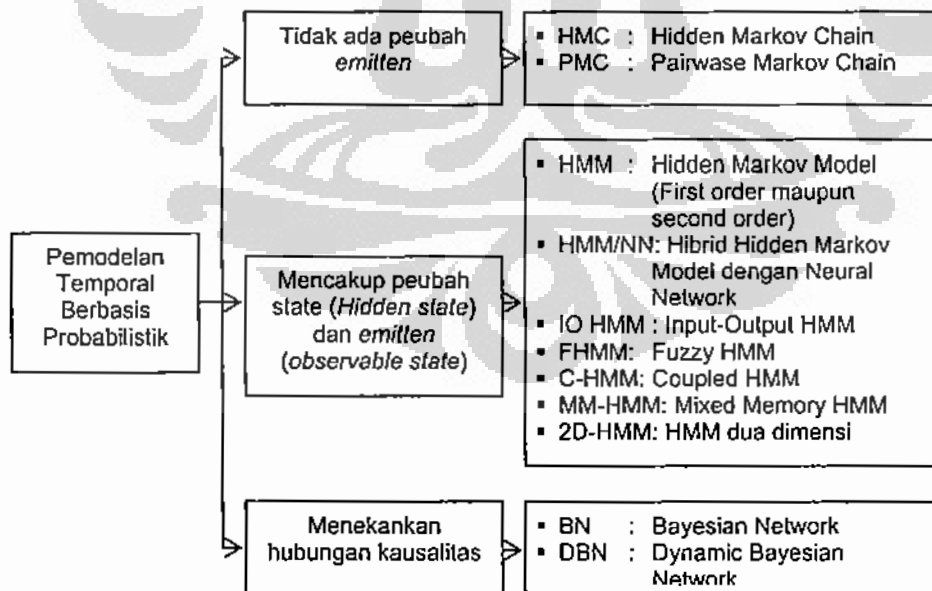


Gambar 3.7. Blok Diagram Sistem Identifikasi Pembicara

Input dari sistem tersebut adalah sinyal suara yang berupa gelombang. Pada bagian ini, dilakukan digitasi energi suara yang berupa gelombang analog untuk menghasilkan sinyal digital, dengan cara sampling, dilanjutkan kuantisasi dan *coding*. Setelah diperoleh nilai digital sinyal suara masukan, maka sebelum masuk ke ekstraksi ciri, dilakukan penghapusan bagian *silence* pada sinyal tersebut, lalu dibaca dari frame ke frame dengan panjang tertentu dan saling overlap. Kepada setiap frame ini dilakukan proses *windowing* dengan fungsi *window* tertentu, dan dilanjutkan dengan proses ekstraksi ciri.

Keluaran ekstraksi ciri ini akan masuk ke sub sistem *classifier* untuk dilakukan proses pengenalan. Dalam hal ini ada dua tipe pengenalan, yang pertama adalah seluruh hasil ekstraksi ciri dari semua frame pada frase ujaran diproses secara bersama-sama (digabungkan atau mungkin juga dirata-ratakan sehingga menjadi satu vektor ciri) menjadi masukan sub sistem *classifier* untuk dikenali. Jenis pengenalan yang melakukan proses seperti ini misalnya adalah *template matching* dan *neural network*. Namun demikian, ada juga tipe pengenalan yang membaca sinyal masukan frame demi frame sesuai periode diujarkannya, dan setelah semua frame diproses, baru diberikan skor bagi sinyal. Tipe kedua ini dilakukan oleh pemodelan temporal/spasial seperti misalnya yang dilakukan oleh model berbasis proses Markov.

Dalam perkembangannya, model berbasis proses Markov menjadi trend dari sistem pemrosesan sinyal, khususnya pada sub sistem *classifier*, sehingga telah dikenal berbagai macam variasi dari model Markov tersebut, seperti disajikan pada **Gambar 3.8**. Pada gambar tersebut, yang dimaksud peubah *emitten* adalah peubah yang kemunculannya sebagai efek dari peubah lain yang tidak dapat diobservasi secara langsung yang disebut peubah *state*.



Gambar 3.8. Klasifikasi Pemodelan Temporal Berbasis Probabilistik

Pada pemodelan yang tidak melibatkan peubah *emitten*, barisan peubah acak dimodelkan mengikuti proses markov, dan hal ini dikenal dengan *Hidden Markov Chain*, HMC. Andaikan peubah acak tersebut muncul secara berpasangan, maka fenomena tersebut bisa dimodelkan dengan model *Pairwise Markov Chain*, PMC. Sesuai dengan [Pie03], PMC didefinisikan sebagai berikut : Kalau $X=(X_1, X_2, X_3, \dots, X_n)$ dan $Y=(Y_1, Y_2, Y_3, \dots, Y_n)$ adalah dua barisan peubah acak temporal berpasangan dan ditulis sebagai $Z=(Z_1, Z_2, Z_3, \dots, Z_n)$ dengan $Z_i=(X_i, Y_i)$, maka Z adalah PMC berkaitan dengan X dan Y jika distribusi Z dapat dirumuskan sebagai :

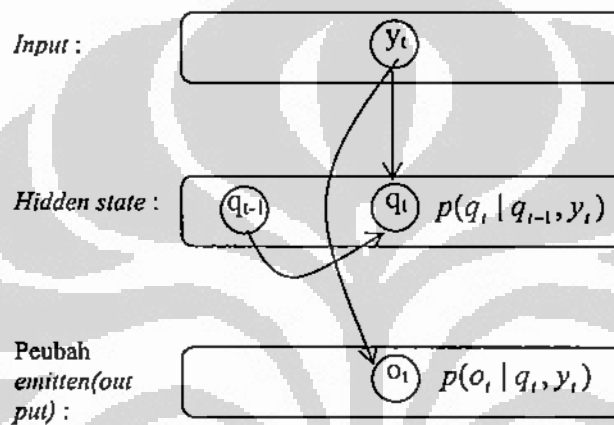
$$p(z) = \frac{p(z_1, z_2)p(z_2, z_3)\dots p(z_{n-1}, z_n)}{p(z_2)p(z_3)\dots p(z_{n-1})}$$

dengan $p(z_i)$ merupakan distribusi marjinal dari $p(z_{i-1}, z_i)$ dan juga sebagai distribusi marjinal dari $p(z_i, z_{i+1})$. Pemakaian model PMC pada kasus segmentasi citra yang terdistorsi oleh *noise* memberikan hasil yang lebih baik dibanding dengan model HMC, seperti yang ditunjukkan pada [DP04].

Pada suatu kondisi tertentu, nilai-nilai peubah X pada proses Markov di atas tidak teramati secara langsung (disebut sebagai *unobservable variable* atau *unobservable state* atau *hidden state*), namun dapat dievaluasi dari peubah lain yang dapat diamati secara langsung (disebut *observable variable* atau peubah *emitten*) yang merupakan efek dari peubah tak teramati X tersebut. Sebagai contoh misalkan pada bidang kesehatan, kondisi jantung (tidak teramati) seorang pasien dapat dievaluasi berdasar tekanan darah, suhu ataupun peubah *emitten* lain yang dapat diobservasi langsung. Untuk situasi seperti ini, model Markov diperluas menjadi Model Markov Tersembunyi (*Hidden Markov Model*, HMM). Oleh karena itu, selain parameter peluang transisi antar *state*, juga diperlukan distribusi untuk peubah *emitten* yang dalam hal ini merupakan peluang bersyarat (*conditional probability*). Jika distribusi peluang untuk peubah observasi (*emitten*) ini diduga dengan menggunakan jaringan syaraf tiruan (*neural network*), maka dikenal menjadi model HMM/NN. Penerapan teknik HMM/NN ini sebagai *classifier* pada *speaker recognition* dapat dilihat pada

[Gan05]. Sedangkan neural network secara tersendiri untuk identifikasi pembicara dapat dilihat pada [PKM05].

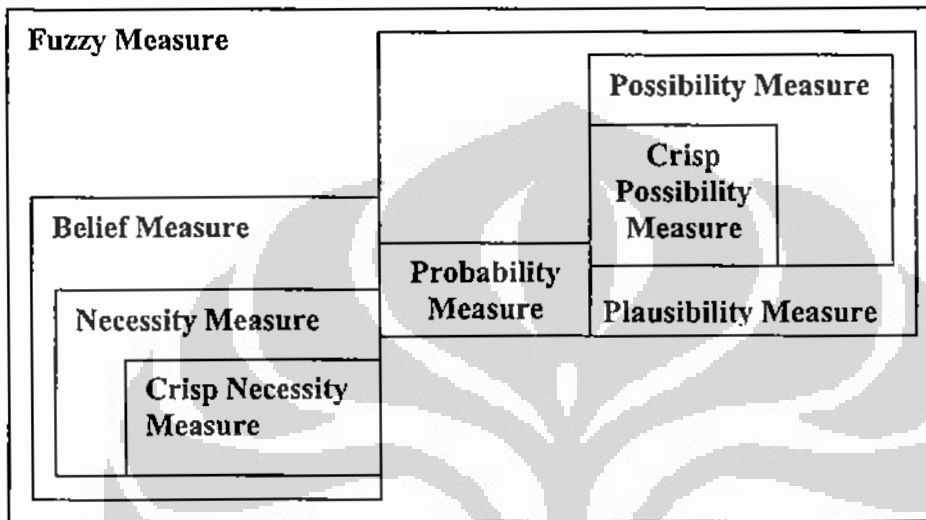
IO-HMM merupakan model HMM yang dalam hal ini distribusi peubah *emitten* selain sebagai *influence* dari peubah *state*, juga tergantung dari input yang diberikan. Oleh karena itu, pada IO-HMM ada tiga barisan peubah, yaitu input, *state* dan *emitten* yang juga disebut sebagai output, seperti disajikan pada Gambar 3.9.



Gambar 3.9. Hubungan Input, Hidden State dan Emitten pada IO-HMM

Variasi lain dari model HMM adalah seperti yang dilakukan oleh Mohamed dan Gader pada [MG00]^a yang menggunakan konsep ukuran kekaburan (*fuzzy measure*) yang dalam hal ini adalah λ -measure sebagai ukuran ketidakpastian (*uncertainty measure*). Model yang dikembangkannya ini disebut sebagai *Fuzzy Hidden Markov Model*, FHMM. Pada HMM biasa, ukuran ketidakpastian menggunakan konsep peluang (*probability measure*), yang merupakan kasus khusus dari ukuran kekaburan secara umum, seperti diperlihatkan pada Gambar 3.10. Oleh karena itu model FHMM yang dikembangkan oleh Magdi dan Gader disebutnya sebagai Generalisasi HMM. Model FHMM ini diimplementasikan untuk melakukan pengenalan terhadap tulisan tangan dan memberikan peningkatan akurasi dari 94.3% menjadi 95.6%, [MG00]^b. Sementara itu Hosseyndoost dan Teshnehlab, 2005, mengembangkan model fuzzy HMM yang berbeda dengan model FHMM sebelumnya, dan dipergunakan untuk klasifikasi fonem dan *phonetic transcription*

dengan persentase kesalahan berkisar dari 30 hingga 39%, [HT05]. Model fuzzy HMM yang dikembangkan pada [HT05] melakukan modifikasi pada peluang distribusi peubah emitten, yaitu dengan memberikan semacam pembobot fuzzy.



Gambar 3.10. Hubungan antara Berbagai Ukuran Ketidakpastian

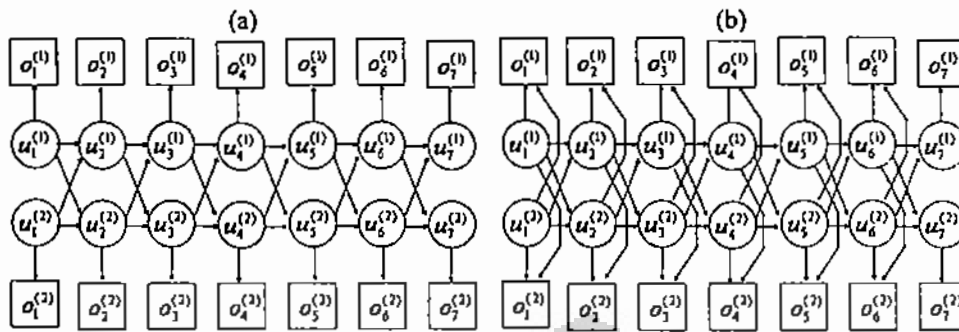
Pada situasi barisan output yang dihasilkan adalah berpasangan, maka model HMM dikembangkan menjadi *Coupled HMM (C-HMM)* atau bisa juga *Mixed Memory HMM (MM-HMM)*. Dalam hal ini barisan $O^{(1)}$ dan $O^{(2)}$, yaitu :

$$O^{(1)} = (o_0^{(1)}, o_1^{(1)}, o_2^{(2)}, \dots, o_{T-1}^{(1)}) \text{ dan } O^{(2)} = (o_0^{(2)}, o_1^{(2)}, o_2^{(2)}, \dots, o_{T-1}^{(2)})$$

dipandang sebagai emitten dari barisan hidden state :

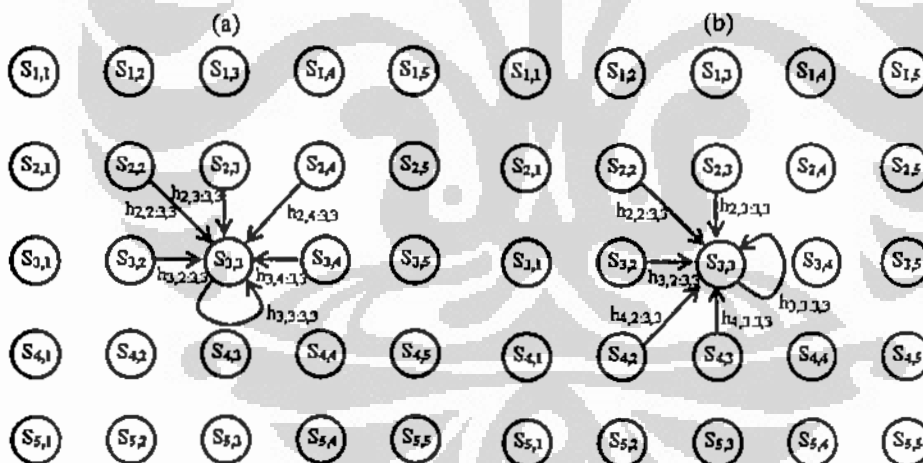
$$U^{(1)} = (u_0^{(1)}, u_1^{(1)}, u_2^{(2)}, \dots, u_{T-1}^{(1)}) \text{ dan } U^{(2)} = (u_0^{(2)}, u_1^{(2)}, u_2^{(2)}, \dots, u_{T-1}^{(2)})$$

Dua pendekatan untuk memodelkan hubungan antara emitten dan hidden state di atas adalah menggunakan C-HMM dan MM-HMM. Gambar 3.11. memberikan ilustrasikan lattice dari model tersebut untuk sekuen dengan panjang tujuh, [PLL04]. Untuk kasus yang sama, Pan dan Liang 2004, [PLL04], mengembangkan model HMM yang disebut sebagai Fused HMM yang pada intinya adalah mengformulasikan peluang observasi gabungan dengan konsep jarak antar distribusi yang diukur dengan entropi relatif menggunakan *Kullback-Leibler divergence*.



Gambar 3.11. Lattice untuk Variasi Model HMM dengan Panjang Tujuh, (a) C-HMM, (b) MM-HMM

Model-model HMM tersebut mengacu pada kasus yang memandang bentuk sekuen pada satu arah (horizontal). Pada kasus citra, sekuen data akan lebih baik dipandang pada dua arah (horizontal dan vertikal). Untuk mengakomodasi dua cara pandang ini, maka dikembangkan 2D HMM. **Gambar 3.12.** menyajikan transisi vertikal dan horizontal pada 2D HMM dengan state 5×5 , [OA03].



Gambar 3.12. Contoh Transisi Vertikal (a) dan Horizontal (b) ke State $S_{3,3}$ ($h_{i,j;x,y}$ adalah peluang transisi dari state $S_{i,j}$ ke state $S_{x,y}$)

Keseluruhan model yang sudah dibahas di atas lebih didasarkan pada dua jenis peubah, yaitu *hidden state* dan *observable state*. Selain itu dikenal model probabilistik lain yang lebih menekankan pada hubungan kausalitas antar peubah yang dikenal dengan *Bayesian Network*, BN. BN merupakan suatu *Directed Acyclic Graph* dengan setiap node mewakili peubah dalam sistem, serta *link(A,B)*

menyatakan adanya pengaruh langsung dari peubah A terhadap peubah B, serta pada setiap node B mempunyai distribusi peluang bersyarat, $P(B|\text{parent}(B))$. Jika model BN ini dikembangkan mengikuti indeks waktu, maka dikenal dengan nama *Dynamic Bayesian Network*, DBN.

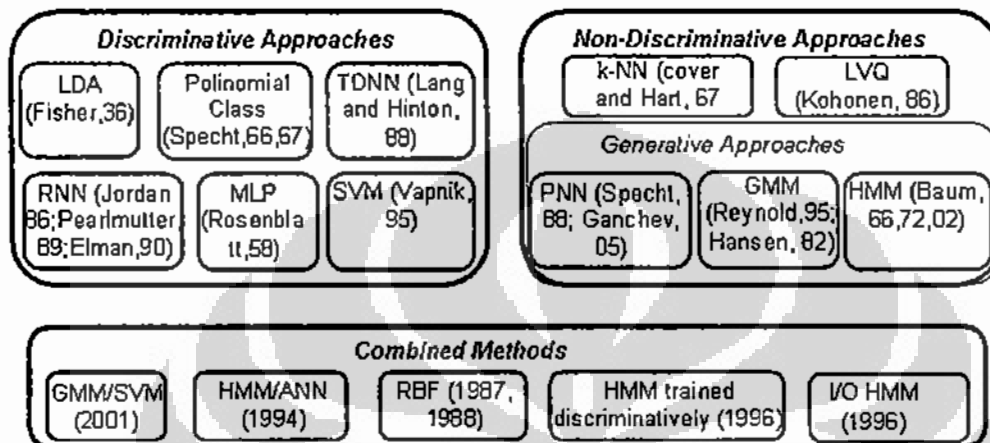
3.4. Perkembangan Penelitian Bidang Identifikasi Pembicara

Seperti sudah disebutkan di atas, bahwa sistem identifikasi pembicara terdiri dari dua bagian besar, yaitu bagian ekstraksi ciri dan bagian *classifier*. Model ekstraksi ciri akan mengkonversi sinyal input ke suatu nilai yang mencirikaninya. Ada beberapa model ekstraksi ciri terhadap sinyal suara, namun yang biasa dipergunakan adalah *Mel Frequency Cepstrum Coefficients* (MFCC). Hal ini dikarenakan MFCC mengakomodasi tingkat kritis telinga manusia terhadap frekuensi suara yang didengar. Dalam hal ini terhadap spektrum sinyal dalam domain frekuensi dilakukan proses *wrapping* menggunakan filter linear pada frekuensi rendah dan filter logaritmik pada frekuensi tinggi, [CH03].

Dari studi literatur yang telah dilakukan, sangat jarang penelitian mengenai pemrosesan sinyal suara difokuskan pada ekstraksi ciri. Hampir semua penelitian yang ada untuk mengatasi permasalahan pada pemrosesan sinyal dengan tujuan akhir menghasilkan sistem yang ideal dengan akurasi tinggi didekati dengan mengembangkan model *classifier*. Secara umum metode *classifier* yang telah dikembangkan terbagi menjadi tiga bagian, seperti yang disajikan pada Gambar 3.13., [Gan05].

Sedangkan menurut [Rey02], ada empat metode *classifier*, yaitu : *template matching*, *nearest neighbor*, *neural network* dan *hidden markov model* (HMM). Pada *template matching* sistem mempunyai cetakan untuk setiap kata/pembicara, sehingga lebih sesuai untuk frase yang *fixed* dan kurang sesuai untuk pengembangan lanjut model pengenalan suara. Pada *nearest neighbor*, sistem harus mempunyai memori yang besar untuk menyimpan data training. Sedangkan pada model *neural network* kurang dapat merepresentasikan bagaimana sinyal suara secara alamiah diproduksi. Pada *hidden markov model*, sinyal suara dimodelkan secara statistik,

sehingga dapat merepresentasikan secara alamiah bagaimana suara dihasilkan. Oleh karena itu, model ini yang pertama kali dipergunakan pada sistem pengenalan pembicara modern, [Rey02].



Gambar 3.13. Klasifikasi Metode *Classifier* pada SIP

Pada tahun 2002 Nelson dan Ejnarsson, [NE02], melaporkan hasil risetnya mengenai pengenalan suara menggunakan MFCC sebagai sebagai ekstraksi ciri dan HMM sebagai *classifier* untuk mengenali 20 target memberikan hasil pengenalan di atas 95%. Namun ketika diberi noise, 10 dB dan 0 dB, terlihat performance sistem turun menjadi sekitar 70% dan 20-30%. Penggunaan HMM sebagai pengenal pola dijumpai pada beberapa literatur, diantaranya adalah pada [XG06], HMM dipergunakan untuk pengenalan tulisan tangan dan memberikan akurasi yang kompetitif dengan teknik lain dengan akurasi mencapai di 99%. Penggunaan HMM untuk menduga *pen trajectory* pada tanda tangan ditunjukkan pada [NPH05] dengan hasil berkisar 90 hingga 100%. Valsan et. al. 2002, [VGS02], menggunakan teknik HMM serta variasinya untuk koversi suara menjadi teks berbahasa Romawi. Pada risetnya, Valsan et. al. membandingkan HMM konvensional, HMM-MLP (HMM-*multilayer perceptron*), HMM-MLP-TD (*discriminative training*), dan Fuzzy HMM sebagai *classifier* pada *speech recognition*. Terlihat bahwa fuzzy HMM yang dikembangkan menggunakan *similarity measure* yang menggantikan *probability measure* pada HMM konvensional mampu menurunkan kesalahan pengenalan sekitar

3% dibanding sistem lainnya. Hasil lain juga ditunjukkan pada [MG00]^b yang menggunakan FHMM untuk pengenalan tulisan tangan dengan 100 kosakata memberikan peningkatan akurasi hingga 2% dibanding HMM konvensional.

Integrasi teknik *support vector machine* (SVM) dengan HMM ditunjukkan pada [GHP04] untuk konversi suara ke teks dengan hasil yang relatif lebih baik dibanding sistem lain seperti RBF, *K-nearest Neighbor*, dan SVM dengan kernel polinomial. Juga disebutkan bahwa SVM/HMM merupakan teknik yang lebih prospektif untuk dikembangkan. Dalam hal ini peran SVM bukan pada ekstraksi ciri, namun untuk melakukan skoring terhadap N sekuen terbaik yang dihasilkan dari HMM.

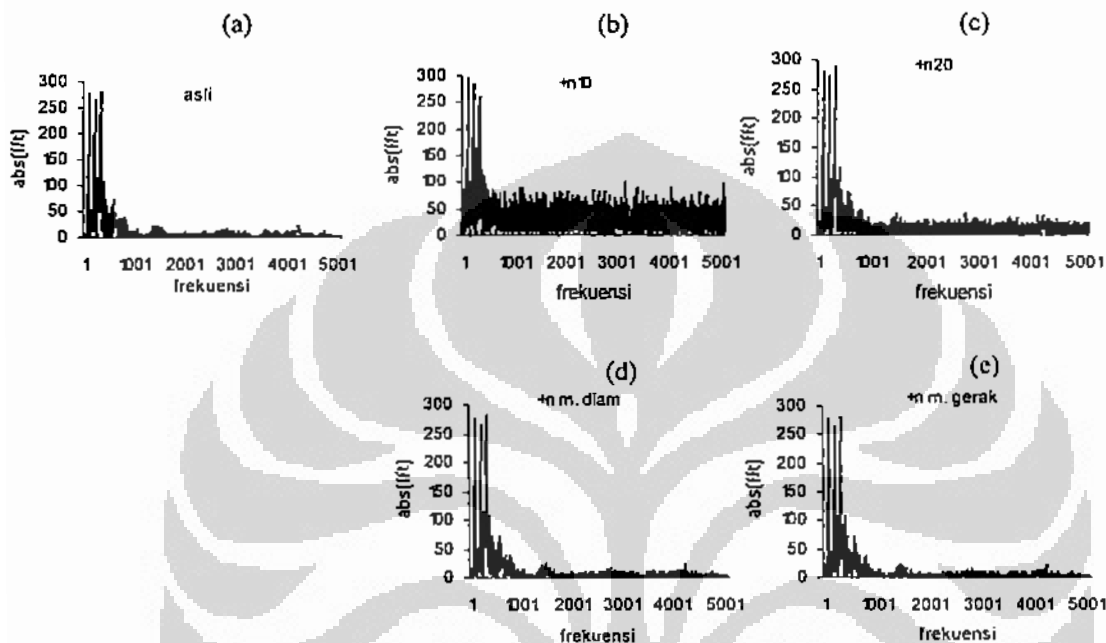
Ganchev pada tahun 2005 dalam disertasinya mengenai pengenalan suara menggunakan *probabilistic neural network* (PNN) sebagai *classifier* dan MFCC serta berbagai variasinya dan transformasi wavelet sebagai ekstraksi ciri memberikan hasil pengenalan sekitar 85% dari 850 target yang dikenali. Setelah melalui proses *denoizing*, akurasi sistem naik menjadi di atas 90 %, dan bahkan 97% untuk *noise* 20 dB. *Noise* yang dipergunakan pada penelitian Ganchev adalah *noise* pabrik dan *noise* pesawat. **Gambar 3.14.** menyajikan perbandingan sinyal *Gaussian noise* dengan *noise* karena suara mesin mobil yang berhenti maupun yang bergerak.



Gambar 3.14. Perbandingan *Gaussian Noise* dengan *Noise* Mesin Mobil

Perbandingan pengaruh ke empat *noise* tersebut terhadap spektrum sinyal asli disajikan pada **Gambar 3.15.** Dari kedua gambar tersebut terlihat bahwa *Gaussian noise* mempunyai pengaruh yang lebih kuat dibanding dengan *noise* karena mobil

berjalan. Hal lainnya adalah bahwa model *classifier* yang dipergunakan oleh Ganchev adalah PNN, maka sistem harus menyimpan seluruh data training. Hal ini akan menjadi masalah pada penyimpanan. Selain itu juga sistem kurang fleksibel untuk pengembangan ke permasalahan pengenalan suara yang bersifat kontinu dengan panjang sinyal suara tidak tetap.



Gambar 3.15. Perbandingan Pengaruh Jenis *Noise* Terhadap Spektrum Sinyal; (a) Spektrum Sinyal Asli; (b), (c), (d), dan (e) adalah Spektrum Sinyal yang sudah Ditambah *Noise*

Pada tahun 1998, Fanani dan Benyamin Kusumoputro mengembangkan model identifikasi pembicara dengan ekstraksi ciri menggunakan *higher order statistic* (HOS). Dengan menggunakan MLP sebagai *classifier*, model ini mampu melakukan pengenalan dengan akurasi 91.25 %, 82.13 % dan 67.81 masing-masing untuk *noise* 20 dB, 10 dB dan 0 dB. Sedangkan Hidayat dan Benyamin Kusumoputro, [HB00], dengan data trispektrum mampu menaikkan akurasi menjadi 79 % untuk *noise* 0 dB. Sedangkan untuk *noise* 20 dB dan 10 dB, akurasi sistem adalah 84.9 % dan 82.8%. Triyanto dan Benyamin Kusumoputro, [TB00], menggunakan bispektrum (HOS orde 3) sebagai pengekraksi ciri dan dan MLP sebagai *classifier* menghasilkan sistem dengan tingkat pengenalan 84.9%, 82.8% dan 79.0 untuk SNR

20 dB, 10 dB, and 0 dB. Sedangkan kalau PNN sebagai *classifier*, maka tingkat pengenalan menjadi 99.2%, 96.8% and 88.4 % untuk SNR 20 dB, 10 dB, and 0 dB. Meskipun hasil pengenalan dengan PNN cukup baik, namun pada metode ini semua data training harus disimpan. Hal ini akan menyebabkan sistem memerlukan *memory* yang besar, sehingga kurang sesuai untuk aplikasi-aplikasi yang bersifat *real time*. Selain itu, meskipun pembacaan data dilakukan per frame, namun input sistem dari metode yang dikembangkan berupa *feature vector* untuk satu suara penuh yang dihitung sebagai rata-rata seluruh frame. Hal ini menyebabkan metode kurang fleksibel terhadap panjang suara dan akan membatasi untuk pengembangan lanjutnya, misal untuk konversi suara ke teks. Kelemahan lain dari model PNN adalah bahwa variasi antar waktu dalam satu suara menjadi tidak terakomodasi.

3.5. Pertanyaan Penelitian

Dari studi literatur yang telah dilakukan terlihat bahwa hampir sebagian besar riset terbaru mengenai pemrosesan sinyal difokuskan pada bagian subsistem pengenalan pola (*classifier*), dengan memodifikasi model peluang yang berbasis proses Markov. Beberapa variasi model Markov yang telah berkembang sebagai pengenalan pola adalah seperti disajikan pada Gambar 3.8. Tiga riset terakhir mengenai penanganan *Gaussian noise*, dan difokuskan pada bagian ekstraksi ciri menggunakan bispektrum sebagai penciri sinyal. Namun, teknik pengenalan pola yang digunakan tidak bersifat temporal, sehingga teknik yang dikembangkan menjadi kurang fleksibel untuk aplikasi pemrosesan sinyal yang lebih luas. Oleh karena itu, pertanyaan utama penelitian ini adalah :

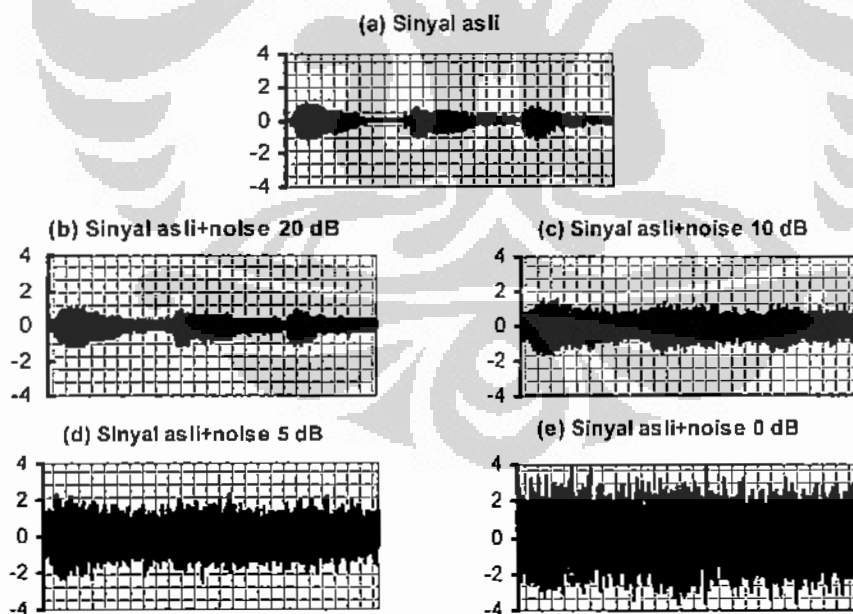
”Bagaimana teknik integrasi bispektrum dengan HMM dilakukan, sehingga dihasilkan sistem yang lebih *robust* terhadap *noise*?”

Permasalahan pada integrasi adalah bispektrum merupakan besaran dua dimensi, sehingga beberapa pertanyaan akan muncul pada proses integrasinya, yaitu :

1. Bagaimana memperluas teknik konvensional ekstraksi ciri MFCC dari satu dimensi menjadi dua dimensi?

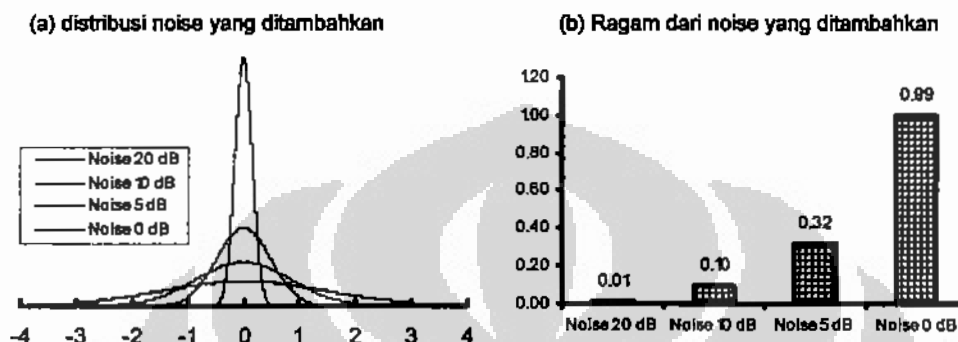
2. Kalau reduksi bispektrum dilakukan dengan kuantisasi, seperti yang dilakukan pada [FB98], [HB99], dan [TB00], statistik apa yang sesuai untuk mewakili nilai bispektrum?
3. Apakah ada kekhasan atau keunggulan metode pada pemrosesan sinyal terkait dengan kualitas sinyal?
4. Parameter apa yang dapat dipergunakan untuk membedakan sinyal sesuai kualitasnya?
5. Bagaimana memanfaatkan parameter (point 4) dan karakteristik metode untuk membangun sistem yang efektif?

Untuk menjawab pertanyaan tersebut, maka serangkaian percobaan dilakukan untuk memberikan bukti empiris, dan disajikan pada Bab 4, 5 dan 6. Data yang dipergunakan adalah seperti yang telah dideskripsikan pada Bab 1, yaitu diarahkan untuk mendapatkan teknik yang lebih *robust* terhadap *Gaussian noise*. **Gambar 3.16.** menyajikan perbandingan sinyal suara asli dengan yang sudah dikontaminasi oleh *noise* yang berdistribusi *Gaussian* dengan level 20 dB, 10 dB, 5 dB, dan 0 dB



Gambar 3.16. Perbandingan Sinyal Asli dengan yang Diberi *Gaussian Noise*

Noise yang diberikan bersifat aditif terhadap sinyal asli, dan sebenarnya yang dilakukan adalah menambahkan suatu nilai tertentu yang berdistribusi Normal (*Gaussian*) dengan rata-rata nol dan ragam (s^2) sebesar 0,1005, 0,3169, 0,56223, dan 0,9974 masing-masing untuk *noise* 20 dB, 10 dB, 5 dB dan 0 dB. Bentuk distribusi *noise* tersebut adalah seperti Gambar 3.17.



Gambar 3.17. Distribusi *Noise* yang Ditambahkan

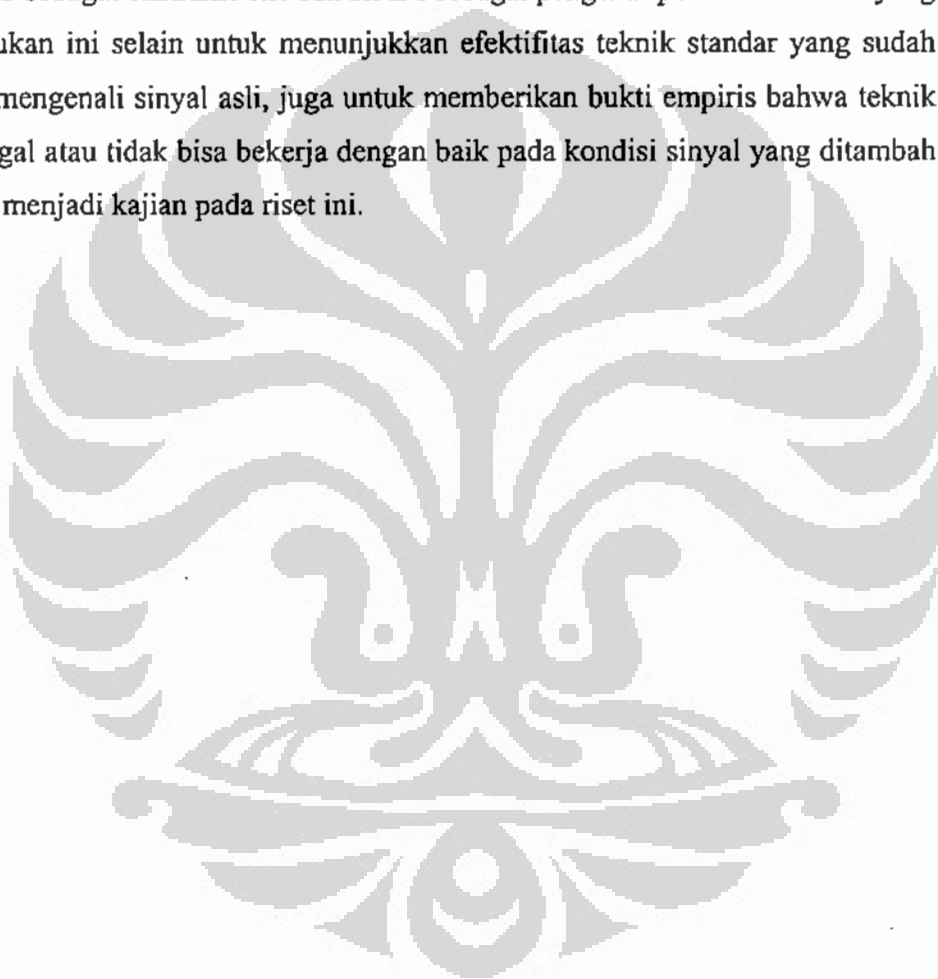
3.6. Beberapa Temuan dan Tahapan Selanjutnya

Dari studi literatur yang dilakukan memperlihatkan beberapa fakta berikut :

- Distorsi sinyal karena *Gaussian noise* lebih berat dibanding dengan *noise* karena mesin mobil, baik saat berhenti maupun jalan. Dalam hal ini *noise* karena mesin mobil dan mobil bergerak setara dengan *Gaussian noise* sebesar 20 dB dalam hal amplitudo. Sedangkan dari aspek frekuensi, pengaruh *Gaussian noise* jauh lebih besar.
- Sebagian besar studi yang ada, baik untuk mengatasi pengaruh *noise* maupun untuk meningkatkan kinerja sistem, pendekatan yang dilakukan adalah dari aspek teknik pengenalan pola. Hanya beberapa riset yang mengfokuskan pada teknik ekstraksi ciri, dengan pengenalan pola menggunakan jaringan syaraf tiruan.
- Teknik pengenalan pola yang berkembang dan menjadi standar pengembangan sistem adalah HMM. Riset-riset yang ada melakukan modifikasi dan pengembangan HMM untuk meningkatkan kinerjanya.

Sedangkan untuk ekstraksi ciri menggunakan metode berbasis power spektrum, dengan teknik terbarunya dan memberikan hasil terbaik adalah *mel-frequency cepstrum coefficients* (MFCC)

Setelah memahami literatur yang ada, maka pada tahap berikutnya akan dilakukan percobaan identifikasi pembicara menggunakan teknik yang sudah ada, yaitu MFCC sebagai ekstraksi ciri dan HMM sebagai pengenalan pola. Percobaan yang akan dilakukan ini selain untuk menunjukkan efektifitas teknik standar yang sudah ada untuk mengenali sinyal asli, juga untuk memberikan bukti empiris bahwa teknik tersebut gagal atau tidak bisa bekerja dengan baik pada kondisi sinyal yang ditambah *noise* yang menjadi kajian pada riset ini.



BAB 4

SISTEM IDENTIFIKASI PEMBICARA BERBASIS POWER SPEKTRUM DALAM DOMAIN FREKUENSI

Telah ditunjukkan pada Bab 3 bahwa sebagian besar riset mengenai pemrosesan suara difokuskan pada sub sistem pengenalan atau *classifier*, dengan HMM standar sebagai *baseline*-nya. Yang dilakukan pada studi-studi tersebut adalah mengkaji model HMM standar dan memberikan usulan teknik baru pengenalan yang merupakan modifikasi atau pengembangan HMM, seperti HMM/NN, IO-HMM, MM-HMM, HMM/SVM, HMM-TD, Coupled HMM, Fuzzy HMM, hingga 2D-HMM. Sedangkan dari aspek ekstraksi ciri relatif tidak mengalami perubahan, yaitu didasarkan pada power spektrum, dengan teknik terbarunya adalah *mel-frequency cepstrum coefficients*, MFCC, seperti dijelaskan pada Bab 2.

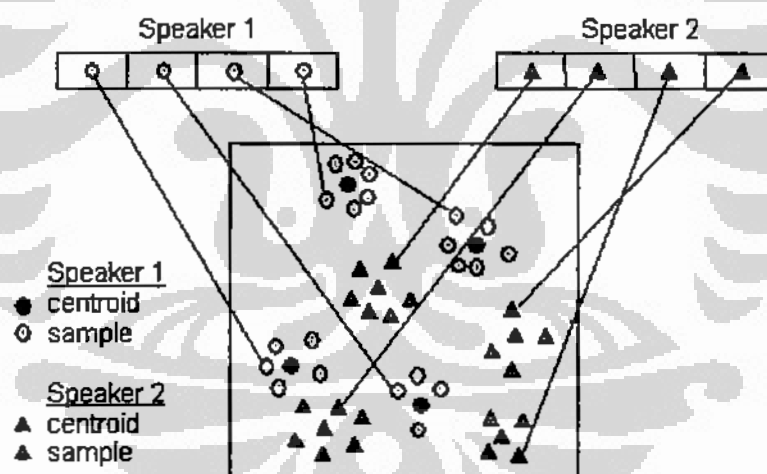
Oleh karena penelitian yang dilakukan pada disertasi ini difokuskan pada sub sistem ekstraksi ciri, dan sub sistem pengenalan hanya sebagai verifikasi hasil, maka dipilih HMM standar sebagai pengenalan pola. Hal ini dengan pertimbangan peningkatan akurasi dari pengembangan HMM tidaklah terlalu besar, misalnya pada Fuzzy HMM, akurasi meningkat sekitar 3% dibanding HMM standar, [MG00]^a dan [MG00]^b.

Bab 4 ini ditujukan untuk memberikan bukti empiris dari keterbatasan atau kelemahan teknik ekstraksi ciri konvensional yang berbasis power spektrum dalam melakukan pengenalan untuk situasi ber-*noise*. Oleh karena itu, pada bab ini disajikan bahasan mengenai percobaan identifikasi pembicara, yang pertama dengan menggunakan template berbasis *codebook* dan yang kedua dengan HMM sebagai pengenalan pola. Untuk meningkatkan kinerja sistem, juga dilakukan percobaan menggunakan *adaptive noise canceling* untuk menangani *noise*. Sedangkan ekstraksi ciri yang digunakan adalah MFCC dengan power spektrum sebagai masukannya yang secara detail telah disajikan pada Bab 2.

4.1. Sistem Identifikasi Pembicara Menggunakan *Codebook*

4.1.1. Kerangka Pikir

Prinsip dasar dari metode ini adalah pada setiap pembicara dibuat sebuah *codebook* yang terdiri dari beberapa *codeword*. *Codebook* adalah kumpulan titik (vektor) yang mewakili distribusi suara dari seorang pembicara tertentu dalam ruang suara. Setiap titik dari *codebook* dikenal sebagai *codeword*. Oleh karena itu, pada setiap pembicara dibuat sebuah *codebook* yang merepresentasikan ciri suara dari pembicara tersebut. Prinsipnya proses pengenalan yang dilakukan adalah setiap suara yang masuk dihitung jarak suara tersebut ke *codebook* setiap pembicara. Jarak sinyal suara masuk dengan *codebook* seorang pembicara dihitung sebagai jumlah jarak setiap frame yang dibaca ke *codeword* terdekat pada *codebook* tersebut. Dari sini akhirnya sinyal masukan diberi label pembicara sesuai jarak *codebook* terkecil. Gambar 4.1. memberikan ilustrasikan tebaran data suara di sekitar *codebook*-nya, [CH03].



Gambar 4.1. Tebaran Suara di Sekitar *Codebook* dari Dua Pembicara

Sesuai dengan deskripsi di atas, maka ada dua tahap yang dilakukan pada percobaan ini, yaitu tahap pembuatan *codebook* dan tahap pengenalan. Tahap pembuatan *codebook* untuk seorang pembicara adalah sebagai berikut :

1. Pada setiap pengucapan (ada n pengucapan sebagai data training),

dilakukan ekstraksi ciri menggunakan teknik MFCC pada setiap frame dengan panjang dan overlap tertentu (pada penelitian ini dipilih panjang frame 256 digit dengan overlap 156 digit).

2. Semua frame dari n pengucapan digabungkan menjadi satu set dan lakukan *unsupervised clustering* menggunakan teknik *Self Organizing Feature Map* (SOFM) dengan memilih banyaknya kluster sesuai dengan jumlah *codeword* yang diinginkan. (Pada penelitian ini dicobakan mulai dari 9 hingga 32).

Sedangkan alur proses untuk tahap pengenalan adalah :

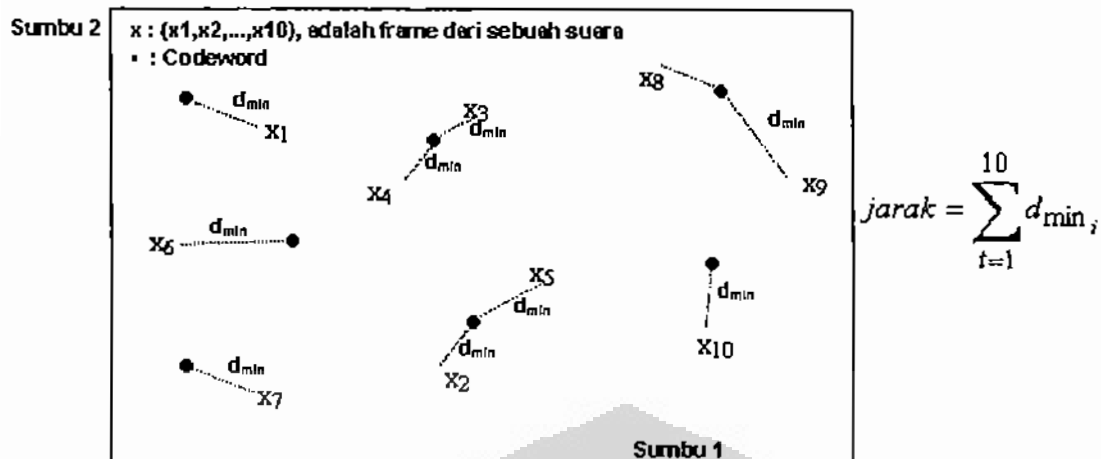
1. Untuk setiap ucapan baru yang masuk ke sistem dibaca frame demi frame, (misalkan jumlah frame yang diperoleh adalah T), dan dilakukan ekstraksi ciri menggunakan teknik MFCC
2. Hitung jarak sinyal input ucapan ini ke *codebook* setiap pembicara yang ada dalam sistem.
3. Keputusan : menetapkan label pada input suara sesuai dengan pembicara dengan jarak *codebook* terkecil.

Jarak input ucapan dengan *codebook* dirumuskan sebagai berikut :

1. Untuk setiap frame dari input ucapan yang masuk, dihitung jarak ke setiap *codeword* dan dipilih *codeword* dengan jarak minimum.
2. Jarak antara input ucapan dengan *codebook* adalah jumlah dari jarak minimum tersebut :

$$\text{jarak}(\text{input}, \text{codebook}) = \sum_{t=1}^T \min_{\forall \text{codeword}_k} [d(\text{frame}_t, \text{codeword}_k)] \quad (4.1.)$$

Gambar 4.2. menyajikan ilustrasi perhitungan jarak antara sinyal dengan panjang 10 frame terhadap *codebook* yang terdiri dari 7 *codeword* berdimensi dua.



Gambar 4.2. Ilustrasi Penghitungan Jarak Input Suara (terdiri 10 frame) dengan Codebook (7 codeword yang berdimensi dua)

4.1.2. Self Organizing Feature Map (SOFM)

Prinsip dasar dari SOFM adalah membagi ruang input menjadi beberapa kluster, dengan setiap kluster mempunyai sebuah vektor perwakilan. Jaringan syaraf tiruan SOFM ini terdiri dari dua lapis neuron, yaitu lapis input dan lapis output, dengan proses pembelajaran tanpa pengarahannya yang dalam hal ini sebanyak k unit pemroses $m_j(t)$, dengan $m_j \in \mathbb{R}^n$, $j=1, 2, 3, \dots, k$. Unit pemroses ini pada waktu t bersifat sensitif terhadap sebuah item tertentu di ruang input.

Pada tahap awal algoritme pembelajaran SOFM dilakukan inisialisasi pada setiap unit pemroses, $m_j(0)$. Dari sini, maka untuk setiap vektor masukan x , suatu unit pemenang m_c akan ditemukan dengan berdasar jarak minimum antara vektor masukan x dengan m_j , $\min\{\|x - m_j\|\} \forall j$ untuk $t=1, 2, 3, \dots$. Setelah itu, unit pemenang dan tetangganya akan diperbaharui dengan aturan :

$$m_j(t+1) = \begin{cases} m_j(t) + \alpha(t)[x - m_j(t)] & \text{jika } j \in N_c(t) \\ m_j(t) & \text{jika } j \notin N_c(t) \end{cases} \quad (4.2.)$$

Dalam hal ini $\alpha(t)$ adalah laju pembelajaran yang bernilai antara 0 dengan 1, dan merupakan fungsi menurun secara geometris terhadap t , [Fau94]. $N_c(t)$ adalah fungsi ketetanggaan yang bisa berbentuk linear, bujursangkar, hexagonal, atau bahkan *Gaussian*, dan senantiasa diperbaharui untuk setiap t , dan juga menurun seiring

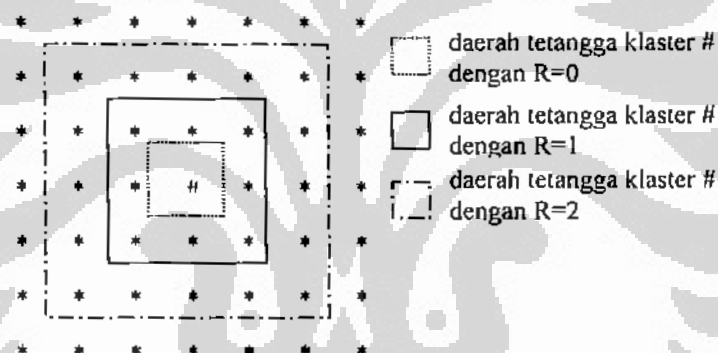
dengan waktu, t . Dalam penelitian ini akan digunakan laju pembelajaran dan fungsi ketetanggaan bujursangkar dengan rumus berikut [TB00] :

$$\alpha(t) = \alpha(0)[1 - t/T] \quad (4.3.)$$

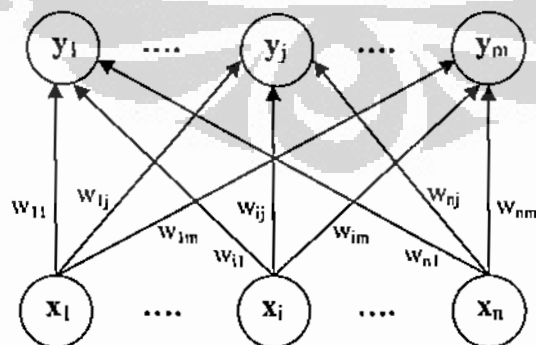
$$N_c(t) = [N_c(0)(1 - t/T)] \quad (4.4.)$$

$t=1, 2, 3, \dots, T$, dan T adalah jumlah iterasi pembelajaran yang akan dilakukan.

Gambar 4.3. memberikan ilustrasi struktur neuron output dengan topologi bujursangkar dengan jumlah perwakilan 7×7 , serta daerah ketetanggaan dengan radius $R=2, 1, 0$. Selama proses pembelajaran, m_j akan cenderung ditempatkan diruang R^n sedemikian hingga akan mendekati fungsi kepekatan peluang (*probability density function, PDF*) dari x . **Gambar 4.4.** menyajikan arsitektur jaringan untuk ruang input berdimensi n dan jumlah kluster m .



Gambar 4.3. Struktur Neuron Output Topologi Bujursangkar 7×7



Gambar 4.4. Arsitektur SOFM (Dimensi Input n dan Jumlah Kluster m)

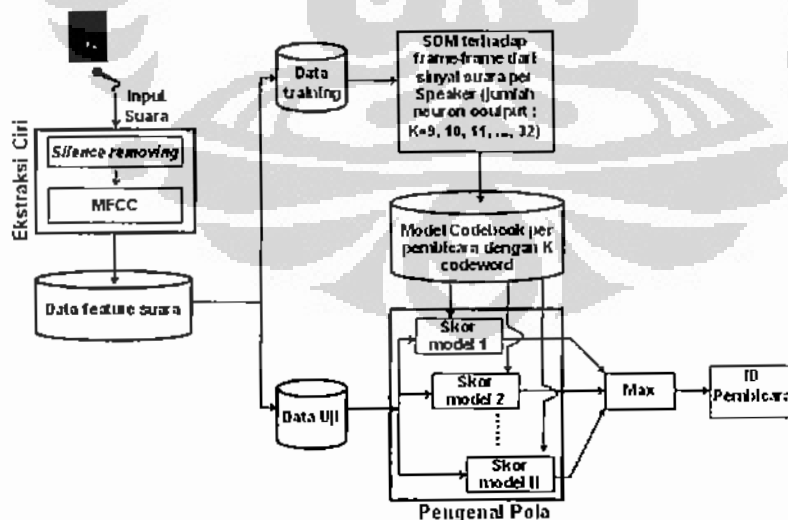
Sedangkan algoritme kuantisasi SOFM adalah seperti pada **Algoritme 2**, [Fau94].

Algoritme 2 : Kuatisasi Vektor Metode SOFM

- Step 0. Inisialisasi bobot w_{ij}
 Set parameter ketetanggaan sesuai topologi yang dipilih
 Set parameter pembelajaran
- Step 1. Selama kondisi pemberhentian belum dipenuhi, lakukan 2-8
- Step 2. untuk setiap vektor input x , lakukan step 3-5
- Step 3. untuk setiap j , hitung :
- $$D(j) = \sum_i (w_{ij} - x_i)^2$$
- Step 4. pilih indeks J , sehingga $D(J)$ minimum
- Step 5. untuk semua unit j dalam lingkup tetangga J , dan untuk semua i , perbaharui bobot w_{ij} dengan rumus :
- $$w_{ij}(\text{new}) = w_{ij}(\text{old}) + \alpha(x_i - w_{ij}(\text{old}))$$
- step 6. Perbaharui laju pembelajaran
 $\alpha(t) = \alpha(0) [1 - t/T]$
- step 7. Perbaharui jari-jari ketetanggaan
 $N_c(t) = [N_c(0)(1 - t/T)]$
- step 8. Uji kondisi pemberhentian

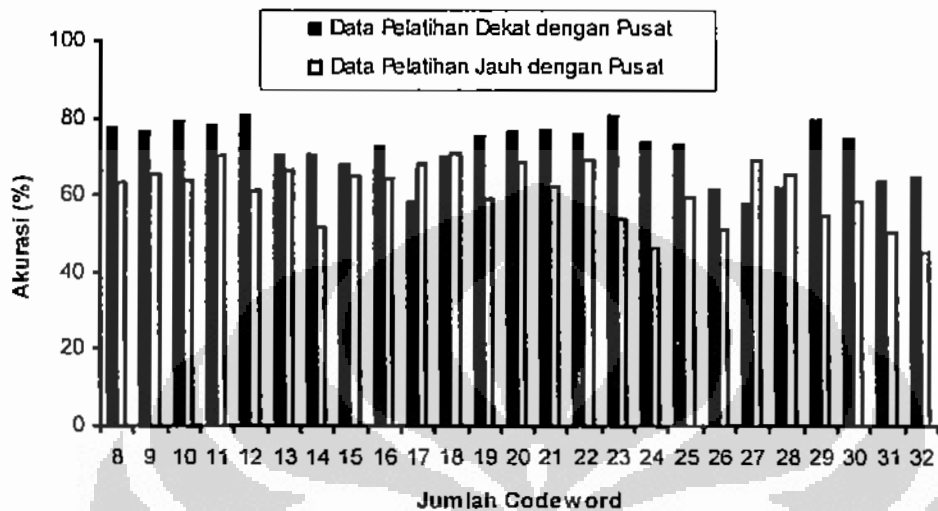
4.1.3. Hasil Percobaan

Percobaan identifikasi pembicara dengan menggunakan *codebook* ini dilaksanakan sesuai **Gambar 4.5**.



Gambar 4.5. Alur Proses Identifikasi Pembicara dengan *Codebook*

Berdasar pendekatan tersebut, maka diperoleh akurasi rata-rata 71% untuk percobaan yang menggunakan data training yang relatif dekat dengan pusat populasi. Sedangkan jika data training berada jauh dari pusat populasi, akurasi adalah 61, seperti pada Gambar 4.6.



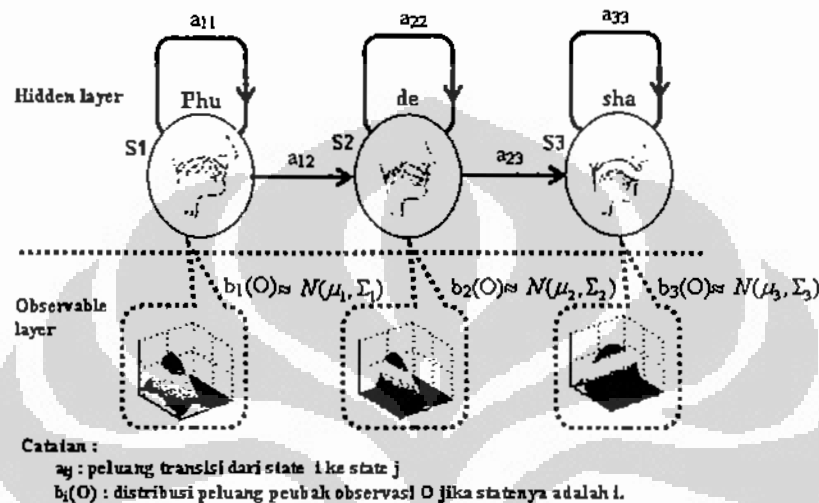
Gambar 4.6. Hasil Akurasi Sistem Berbasis *Codebook* (Data Suara Asli)

Terlihat bahwa metode berbasis *codebook* dapat diterapkan pada sistem identifikasi pembicara, namun memberikan hasil akurasi yang rendah, dengan kisaran 60 hingga 80%. Terlebih lagi kalau data pelatihan yang terambil adalah yang jauh dari pusat distribusi.

4.2. Sistem Identifikasi Pembicara Menggunakan HMM

Sebelumnya telah diperlihatkan hasil percobaan dengan teknik *template matching* berbasis *codebook* (merupakan teknik pada awal mula sistem identifikasi dikembangkan) dengan akurasi sistem hanya mencapai rata-rata sekitar 71% untuk sinyal asli tanpa penambahan *noise*. Pada bagian ini akan disajikan sistem identifikasi pembicara dengan teknik pengenalan pola yang umum dipergunakan, yaitu HMM, dengan ekstraksi cirinya MFCC. Perpaduan kedua teknik tersebut menjadi standar sistem pemrosesan suara sekarang ini. Teknik HMM serta MFCC yang dipergunakan pada percobaan adalah sesuai yang dijabarkan pada Bab 2 sebelumnya.

Pada penelitian ini, identifikasi pembicara didasarkan pada ujaran yang bersifat tetap, yaitu frase "PUDESHA" dan vektor pengamatan pada setiap periode t , O_t , adalah kontinyu yang didekati dengan distribusi *Gaussian*. Oleh karena itu model HMM yang sesuai adalah model left-right HMM dengan distribusi *emitten* bersifat *Gaussian*. Bentuk model ini disajikan pada Gambar 4.7.



Gambar 4.7. Model *Left-Right* HMM dengan Tiga State yang Dipergunakan pada Penelitian

Oleh karena itu, parameter-parameter HMM pada Gambar 4.7. di atas adalah :

$$\lambda = \left(A = \begin{bmatrix} a_{11} & a_{12} & 0 \\ 0 & a_{22} & a_{23} \\ 0 & 0 & a_{33} \end{bmatrix}, B = \begin{bmatrix} N(\mu_1, \Sigma_1) \\ N(\mu_2, \Sigma_2) \\ N(\mu_3, \Sigma_3) \end{bmatrix}, \Pi = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \right)$$

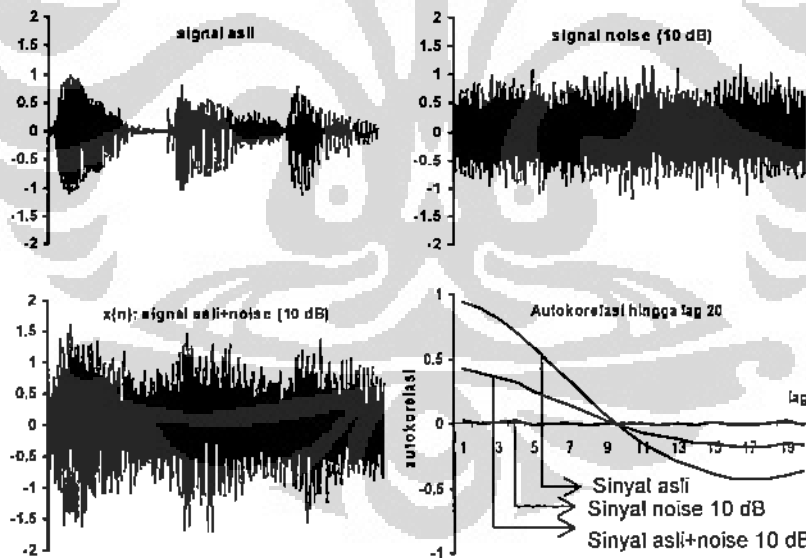
Percobaan identifikasi akan dilakukan baik pada sinyal suara asli tanpa penambahan *noise* juga pada sinyal suara asli dengan penambahan *noise*. Hal ini untuk melihat sejauh mana ketegaran teknik standar yang ada, yaitu MFCC-HMM terhadap distorsi karena *noise* (*Gaussian*). Untuk meningkatkan kinerja sistem, maka pada percobaan juga akan ditambahkan bagian yang disebut *adaptive noise canceling* untuk menangani pengaruh *noise*.

4.2.1. Adaptive Noise Canceling (ANC)

Proses penghapusan *noise* (*noise canceling*, NC) ditujukan untuk mengoreksi sinyal masukan menjadi sinyal keluaran yang menghilangkan (menekan) pengaruh *noise* yang bersifat aditif (menambahkan pada sinyal asli), seperti ditunjukkan persamaan berikut :

$$x(k)=s(k)+n(k) \quad (4.5)$$

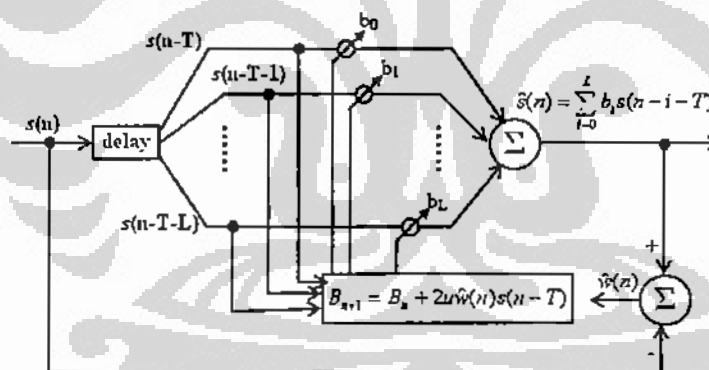
dalam hal ini s adalah sinyal suara dan n adalah *noise* yang ditambahkan pada sinyal suara untuk menghasilkan sinyal yang terkontaminasi, x . Ada dua teknik penghapusan *noise* yang dikenal, yaitu dengan menggunakan teknik kuadrat error terkecil yang bersifat adaptif, dan yang ke dua menggunakan teknik spektral sinyal. Teknik yang berbasis kuadrat error terkecil menggunakan fakta bahwa autokorelasi pada sinyal suara membentuk pola tertentu dan berbeda nyata dengan nol. Sementara itu, untuk sinyal *noise*, nilai autokorelasinya adalah nol, seperti yang ditunjukkan pada Gambar 4.8.



Gambar 4.8. Perbandingan Autokorelasi antara Sinyal Asli, Sinyal *Noise* dan Sinyal Asli Terkontaminasi *Noise*

Sementara itu, teknik yang berbasis spektral sinyal, menggunakan sifat matematika dari formulasi spektral untuk sinyal gabungan sebagai penjumlahan dari spektral setiap komponennya, sehingga teknik ini dikenal dengan *spectral subtraction*, [Bol79].

Dari Gambar 4.8. tersebut dapat dikatakan bahwa nilai autokorelasi dari sinyal yang terkontaminasi *noise* adalah mencerminkan pola hubungan antar simpangan sinyal pada lag tertentu dari sinyal suara (bukan dari sinyal *noise*). Oleh karena itu, dengan asumsi bahwa *noise* aditif tersebut bersifat saling bebas, maka setiap simpangan pada unit waktu t dapat dirumuskan sebagai fungsi dari simpangan-simpangan pada unit waktu sebelumnya. Parameter dari fungsi ini ditentukan dengan menggunakan kriteria rata-rata kuadrat error terkecil (*Least Mean Square Error*, LMSE), dan dilakukan secara *adaptive* dari frame ke frame. Gambar 4.9. menyajikan diagram dari proses *adaptive noise canceling* tersebut, [Wid75]. Dalam hal ini $B_n = (b_0 \ b_1 \ \dots \ b_L)_n$ dan $\hat{s}(n) = b_0 s(n-T) + b_1 s(n-1-T) + \dots + b_L s(n-L-T)$. Pengaruh dari teknik NC di atas secara visual diperlihatkan pada Gambar 4.10.



Gambar 4.9. Blok Diagram *Adaptive Noise Canceling*

Oleh karena *noise* pada percobaan ini dibangkitkan secara bebas dari distribusi *Gaussian* dan diberikan pada sinyal secara aditif, maka teknik NC yang dipergunakan adalah *adaptive noise canceling*. Kalau $s(n)$ adalah sinyal suara asli (diasumsikan sebagai *clean signal*) dan $w(n)$ adalah *noise* yang bersifat aditif, maka dugaan bagi $s(n)$, yaitu $z(n)$, dapat dimodelkan sebagai :

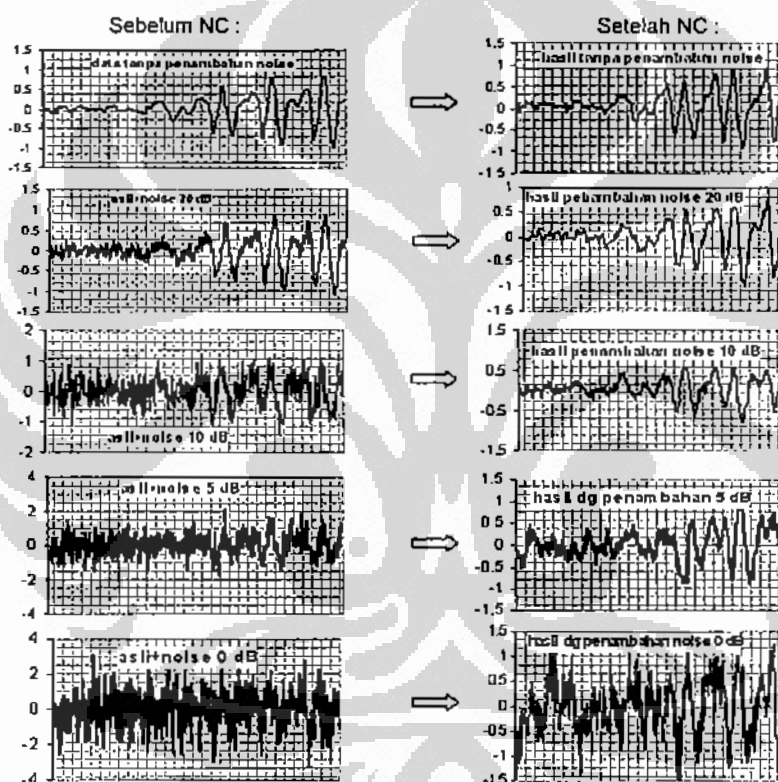
$$z(n) = s(n) + w(n) - \hat{w}(n) \quad (4.6)$$

dengan $\hat{w}(n)$ sebagai dugaan bagi $w(n)$. Untuk mendapatkan dugaan $z(n)$, kriteria yang dipakai adalah meminimumkan rata-rata kuadrat error dengan rumus :

$$E\{(z(n) - s(n))^2\} \quad (4.7)$$

Strategi yang dilakukan adalah dengan meminimumkan rata-rata kuadrat error untuk sinyal *noise* seperti formula berikut, [Wid75] :

$$E\{(w(n) - \hat{w}(n))^2\} \quad (4.8)$$



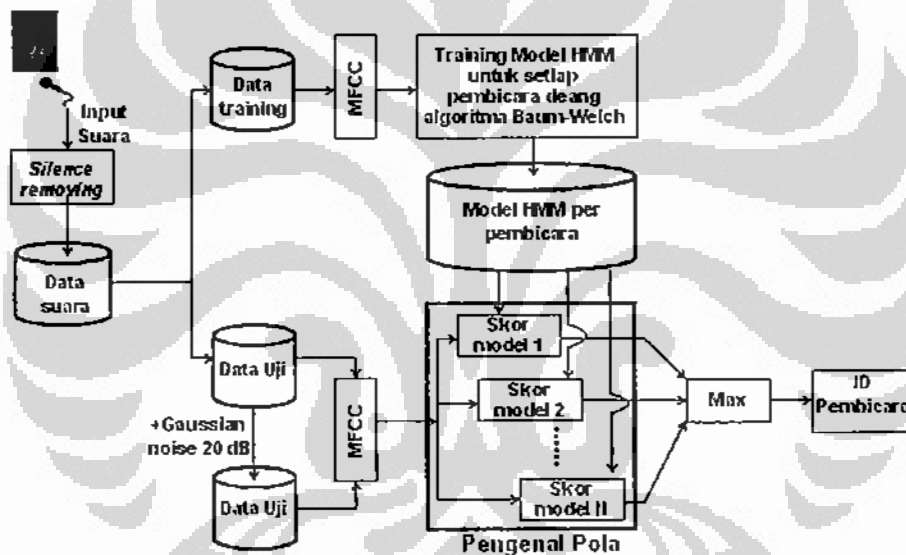
Gambar 4.10. Perbandingan Sinyal antara Sebelum dengan Setelah ANC

4.2.2. Seting Percobaan dan Hasil

Seting Percobaan

Percobaan identifikasi pembicara yang dilakukan mengikuti alur tahapan seperti pada Gambar 4.11. Data yang digunakan berasal dari 10 pembicara masing-

masing 80 pengucapan. Dalam hal ini dicobakan tiga set perbandingan jumlah data pelatihan dengan jumlah data pengujian, yaitu 20:60, 40:40, dan 60:20. Ada tiga tahapan dalam percobaan ini, yaitu tahap ekstraksi ciri, tahap pelatihan model dan tahapan pengenalan. Pada tahap ekstraksi ciri, semua ucapan dari setiap pembicara dihitung nilai cirinya yang dibaca frame demi frame dengan panjang 256 dan overlap antar frame 156 menggunakan teknik MFCC. Berikutnya data yang telah dihitung cirinya dikelompokkan menjadi dua, satu untuk pelatihan dan sisanya untuk pengujian (pengenalan) sesuai proporsi di atas. Selanjutnya adalah tahap pelatihan. Pada tahap ini dilakukan pendugaan parameter HMM dengan algoritme Baum Welch [Rab89] dari set data pelatihan. Setelah model diperoleh, dilanjutkan tahap pengenalan pembicara untuk set data pengujian yang sudah ditentukan.



Gambar 4.11. Alur Proses Pengenalan Pembicara Menggunakan MFCC-HMM

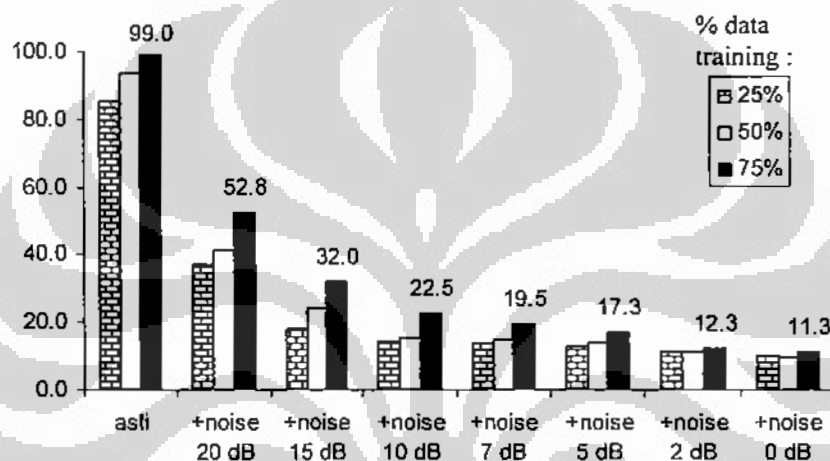
Hasil Percobaan

Hasil pengenalan pembicara dengan metode MFCC+HMM untuk sinyal asli disajikan pada **Tabel 4.1**. Dari tabel tersebut terlihat bahwa teknik yang dipakai berhasil dengan baik mengenali pembicara, yaitu sekitar 99% untuk data asli pada proporsi data training 75%. Sedangkan jika proporsi data training berkurang menjadi 50% dan 25%, maka akurasi turun cukup berarti, yaitu menjadi 93.8% dan 85.5%.

Tabel 4.1. Akurasi (%) dengan Teknik MFCC+HMM pada Berbagai *Noise*

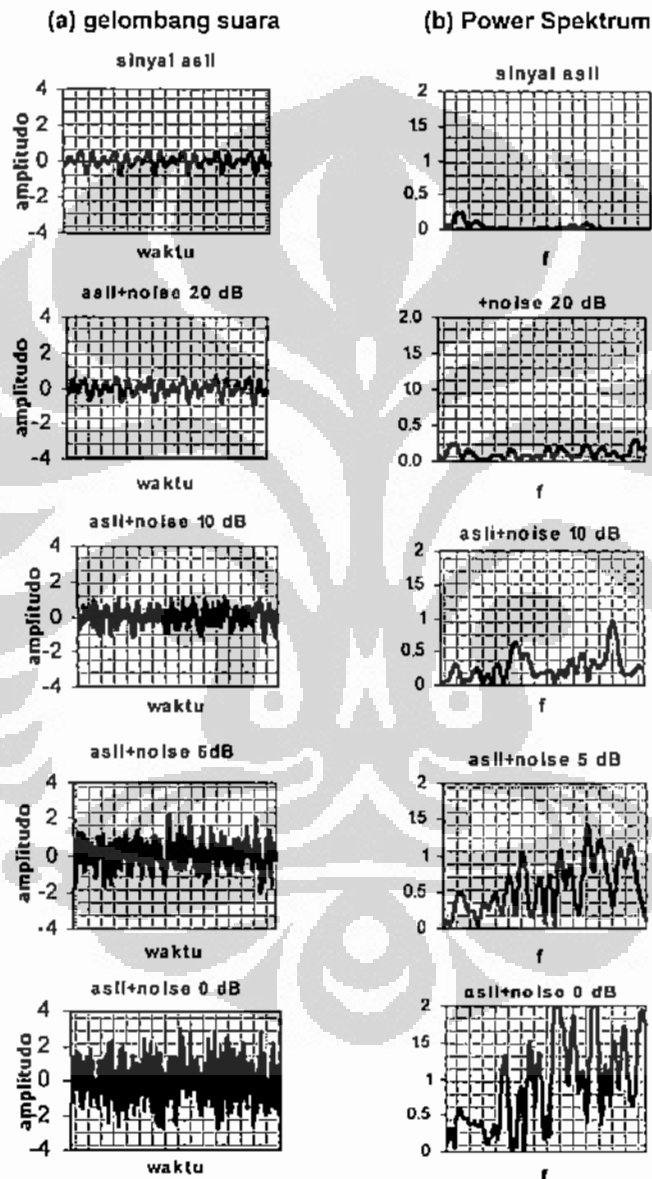
%data training	asli	+noise 20 dB	+noise 15 dB	+noise 10 dB	+noise 7 dB	+noise 5 dB	+noise 2 dB	+noise 0 dB
25%	85.5	37.0	18.0	14.4	13.9	12.7	11.2	10.4
50%	93.8	41.1	24.3	15.4	14.9	13.8	11.2	10.0
75%	99.0	52.8	32.0	22.5	19.5	17.3	12.3	11.3

Hal ini menunjukkan bahwa untuk memberikan hasil yang optimum, maka teknik HMM memerlukan jumlah data pelatihan yang mencukupi. Dengan 60 data pelatihan ini, yang setiap suara mempunyai panjang rata-rata sekitar 50 frame, panjang total barisan yang dipergunakan untuk pelatihan adalah sekitar 3000 frame.

**Gambar 4.12.** Akurasi Sistem (%) Menggunakan Teknik MFCC+HMM untuk Berbagai Data Uji dengan Berbagai *Noise*

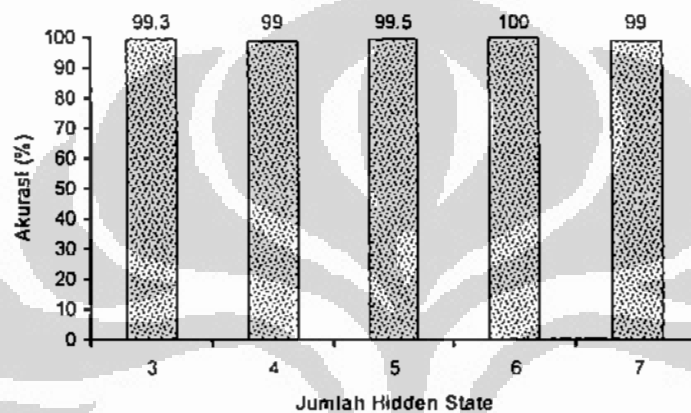
Dari tabel tersebut juga terlihat bahwa dengan bertambahnya *noise*, akurasi turun secara drastis, yaitu menjadi 52% untuk *noise* 20 dB, dan untuk *noise* yang lebih tinggi, akurasi jauh di bawah 50%. Hal tersebut terlihat jelas secara visual seperti yang ditunjukkan pada **Gambar 4.12**. Gagalnya sistem ini disebabkan oleh nilai power spektrum yang sensitif terhadap gangguan *noise*, seperti ditunjukkan pada **Gambar 4.13**. Bagian kiri, (a), menunjukkan dengan jelas bahwa bentuk gelombang sinyal suara mengalami perubahan dengan adanya tambahan *noise* mulai dari terkecil 20 dB hingga terbesar 0 dB. Meskipun, secara alamiah ciri sinyal asli dengan sinyal asli yang sudah ditambah *noise* tidaklah berubah, namun pola power spektrum untuk sinyal yang telah mengalami penambahan *noise* semakin berbeda dengan dengan pola

power spektrum sinyal aslinya. Perbedaan ini sudah mulai terlihat pada noise 20 dB, dan secara nyata mulai terlihat pada *noise* 10 dB. Fakta empiris yang ditunjukkan pada **Gambar 4.13**, mengatakan bahwa power spektrum sebagai penentu ciri sinyal suara sangat sensitif terhadap gangguan *noise*. Dengan demikian, maka nilai *feature* yang diperoleh dari ekstraksi ciri telah mengalami distorsi yang cukup besar dari sinyal aslinya, yang pada akhirnya akan menurunkan akurasi sistem.



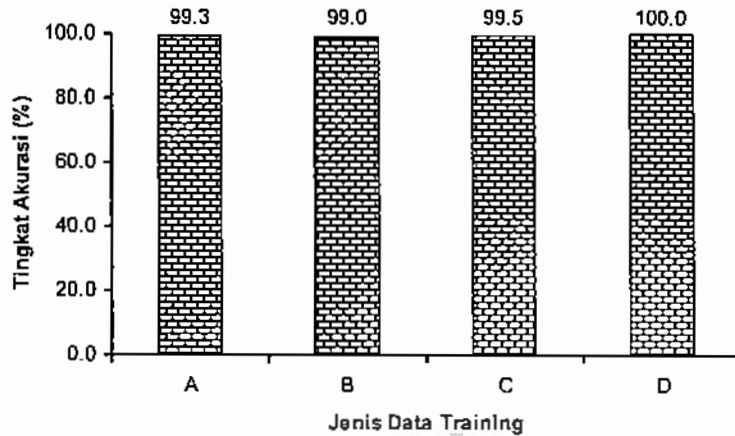
Gambar 4.13. Perbandingan Pola Power Spektrum antara Sinyal Asli dengan Sinyal asli yang Ditambah *Noise*

Sedangkan kalau diperhatikan dari jumlah *hidden state* dari model HMM, terlihat adanya perbedaan kecil tingkat akurasi dari jumlah *hidden state* 3 hingga 7, dengan nilai tertinggi pada jumlah *hidden state* 6, yaitu mencapai 100% (untuk data asli/tanpa penambahan *noise*), seperti ditunjukkan pada **Gambar 4.14**. Meskipun demikian, secara absolut, nilai akurasi dari setiap jumlah *hidden state* tersebut tidaklah terlalu berbeda nyata (semuanya di atas 99%). Hal ini menunjukkan bahwa pemilihan jumlah *hidden state* pada HMM tidaklah memberikan pengaruh yang nyata terhadap hasil akurasi sistem.



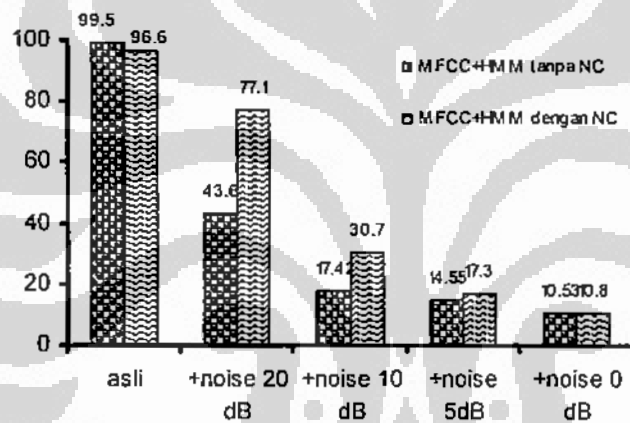
Gambar 4.14. Perbandingan Akurasi pada Berbagai Jumlah Hidden State

Dari aspek pemilihan data training, terlihat fakta bahwa pemilihan secara acak terhadap ucapan sebagai data pelatihan akan memberikan akurasi sistem yang lebih baik, seperti disajikan pada **Gambar 4.15**. Dari gambar tersebut terlihat bahwa sistem yang dikembangkan dengan data training yang dipilih dari suara yang secara relatif berada jauh dari pusat distribusinya akan memberikan akurasi yang rendah, (B). Untuk data training berasal dari suara yang menyebar di sekitar pusat distribusi, (A), juga kurang memberikan hasil yang bagus. Kalau dipilih suara secara sistematis dari yang dekat hingga yang terjauh dari pusat distribusi suara, maka akurasi lebih tinggi dibanding (A) dan (B). Hasil tertinggi adalah kalau data training dipilih secara acak, seperti yang ditunjukkan pada (D), yaitu hampir mencapai 100%. Meskipun demikian, dari nilai absolut akurasi tersebut masih cukup baik untuk semuanya (diatas 98.8%).



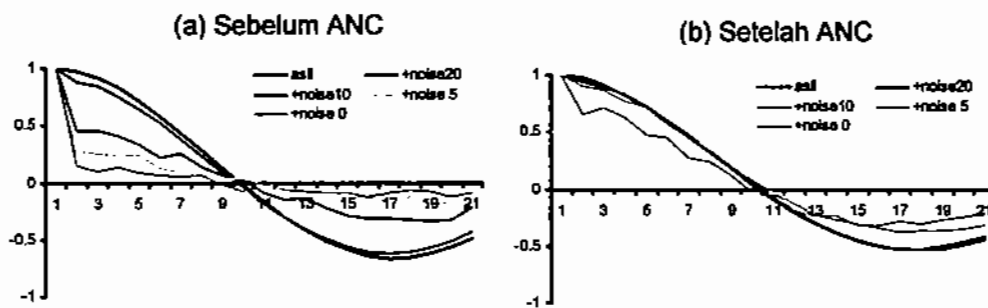
Gambar 4.15. Perbandingan Akurasi dari Empat Jenis Data Training

Setelah melalui proses *adaptive noise canceling*, kinerja sistem meningkat, terutama untuk sinyal dengan penambahan *noise*, seperti ditunjukkan pada **Gambar 4.16**.



Gambar 4.16. Perbandingan Akurasi untuk Sistem dengan dan tanpa NC pada Berbagai *Noise*

Untuk sinyal asli tanpa penambahan *noise*, sistem dengan NC memberikan akurasi 96.6%, sekitar 3% di bawah sistem tanpa melalui NC. Sedangkan untuk sinyal dengan penambahan *noise*, dari gambar di atas terlihat bahwa *adaptive noise canceling* mampu meningkatkan tingkat *robustness* sistem hingga *noise* 20 dB dengan akurasi 77.1%. Untuk *noise* yang lebih besar, sistem gagal bekerja dengan baik. Kemampuan teknik NC dalam meningkatkan *robustness* sistem ini disebabkan teknik tersebut dapat mempertahankan nilai autokorelasi sinyal suara seperti diperlihatkan pada **Gambar 4.17**.



Gambar 4.17. Autokorelasi Sinyal Suara antara Sebelum dan Setelah ANC

4.3. Beberapa Temuan dan Tahapan Selanjutnya

Beberapa fakta yang diperoleh pada bagian ini adalah :

1. Teknik MFCC dan HMM standar dapat diterapkan untuk pengenalan pola suara dengan baik (akurasi di atas 99%) untuk data tanpa penambahan *noise*.
2. Secara visual, perbedaan jumlah *hidden state* pada HMM tidak memberikan pengaruh yang berarti bagi kinerja sistem. Nilai akurasi yang diberikan sekitar 99%, jauh di atas akurasi yang dihasilkan dengan menggunakan *codebook* yang dibentuk menggunakan SOFM yang hanya berkisar antara 60 hingga 80%.
3. Akurasi sistem dipengaruhi oleh jenis data suara yang dipakai sebagai data pelatihan. Pemilihan secara acak data untuk pelatihan akan memberikan hasil pengenalan yang lebih baik dibanding kalau data pelatihan dipilih tidak acak. Namun demikian, perbedaan nilai akurasi yang diberikan secara absolut tidak berarti, semuanya di sekitar nilai 99%.
4. Jumlah data training memberikan perbedaan akurasi yang signifikan, yaitu 99%, 92% dan 89% untuk sinyal tanpa penambahan *noise*, masing-masing untuk jumlah data training 60 suara (75%), 40 suara (50%), dan 20 suara (25%).

5. Akurasi sistem turun menjadi sekitar 50% untuk data suara ber-*noise* 20 dB. Untuk *noise* yang lebih besar (10 dB hingga 0 dB), akurasi turun drastis di bawah 50% hingga mencapai sekitar 10%.
6. *Denoizing* menggunakan *adaptive noise cancelling* memberikan perbaikan hingga noise 20 dB, dengan akurasi mencapai 77.1% dari semula hanya 43.6%. Perbaikan ini dikarenakan NC mampu mempertahankan nilai autokorelasi sinyal suara yang sudah terdistorsi oleh *noise*.
7. Kegagalan sistem MFCC+HMM ini lebih dikarenakan nilai power spektrum sebagai penciri sinyal yang merupakan input bagi ekstraksi ciri MFCC bersifat sangat sensitif terhadap *noise*.

Dari hasil yang diperoleh pada bab ini, maka tahap selanjutnya adalah melakukan percobaan dengan menggunakan komposisi data training sebanyak 60 suara (dari total 80) dan dipilih secara acak. Dari aspek metode, untuk pengenalan pola menggunakan HMM dengan jumlah *hidden state* 3 dengan distribusi *unimodal*. Sedangkan untuk ekstraksi ciri adalah mengadopsi teknik MFCC, namun sebagai input menggunakan besaran yang relatif lebih *robust* terhadap *noise* dibanding dengan power spektrum. Dari studi pustaka pada Bab 2, terlihat bispektrum sinyal asli dengan sinyal asli yang sudah ditambah *noise*, secara visual lebih stabil dibanding dengan power spektrum. Namun demikian, bispektrum ini berdimensi dua. Oleh karena itu, tahapan selanjutnya adalah melakukan ekstensi teknik MFCC dari satu dimensi menjadi dua dimensi, sehingga teknik ini dapat bekerja untuk nilai bispektrum sinyal.

BAB 5

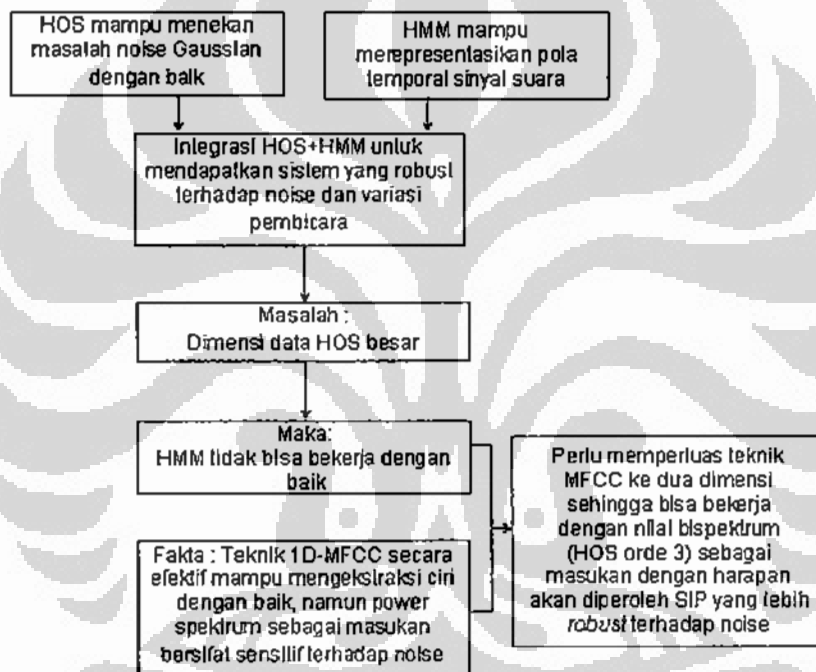
SISTEM IDENTIFIKASI PEMBICARA MENGGUNAKAN 2D-MFCC SEBAGAI EKSTRAKSI CIRI DAN HMM SEBAGAI *CLASSIFIER*

Pada bagian sebelumnya, sistem pengenalan pembicara yang telah dibahas adalah yang berbasis pada power spektrum dalam domain frekuensi sebagai penentu ciri sinyal suara. Satu teknik yang berbasiskan pada power spektrum dan telah dikenal luas pemakaiannya adalah MFCC seperti yang disajikan pada bagian sebelumnya. Dari Bab 4 tersebut ditunjukkan bahwa sistem yang dihasilkan mampu mengenali dengan baik untuk data suara tanpa penambahan *noise*. Untuk data suara yang telah ditambah *noise*, terlihat bahwa akurasi turun secara drastis. Adanya *noise canceling* mampu meningkatkan kinerja untuk sinyal yang ditambah *noise*, namun hanya bekerja hingga level 20 dB, itupun dengan akurasi 77.1%. Hal ini dikarenakan power spektrum sebagai penentu ciri bersifat sensitif terhadap gangguan *noise* seperti yang sudah ditunjukkan sebelumnya.

Dari fakta ini dapat disimpulkan bahwa rendahnya akurasi disebabkan oleh karena kelemahan dari besaran yang dipergunakan untuk menangkap ciri sinyal. Oleh karena itu pada bagian ini akan dikembangkan sistem identifikasi pembicara yang berbasis pada HOS sebagai besaran penentu ciri sinyal suara. Satu bentuk HOS yang memiliki sifat tersebut adalah spektrum order 3, yang dikenal dengan nama bispektrum. Secara empiris, penggunaan bispektrum ini telah menunjukkan hasil yang optimal pada sinyal ber-*noise* seperti yang ditunjukkan oleh [FB98], [HB99] dan [TB00]. Namun demikian ketiga penelitian tersebut menggunakan PNN sebagai *classifier* sehingga ada beberapa kelemahan, seperti kurang fleksibelnya sistem untuk bisa diterapkan pada berbagai bidang terapan. Selain itu, dengan PNN, memerlukan *memory* yang cukup besar dikarenakan seluruh data training harus disimpan.

Dari hal di atas, maka pada bagian ini akan dikaji lebih lanjut penggunaan bispektrum sebagai penentu ciri sinyal dan bagaimana integrasinya dengan HMM sebagai pengenalan pola. Seperti telah disebutkan sebelumnya bahwa permasalahan MFCC sebagai ekstraksi ciri adalah pada besaran penentu ciri sinyal suara, dan bukan

pada teknik MFCC sebagai pengekstraksi ciri. Masukan dari MFCC tersebut adalah power spektrum yang secara empiris terlihat sangat sensitif terhadap *noise*. Oleh karena itu, pada bagian ini teknik MFCC yang bekerja pada power spektrum dalam ruang satu dimensi (1D-MFCC) akan diperluas menjadi dua dimensi untuk memproses nilai bispektrum yang merupakan besaran dua dimensi, dan berikutnya disingkat sebagai 2D-MFCC. Output 2D-MFCC ini berupa K koefisien seperti juga pada 1D-MFCC, sehingga integrasinya dengan HMM diharapkan bisa berjalan dengan baik. Pola pikir di atas didasarkan pada kerangka pemikiran seperti yang disajikan pada **Gambar 5.1**.

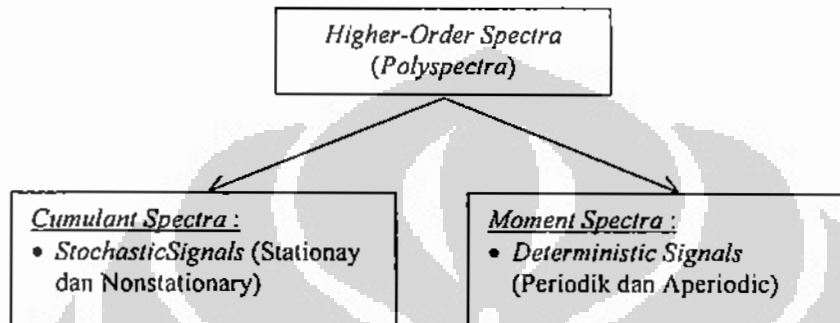


Gambar 5.1. Kerangka Pikir Pengembangan 1D-MFCC menjadi 2D-MFCC

Bab ini akan diawali dengan konsep HOS, dilanjutkan dengan rumusan memperluas teknik MFCC dari satu dimensi ke dua dimensi, formulasi pembentukan filter dengan algoritme genetika, rancangan percobaan yang dilakukan serta hasilnya dan ditutup dengan menyajikan beberapa temuan serta tahapan selanjutnya.

5.1. Higher Order Statistics

Dari suatu gugus yang terdiri n peubah acak, $\{x_1, x_2, x_3, \dots, x_n\}$, dikenal dua statistik orde tinggi (*higher-order statistics*, HOS) yaitu *Moment* dan *Cumulant*. *Moment* lebih sesuai untuk analisis terhadap sinyal deterministik, dan *cumulant* untuk sinyal stokastik (Gambar 5.2.), [NP93].



Gambar 5.2. Klasifikasi Penggunaan HOS

Nikias dan Raghuveer, 1987 dalam [NP93] menyatakan secara umum ada tiga alasan mengapa menggunakan HOS, yaitu :

- a. Untuk menekan pengaruh sinyal *Gaussian Noise* pada masalah pendeteksian, pendugaan parameter dan klasifikasi. HOS orde 3 (*Bispectrum*) juga mampu menekan pengaruh *non-Gaussian Noise* yang mempunyai fungsi distribusi yang simetrik.
- b. Merekonstruksikan respon *magnitude* dan *phase* dari sinyal atau sistem.
- c. Mendeteksi dan mengkarakterisasi adanya *nonlinearitas* data deret waktu.

Keuntungan yang pertama didasarkan pada rumusan matematika bahwa *cumulant* orde tiga ke atas untuk proses *Gaussian* adalah nol. Oleh karena itu, suatu sinyal *non-Gaussian* yang terkontaminasi oleh *Gaussian noise* setelah menggunakan transformasi *cumulant*, secara teori *noise* tersebut akan di-eliminasi. Keuntungan yang kedua didasarkan pada kemampuan HOS orde di atas dua yang mampu menangkap fase sinyal, yang diperlukan pada pemodelan data deret waktu untuk merekonstruksi sinyal. Sementara itu, pemodelan yang hanya didasarkan pada orde

kedua (power spektrum) dengan statistik autokorelasi, sifat fase sinyal tidak bisa ditangkap. Albera et. Al., [AFCC05], memanfaatkan sifat pertama di atas untuk membangun *Independent Component Analysis* berbasis *Redundancy* pada *quadraticovariance* (ICAR) untuk separasi sinyal campuran beberapa sumber (*Blind Source Separation*) dalam situasi ber-noise. Teknik yang dikembangkan berbasis statistik orde empat ini (trispektrum) mampu memberikan hasil yang jauh lebih baik dari metode sebelumnya yang berbasis statistik orde dua (power spektrum). Hal ini juga ditunjukkan oleh Blanco et. al. pada [BRHC04].

5.1.1. Sinyal Stokastik

Untuk sinyal stokastik, maka spektrum *cumulant* lebih menguntungkan dibanding dengan spektrum *moment*. Tiga alasan untuk hal ini adalah, [NP93]:

- Spektrum *cumulant* orde $n > 2$ untuk proses *Gaussian* adalah nol, dan nilai tidak nol memberikan ukuran bagi penyimpangannya terhadap *Gaussian*.
- Spektrum *cumulant* memberikan ukuran penyimpangan keterkaitan statistik dalam data deret waktu.
- Spektrum *cumulant* dari sinyal yang merupakan penjumlahan dua sinyal yang saling bebas, rata-rata tidak nol, dan proses stokastik stasioner adalah jumlah dari spektrum *cumulant* masing-masing sinyal. Sifat ketiga ini tidak dimiliki oleh spektrum *moment*.

Kalau $\{x(t)\}$, $t=0, \pm 1, \pm 2, \pm 3, \dots$ adalah proses acak stasioner dan *moment* orde ke n ada, maka, [NP93] :

$$\text{Mom}[X(t), X(t + \tau_1), \dots, X(t + \tau_{n-1})] = E\{x(t)x(t + \tau_1)x(t + \tau_2) \dots x(t + \tau_{n-1})\} \quad (5.1)$$

Nilai *moment* ini hanya tergantung pada lag $\tau_1, \tau_2, \dots, \tau_{n-1}$, $\tau_i=0, \pm 1, \pm 2, \pm 3, \dots$. Oleh karena itu, rumus *moment* untuk sinyal stasioner menjadi :

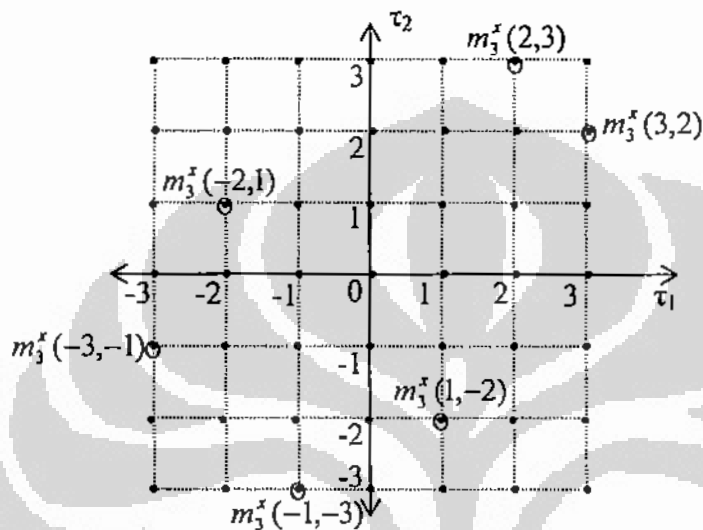
$$m_n^x(\tau_1, \tau_2, \dots, \tau_{n-1}) = E\{x(t)x(t + \tau_1)x(t + \tau_2) \dots x(t + \tau_{n-1})\} \quad (5.2)$$

Formula tersebut berimplikasi bahwa sifat *moment* yang adalah simetrik. Sebagai contoh untuk orde 3, maka terdapat kesamaan :

$$\begin{aligned}
 m_3^x(\tau_1, \tau_2) &= m_3^x(\tau_2, \tau_1) = m_3^x(\tau_2 - \tau_1, -\tau_1) \\
 &= m_3^x(-\tau_1, \tau_2 - \tau_1) = m_3^x(-\tau_2, \tau_1 - \tau_2) = m_3^x(\tau_1 - \tau_2, -\tau_2)
 \end{aligned}$$

Gambar 5.3. berikut adalah contoh sifat simetrik untuk *moment* orde 3 untuk $\tau_1 = 3, \tau_2 = 2$, yang dalam hal ini berlaku :

$$m_3^x(3,2) = m_3^x(2,3) = m_3^x(-1,-3) = m_3^x(-3,-1) = m_3^x(-2,1) = m_3^x(1,-2)$$



Gambar 5.3. Sifat Simetrik *Moment* (kasus orde 3, dengan $\tau_1=3, \tau_2=2$.)

Sedangkan *cumulant* dari $\{x(t)\}$ adalah, [NP93] :

$$c_n^x(\tau_1, \tau_2, \dots, \tau_{n-1}) = \text{cum}\{x(t)x(t+\tau_1)x(t+\tau_2)\dots x(t+\tau_{n-1})\} \quad (5.3)$$

dengan $\text{cum}(x_1, x_2, \dots, x_n)$ adalah *cumulant* dari set peubah acak $\{x_1, x_2, \dots, x_n\}$, yang dirumuskan sebagai :

$$\text{Cum}(X_1, X_2, \dots, X_n) = \sum_{p=1}^n (-1)^{p-1} (p-1)! E \left(\prod_{i \in S_1} X_i \right) \left(\prod_{i \in S_2} X_i \right) \dots E \left(\prod_{i \in S_p} X_i \right) \quad (5.4)$$

Dalam hal ini $p=1,2,3,\dots,n$, yaitu banyaknya sekatan dari himpunan $\{x_1, x_2, \dots, x_n\}$ dan s_i adalah sekatan ke i pada p tertentu. Sebagai ilustrasi misalkan akan ditentukan *cumulant* order 3, yaitu $C_3^x(X_1, X_2, X_3)$:

$$p=1 : S=\{X_1, X_2, X_3\} \quad \rightarrow (-1)^0 0! E(X_1 X_2 X_3) = E(X_1 X_2 X_3)$$

$$\begin{aligned}
p=2 : S_1=\{X_1, X_2\}, S_2=\{X_3\} &\rightarrow (-1)^1 1! E(X_1 X_2) E(X_3) = -E(X_1 X_2) E(X_3) \\
S_1=\{X_1, X_3\}, S_2=\{X_2\} &\rightarrow (-1)^1 1! E(X_1 X_3) E(X_2) = -E(X_1 X_3) E(X_2) \\
S_1=\{X_2, X_3\}, S_2=\{X_1\} &\rightarrow (-1)^1 1! E(X_2 X_3) E(X_1) = -E(X_2 X_3) E(X_1) \\
p=3 : S_1=\{X_1\}, S_2=\{X_2\}, S_3=\{X_3\} &\rightarrow (-1)^2 2! E(X_1) E(X_2) E(X_3) = 2E(X_1) E(X_2) E(X_3)
\end{aligned}$$

dengan menjumlahkan seluruh komponen untuk $p=1$ sampai dengan 3, diperoleh

$C_3^x(X_1, X_2, X_3)$, yaitu :

$$E(X_1 X_2 X_3) - E(X_1 X_2) E(X_3) - E(X_1 X_3) E(X_2) - E(X_2 X_3) E(X_1) + 2E(X_1) E(X_2) E(X_3)$$

Atau :

$$C_3^x = m_3 - 3m_2 m_1 + 2m_1^3 \quad \text{dengan } m_i \text{ adalah } \textit{moment} \text{ ke } i.$$

Berdasar sifat simetrik dari *moment* tersebut di atas, maka *cumulant* orde 3 juga memenuhi sifat simetri berikut :

$$\begin{aligned}
c_3^x(\tau_1, \tau_2) &= c_3^x(\tau_2, \tau_1) = c_3^x(\tau_2 - \tau_1, -\tau_1) \\
&= c_3^x(-\tau_1, \tau_2 - \tau_1) = c_3^x(-\tau_2, \tau_1 - \tau_2) = c_3^x(\tau_1 - \tau_2, -\tau_2)
\end{aligned}$$

Jika sinyal stasioner pada rata-rata nol, yang berarti bahwa $E(X_i)=0$, maka nilai *cumulant* tersebut menjadi :

$$c_3^x(X_1, X_2, X_3) = E(X_1 X_2 X_3) = m_3$$

Ini berarti bahwa *cumulant* orde 3 adalah sama dengan *moment* orde 3 untuk sinyal dengan rata-rata nol.

Beberapa sifat *moment* dan *cumulant* sinyal stokastik adalah sebagai berikut, [NP93] :

a. Untuk $a_1, a_2, a_3, \dots, a_n$ adalah konstanta berlaku :

- ✓ $Mom[a_1 x_1, a_2 x_2, a_3 x_3, \dots, a_n x_n] = a_1 a_2 a_3 \dots a_n Mom[x_1, x_2, x_3, \dots, x_n]$
- ✓ $Cum[a_1 x_1, a_2 x_2, a_3 x_3, \dots, a_n x_n] = a_1 a_2 a_3 \dots a_n Cum[x_1, x_2, x_3, \dots, x_n]$

- b. *Moment* dan *cumulant* adalah fungsi yang bersifat simetrik terhadap argumennya, yaitu :

$$Mom[x_1, x_2, x_3] = Mom[x_2, x_1, x_3] = Mom[x_3, x_1, x_2] = \dots \text{ dan seterusnya.}$$

- c. Jika peubah acak $\{x_1, x_2, \dots, x_n\}$ dapat disekat menjadi dua atau lebih himpunan yang saling bebas, maka *cumulant* orde n adalah $Cum[x_1, x_2, \dots, x_n] = 0$, dan secara umum $Mom[x_1, x_2, \dots, x_n] \neq 0$.
- d. Jika himpunan $\{x_1, x_2, \dots, x_n\}$ dan $\{y_1, y_2, \dots, y_n\}$ saling bebas, maka :

$$Cum[x_1+y_1, x_2+y_2, \dots, x_n+y_n] = Cum[x_1, x_2, \dots, x_n] + Cum[y_1, y_2, \dots, y_n]$$

Sedangkan untuk *moment*, hal tersebut secara umum tidak berlaku.

- e. Jika distribusi bersama dari $\{x_1, x_2, \dots, x_n\}$ adalah *Gaussian*, maka semua informasi mengenai distribusi adalah hanya tersedia pada *moment* orde $n \leq 2$.

Kalau $\{x(t)\}$, $t=0, \pm 1, \pm 2, \pm 3, \dots$ adalah real dan stasioner dengan barisan *cumulant*, $\{c_n^x\}$, seperti dirumuskan sebelumnya, dan memenuhi kondisi :

$$\sum_{\tau_1=-\infty}^{+\infty} \dots \sum_{\tau_{n-1}=-\infty}^{+\infty} |c_n^x(\tau_1, \tau_2, \dots, \tau_{n-1})| < \infty$$

maka spektrum dari *cumulant* orde n (disebut juga *Higher-order spectrum*) ada dan kontinyu. Spektrum *Cumulant* tersebut didefinisikan sebagai transformasi Fourier dimensi $(n-1)$ dari barisan *cumulant* orde n , dan dirumuskan sebagai :

$$C_n^x(\omega_1, \omega_2, \dots, \omega_{n-1}) = \sum_{\tau_1=-\infty}^{+\infty} \dots \sum_{\tau_{n-1}=-\infty}^{+\infty} c_n^x(\tau_1, \tau_2, \dots, \tau_{n-1}) \exp\{-j(\omega_1 \tau_1, \omega_2 \tau_2, \dots, \omega_{n-1} \tau_{n-1})\} \quad (5.5)$$

dengan $|\omega_i| \leq \pi$ untuk $i=1, 2, 3, \dots, n-1$ dan $|\omega_1 + \omega_2 + \dots + \omega_{n-1}| \leq \pi$. Definisi alternatif dari spektrum *cumulant* adalah :

$$Cum[dZ(\omega_1), dZ(\omega_2), \dots, dZ(\omega_n)] = \begin{cases} C_n^x(\omega_1, \dots, \omega_{n-1}) d\omega_1 \dots d\omega_{n-1}, & \omega_1 + \dots + \omega_n = 0 \\ 0 & \omega_1 + \dots + \omega_n \neq 0 \end{cases} \quad (5.6)$$

Magnitudo dan fase spektrum *cumulant* adalah :

$$C_n^x(\omega_1, \omega_2, \dots, \omega_{n-1}) = |C_n^x(\omega_1, \omega_2, \dots, \omega_{n-1})| \exp\{j\Psi_n^x(\omega_1, \omega_2, \dots, \omega_{n-1})\} \quad (5.7)$$

dan periodik dengan periode 2π .

Bispektrum dari barisan peubah acak $\{x(t)\}$ adalah spektrum *cumulant* order 3 dan dirumuskan sebagai :

$$C_3^x(\omega_1, \omega_2) = \sum_{\tau_1=-\infty}^{+\infty} \sum_{\tau_2=-\infty}^{+\infty} c_3^x(\tau_1, \tau_2) \exp\{-j(\omega_1\tau_1, \omega_2\tau_2)\} \quad (5.8)$$

dengan $|\omega_1| \leq \pi$, $|\omega_2| \leq \pi$, $|\omega_1 + \omega_2| \leq \pi$. Dalam hal ini $c_3^x(\tau_1, \tau_2)$ adalah barisan *cumulant* order 3 dari barisan peubah acak $\{x(t)\}$ tersebut. Beberapa sifat dari bispektrum adalah sebagai berikut :

- $C_3^x(\omega_1, \omega_2)$ adalah kompleks yang bisa dituliskan dalam magnitudo dan fase:

$$C_3^x(\omega_1, \omega_2) = |C_3^x(\omega_1, \omega_2)| \exp(j\psi_C(\omega_1, \omega_2))$$

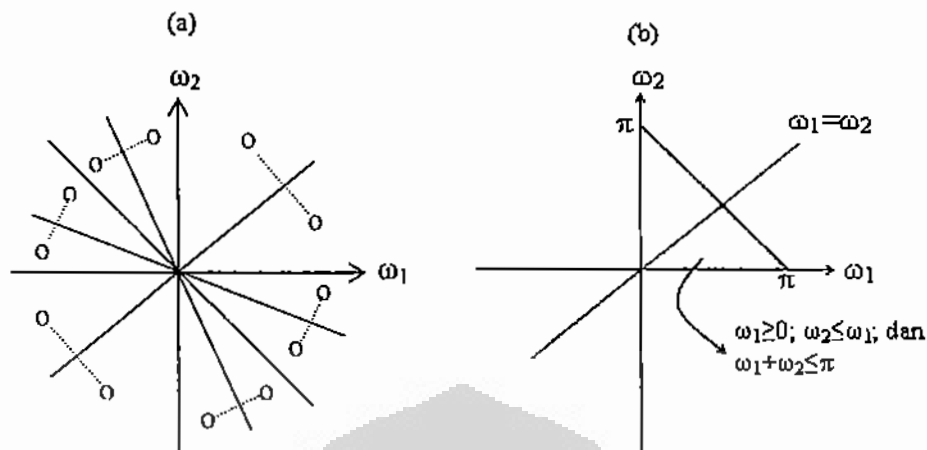
- periode dari $C_3^x(\omega_1, \omega_2)$ adalah 2π , sehingga :

$$C_3^x(\omega_1, \omega_2) = C_3^x(\omega_1 + 2\pi, \omega_2 + 2\pi)$$

- bersifat simetri :

$$\begin{aligned} C_3^x(\omega_1, \omega_2) &= C_3^x(\omega_2, \omega_1) = C_3^{x^*}(-\omega_2, -\omega_1) \\ &= C_3^x(-\omega_1 - \omega_2, \omega_2) \\ &= C_3^x(\omega_1, -\omega_1 - \omega_2) = C_3^x(-\omega_1 - \omega_2, \omega_1) \\ &= C_3^x(\omega_2, -\omega_1 - \omega_2) \end{aligned}$$

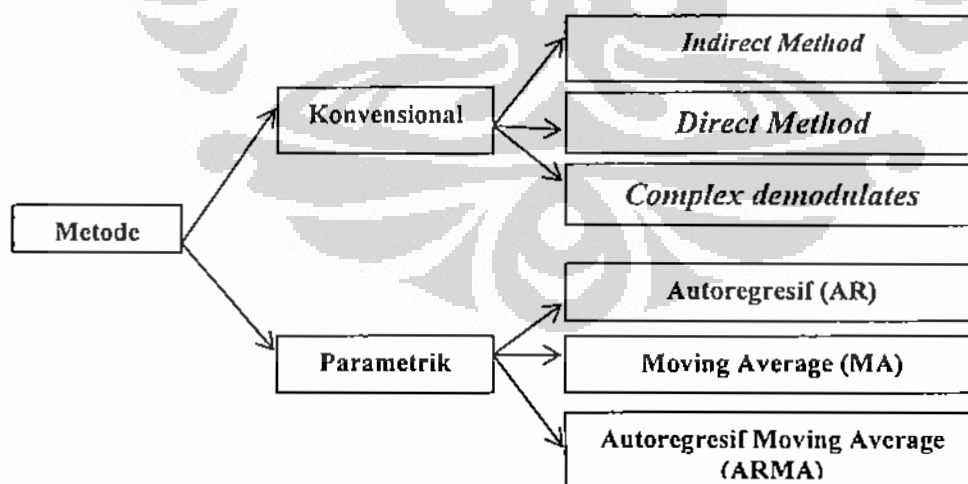
Dengan sifat simetrik di atas, maka daerah *support* untuk bispektrum adalah seperti disajikan pada **Gambar 5.4**. Terlihat ada 12 daerah yang meng-*support* bispektrum dan memberikan nilai yang sama, sehingga pada pendugaan bispektrum dapat digunakan salah satu daerah saja. Sedangkan berdasar batasan pada formula (5.3), maka daerah definisi bispektrum adalah sesuai **Gambar 5.4b**.



Gambar 5.4. Daerah *Support* Bispektrum (a), Definisi Bispektrum (b)

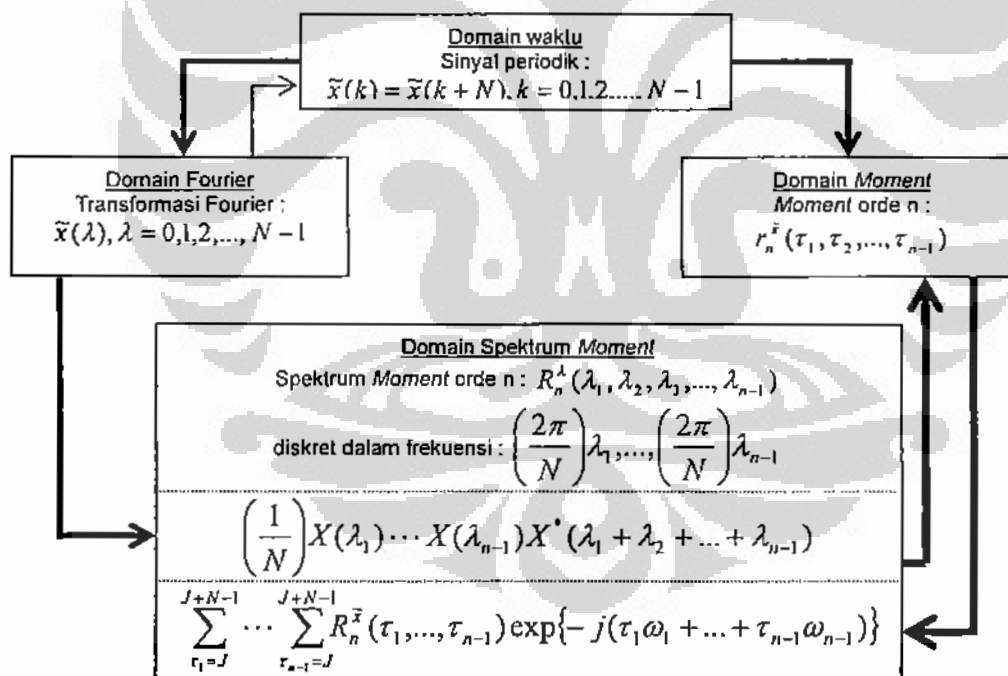
5.1.2. Pendugaan Bispektrum

Ada dua pendekatan utama untuk menduga nilai spektrum orde tinggi, yaitu metode konvensional yang berbasis transformasi Fourier dan metode parametrik yang berbasis *Autoregresif* (AR), *Moving Average* (MA), *Autoregresif Moving Average* (ARMA), dan model Volterra, [NP93], seperti disajikan pada Gambar 5.5. Metode konvensional terdiri dari tiga kelas, yaitu *indirect class* (tidak langsung) yang berbasis (5.5), *direct class* (langsung) yang berbasis (5.6), dan *demodulate class*.



Gambar 5.5. Klasifikasi Metode Pendugaan Spektrum Orde Tinggi

Pada metode tidak langsung, sinyal dibaca per segmen dan dihitung nilai *moment*-nya pada berbagai lag. *Moment* sinyal diduga dengan merata-ratakan *moment* yang diperoleh dari semua segmen yang ada. Dari sini dihitung *cumulant* dan melalui transformasi Fourier diperoleh spektrum *cumulant*-nya. Pada metode langsung, *moment* setiap segmen sinyal diduga melalui perata-rataan dalam domain frekuensi. Ini berarti sinyal dibaca per segmen dan dilakukan transformasi Fourier. Rata-rata dari *triple-product* hasil transformasi Fourier tersebut sebagai dugaan bagi *moment* segmen tersebut. *Moment* sinyal diduga sebagai rata-rata *moment* seluruh segmen. Teknik ketiga adalah mirip dengan metode langsung, namun setelah transformasi Fourier, dilakukan konvolusi kompleks untuk mentransformasi kembali ke domain waktu. Dari sini dihitung *moment* per segmen sebagai rata-rata dari *triple-product* hasil konvolusi kompleks. Secara garis besar, kaitan antara waktu, transformasi Fourier, *moment* dan spektrum *moment* diilustrasikan pada **Gambar 5.6**.



Gambar 5.6. Hubungan antara Domain Waktu, Fourier, *Moment* dan Spektrum *Moment*, [NP93].

Sesuai dengan [NP93], dugaan terhadap HOS dengan teknik konvensional selain sederhana, juga secara asimtotik tidak berbias dan konsisten. Kelemahannya adalah pada "uncertainty principle" dari transformasi Fourier yang menyebabkan teknik konvensional kurang bisa diandalkan untuk menduga komponen harmonik sinyal. Teknik ini sesuai untuk data sinyal yang "cukup panjang". Sebagai ilustrasi, dengan jumlah segmen 16 yang berukuran 128 data, teknik konvensional memberikan dugaan yang lebih baik dibanding dengan teknik berbasis parametrik. Sementara itu, jika asumsi terhadap sinyal dipenuhi, maka penggunaan teknik parametrik akan memberikan hasil yang lebih baik, [RN85]. Pada penelitian ini pendugaan bispektrum dilakukan dengan teknik konvensional menggunakan metode tidak langsung yang diimplementasikan menggunakan Matlab yang ditulis oleh Swami, Mendel dan Nikeas, 2001, [SMN01]. Andaikan $\{X(1), X(2), \dots, X(N)\}$ adalah gugus data sebagai realisasi dari suatu proses stokastik, maka algoritme konvensional tidak langsung adalah, [NP93]:

Algoritme 3 : Tahapan Menduga Nilai Bispektrum

1. Pecah gugus data ke dalam K segmen yang masing-masing terdiri M data ($N=KM$).
2. Kurangkan nilai data pada setiap dengan rata-ratanya
3. Misal $\{x^{(i)}(k), k=0, 1, \dots, M-1\}$ adalah data pada tiap segmen ($i=1, 2, \dots, K$), maka dugaan moment order tinggi adalah :

$$m_n^{(i)}(\tau_1, \dots, \tau_{n-1}) = \frac{1}{M} \sum_{k=s_1}^{s_2} x^{(i)}(k) x^{(i)}(k + \tau_1) \dots x^{(i)}(k + \tau_{n-1}) \quad (5.9)$$

Dengan $n=2, 3, \dots; i=1, 2, \dots, K; \tau_k=0, \pm 1, \pm 2, \dots; s_1=\max(0, -\tau_1, \dots, -\tau_{n-1}); s_2=\min(M-1, M-1-\tau_1, \dots, M-1-\tau_{n-1})$ dan $|\tau_k| \leq L_n$, dengan L_n adalah daerah yang mendukung pendugaan fungsi moment orde n.

4. Rata-ratakan momen yang diperoleh dari semua segmen :

$$\hat{m}_n^x(\tau_1, \dots, \tau_{n-1}) = \frac{1}{K} \sum_{i=1}^K m_n^{(i)}(\tau_1, \dots, \tau_{n-1}) \quad n=2, 3, \dots; |\tau_k| \leq L_n \quad (5.10)$$

5. Untuk sinyal stokastik, bangkitkan nilai cumulant $c_n^x(\tau_1, \tau_2, \dots, \tau_{n-1})$ menggunakan formula (2.3.4.). Jika nilai data telah dikurangi dengan rata-ratanya, maka nilai cumulant orde 3 sama dengan nilai moment orde 3 seperti pada langkah 4 :

$$\hat{c}_3^x(\tau_1, \tau_2) = \hat{m}_3^x(\tau_1, \tau_2) = \frac{1}{K} \sum_{i=1}^K m_3^{(i)}(\tau_1, \tau_2) \quad (5.11)$$

6. Spektrum cumulant orde n adalah :

$$\hat{C}_n^x(\omega_1, \omega_2, \dots, \omega_{n-1}) = \sum_{\tau_1=-L_n}^{+L_n} \dots \sum_{\tau_{n-1}=-L_n}^{+L_n} \hat{c}_n^x(\tau_1, \tau_2, \dots, \tau_{n-1}) w(\tau_1 \Delta_n, \tau_2 \Delta_n, \dots, \tau_{n-1} \Delta_n) \exp\{-j(\omega_1 \tau_1, \omega_2 \tau_2, \dots, \omega_{n-1} \tau_{n-1})\} \quad (5.12)$$

$w(u_1, u_2, \dots, u_{n-1})$ adalah fungsi window kontinyu dan $\Delta_n = 1/L_n$.

Fungsi window ini diperlukan agar nilai spektrum dugaan bersifat mulus. Fungsi window ini harus memenuhi sifat, (Rosenblatt dan Van Ness, 1965; Sasaki, et al., 1975; Rao dan Gabr, 1984; dalam [NP93]) :

1. Memenuhi sifat simetri seperti pada *moment* dan *cumulant*. Sebagai contoh untuk kasus bispektrum sinyal real, maka harus dipenuhi $w(m,n)=w(n,m)=w(-m,n-m)=w(m,-n,-n)$
2. Bernilai 0 untuk daerah diluar *support* dari pendugaan, yaitu : $w(\tau_1 \Delta_n, \tau_2 \Delta_n, \dots, \tau_{n-1} \Delta_n) = 0$ untuk $|\tau_k| > L_n, k=1,2,3,\dots,n-1$
3. Bernilai satu pada titik asal, $w(0,0,\dots,0)=1$
4. Mempunyai nilai transformasi Fourier yang real non negatif, yaitu : $w(\omega_1, \omega_2, \dots, \omega_{n-1}) \geq 0$ untuk $|\omega_k| \leq \pi, k=1,2,3,\dots,n-1$.

Window dimensi $(n-1)$, $n > 1$, yang memenuhi sifat-sifat di atas dapat dibangkitkan dari window dimensi 1 dengan formula :

$$w(\tau_1 \Delta_n, \tau_2 \Delta_n, \dots, \tau_{n-1} \Delta_n) = d(\tau_1 \Delta_n) \dots d(\tau_{n-1} \Delta_n) d((\tau_1 + \dots + \tau_{n-1}) \Delta_n) \quad (5.13)$$

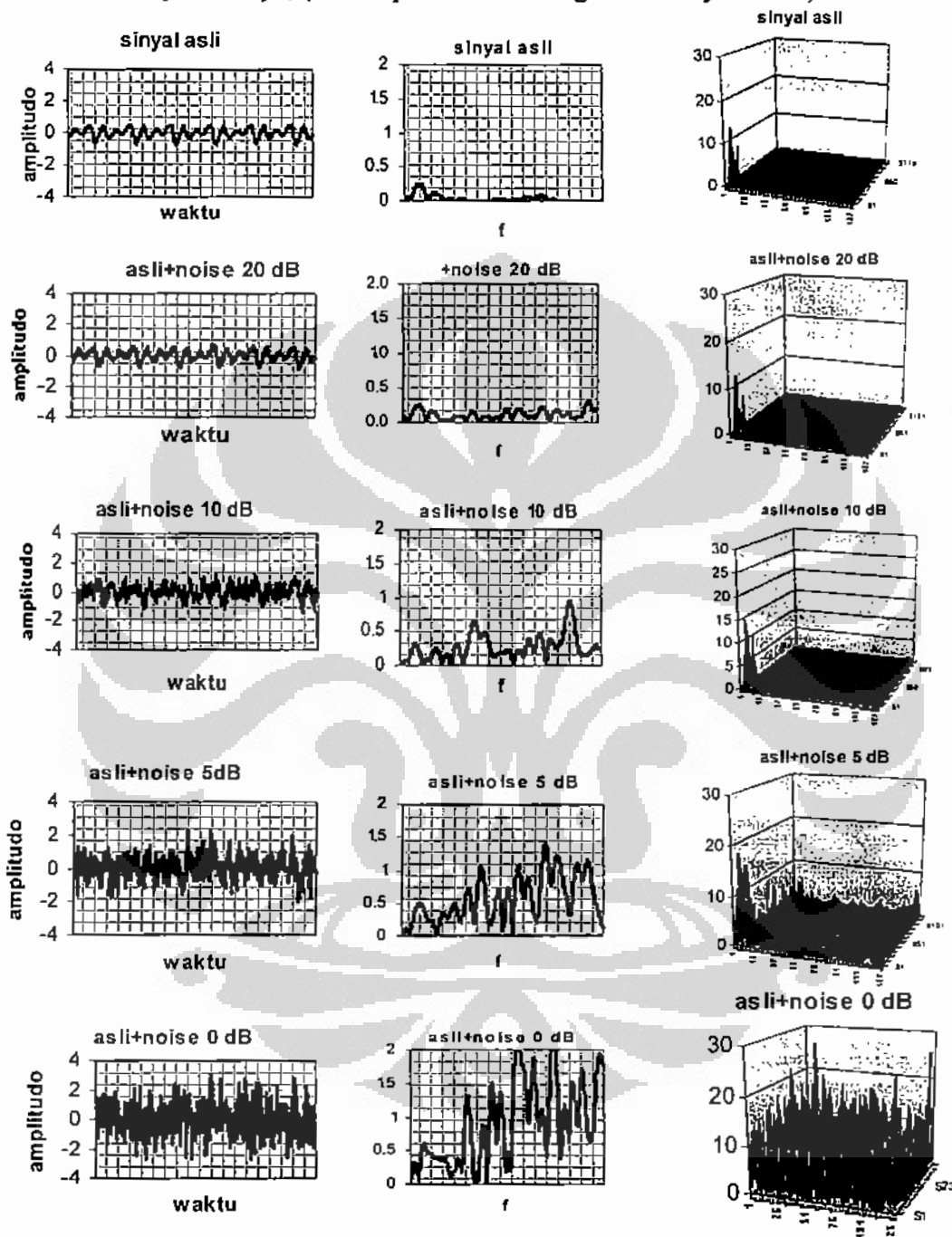
Dengan $d(\tau \Delta_n)$ adalah window dimensi 1 satu dengan sifat :

- a. $d(\tau \Delta_n) = d(-\tau \Delta_n)$
- b. $d(\tau \Delta_n) = 0$ untuk $|\tau| > L_n$
- c. $d(0) = 1$
- d. $D(\omega) \geq 0$ untuk $|\omega| \leq \pi$

Beberapa window yang memenuhi sifat tersebut telah disajikan pada Bab 2.

Gambar 5.7. memberikan perbandingan secara visual antara power spektrum dengan bispektrum untuk data asli dan data yang sudah ditambah *noise*. Dari gambar tersebut terlihat bahwa nilai power spektrum (kolom yang ditengah) mengalami

perbedaan yang berarti akibat dari penambahan *noise*. Sedangkan pada bispektrum (kolom terakhir), hingga *noise* di atas 0 dB, masih memiliki kesamaan dengan bispektrum dari sinyal aslinya, (kolom pertama adalah gambar sinyal suara).



Gambar 5.7. Perbandingan Pola Spektrum (1D dengan 2D) dari Sinyal Asli dengan Sinyal yang Ditambah *Noise*

5.2. Perluasan 1D-MFCC ke 2D-MFCC

Pada banyak aplikasi pemrosesan sinyal, sinyal input dibaca frame demi frame. Kemudian, pada setiap frame dilakukan proses ekstraksi ciri. Pada sistem dengan 1D-MFCC sebagai pengekstraksi ciri, komponen yang diproses dari setiap frame adalah nilai spektrum atau periodogram yang diperoleh dari transformasi Fourier dan berupa vektor dengan N elemen. Pada 2D-MFCC, periodogram tersebut digantikan oleh bispektrum yang berupa array dua dimensi dengan ukuran $N \times N$ (N adalah banyaknya titik frekuensi yang disampling), dan setiap unsurnya dinotasikan dengan $B(f_1, f_2)$. Dalam hal ini $B(f_1, f_2)$ adalah nilai bispektrum pada frekuensi (f_1, f_2) , dengan f_1 dan f_2 adalah $1, 2, 3, \dots, N$. Oleh karena itu, teknik MFCC yang bekerja pada data satu dimensi (1D-MFCC) seperti yang dijelaskan pada Bab 2 akan diperluas sehingga bisa bekerja pada data dua dimensi (2D-MFCC).

Perluasan 1D-MFCC menjadi 2D-MFCC ini terletak pada dimensi filter yang dipergunakan. Hal ini dikarenakan pada 2D-MFCC, komponen yang masuk ke proses filtering adalah nilai bispektrum yang merupakan data dua dimensi dengan $N \times N$ komponen. Oleh karena itu, filter pada 1D-MFCC perlu diperluas dari 1D menjadi 2D. Kalau pada 1D-MFCC proses wrapping dilakukan pada satu sumbu oleh M filter satu dimensi, maka pada 2D-MFCC proses tersebut dikenakan pada bidang oleh $M \times M$ filter oleh filter dua dimensi. Oleh karena bispektrum bersifat simetrik, maka pusat untuk $M \times M$ filter dua dimensi ini adalah $(f_{1i}, f_{2j}) \in F_1 \times F_2$, dengan $F_1 = \{f_{11}, f_{12}, f_{13}, \dots, f_{1M}\}$ dan $F_2 = \{f_{21}, f_{22}, f_{23}, \dots, f_{2M}\}$. Dalam hal ini $f_{21} = f_{11}$, $f_{22} = f_{12}$, $f_{23} = f_{13}$, ..., $f_{2M} = f_{1M}$, dengan $f_{11}, f_{12}, f_{13}, \dots, f_{1M}$ adalah pusat filter satu dimensi.

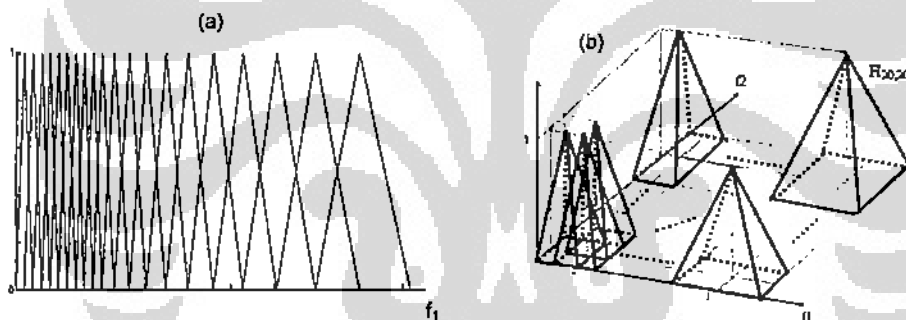
Seperti juga pada filter 1D-MFCC, filter 2D-MFCC ini dapat dibentuk sesuai dengan jenis-jenis filter yang dikenal, yaitu yang disarankan oleh Davis dan Melmerstein, Young, et. al., Slaney, ataupun oleh Skowronski dan Harris yang disebut sebagai HFCC (seperti yang dijelaskan pada Bab 2). Pada penelitian ini, filter dua dimensi yang dibentuk adalah mengikuti Davis dan Melmerstein. Hal ini dengan pertimbangan pada hasil yang diperoleh oleh Ganchev, [Gan05], pemilihan filter ini

dapat menghasilkan sistem dengan akurasi yang baik pada kasus 1D-MFCC. Oleh karena itu akan diperoleh $20 \times 20 = 400$ filter dua dimensi yang berupa limas segi empat dengan tinggi satu. Tahapan detail membentuk $M \times M$ filter ($M=20$) tersebut adalah seperti disajikan pada algoritme 4.

Algoritme 4 : Pembentukan Filter Dua Dimensi (2D Filter)

- Menggunakan algoritme 1, tentukan pusat filter ke i untuk sumbu $F1$ (yaitu $f1_i$) dan juga untuk sumbu $F2$ (yaitu $f2_j$).
- Filter dimensi dua berupa limas dengan tinggi satu dan proyeksi titik puncak ke bidang alas pada titik $(f1_i, f2_j)$, dan sudut bidang alasnya adalah titik $(f1_{i-1}, f2_{j-1})$, $(f1_{i+1}, f2_{j-1})$, $(f1_{i-1}, f2_{j+1})$, dan $(f1_{i+1}, f2_{j+1})$.

Output algoritme tersebut adalah $M \times M$ filter dua dimensi yang berupa limas yang saling tumpang tindih pada bidang $(F1 \times F2)$. **Gambar 5.8.** menyajikan perbandingan filter satu dimensi dengan filter dua dimensi.



Gambar 5.8. Filter untuk Proses *Wrapping*, (a) 1D-MFCC, (b) 2D-MFCC pada pada Ruang $(F1, F2)$

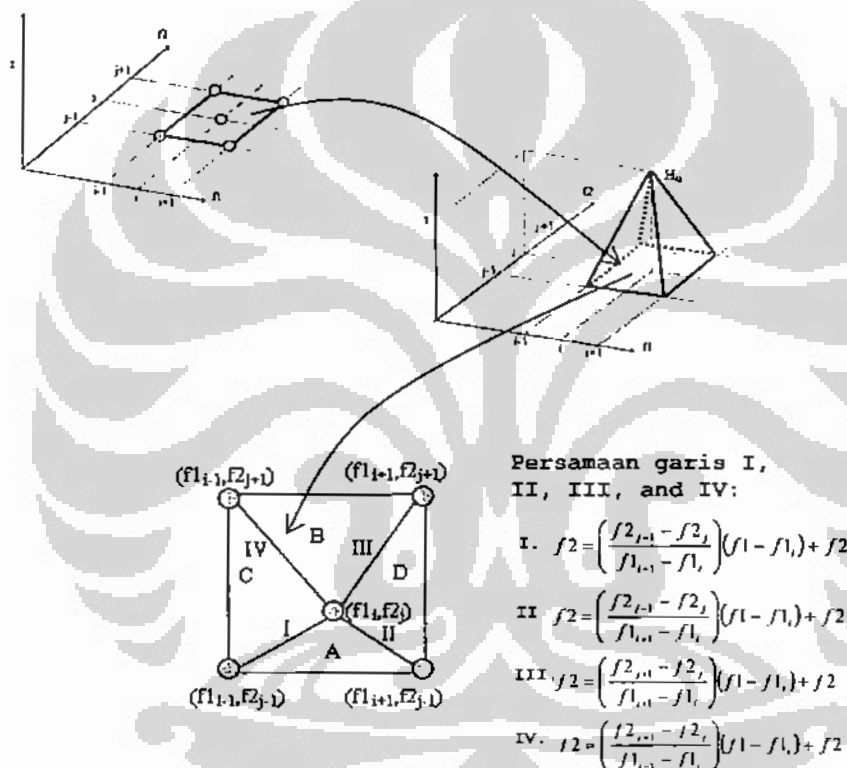
Dengan filter 2D ini, nilai koefisien *mel-spektrum* (*mel-spectrum coefficients*) pada formula (2.27) akan menyesuaikan dan berubah menjadi :

$$X(i, j) = \log \left[\sum_{f1=1}^{128} \sum_{f2=1}^{128} B(f1_n, f2_n) * H_{i,j}(f1_n, f2_n) \right] \quad (5.14)$$

Nilai ini disebut sebagai koefisien *mel-bispectrum* (*mel-bispectrum coefficients*). Pada formula 5.14, $X(i,j)$ adalah nilai koefisien *mel-bispectrum* untuk filter $H_{i,j}$, dengan $i,j=1,2,3,\dots,M$, dan M adalah banyaknya sekatan untuk filter pada sumbu $F1$

atau $F2$. $B(f1_m, f2_n)$ adalah nilai bispektrum pada titik $(f1_m, f2_n)$. Sementara itu $H_{i,j}(f1_m, f2_n)$ adalah tinggi filter $H_{i,j}$ pada titik $(f1_m, f2_n)$, dan diformulasikan dengan algoritme 5. **Gambar 5.9.** memberikan ilustrasi alur proses penentuan tinggi filter sesuai algoritme 5.

Pada **Gambar 5.9.** terlihat bahwa bidang filter terbagi menjadi 4 daerah, yang masing-masing sebagai proyeksi sisi filter ke bidang alas, yang dipisahkan empat garis, yaitu garis I, II, III, dan IV. Tinggi filter pada setiap bidang tersebut sesuai dengan formula yang juga disajikan pada **Gambar 5.9.**



Gambar 5.9. Konstruksi Filter $H_{i,j}$ pada Ruang 2D dengan Pusat $(f1_i, f2_j)$

Algoritme 5: Penghitungan $H_{i,j}$ di titik $(f1_n, f2_n)$

- a. Bidang alas filter $H_{i,j}$ disekat menjadi empat bidang (A, B, C, D) yang merupakan proyeksi dari setiap sisi filter limas ke bidang alas.
- b. Tentukan batas-batas daerah pada tahap a.
- c. Tentukan daerah letak titik $(f1_n, f2_n)$ berada dengan aturan :

Jika:

1. $f2_n > f2_{j-1}$, and

$$2. f2_n < \left(\frac{f2_{j+1} - f2_j}{f1_{i+1} - f1_i} \right) (f1_n - f1_i) + f2_j$$

$$3. f2_n < \left(\frac{f2_{j+1} - f2_j}{f1_{i+1} - f1_i} \right) (f1_n - f1_i) + f2_j$$

Maka $(f1_n, f2_n) \in A$

Jika:

$$1. f2_n < f2_{j+1}, \text{ and}$$

$$2. f2_n > \left(\frac{f2_{j+1} - f2_j}{f1_{i+1} - f1_i} \right) (f1_n - f1_i) + f2_j$$

$$3. f2_n > \left(\frac{f2_{j+1} - f2_j}{f1_{i+1} - f1_i} \right) (f1_n - f1_i) + f2_j$$

maka $(f1_n, f2_n) \in B$

Jika:

$$1. f1_n > f1_{i+1}, \text{ and}$$

$$2. f2_n > \left(\frac{f2_{j+1} - f2_j}{f1_{i+1} - f1_i} \right) (f1_n - f1_i) + f2_j$$

$$3. f2_n < \left(\frac{f2_{j+1} - f2_j}{f1_{i+1} - f1_i} \right) (f1_n - f1_i) + f2_j$$

maka $(f1_n, f2_n) \in C$

Jika:

$$1. f1_n < f1_{i+1}, \text{ and}$$

$$2. f2_n > \left(\frac{f2_{j+1} - f2_j}{f1_{i+1} - f1_i} \right) (f1_n - f1_i) + f2_j$$

$$3. f2_n < \left(\frac{f2_{j+1} - f2_j}{f1_{i+1} - f1_i} \right) (f1_n - f1_i) + f2_j$$

maka $(f1_n, f2_n) \in D$

d. Hitung nilai $H_{i,j}(f1_n, f2_n)$ sesuai dengan daerahnya, dengan menggunakan aturan berikut:

$$\text{Jika } (f1_n, f2_n) \in A \text{ maka: } H_{i,j}(f1_n, f2_n) = \frac{f2_n - f2_{j+1}}{f2_j - f2_{j+1}}$$

$$\text{Jika } (f1_n, f2_n) \in B \text{ maka: } H_{i,j}(f1_n, f2_n) = \frac{f2_{j+1} - f2_n}{f2_{j+1} - f2_j}$$

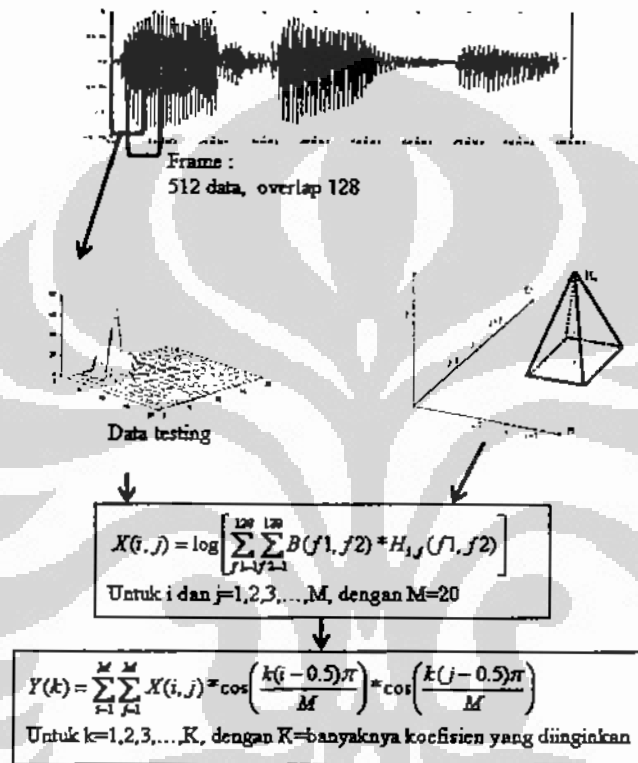
$$\text{Jika } (f1_n, f2_n) \in C \text{ maka: } H_{i,j}(f1_n, f2_n) = \frac{f1_n - f1_{i+1}}{f1_i - f1_{i+1}}$$

$$\text{Jika } (f1_n, f2_n) \in D \text{ maka: } H_{i,j}(f1_n, f2_n) = \frac{f1_{i+1} - f1_n}{f1_{i+1} - f1_i}$$

Tahap akhir dari 2D-MFCC ini adalah mengkonversi koefisien *mel-bispectrum* yang berdimensi M*M dari domain frekuensi ke dalam domain waktu menggunakan transformasi kosinus. Kalau pada 1D-MFCC, transformasi ini adalah orde satu (formula 2.28), maka pada 2D-MFCC menggunakan formula :

$$Y(k) = \sum_{i=1}^M \sum_{j=1}^M X(i, j) * \cos\left(\frac{k(i-0.5)\pi}{M}\right) * \cos\left(\frac{k(j-0.5)\pi}{M}\right) \quad (5.15)$$

$Y(k)$ adalah koefisien ke k dari 2D-MFCC, dengan $k=1,2,3,\dots,K$; K adalah banyaknya koefisien, dan $X(i,j)$ adalah koefisien *mel-bispectrum* dari formula (5.2.14.). **Gambar 5.10.** menyajikan ilustrasi alur proses ekstraksi ciri menggunakan 2D-MFCC.



Gambar 5.10. Ilustrasi Proses Ekstraksi Ciri dengan 2D-MFCC

Di atas telah dijelaskan bagaimana mengembangkan 1D-MFCC menjadi 2D-MFCC yang pada intinya adalah memperluas filter pada proses *wrapping* dari satu dimensi menjadi dua dimensi. Pembentukan filter dua dimensi di atas didasarkan pada model filter satu dimensi yang sudah dikenal, seperti MFCC 20 koefisien, 24 koefisien, 29 koefisien ataupun 40 koefisien. Hal ini dapat dipandang sebagai permasalahan optimisasi, yaitu mencari bentuk filter yang optimum yang mampu melakukan proses *wrapping* sehingga dihasilkan "kemiripan" antara vektor ciri untuk sinyal asli dengan vektor sinyal asli yang ditambah *noise*. Oleh karena itu, bagian

berikutnya akan disajikan pendekatan dengan algoritme genetika untuk mencari solusi masalah tersebut.

5.3. Algoritme Genetika untuk Membentuk Filter Dua Dimensi

Pada bagian sebelumnya telah disajikan proses ekstensi MFCC ke dua dimensi, yang pada intinya adalah memperluas filter pada proses *wrapping* satu dimensi menjadi dua dimensi. Filter dua dimensi yang dibangun mengikuti formula yang dirumuskan dengan mengakomodasi persepsi telinga manusia dalam menerima frekuensi bunyi. Bagian ini akan menyajikan rumusan pengembangan filter melalui algoritme genetika. Dengan pendekatan ini diharapkan filter yang dibentuk secara *data driven* ini dapat mengikuti perubahan dinamis dari data.

5.3.1. Algoritme Genetika

Algoritme genetika (*Genetics Algorithm*, GA) merupakan satu kelas dari pemrograman evolusi. Teknik ini menduga solusi permasalahan atau menduga nilai parameter yang optimal melalui pencarian (*searching*) pada domain solusi dengan mengikuti prinsip evolusi alami. Secara umum, struktur pemrograman evolusi adalah seperti yang disajikan Algoritme 6, [Zbi96].

```

Algoritme 6 :   Struktur Pemrograman Evolusi
Begin
  t←0
  initialize P(t)
  evaluate P(t)
  while (not termination-condition) do
    begin
      t←t+1
      select P(t) from P(t-1)
      alter P(t)
      evaluate P(t)
    end
  end
end

```

Pemrograman evolusi merupakan algoritme probabilistik yang menjaga individu-individu populasi pada generasi ke t , $P(t) = \{x'_1, x'_2, \dots, x'_n\}$, berada pada kondisi yang '*fit*' dibanding individu-individu populasi pada generasi sebelumnya. Setiap individu, x'_i , yang merepresentasikan suatu solusi ini dievaluasi dengan

suatu fungsi yang menunjukkan tingkat *fitness*-nya. Berikutnya populasi baru pada generasi ke $t+1$ dibentuk melalui penyeleksian berbasis tingkat *fitness*-nya dari individu pada generasi ke t (tahap seleksi). Beberapa individu baru akan mengikuti tahapan transformasi melalui operator-operator genetika yang akhirnya akan membentuk populasi pada generasi berikutnya. Operator-operator genetika tersebut adalah transformasi *unary* (dikenal dengan mutasi, m_i) dan transformasi orde tinggi (dikenal dengan *crossover*, c_j). Transformasi *unary* akan membentuk individu baru dengan merubah satu bagian dari individu lama, dan dinotasikan dengan $m_i: S \rightarrow S$. Sedangkan transformasi orde tinggi akan membentuk beberapa individu baru melalui kombinasi bagian dari beberapa individu lama, dan dinotasikan $c_j: S \times S \times \dots \times S \rightarrow S$. Setelah melalui beberapa generasi, solusi akan konvergen, dan diharapkan akan memberikan solusi yang hampir optimum (*near-optimum*).

5.3.2. Ide Dasar Pembentukan Filter Dua Dimensi dengan GA

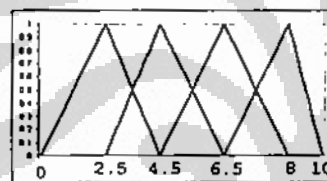
Pada proses ekstraksi ciri menggunakan 2D-MFCC, diperlukan filter dua dimensi untuk melakukan *wrapping* nilai bispektrum dari suatu frame tertentu. Meskipun ada beberapa jenis filter yang bisa dibentuk, pada penelitian ini difokuskan pada filter sigitiga dari sudut pandang satu sumbu (F_1 dan F_2), atau berbentuk limas pada bidang $F_1 \times F_2$, dengan pusat (f_1, f_2) , dengan $f_1 \in CF_1$ and $f_2 \in CF_2$. Beberapa variasi yang bisa dilakukan adalah membedakan antara pencarian filter optimal pada frekuensi rendah dengan frekuensi tinggi, filter dengan luas tetap (sehingga tingginya berubah), atau filter *Gaussian*. Seperti pada bagian 5.2., filter yang akan dibentuk adalah yang mempunyai tinggi tetap, yaitu satu.

Untuk pembentukan filter dua dimensi yang berupa limas dengan tinggi satu, yang dilakukan adalah menentukan M pusat filter pada sumbu F_1 . Oleh karena bispektrum bersifat simetrik, maka pusat dari filter-filter pada F_2 bisa diambil nilai yang sama dengan pusat filter pada sumbu F_1 . Dari M titik pada F_1 dan M titik pusat pada F_2 , dapat dibentuk $M \times M$ filter dua dimensi. Kalau

frekuensi terbesar adalah F , dan x_i adalah jarak antar pusat filter segitiga ke i dengan filter ke $i+1$ untuk $i=2,3,4,\dots,M$, maka $M \times M$ filter ini dapat dibentuk dengan menentukan $x_1, x_2, x_3, \dots, x_{M+1}$ sedemikian sehingga $x_1+x_2+x_3+\dots+x_{M+1}=F$.

Representasi Kromosom

Dari penjabaran di atas terlihat bahwa filter dua dimensi pada bidang $F_1 \times F_2$ dapat dinyatakan atau diidentifikasi dari tuple berukuran $M+1$, $X=(x_1, x_2, x_3, \dots, x_{M+1})$. Dengan algoritme genetika, nilai X ini akan dioptimalkan melalui fungsi *fitness* tertentu. Untuk keperluan ini maka X dikodekan menggunakan simbol biner, yang dalam hal ini ada n digit untuk setiap $x_i \in X$. Oleh karena itu, setiap set filter dua dimensi yang terdiri dari $M \times M$ filter dapat dinyatakan menggunakan simbol biner dengan panjang $n \times (M+1)$. Sebagai ilustrasi, **Gambar 5.11.** menyajikan contoh empat filter satu dimensi dan kode binernya.



Jarak antar dua filter yang berurutan :
 $x_1=2.5$ $x_2=2$ $x_3=2$ $x_4=1.5$ $x_5=2$

LOCUS KE								X
1	0	0	0	0	1	0	1	2.5
2	0	0	0	0	1	0	0	2
3	0	0	0	0	1	0	1	2
4	0	0	0	0	0	1	1	1.5
5	0	0	0	0	1	0	0	2

kromosom:
 00001010000100000010100000110000100

Gambar 5.11. Ilustrasi Pengkodean Kromosom dari Satu Set Filter

Pada **Gambar 5.11.**, dengan frekuensi maksimum 10, set filter segitiga terdiri dari empat filter dengan jarak antar pusat filter 2.5, 4.5, 6.5 dan 8. Ini berarti

$x_1=2.5$, $x_2=4.5-2.5=2$, $x_3=6.5-4.5=2$, $x_4=8-6.5=1.5$, dan $x_5=10-8=2$. Oleh karena itu, kromosom yang dibentuk terdiri dari 5 lokus, yaitu x_1 , x_2 , x_3 , x_4 , and x_5 , dengan setiap lokus dikodekan dengan 7 digit biner, sehingga panjang kromosom adalah $5*7=35$ digit, seperti yang ditunjukkan pada **Gambar 5.11**.

Fungsi Fitness

Fungsi fitness dibentuk sedemikian sehingga untuk sembarang input sinyal suara (dengan atau tanpa penambahan *noise*), output dari ekstraksi ciri setelah filtering adalah sedekat mungkin dengan hasil ekstraksi ciri untuk sinyal tanpa penambahan *noise*. Oleh karena itu, dua kriteria yang ingin dioptimumkan adalah :

- Jarak antara vektor ciri dari sinyal asli (tanpa penambahan *noise*) dengan vektor ciri dari signal asli yang sudah ditambah *noise* adalah sekecil mungkin.
- Jarak antara vektor ciri dari sinyal asli tanpa penambahan *noise* dengan vektor ciri dari sinyal *noise* adalah sebesar mungkin.

Untuk memenuhi dua hal di atas, maka fungsi fitness yang memenuhi adalah :

$$f(i) = \frac{d(s1,s3)*d(s2,s4)}{d(s1,s2)*d(s3,s3)} \quad (5.16)$$

dengan:

$s1$: bispektrum sinyal tanpa penambahan *noise*

$s2$: bispektrum sinyal yang sudah ditambah *noise* of 20 dB

$s3$: selisih $s2$ dengan $s1$

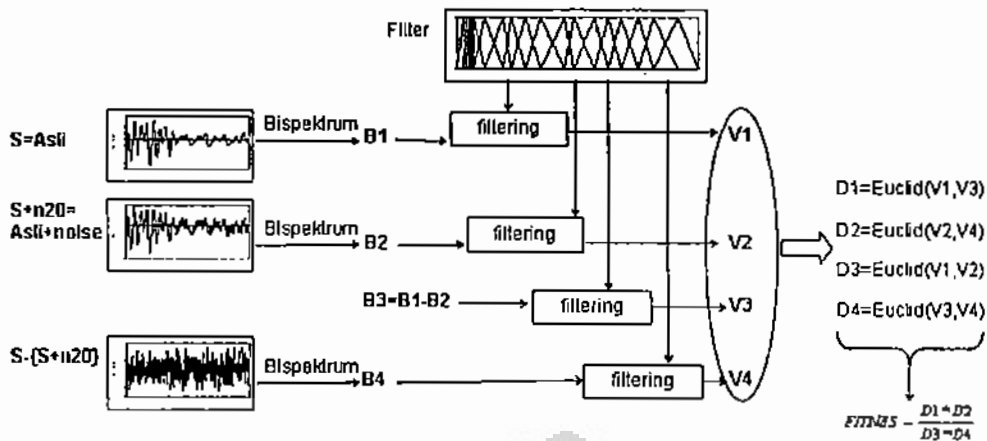
$s4$: bispektrum dari sinyal *noise* 20 dB

$d(a,b)$ jarak antara vektor ciri bispektrum a dengan vektor ciri bispektrum b.

Gambar 5.12. memberikan ilustrasi penentuan nilai fitness.

Seleksi

Untuk seleksi individu atau kromosom terpilih pada generasi ke t , $P(t)$, digunakan aturan putaran Roulette. Kesempatan sembarang kromosom untuk terpilih adalah proporsional dengan nilai *fitness*-nya.



Gambar 5.12. Alur Proses Penghitungan Fitness

Crossover

Operator crossover digunakan untuk menghasilkan individu baru (*offspring*) dengan cara melakukan penyilangan segmen dari dua kromosom induk yang terpilih. Sesuai batasan kromosom di atas, maka operator yang dipilih adalah metode aritmetika. Pembentukan anak dengan metode aritmetika adalah sebagai berikut :
Andaikan X dan Y adalah dua kromosom induk dengan notasi X dan Y :

$$X = (x_1, x_2, x_3, \dots, x_{M+1})$$

$$Y = (y_1, y_2, y_3, \dots, y_{M+1})$$

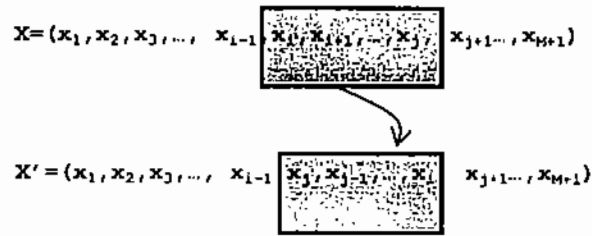
maka dengan $a \in (0, 1)$, dua individu baru yang terbentuk adalah :

$$X' = \{ [ax_1 + (1-a)y_1], [ax_2 + (1-a)y_2], [ax_3 + (1-a)y_3], \dots, [ax_M + (1-a)y_M] \}$$

$$Y' = \{ [ay_1 + (1-a)x_1], [ay_2 + (1-a)x_2], [ay_3 + (1-a)x_3], \dots, [ay_M + (1-a)x_M] \}$$

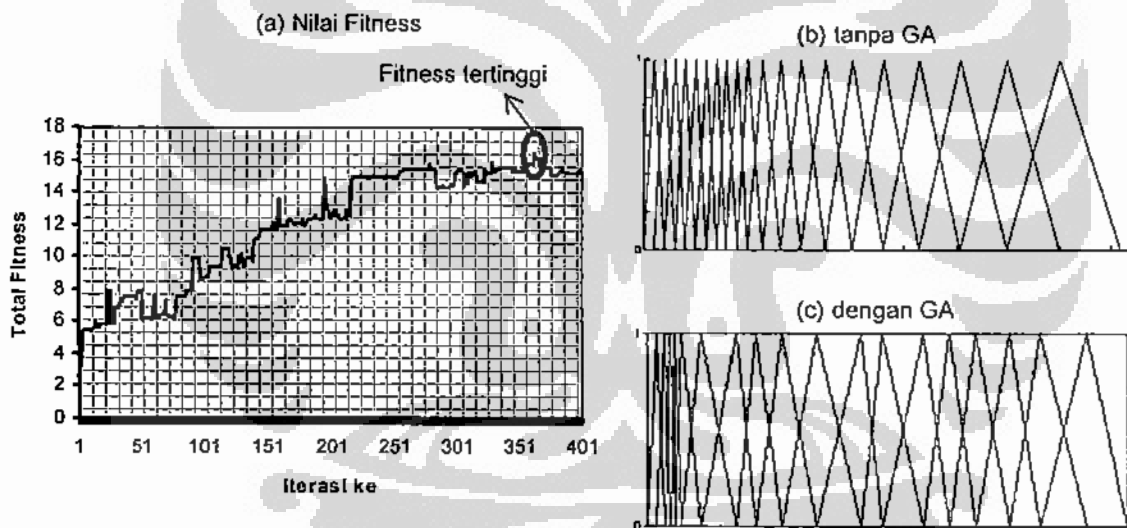
Mutasi

Mutasi merupakan proses mentransformasi satu individu sebagai induk menjadi satu individu baru melalui perubahan internal dari kromosom induk. Perubahan internal ini dilakukan melalui operator *unary*. Salah satu operator *unary* yang sering digunakan adalah teknik inversi. Teknik ini dimulai dengan menyeleksi sembarang kromosom untuk dipilih terjadinya mutasi. Kemudian, dibangkitkan secara acak dua bilangan asli, misalkan p dan q , dengan $p, q \in [0, M+1]$, dan M adalah banyaknya filter yang akan dibuat. Proses mutasi dilakukan dengan membalik urutan lokus yang terletak pada dua bilangan p dan q tersebut, **Gambar 5.13**.



Gambar 5.13. Proses Mutasi Kromosome X menjadi X'

Coding lengkap pembentukan filter dengan GA ini disajikan pada Lampiran 6. Dengan bertambahnya iterasi, nilai total fitness makin besar, dan setelah konvergen atau jumlah iterasi dipenuhi algoritme berhenti. **Gambar 5.14a.** menyajikan perubahan total nilai fitness, dan terlihat bahwa nilai tertinggi pada generasi ke 380. Oleh karena itu, dipilih kromosom dengan fitness tertinggi pada generasi ke 380, dan hasil filter segitiganya disajikan pada **Gambar 5.14b.** dan **Gambar 5.14c.**



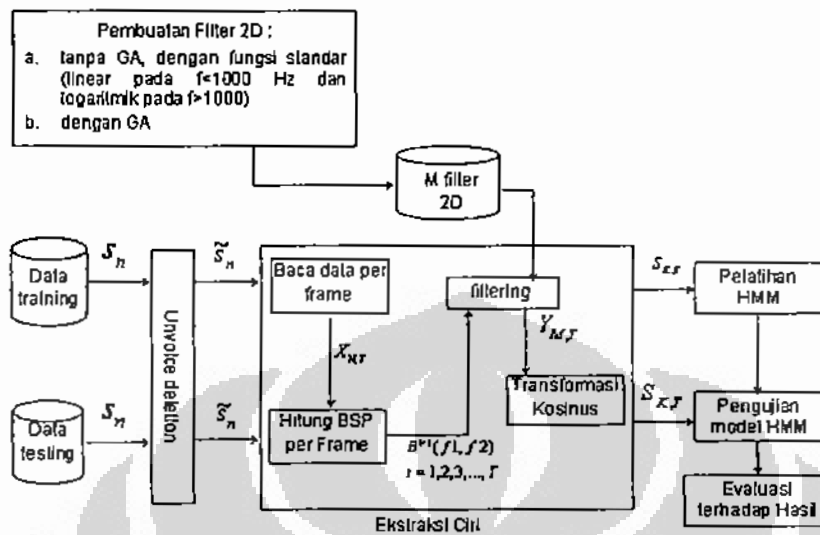
Gambar 5.14. Nilai Total Fitness (a) dan Bentuk Filter Segitiga (b dan c)

5.4. Rancangan Percobaan dan Hasil

5.4.1. Rancangan Percobaan

Bagian ini akan menyajikan hasil pengenalan dari dua percobaan identifikasi pembicara, yang pertama adalah menggunakan 2D-MFCC sebagai ekstraksi ciri tanpa melalui algoritme genetika. Kedua adalah identifikasi pembicara menggunakan 2D-

MFCC yang pembentukan filternya melalui algoritme genetika. **Gambar 5.15.** memberikan diagram mengenai tahapan percobaan yang dilakukan.



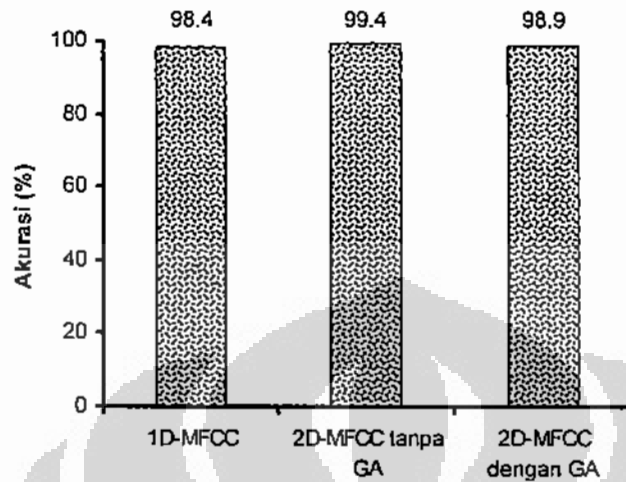
Gambar 5.15. Alur Percobaan dengan 2D-MFCC sebagai Ekstraksi Ciri

Dari **Gambar 5.15.** terlihat bahwa pada setiap data training dilakukan praproses untuk menghapus bagian *unvoice*, dilanjutkan dengan ekstraksi ciri setiap frame sinyal, dan berikutnya menghitung nilai bispektrumnya. Nilai bispektrum ini di-*wrapping* dengan filter. Percobaan pertama dengan filter baku dan kedua dengan filter yang dibentuk dengan algoritme genetika. Output dari proses *wrapping* ini dikenakan transformasi kosinus untuk membalikkan kembali ke domain waktu, dan dipilih K koefisien (dalam percobaan ini diambil $K=13$). Setiap frame yang sudah dikonversi menjadi vektor ciri berdimensi K ini selanjutnya dipergunakan untuk pembentukan model HMM setiap pembicara.

Untuk data uji, setiap sinyal suara melalui tahap praproses dan ekstraksi ciri yang sama dengan data latih untuk memperoleh vektor ciri. Setelah diperoleh vektor ciri, selanjutnya dijalankan algoritme *forward* untuk menghitung skor sinyal masukan tersebut ke setiap pembicara menggunakan model HMM yang telah diperoleh melalui pelatihan, dan diberi label sesuai skor tertingginya. Setelah semua data uji diberi label, maka dilanjutkan dengan evaluasi hasil pengenalan terhadap set data uji.

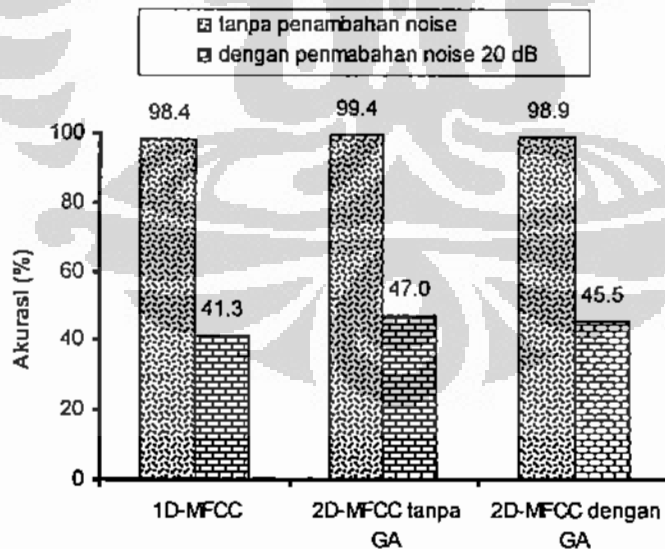
5.4.2. Hasil

Hasil pengenalan untuk data asli disajikan pada **Gambar 5.16**.



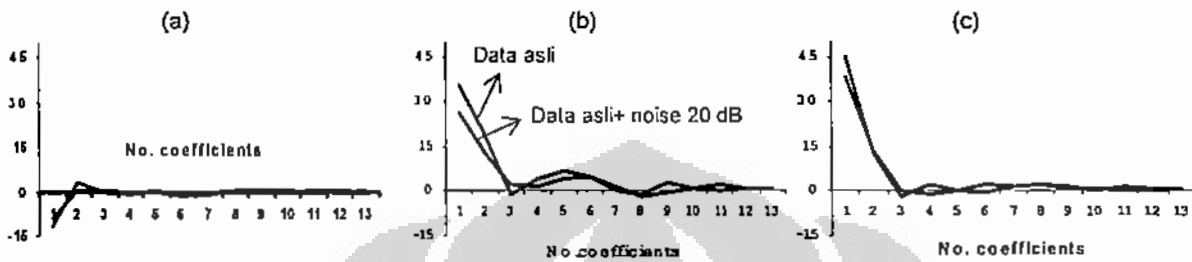
Gambar 5.16. Perbandingan Akurasi Sistem (%) untuk Data Tanpa Penambahan *Noise* dari Tiga Metode Ekstraksi Ciri dengan HMM sebagai *Classifier*

Terlihat bahwa untuk data asli, ketiga metode ekstraksi ciri, yaitu 1D-MFCC, 2D-MFCC standar dan 2D-MFCC dengan GA memberikan akurasi yang relatif tinggi, yaitu di atas 98%. Namun demikian, untuk data yang ditambah *noise*, akurasi sistem turun drastis untuk ketiga metode tersebut, seperti ditunjukkan pada **Gambar 5.17**.



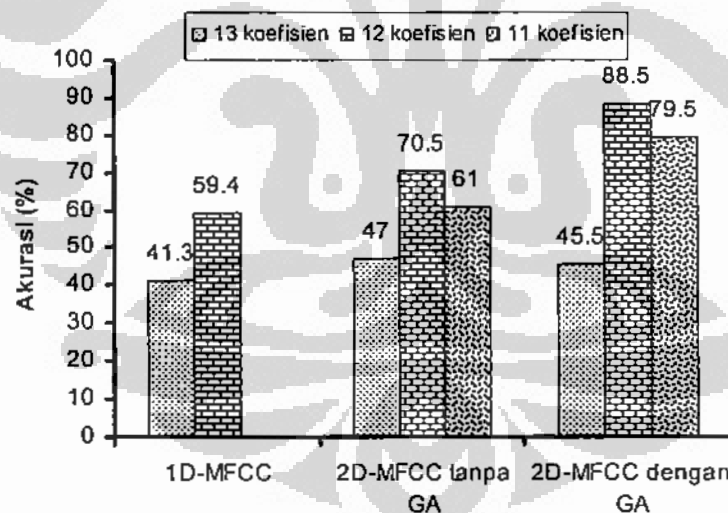
Gambar 5.17. Perbandingan Akurasi Sistem untuk Data Asli dan Data dengan Penambahan *Noise* 20 dB

Terlihat bahwa, meskipun akurasi sistem tinggi untuk data tanpa penambahan noise (>98%), untuk data dengan penambahan noise 20 dB, akurasi langsung drop menjadi 41.3%, 47.0% dan 45.5%, masing-masing untuk 1D-MFCC, 2D-MFCC tanpa dan juga pada 2D-MFCC dengan GA. Pengaruh penambahan *noise* terutama pada koefisien pertama dari vektor ciri, seperti ditunjukkan pada **Gambar 5.18**.



Gambar 5.18. Pola Vektor Ciri dari 13 Koefisien, (a) 1D-MFCC, (b) 2D-MFCC tanpa GA, dan (c) 2D-MFCC dengan GA

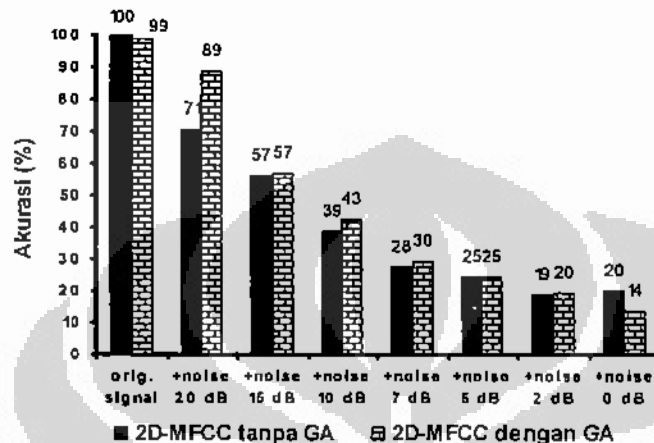
Terlihat bahwa koefisien awal cenderung lebih sensitif terhadap *noise*. Oleh karena itu, dicoba untuk membentuk sistem dengan membuang nilai-nilai koefisien pertama, kedua dan ketiga, dan hasilnya disajikan pada **Gambar 5.19**.



Gambar 5.19. Perbandingan Akurasi (%) dari Tiga Teknik Ekstraksi Ciri untuk Data ber-*noise* 20 dB dengan jumlah Koefisien 13, 12 dan 11

Dari **Gambar 5.19**, dapat dikatakan bahwa jumlah koefisien 12 akan memberikan akurasi tertinggi (88%) untuk data bernoise 20 dB. Dengan 12 koefisien ini, terjadi peningkatan akurasi dari 41.3% menjadi 59.4% untuk 1D-MFCC, 47% menjadi

70.5% untuk 2D-MFCC tanpa GA dan dari 45.5% menjadi 88.5% untuk 2D-MFCC dengan GA. Juga terlihat bahwa metode 2D-MFCC yang dioptimasi dengan algoritme genetika memberikan hasil relatif lebih baik dibanding dua metode lainnya. Namun demikian, untuk data dengan *noise* lebih besar dari 20 dB, ternyata akurasi sistem turun secara nyata, seperti disajikan pada Gambar 5.20.



Gambar 5.20. Perbandingan Akurasi Sistem antara Teknik 2D-MFCC tanpa dan dengan GA pada Jumlah Koefisien 12

Hal ini menunjukkan bahwa teknik yang dikembangkan, meskipun sudah menunjukkan adanya perbaikan dibanding teknik sebelumnya (1D-MFCC dan 2D-MFCC), namun untuk data dengan *noise* yang cukup tinggi, hasil akurasi sistem masih jauh dari yang diharapkan.

5.5. Beberapa Temuan dan Tahapan Selanjutnya

Beberapa temuan dari percobaan yang sudah dilakukan di antaranya adalah :

1. Telah berhasil dilakukan perluasan metodologi dari 1D-MFCC menjadi 2D-MFCC.
2. Sistem yang dikembangkan dengan teknik 2D-MFCC mampu mengenali pembicara dengan baik untuk data tanpa penambahan *noise*. Hal ini juga berlaku untuk teknik 1D-MFCC (>98%)

3. Juga telah berhasil dikembangkan 2D-MFCC yang pembentukan filternya menggunakan algoritme genetika. Sistem dengan 2D-MFCC yang dioptimasi dengan algoritme genetika mampu mengenali pembicara dengan baik untuk data tanpa penambahan *noise*.
4. Koefisien pertama, kedua dan ketiga adalah koefisien yang paling sensitif terhadap *noise*. Dengan mengabaikan koefisien ini, akurasi sistem naik secara nyata, khususnya untuk 2D-MFCC dengan GA. Untuk 1D-MFCC akurasi meningkat dari 41.3% menjadi 59.4%, 47% menjadi 70.5% untuk 2D-MFCC tanpa GA dan 45.5% menjadi 88.5% untuk 2D-MFCC dengan GA.
5. Namun demikian, untuk data dengan penambahan *noise* yang lebih besar, ketiga teknik masih gagal untuk mengenali dengan baik.

Dari hasil percobaan pada bab ini, terlihat salah satu permasalahannya adalah pada variabilitas nilai bispektrum menjadi tinggi saat sinyal yang masuk sudah terkontaminasi oleh *noise*. Untuk itu, maka pada percobaan berikutnya akan dilakukan proses untuk merepresentasikan nilai bispektrum sebelum masuk ke tahap *wrapping*. Teknik representasi yang akan dilakukan adalah dengan kuantisasi terhadap bispektrum. Dengan rerepresentasi ini diharapkan variabilitas bispektrum karena *noise* menjadi lebih kecil. Dua teknik kuantisasi yang akan dilakukan adalah kuantisasi skalar dan kuantisasi vektor. Pada kuantisasi skalar, pemilihan set frekuensi pada domain dilakukan secara tetap. Dengan teknik ini distribusi spasial nilai bispektrum kurang terakomodasi, sehingga kinerja sistem kemungkinan besar masih belum optimum. Kelemahan ini akan diperbaiki dengan kuantisasi vektor yang pemilihan set frekuensi pada domain didasarkan nilai relatif bispektrum. Dengan cara ini, maka distribusi spasial bispektrum terakomodasi, sehingga diharapkan sistem yang dihasilkan akan lebih baik.

BAB 6

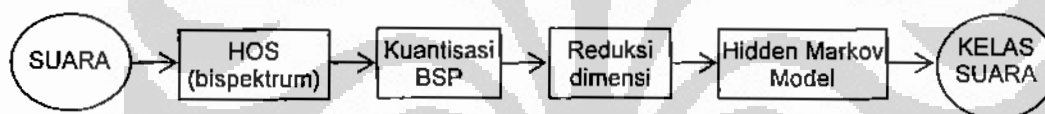
TEKNIK KUANTISASI UNTUK REPRESENTASI NILAI BISPEKTRUM SEBAGAI PENCIRI SINYAL PADA SIP DENGAN HMM SEBAGAI CLASSIFIER

Pada bagian sebelumnya telah diperlihatkan bahwa SIP yang berbasis power spektrum dengan teknik 1D-MFCC sebagai ekstraksi ciri memberikan hasil pengenalan dengan baik (>98%) untuk sinyal asli tanpa penambahan *noise*. Hal ini juga terjadi untuk SIP yang berbasis bispektrum dengan teknik 2D-MFCC sebagai pengekstraksi ciri (dengan ataupun tanpa GA). Untuk sinyal yang ditambah *noise* 20 dB, akurasi sistem dengan 1D-MFCC hanya mampu mencapai 59.6%, sedangkan dengan 2D-MFCC mencapai 70.5% (tanpa GA), dan 88.5% (dengan GA). Namun demikian, dengan bertambahnya *noise*, akurasi turun hingga di bawah 50%. Fakta ini membuktikan secara empiris bahwa bispektrum memiliki keunggulan dibanding dengan power spektrum sebagai pengenalan ciri.

Kalau diperhatikan kembali **Gambar 5.7.** pada Bab 5 sebelumnya, terlihat bahwa masih ada kemiripan bentuk bispektrum untuk sinyal asli maupun sinyal asli yang sudah mengalami penambahan *noise*. Hal ini berbeda dengan power spektrum yang sudah mengalami perubahan yang besar, bahkan untuk *noise* 20 dB. Fakta ini menunjukkan bahwa ekstraksi yang dilakukan dengan 2D-MFCC (pada Bab 5) belum secara efektif mampu menangkap ciri yang terkandung pada bispektrum. Salah satu kemungkinannya adalah pada pembacaan data bispektrum yang dibaca per titik (f_1, f_2) . Dengan cara seperti ini, maka variasi bispektrum antar titik akan memiliki pengaruh yang besar, sehingga pola spasial data bispektrum kurang terlihat, dan akibatnya hasil ekstraksi ciri lebih menyebar.

Pada Bab 6 ini akan dilakukan satu pendekatan untuk merepresentasikan nilai bispektrum yang selain mampu menangkap pola spasialnya juga mereduksi dimensi data. Teknik yang dilakukan adalah dengan melakukan kuantisasi nilai bispektrum, dan bekerja pada bidang $F_1 \times F_2$. Bagian pertama akan disajikan kuantisasi skalar, yaitu nilai bispektrum dibaca sebagai rata-rata dari sejumlah tertentu titik (f_1, f_2) pada

domain. Teknik ini diharapkan mampu meningkatkan tingkat *robustness* sistem terhadap *noise* dibanding dengan teknik sebelumnya. Namun demikian, oleh karena pemilihan sejumlah titik tersebut dilakukan secara tetap tanpa memperhatikan distribusi spasial nilai bispektrum, maka teknik kuantisasi skalar ini besar kemungkinannya kurang tegar untuk *noise* yang lebih besar. Untuk mengatasi hal ini, maka percobaan akan dilanjutkan dengan kuantisasi vektor yang dalam hal ini pemilihan jumlah titik (f_1, f_2) tidak tetap, namun mengikuti pola spasial nilai bispektrum. Pada dua teknik tersebut, nilai bispektrum di suatu titik kuantisasi akan dihitung dengan menggunakan tiga statistik, yaitu rata-rata, median, rata-rata nilai bispektrum di atas kuartil ke tiga. Bagian ke tiga adalah penentuan parameter optimal dari teknik kuantisasi vektor dengan melakukan percobaan faktorial yang melibatkan empat faktor, yaitu jenis peubah yang dikenakan proses *wrapping*, jumlah filter linear dan logaritma, lebar filter linear, lebar filter logaritma. **Gambar 6.1.** memperlihatkan alur proses pengenalan dengan ekstraksi ciri melalui teknik kuantisasi terhadap nilai bispektrum.

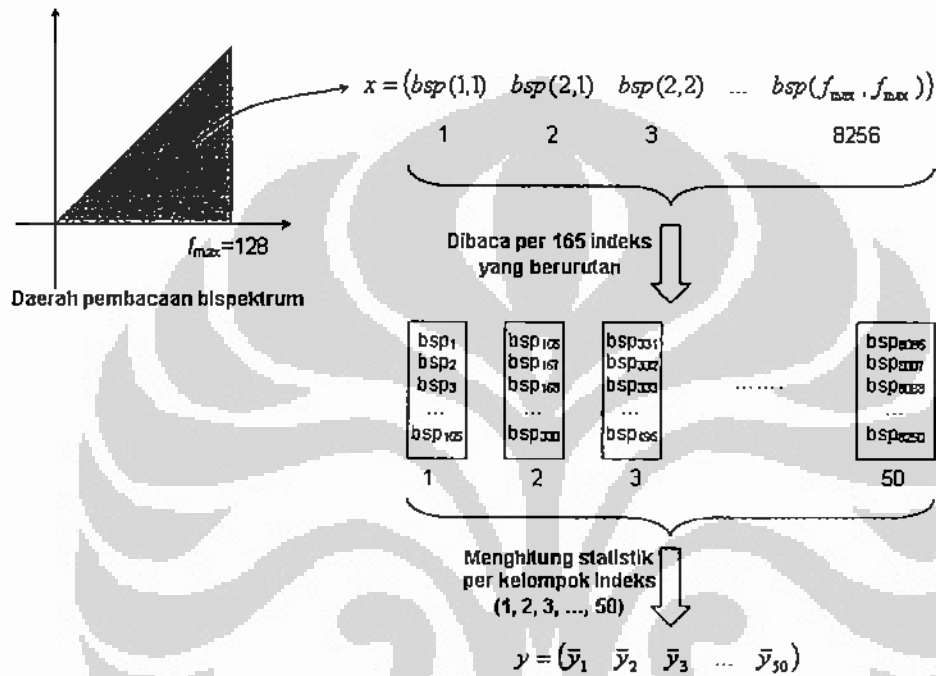


Gambar 6.1. Alur Proses Percobaan Identifikasi Pembicara

6.1. Kuantisasi Skalar

Teknik kuantisasi skalar yang dilakukan adalah seperti yang dijelaskan pada [FB98], seperti diperlihatkan pada **Gambar 6.2.** Oleh karena nilai bispektrum bersifat simetrik, maka pembacaan hanya dilakukan pada daerah segitiga dari ruang domain bispektrum. Pada absis i , pembacaan ordinatnya dilakukan dari 1 hingga i , untuk $i=1, 2, 3, \dots, f_{\max}$ dengan f_{\max} adalah frekuensi maksimum dari domain bispektrum. Oleh karena itu, daerah pembacaan bispektrum yang berbentuk segitiga tersebut diubah menjadi vektor dengan indeks $1, 2, 3, \dots, f_{\max}(f_{\max} + 1)/2$. Pada penelitian ini, nilai f_{\max} adalah 128, sehingga vektor daerah pembacaan tersebut mengandung 8256 elemen. Kuantisasi skalar dilakukan dengan membaca sejumlah

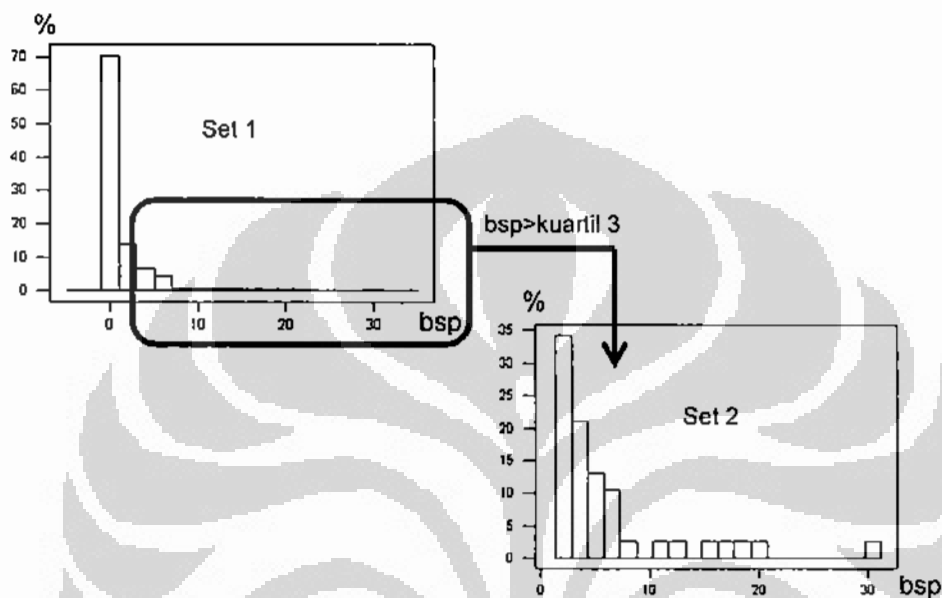
indeks yang berurutan dan mengubahnya menjadi satu nilai dengan salah satu cara, yaitu cara merata-ratakan, median, atau rata-rata setelah persentil 75%. Hasil dari kuantisasi skalar dari sebuah frame suara ini adalah satu vektor yang disebut sebagai vektor perwakilan. Banyaknya unsur pada vektor perwakilan disebut sebagai jumlah channel yang nilainya tergantung dari jumlah indeks yang dibaca pada pembacaan. Pada Gambar 6.2., setiap vektor perwakilan mengandung 50 channel.



Gambar 6.2. Kuantisasi Skalar dengan Jumlah Channel 50 untuk Bispektrum dengan $f_{max}=128$ (jumlah indeks per kelompok $[8256/50]=165$)

Gambar 6.3. menunjukkan distribusi nilai bispektrum (sebelum dan setelah persentil 75%) yang diperoleh dari kelompok indeks tertentu yang terdiri dari 165 nilai bispektrum. Terlihat bahwa distribusi nilai bispektrum menjulur ke nilai-nilai yang besar, yang berarti bahwa mayoritas nilai BSP adalah kecil, namun ada beberapa titik (f_1, f_2) tertentu mempunyai nilai BSP yang besar. Dengan kondisi seperti ini, maka pemilihan statistik untuk nilai BSP pada suatu indeks haruslah sesuai. Sebagai contoh, untuk data BSP sesuai Gambar 6.3. maka hasil kuantisasi untuk 165 nilai bispektrum tersebut adalah sebuah channel dengan nilai 1.705 jika yang dipilih adalah dengan merata-ratakan nilai bsp (set 1), 0.2581 jika yang dipergunakan median nilai

bispektrum (set 1) atau 6.0701 jika menggunakan rata-rata setelah kuartil 3 (set 2). Oleh karena itu, percobaan akan dilakukan untuk mengetahui jenis statistik yang paling optimum, sehingga dihasilkan sistem yang lebih baik. Percobaan akan dilakukan dengan menetapkan beberapa pilihan jumlah channel, yaitu sebanyak 128, 250, 400 dan 600.



Gambar 6.3. Distribusi Bispektrum Sebelum dan Setelah Persentil 75%.

Proses kuantisasi skalar tersebut diimplementasikan dengan Algoritme 7, [FB98] :

Algoritme 7. Proses Kuantisasi Skalar

```

Input : BSP[128:128];
Output: Perwakilan[1:channel];
%membaca ½ domain BSP
k=0;
for i=1:128
    for j=1:i
        k=k+1;
        MAG(k)=BSP(i,j);
    end
end
%Menghitung kuantisasi skalar
channel=p;
k=floor(length(MAG)/channel);
offset=0;
for i=1:channel
    t=0;
    for j=(1+offset):(k+offset)
        t=t+1;
        tem(t)=MAG(j);
    end
end

```

```

end
Pewakil(i)=mean(tem);
end

```

Hasil kuantisasi skalar ini adalah sebuah vektor perwakilan dengan sejumlah tertentu channel, yaitu 128, 250, 400 atau 600. Nilai dimensi sesuai jumlah channel tersebut masih terlalu besar, sehingga perlu dilakukan reduksi dimensi. Pada penelitian ini, akan dicobakan dua cara reduksi dimensi, yaitu dengan analisis komponen utama, dan yang kedua adalah dengan teknik seperti yang dilakukan pada MFCC, yaitu *wrapping* dan transformasi kosinus.

6.1.1. Reduksi Dimensi

Transformasi Komponen Utama

Analisis komponen utama merupakan salah satu analisis peubah ganda yang bertujuan untuk mereduksi dimensi data tanpa harus kehilangan informasi secara berarti. Peubah hasil transformasi ini merupakan kombinasi linear dari peubah asli, tidak berkorelasi antar sesama, serta tertata berdasar informasi yang dikandungnya.

Andaikan peubah asli adalah vektor x yang berdimensi p :

$$x = (x_1, x_2, \dots, x_p)^T,$$

maka peubah hasil transformasi adalah y yang berdimensi q :

$$y = (y_1, y_2, \dots, y_q),$$

dengan q jauh lebih kecil dibanding p . Dalam hal ini y , sebagai kombinasi linear dari x , dan dirumuskan sebagai :

$$y_1 = a_{11}x_1 + a_{12}x_2 + \dots + a_{1p}x_p = a_1^T x$$

$$y_2 = a_{21}x_1 + a_{22}x_2 + \dots + a_{2p}x_p = a_2^T x$$

.....

$$y_q = a_{q1}x_1 + a_{q2}x_2 + \dots + a_{qp}x_p = a_q^T x$$

(6.1)

Kalau matriks koragam (*covariance matrix*) dari vektor x adalah Σ , maka ragam (*variance*) y_i dirumuskan sebagai :

$$\text{ragam}(y_i) = \sigma_{y_i}^2 = \mathbf{a}_i^T \Sigma \mathbf{a}_i \quad (6.2)$$

Dari penjabaran di atas terlihat bahwa permasalahan transformasi adalah bagaimana memilih koefisien dari kombinasi linear tersebut sehingga :

$$\text{informasi } y_1 > \text{informasi } y_2 > \dots > \text{informasi } y_q$$

dengan kata lain :

$$\text{ragam}(y_1) > \text{ragam}(y_2) > \dots > \text{ragam}(y_q)$$

Dari sudut pandang geometrik, unsur-unsur dalam vektor \mathbf{a}_i merupakan komponen-komponen penyusun sumbu koordinat. Oleh karenanya dapat dipilih vektor \mathbf{a}_i yang mempunyai panjang satu dan saling ortogonal (sumbu yang ortonormal). Dengan demikian permasalahan ini menjadi masalah optimisasi dengan fungsi tujuan memaksimalkan ragam(y_i) dengan kendala $\mathbf{a}_i^T \mathbf{a}_i = 1$ dan $\text{cov}(\mathbf{a}_i, \mathbf{a}_j) = 0$ untuk $i \neq j$.

penentuan \mathbf{a}_1

Masalah optimisasi : Maksimumkan: $\text{ragam}(y_1) = \mathbf{a}_1^T \Sigma \mathbf{a}_1$

Kendala : $\mathbf{a}_1^T \mathbf{a}_1 = 1$

Melalui pengganda *lagrange*, fungsi yang dimaksimalkan adalah :

$$f(\mathbf{a}_1) = \mathbf{a}_1^T \Sigma \mathbf{a}_1 - \lambda (\mathbf{a}_1^T \mathbf{a}_1 - 1) \quad (6.3)$$

Pengoptimuman dilakukan dengan cara menurunkan fungsi f terhadap peubah-peubah yang dicari, dan diperoleh :

$$\frac{\partial f}{\partial \mathbf{a}_1} = 2\Sigma \mathbf{a}_1 - 2\lambda \mathbf{a}_1 = 0 \Leftrightarrow (\Sigma \mathbf{a}_1 - \lambda \mathbf{a}_1) = 0$$

Ini berarti \mathbf{a}_1 adalah vektor eigen dari matriks Σ dengan nilai eigen (*eigen value*) λ . Berdasar hasil di atas, maka :

$$(\Sigma \mathbf{a}_1 - \lambda \mathbf{a}_1) = 0 \Leftrightarrow \Sigma \mathbf{a}_1 = \lambda \mathbf{a}_1 \Leftrightarrow \mathbf{a}_1^T \Sigma \mathbf{a}_1 = \mathbf{a}_1^T \lambda \mathbf{a}_1 = \mathbf{a}_1^T \mathbf{a}_1 \lambda = 1 \lambda = \lambda \quad (6.4)$$

Ini berarti ragam(y_1) adalah λ yang merupakan nilai eigen matriks Σ . Karena diinginkan peubah hasil transformasi tertata berdasar 'pentingnya' maka vektor \mathbf{a}_1 adalah vektor eigen yang bersesuaian dengan nilai eigen terbesar pertama.

penentuan \mathbf{a}_2

maksimumkan : ragam(y_2) = $\mathbf{a}_2^T \Sigma \mathbf{a}_2$

$$\text{kendala} \quad : \mathbf{a}_2^T \mathbf{a}_2 = 1 \text{ dan } \mathbf{a}_1^T \mathbf{a}_2 = 0$$

Melalui pengganda lagrange, fungsi yang dimaksimumkan adalah :

$$f(\mathbf{a}_1) = \mathbf{a}_2^T \Sigma \mathbf{a}_2 - \lambda_2 (\mathbf{a}_2^T \mathbf{a}_2 - 1) - \delta (\mathbf{a}_1^T \mathbf{a}_2) \quad (6.5)$$

Setelah didiferensialkan, diperoleh :

$$\frac{\partial f}{\partial \mathbf{a}_2} = 2\Sigma \mathbf{a}_2 - 2\lambda_2 \mathbf{a}_2 - \delta \mathbf{a}_1 = 0$$

Dengan mengalikan \mathbf{a}_2^T pada ruas kiri dan kanan diperoleh :

$$2\mathbf{a}_2^T \Sigma \mathbf{a}_2 - 2\lambda_2 \mathbf{a}_2^T \mathbf{a}_2 - \delta \mathbf{a}_2^T \mathbf{a}_1 = 0 \Leftrightarrow \mathbf{a}_2^T \Sigma \mathbf{a}_2 = \lambda_2 \quad (6.6)$$

Oleh karena itu $\Sigma \mathbf{a}_2 = \lambda_2 \mathbf{a}_2$ yang berarti bahwa vektor \mathbf{a}_2 merupakan vektor eigen dari Σ yang bersesuaian dengan nilai eigen terbesar kedua, λ_2 .

Penentuan \mathbf{a}_i

Analog cara di atas, maka vektor \mathbf{a}_i merupakan vektor eigen dari matriks Σ yang bersesuaian dengan nilai eigen terbesar ke i , λ_i . Atau dengan kata lain berlaku :

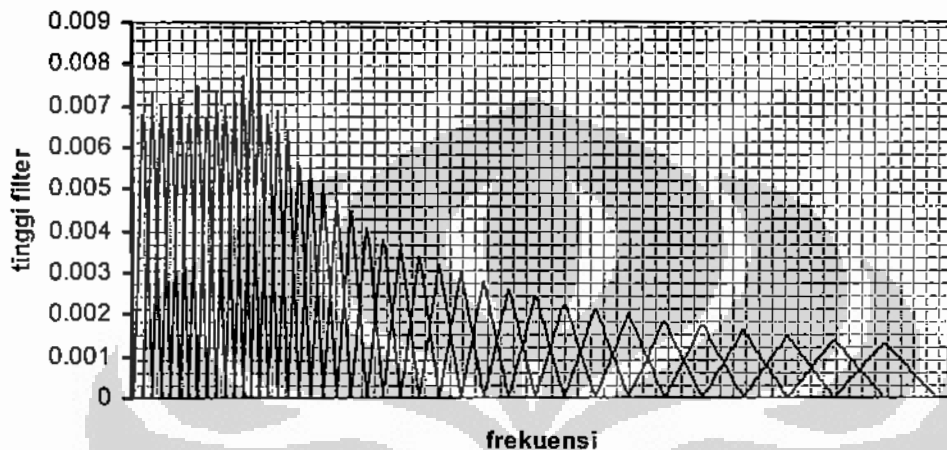
$$\Lambda = \mathbf{A}^T \Sigma \mathbf{A} \quad (6.7)$$

dengan matriks $\Lambda = \text{diag}\{\lambda_i\}$ dan $\mathbf{A} = [\mathbf{a}_1 \ \mathbf{a}_2 \ \dots \ \mathbf{a}_p]^T$

Wrapping dan Transformasi Kosinus

Proses kuantisasi bispektrum pada setiap frame input akan menghasilkan vektor perwakilan dengan dimensi sesuai jumlah channel yang diinginkan. Berikutnya pada vektor perwakilan ini dilakukan proses *wrapping* menggunakan sejumlah filter.

Filter yang dipergunakan seperti pada 1D-MFCC pada bagian sebelumnya, [Gan05]. Bentuk filter adalah segitiga yang saling tumpang tindih sebanyak 40 filter (13 filter linear dan 27 filter logaritma) dengan tinggi berbanding terbalik dengan lebar filter. **Gambar 6.4.** adalah 40 filter yang dipergunakan untuk proses *wrapping* terhadap vektor perwakilan.



Gambar 6.4. Empat Puluh Filter pada Proses *Wrapping*

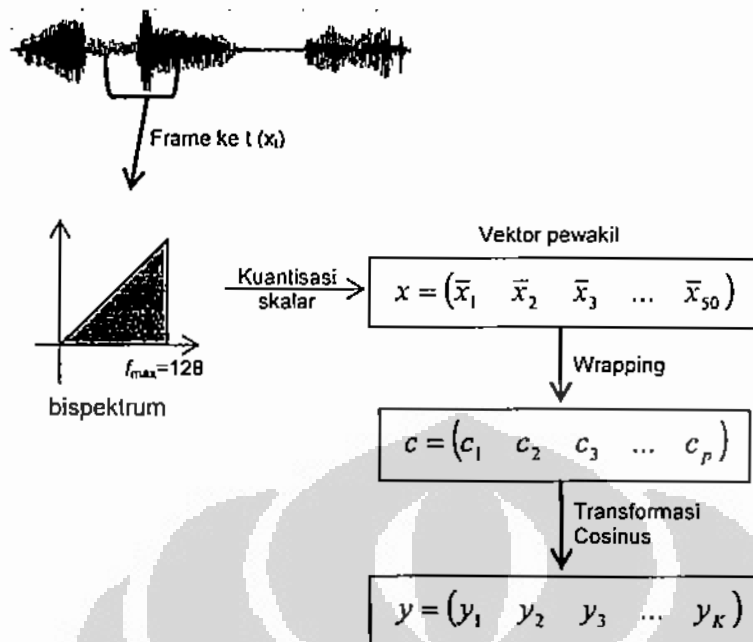
Proses *wrapping* terhadap vektor perwakilan, x , yang berdimensi p (p adalah banyaknya channel) menggunakan formula, [Gan05] :

$$c(i) = \log \left[\sum_{f=1}^p x(f) * h_i(f) \right] \text{ dengan } i \text{ adalah } 1, 2, 3, \dots, 40 \quad (6.8)$$

Dalam hal ini $h_i(f)$ adalah nilai filter ke i untuk dimensi ke f pada vektor perwakilan. Oleh karena itu, setiap vektor perwakilan yang berdimensi p akan ditransformasi menjadi vektor baru yang berdimensi 40. Berikutnya vektor hasil *wrapping* ini akan ditransformasi menggunakan transformasi kosinus dengan formula :

$$y(k) = \sum_{i=1}^{40} \cos[2(i-1)k\pi/40] \text{ dengan } k=1, 2, 3, \dots, 13 \quad (6.9)$$

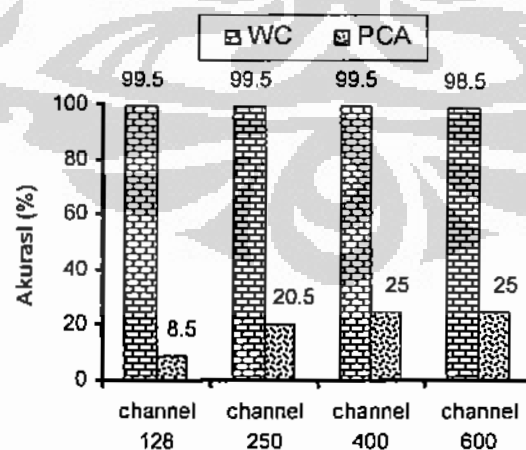
Gambar 6.5. adalah alur proses ekstraksi ciri dari sebuah frame pada sinyal masukan menjadi sebuah vektor berdimensi 13 menggunakan teknik kuantisasi dilanjutkan dengan *wrapping* dan transformasi kosinus, seperti yang dijelaskan di atas.



Gambar 6.5. Alur Proses Ekstraksi Ciri Menggunakan Teknik Skalar Kuantisasi – *Wrapping* dan Transformasi Kosinus (WC)

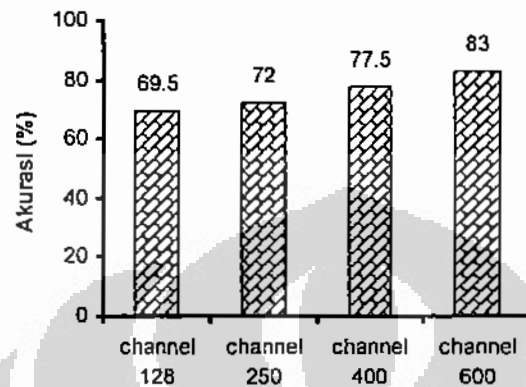
6.1.2. Hasil Percobaan

Seperti pada percobaan sebelumnya, data yang dipakai terdiri 10 pembicara dengan masing-masing mengucapkan kata "PUDESHA" sebanyak 80 kali. Dari 80 ini, 60 dipergunakan sebagai data training dan 20 sebagai data uji. Hasil pengenalan terhadap 20 data uji untuk setiap pembicara ini disajikan pada Gambar 6.6.



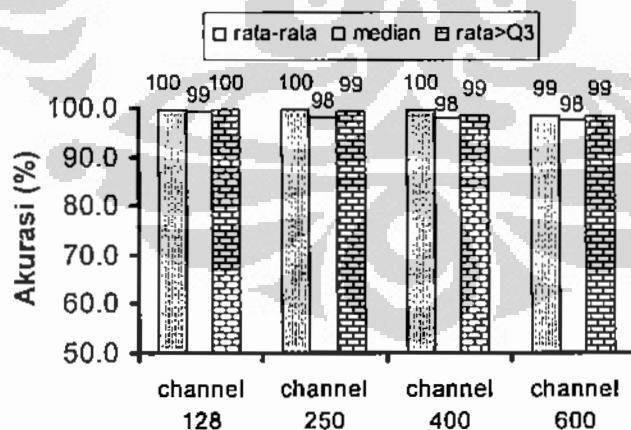
Gambar 6.6. Perbandingan Akurasi antara Teknik WC dengan PCA untuk Data Tanpa Penambahan *Noise*

Terlihat bahwa dengan melakukan *wrapping* terhadap vektor perwakilan hasil kuantisasi skalar dan dilanjutkan dengan transformasi kosinus akan memberikan sistem dengan akurasi yang jauh lebih baik (akurasi sekitar 99%) dibanding dengan menggunakan transformasi komponen utama (akurasi hanya sekitar 20%).



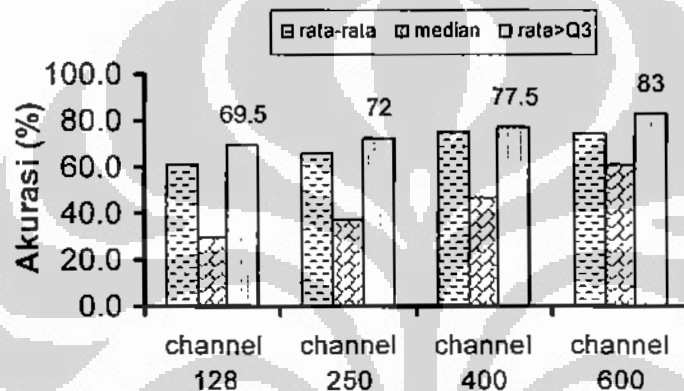
Gambar 6.7. Perbandingan Akurasi antar Berbagai Jumlah Channel dengan Kuantisasi Skalar-WC untuk Sinyal Suara yang Ditambah *Noise* 20 dB

Juga terlihat bahwa untuk sinyal yang ditambah *noise*, ternyata ada kecenderungan akurasi meningkat dengan bertambahnya channel, seperti ditunjukkan pada **Gambar 6.7.**, dengan nilai tertinggi mencapai 83%. Nilai akurasi pada **Gambar 6.7.** merupakan rata-rata dari tiga cara penentuan nilai bispektrum saat kuantisasi, yaitu statistik rata-rata, statistik median dan statistik rata-rata setelah persentil 75%.



Gambar 6.8. Perbandingan Akurasi antara Statistik Rataan, Median dan Rataan BSP di atas Persentil 75% pada Berbagai Channel untuk Sinyal Asli

Untuk melihat efektifitas tiga cara di atas perhatikan **Gambar 6.8.** yang menyajikan perbandingan akurasi sistem antara tiga jenis statistik yang dipergunakan untuk menentukan nilai bispektrum pada setiap channel. Terlihat akurasi yang diberikan semuanya lebih besar atau sama dengan 98%. Ini menunjukkan bahwa untuk sinyal tanpa penambahan *noise*, ketiga jenis statistik memberikan hasil yang baik ($\geq 98\%$) dan juga jumlah channel yang diperlukan tidak lebih dari 128. Namun demikian, untuk sinyal yang terkontaminasi dengan *noise* 20 dB, statistik Rataan bispektrum di atas persentil 75% memberikan akurasi yang lebih baik dibanding dua statistik lainnya, seperti ditunjukkan pada **Gambar 6.9.**

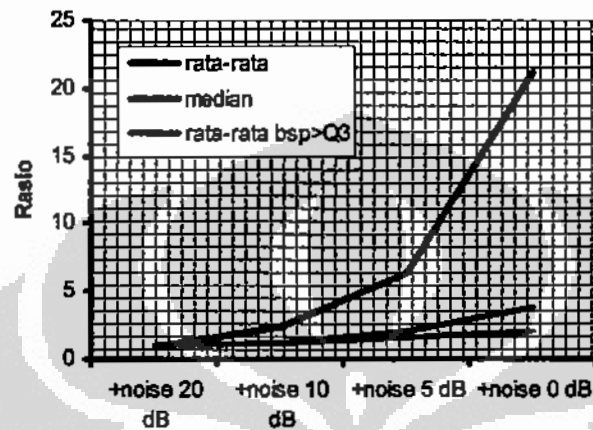


Gambar 6.9. Perbandingan Akurasi antara Statistik Rataan, Median dan Rataan BSP di atas Persentil 75% pada Berbagai Channel untuk Sinyal Asli dengan Penambahan *Noise* 20 dB

Dari ketiga statistik tersebut, median memberikan akurasi yang paling rendah di antara ke tiga statistik di atas, baik pada channel 128, 250, 400, hingga 600.

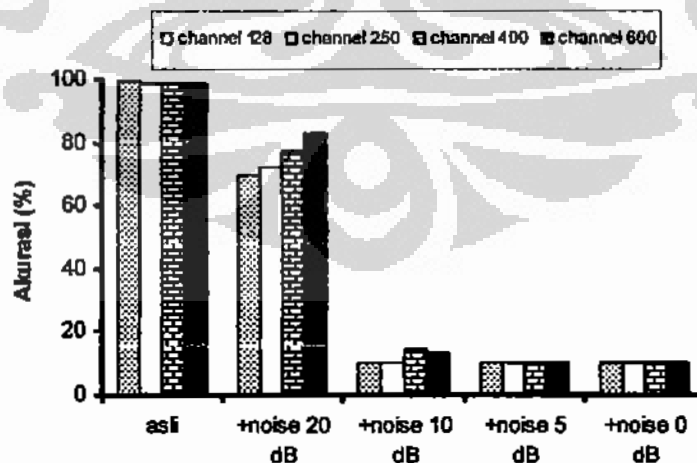
Hasil percobaan yang ditunjukkan pada **Gambar 6.8.** dan **Gambar 6.9.** memberikan bukti empiris bahwa nilai rata-rata bispektrum di atas persentil 75% bersifat lebih *robust* (kurang sensitif) terhadap pengaruh *noise*, dibandingkan dengan rata-rata maupun median, seperti ditunjukkan pada **Gambar 6.10.** **Gambar 6.10.** menampilkan perbandingan nilai statistik pada berbagai *noise* terhadap nilai statistik besaran tersebut saat tidak diberi tambahan *noise*. Pada *noise* 20 dB, ketiga statistik mempunyai nilai yang relatif sama dengan nilainya untuk sinyal tanpa penambahan *noise*, dengan nilai rasio 1.000059, 0.899651 dan 0.999753, masing-masing untuk rata-rata, median dan rata-rata bsp di atas persentil 75%. Dengan bertambahnya

noise, rasio ini meningkat. Hal ini menunjukkan bahwa *noise* yang diberikan menaikkan nilai bsp, yang pada akhirnya berpengaruh pada statistik yang dipakai. Pada penambahan *noise* sebesar 0 dB, statistik median meningkat tajam, yaitu 21.1 kali, rata-rata meningkat 3.9 kali, dan rata-rata bsp setelah Q3 relatif lebih baik, yaitu hanya meningkat menjadi 2.1 kali nilainya saat kondisi tanpa penambahan *noise*.



Gambar 6.10. Rasio Nilai Statistik pada Berbagai Penambahan *Noise* terhadap Nilainya pada Kondisi Tanpa Penambahan *Noise* untuk Ketiga Cara Penghitungan

Gambar 6.11. menyajikan akurasi sistem untuk berbagai *noise* dan berbagai channel dengan menggunakan statistik rata-rata bsp di atas Q3. Meskipun secara empiris, statistik rata-rata bsp di atas Q3 relatif tidak sensitif terhadap *noise*, namun akurasi sistem untuk noise 10 dB hingga 0 dB turun secara drastis.



Gambar 6.11. Perbandingan Akurasi antar Channel untuk Berbagai *Noise*

Hal ini menunjukkan bahwa teknik kuantisasi yang dilakukan masih perlu dikembangkan lagi dengan menentukan pusat yang sesuai, sehingga vektor perwakilan yang dihasilkan bersifat lebih *robust* terhadap adanya *noise*. Penentuan pusat channel pada kuantisasi skalar ini dilakukan dengan membagi rata dari semua sampel bispektrum yang ada. Oleh karena itu pusat channel tidak mencerminkan distribusi spasial data bispektrum.

6.1.3. Beberapa Temuan dan Tahapan Selanjutnya

Beberapa hal yang bisa diutarakan berdasar hasil percobaan identifikasi dengan kuantisasi skalar adalah :

1. Sistem yang dikembangkan dengan teknik kuantisasi skalar dikombinasikan dengan transformasi *wrapping* dan kosinus mampu melakukan pengenalan dengan akurasi yang baik untuk sinyal tanpa penambahan *noise* (>98%). Namun pada sinyal yang ditambah *noise*, akurasi sistem turun secara drastis untuk *noise* 10 dB hingga 0 dB.
2. Transformasi analisis komponen utama yang dipergunakan untuk mereduksi dimensi vektor perwakilan tidak mampu menghasilkan sistem dengan akurasi yang baik.
3. Statistik rata-rata nilai bispektrum di atas persentil 75% relatif lebih *robust* dibanding dengan statistik rata-rata maupun median.

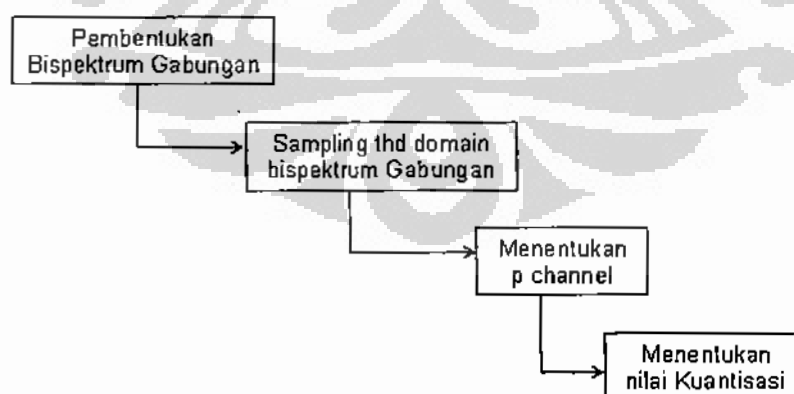
Dari hasil percobaan terlihat bahwa salah satu kelemahan kuantisasi skalar adalah pada pemilihan channel yang bersifat tetap, sehingga hal ini mengabaikan distribusi empiris dari bispektrum. Oleh karena itu, pada tahap selanjutnya akan dilakukan percobaan yang menggunakan teknik kuantisasi yang mengakomodasi distribusi empiris data bispektrum dalam domain frekuensi. Untuk itu, pemilihan titik channel akan mengacu pada besarnya nilai bispektrum. Atau dengan kata lain, kemungkinan titik (f_1, f_2) pada bidang $F_1 \times F_2$ untuk terambil sebagai pusat channel sebanding dengan nilai bispektrum di titik tersebut. Teknik kuantisasi seperti ini dikenal dengan kuantisasi vektor.

6.2. Kuantisasi Vektor

Teknik kuantisasi vektor yang dilakukan adalah seperti yang dijelaskan pada [FB98]. Seperti pada bagian sebelumnya, oleh karena nilai bispektrum bersifat simetrik, maka pembacaan hanya dilakukan pada daerah segitiga dari ruang domain bispektrum. Inti dari kuantisasi vektor adalah penentuan titik dalam domain bispektrum sebagai pusat channel. Domain dari bispektrum adalah bidang dimensi dua dengan masing-masing sumbunya adalah frekuensi. Pusat channel ditentukan sedemikian sehingga titik dengan nilai bispektrum yang tinggi akan berpeluang terpilih lebih besar sebagai pusat channel dibanding titik lain dengan nilai bispektrum yang kecil. Oleh karena itu, pusat channel akan mengumpul pada daerah (f_1, f_2) dengan nilai bispektrum yang besar dan untuk daerah dengan nilai bispektrum kecil akan memiliki sedikit pusat channel. Dengan konsep tersebut, maka pusat channel dari vektor perwakilan ditentukan dengan melakukan sampling terhadap titik-titik pada bidang (f_1, f_2) . Sampling dilakukan dengan cara mengambil sembarang titik pada domain dan untuk setiap titik ini dibangkitkan bilangan acak $[0,1]$. Jika bilangan acak ini lebih kecil dari rasio bispektrum di titik tersebut dengan bispektrum maksimum, maka titik tersebut akan terpilih sebagai penentu pusat channel. Untuk hal lainnya, maka titik tersebut diabaikan.

6.2.1. Tahapan Kuantisasi Vektor

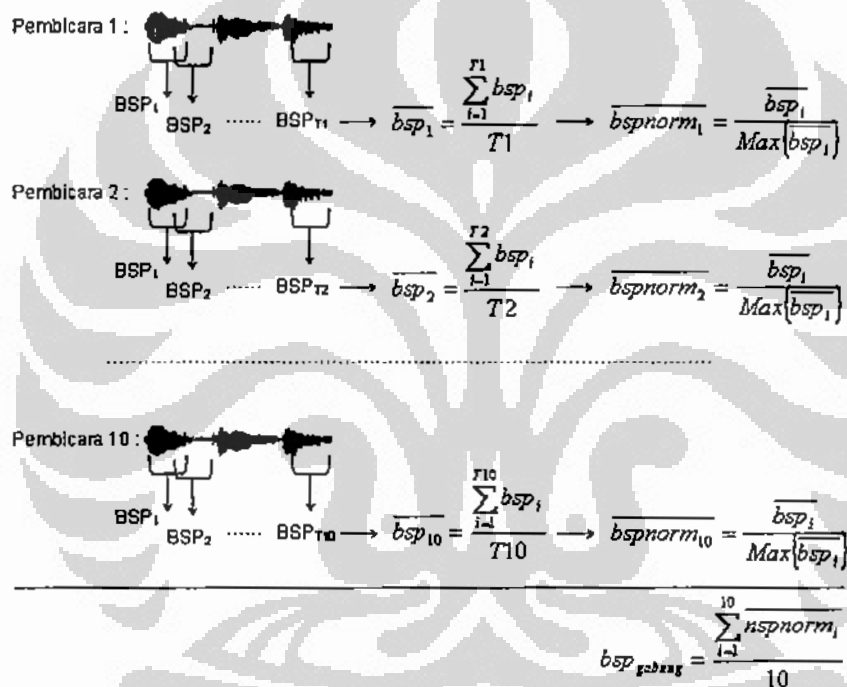
Tahapan kuantisasi vektor adalah seperti disajikan pada Gambar 6.12.



Gambar 6.12. Alur Proses Kuantisasi Vektor

Dari gambar tersebut terlihat ada empat tahap untuk melakukan kuantisasi vektor, yaitu : pembentukan bispektrum gabungan dari semua pembicara, sampling terhadap titik domain, (f_1, f_2) , dengan dasar nilai bispektrum gabungan pada titik tersebut, menentukan pusat channel dari hasil sampling sebelumnya, dan terakhir adalah menghitung nilai bispektrum pada setiap channel.

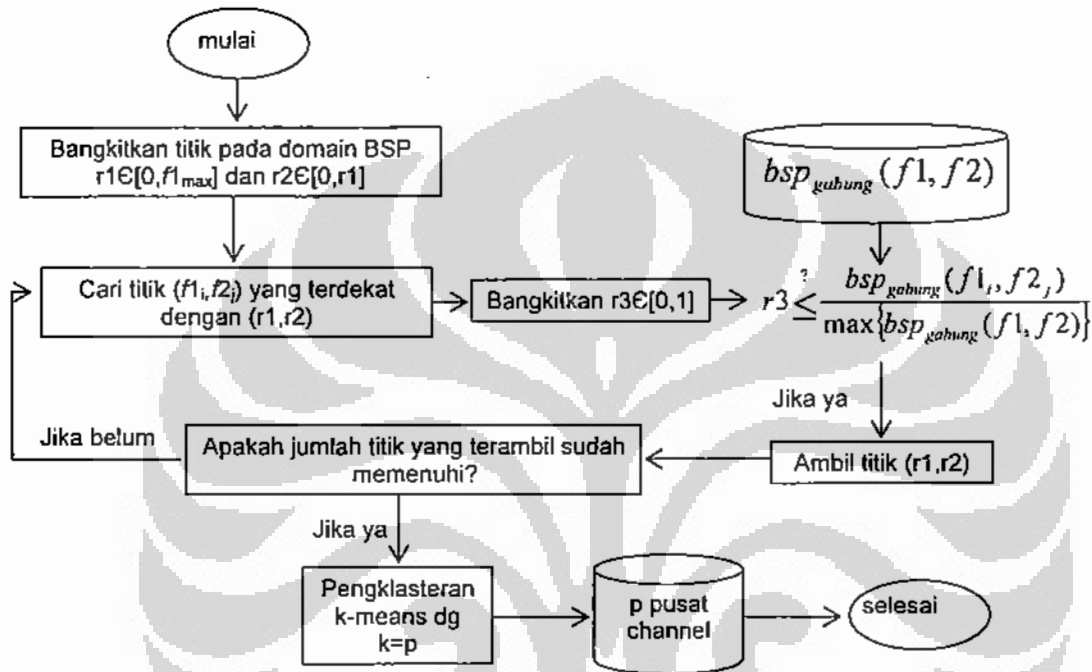
Pembentukan bispektrum gabungan dilakukan dengan menjumlahkan nilai bispektrum yang sudah dinormalisasikan dari setiap pembicara. Sementara itu, nilai bispektrum setiap pembicara diperoleh dengan merata-ratakan nilai bispektrum setiap frame dari sebuah pengucapan, seperti diilustrasikan pada Gambar 6.13.



Gambar 6.13. Alur Proses Pembentukan Bispektrum Gabungan

Setelah itu dilanjutkan dengan melakukan sampling terhadap titik-titik dalam domain bispektrum gabungan tersebut sedemikian sehingga titik dengan bispektrum besar, lebih mungkin terambil dibanding dengan titik lain yang nilai bispektrumnya lebih kecil. Pendekatan yang dilakukan adalah dengan mengambil sembarang titik pada domain secara acak dan memutuskan apakah titik yang diambil sembarang ini dipilih atau tidak berdasar nilai bispektrum terdekat, misalkan $bsp(i,j)$, dibanding dengan

bispektrum maksimumnya. Untuk itu dibangkitkan bilangan acak $r \in [0,1]$. Jika $r < \frac{bsp(i, j)}{bsp_{\max \text{ intum}}}$ maka titik tersebut dipilih sebagai pembentuk pusat channel. Untuk hal lainnya, titik tersebut diabaikan. Proses ini dilakukan hingga diperoleh sejumlah titik. **Gambar 6.14.** menyajikan blok diagram sampling titik-titik domain bispektrum.

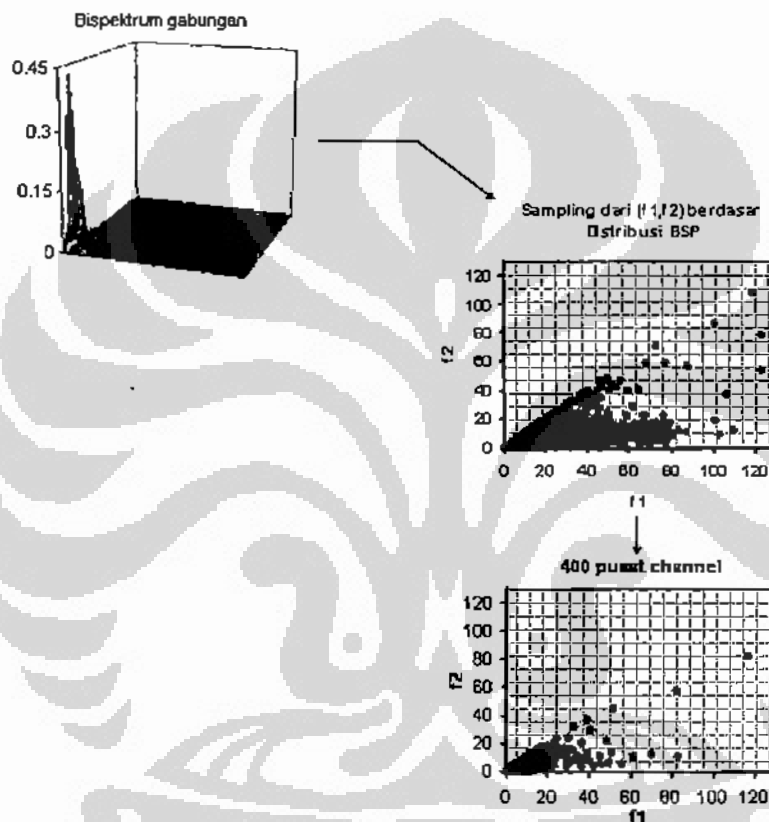


Gambar 6.14. Blok Diagram Sampling Titik-Titik Domain Bispektrum

Berdasar titik-titik yang terpilih inilah dilakukan penentuan pusat channel. Untuk penentuan pusat channel dari titik-titik terpilih tersebut dilakukan teknik pengklasteran *k-means*, dengan k adalah banyaknya channel yang diinginkan dari vektor perwakilan. **Gambar 6.15.** menyajikan ilustrasi tahapan pembentukan pusat channel, dimulai dari bispektrum gabungan, hasil sampling titik-titik domain bispektrum dan 400 pusat channel yang diperoleh dengan algoritme *k-means*.

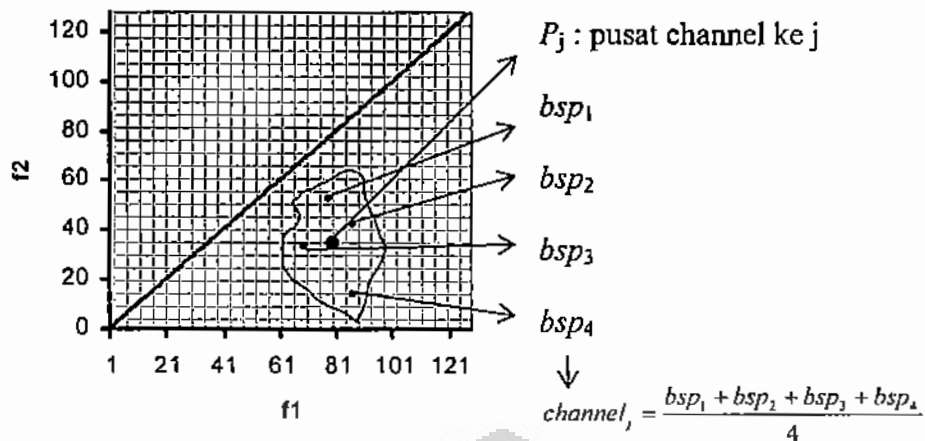
Tahap ke empat dari kuantisasi vektor adalah penentuan bispektrum pada setiap channel. Proses penentuan nilai setiap channel dari vektor perwakilan dimulai dengan memberikan label setiap titik $(f1, f2)$, dengan $f1=1, 2, 3, \dots, f_{\max}$ dengan f_{\max}

adalah frekuensi maksimum domain bispektrum. Oleh karena hanya $\frac{1}{2}$ daerah domain yang dipakai, maka $f_2=1, 2, 3, \dots, \dots, f_1$. Label titik (f_1, f_2) adalah i , jika jarak (f_1, f_2) ke pusat channel ke i adalah terkecil dibanding ke pusat channel yang lain. Setelah setiap titik diberi label sesuai channel yang ada, berikutnya adalah menentukan nilai setiap channel dari vektor perwakilan, yang menggunakan rata-rata, median, rata-rata bispektrum yang lebih besar dari kuartil tiga (persentil 75) atau rata-rata bispektrum yang lebih besar dari persentil ke 90.



Gambar 6.15. Ilustrasi Tahapan Pembentukan Pusat Channel

Gambar 6.16. memberikan ilustrasi penghitungan nilai setiap channel dari sebuah vektor perwakilan. Nilai channel ke j pada **Gambar 6.16.** diperoleh dengan merata-ratakan nilai bispektrum pada titik yang berlabel j yang berada pada daerah yang dilingkari, yang dalam hal ini ada empat titik. Untuk teknik lainnya pada prinsipnya sama, hanya nilai rata-rata tersebut diganti dengan statistik yang diinginkan.



Gambar 6.16. Ilustrasi Penentuan Nilai Channel ke j yang Mempunyai Empat Titik Berlabel j dengan Statistik Rataan

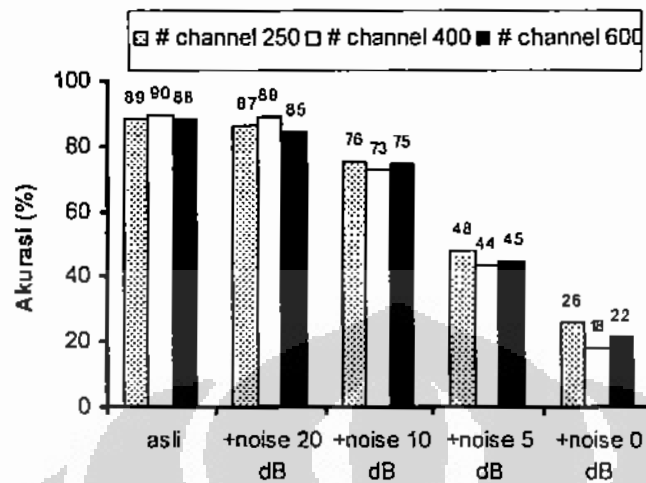
Setelah diperoleh vektor perwakilan (yang terdiri dari p channel) dari suatu frame, maka berikutnya adalah mereduksi dimensi vektor perwakilan menggunakan transformasi WC seperti yang sudah dijelaskan pada bagian 6.1. Algoritme lengkap proses kuantisasi vektor disajikan pada Lampiran 6.

6.2.2. Hasil Percobaan

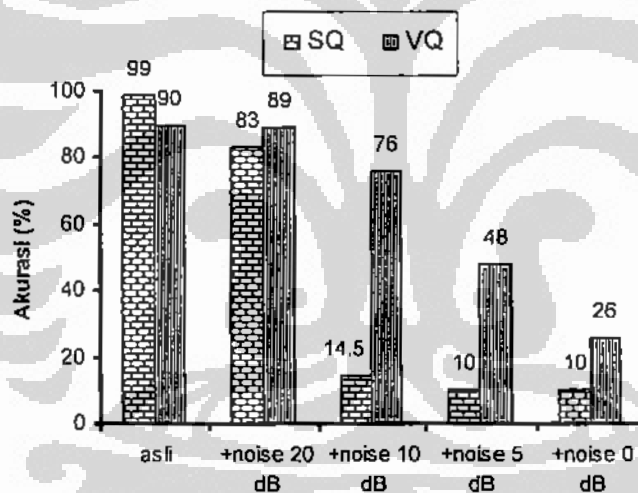
Data suara yang dipergunakan untuk training dan pengujian pada bab ini adalah sama dengan yang dipergunakan pada Bab 6 sebelumnya. **Gambar 6.17.** menyajikan perbandingan akurasi sistem dengan kuantisasi vektor menggunakan jumlah channel 250, 400 dan 600 yang dilanjutkan wrapping dan transformasi kosinus untuk reduksi dimensi channel. Dari **Gambar 6.17.** terlihat bahwa teknik kuantisasi vektor dapat diterapkan sebagai ekstraksi ciri dan memberikan akurasi sistem rata-rata sebesar 90%, 89%, 75% untuk sinyal asli, sinyal asli ditambah *noise* 20 dB dan sinyal asli ditambah *noise* 10 dB. Untuk level *noise* 5 dB dan 0 dB, sistem telah gagal melakukan pengenalan dengan baik. Dari gambar tersebut juga terlihat bahwa jumlah channel kurang memberikan perbedaan pengaruh yang berarti.

Kalau dibandingkan dengan kuantisasi skalar, efektifitas teknik kuantisasi vektor ini untuk sinyal asli lebih rendah. Hal ini ditunjukkan oleh akurasi sistem yang hanya 90%, sementara pada kuantisasi skalar mencapai 99%. Namun demikian

untuk sinyal yang ditambah *noise*, terlihat teknik ini memberikan hasil yang lebih baik, seperti ditunjukkan pada **Gambar 6.18**.



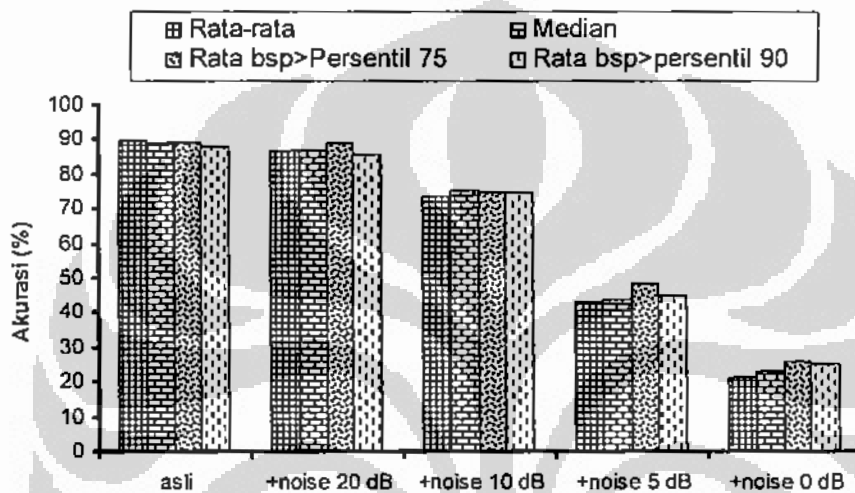
Gambar 6.17. Perbandingan Akurasi SIP dengan Kuantisasi Vektor dan Transformasi *Wrapping* dan Kosinus



Gambar 6.18. Perbandingan Akurasi SIP antara Kuantisasi Vektor dengan Kuantisasi Skalar

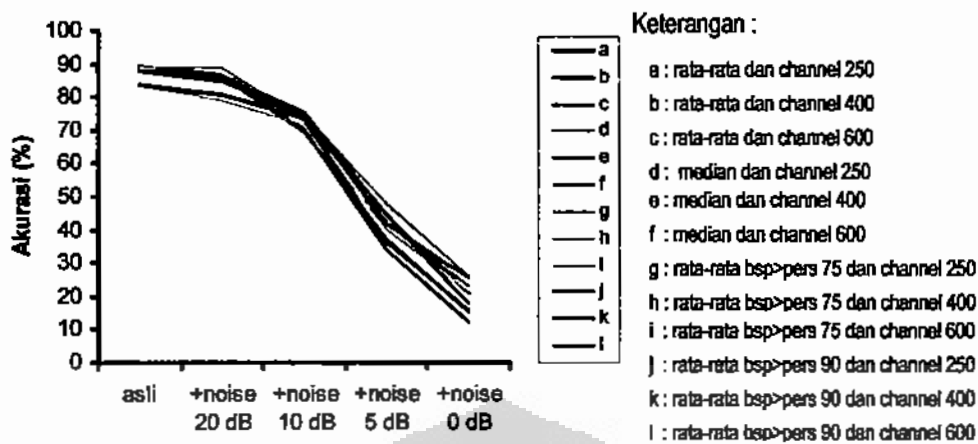
Dari gambar tersebut juga terlihat bahwa sistem dengan kuantisasi vektor bersifat lebih *robust* terhadap *noise* dibanding sistem dengan kuantisasi skalar. Hal ini ditunjukkan bahwa akurasi sistem dengan kuantisasi vektor pada *noise* 20 dB, 10 dB, 5 dB dan 0 dB yang berturut-turut 89%, 76%, 48% dan 26%, berada di atas akurasi sistem dengan kuantisasi skalar, yang hanya 83%, 14.5%, 10% dan 10%.

Gambar 6.19. menyajikan perbandingan akurasi antara berbagai jenis statistik yang dipergunakan untuk menghitung nilai bispektrum pada setiap channel. Dari gambar tersebut terlihat bahwa untuk sinyal asli, ke empat jenis statistik tidak memberikan perbedaan akurasi yang berarti. Namun untuk data ber-*noise*, terlihat kecenderungan bahwa statistik rata-rata nilai bispektrum di atas kuartil ke tiga (atau di atas persentil 75) secara relatif lebih baik dibanding dengan tiga statistik lainnya, yaitu rata-rata, median, dan rata-rata setelah persentil 90%.



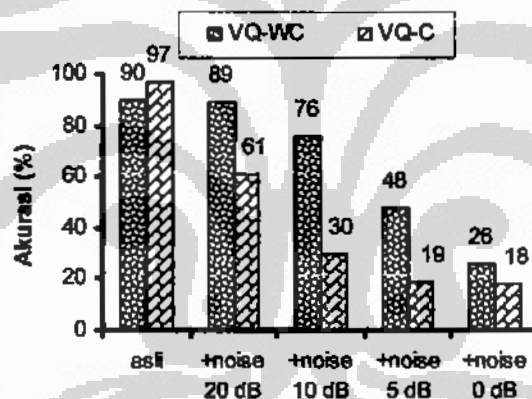
Gambar 6.19. Perbandingan Akurasi antar Berbagai Jenis Statistik pada Sistem dengan Kuantisasi Vektor dan Transformasi WC

Untuk membandingkan akurasi sistem dikaitkan dengan jumlah channel dan jenis statistik yang dipergunakan, perhatikan **Gambar 6.20**. Terlihat bahwa sistem dengan jumlah channel 400 dan statistik rata-rata nilai bispektrum di atas persentil 75% memberikan akurasi lebih layak dibanding dengan kombinasi lainnya. Dari **Gambar 6.20**, terlihat bahwa sistem masih mampu bertahan hingga level *noise* 10 dB, dengan akurasi di atas 70%. Sementara itu, dengan teknik lainnya, pada *noise* 10 dB, akurasi sistem jauh berada di bawah 50%. Namun demikian, untuk sinyal dengan kualitas yang relatif baik, yaitu sinyal asli, akurasi sistem hanya berkisar sekitar 90%. Nilai ini jauh di bawah sistem sebelumnya, yaitu yang semuanya mencapai hampir 99%.



Gambar 6.20. Perbandingan Akurasi SIP antara Berbagai Jenis Statistik dan Jumlah Channel pada Sistem dengan Kuantisasi Vektor dan Transformasi WC

Untuk melihat efektifitas transformasi *wrapping* pada sistem dengan kuantisasi vektor, perhatikan **Gambar 6.21**.

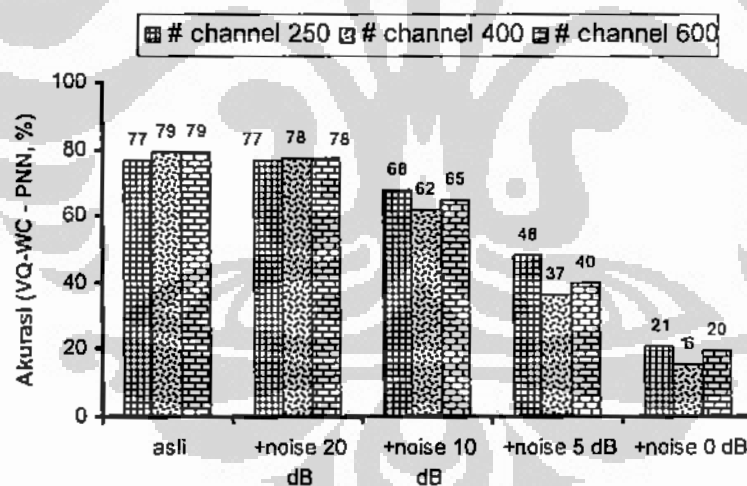


Gambar 6.21. Perbandingan Efektifitas Transformasi *Wrapping*-Kosinus dengan Kosinus dalam Menangani Masalah *Noise* pada SIP

Dari gambar tersebut, untuk sinyal asli tanpa menggunakan proses *wrapping*, terlihat sistem mampu mengenali dengan akurasi 97%. Nilai ini lebih tinggi dibanding dengan proses *wrapping* yang hanya 90%. Namun untuk sinyal ber-*noise*, teknik *wrapping* mampu mempertahankan akurasi untuk level 20dB, yaitu tetap 89%. Sedangkan tanpa *wrapping*, akurasi langsung drop menjadi sekitar 60%. Untuk *noise* yang lebih besar lagi, yaitu 10 dB, 5 dB dan 0dB, teknik reduksi dimensi tanpa *wrapping* menjadi sekitar 30%, 19% dan 18%. Sementara itu teknik *wrapping* masih

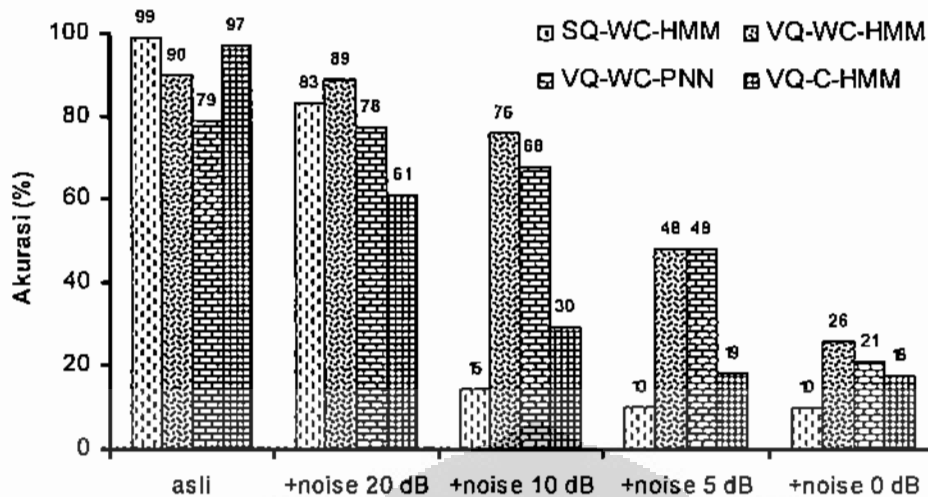
mampu mempertahankan pada level 76%, 48% dan 26%. Hal ini menunjukkan bahwa proses *wrapping* mampu mempertahankan *robustness* sistem terhadap noise.

Untuk membandingkan efektifitas teknik HMM, maka pada bagian ini juga dilakukan percobaan identifikasi pembicara dengan menggunakan *Probabilistic Neural Network* (PNN) sebagai pengenalan pola pada SIP. Akurasi sistem yang dikembangkan menggunakan bispektrum sebagai pengenalan ciri, kuantisasi vektor dan transformasi WC sebagai pereduksi dimensi dan PNN sebagai pengenalan pola disajikan pada **Gambar 6.22**. Dari gambar tersebut terlihat bahwa akurasi sistem memang masih di bawah 80% untuk sinyal asli. Nilai ini sekitar 10% lebih rendah jika dibandingkan dengan sistem yang menggunakan HMM sebagai *classifier* dengan teknik ekstraksi ciri yang sama. Pada gambar tersebut terlihat satu hal yang menarik, yaitu bahwa turunnya akurasi seiring dengan naiknya level *noise* tidak langsung drop, namun bersifat gradual. Hal ini sekali lagi menunjukkan bahwa teknik kuantisasi vektor yang dikembangkan mampu menghasilkan sistem yang relatif lebih *robust* terhadap *noise*.



Gambar 6.22. Akurasi SIP Berbasis Data Bispektrum dan Kuantisasi Vektor dengan Transformasi *Wrapping*-Kosinus sebagai Pereduksi Dimensi dan PNN sebagai *Classifier*

Untuk lebih menunjang pernyataan tersebut, perhatikan **Gambar 6.23**. dan data lengkapnya disajikan pada **Tabel 6.1**.



Gambar 6.23. Perbandingan Akurasi SIP Berbasis Data Bispektrum dengan Beberapa Metode yang telah Dikembangkan

Dari gambar tersebut terlihat bahwa kuantisasi skalar memberikan akurasi yang sangat baik untuk data sinyal tanpa penambahan *noise* (99%). Sementara metode dengan kuantisasi vektor bervariasi dari 79% hingga 97%. Untuk data dengan *noise* 20 dB, akurasi sistem dengan kuantisasi vektor relatif bertahan, khususnya yang menggunakan HMM sebagai *classifier*. Jika nilai *noise* bertambah, akurasi sistem dengan kuantisasi skalar langsung drop. Sedangkan untuk sistem dengan kuantisasi vektor penurunan bersifat gradual. Dari aspek pengenalan pola, terlihat bahwa sistem yang dikembangkan dengan kuantisasi vektor dan PNN sebagai *classifier* mengalami penurunan akurasi yang berarti (sekitar 10%) dibanding sistem yang menggunakan HMM. Ini menunjukkan bahwa HMM sebagai *classifier* relatif lebih baik dibanding PNN.

Tabel 6.1. Perbandingan Persentase Akurasi Sistem antara Metode Kuantisasi Skalar dan Vektor dengan *Classifier* HMM dan PNN

	asli	+noise 20 dB	+noise 10 dB	+noise 5 dB	+noise 0 dB
SQ-WC-HMM	99	83	14.5	10	10
VQ-WC-HMM	90	89	76	48	26
VQ-WC-PNN	79	77.5	68	48	21
VQ-C-HMM	97	61	29.5	18.5	18

Dari Tabel 6.1. juga bisa disebutkan bahwa transformasi kosinus (tanpa melalui proses *wrapping*) akan memberikan hasil yang baik untuk data asli dengan akurasi mencapai 97%. Untuk data dengan penambahan *noise*, akurasi sistem turun secara drastis, khususnya untuk *noise* lebih besar dari 20 dB. Dari fakta ini bisa dikatakan bahwa teknik kuantisasi vektor digabungkan dengan proses *wrapping* akan menghasilkan sistem yang lebih *robust* terhadap *noise* dibanding teknik lainnya.

6.2.3. Beberapa Temuan dan Tahapan Selanjutnya

Beberapa temuan yang bisa disampaikan terkait hasil percobaan dengan teknik kuantisasi vektor yang telah dilakukan adalah :

1. Teknik kuantisasi vektor digabungkan dengan transformasi *wrapping* dan kosinus (WC) dapat dipergunakan sebagai pengekstraksi ciri pada sistem pengenalan pembicara, dengan akurasi sebesar 90% untuk data sinyal tanpa penambahan *noise*. Penurunan akurasi dengan nilai *noise* yang makin tinggi bersifat gradual dan tidak setajam pada teknik kuantisasi skalar.
2. Jumlah channel yang sesuai pada teknik kuantisasi vektor adalah 400 dengan nilai bispektrum dihitung menggunakan rata-rata nilai bispektrum di atas persentil 75%.
3. HMM sebagai pengenal pola lebih baik dibandingkan dengan PNN. Untuk data tanpa penambahan *noise*, HMM mampu mengenali sekitar 10% lebih tinggi dibanding PNN.
4. Sistem yang dikembangkan dengan kuantisasi vektor digabungkan dengan transformasi *wrapping* dan kosinus dengan HMM sebagai pengenal pola menghasilkan sistem yang bersifat lebih *robust*. Namun akurasi yang dihasilkan untuk data sinyal tanpa penambahan *noise* masih belum terlalu baik, yaitu 90%, sehingga masih perlu dilakukan optimisasi terhadap parameter pada transformasi WC untuk meningkatkan akurasi.

Berdasar hasil percobaan yang dilakukan, terlihat bahwa pada kuantisasi vektor terhadap nilai bispektrum dilanjutkan dengan transformasi *wrapping* dan kosinus akan melibatkan beberapa parameter, seperti lebar dan jumlah filter (linear dan logaritma), dan posisi titik saat pemetaan ke sumbu frekuensi untuk dilakukan *wrapping* (apakah diurut biasa sesuai posisinya terhadap pusat sumbu, atau sesuai dengan jaraknya terhadap pusat sumbu). Oleh karena itu, pada tahap selanjutnya akan dilakukan percobaan faktorial yang melibatkan variabel-variabel tersebut pada beberapa level. Hal ini ditujukan untuk mengetahui posisi atau kombinasi parameter yang optimum yang mampu menghasilkan sistem dengan akurasi yang lebih baik.

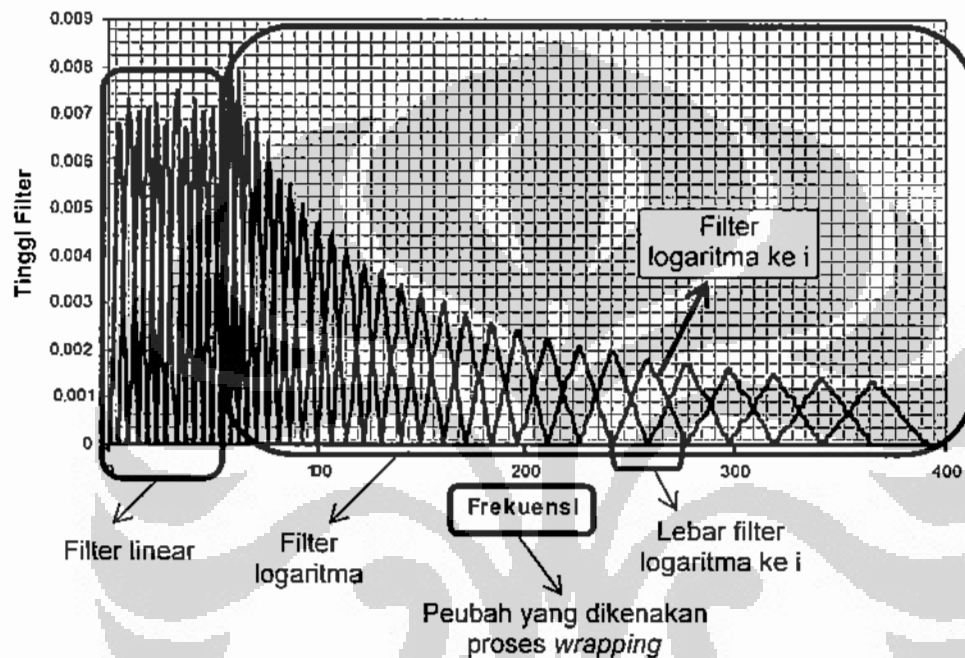
6.3. Kaji Lanjut Parameter pada Teknik Kuantisasi Vektor

Pada bagian sebelumnya telah ditunjukkan bahwa sistem dengan bispektrum sebagai penentu ciri sinyal dilanjutkan kuantisasi vektor dan transformasi WC sebagai pengestraksi ciri akan bersifat lebih *robust* terhadap *noise* dibandingkan metode lainnya. Namun demikian, akurasi yang diberikan masih belum begitu tinggi, yaitu 90% untuk sinyal asli. Juga dari hasil pada bab-bab sebelumnya dapat disebutkan bahwa proses *wrapping* adalah sangat berpengaruh pada tingkat *robustness* sistem terhadap *noise*. Pada proses *wrapping* melibatkan beberapa faktor, seperti lebar filter linear, lebar filter logaritma, serta perbandingan jumlah filter linear dan logaritma. Selain itu juga pada variabel yang dikenakan proses *wrapping*. Oleh karena itu, pada bagian ini akan dilakukan percobaan untuk mengkaji lebih lanjut peran setiap parameter dalam menentukan akurasi sistem serta tingkat *robustness* sistem terhadap *noise*.

Dari hasil sebelumnya juga sudah diperlihatkan bahwa HMM sebagai pengenalan pola memberikan hasil yang lebih baik dibanding PNN. Oleh karena itu percobaan identifikasi pembicara pada bab ini akan menggunakan HMM sebagai pengenalan pola dengan jumlah state sebanyak 3. Pemilihan ini didasarkan pada fakta sebelumnya bahwa jumlah state tidak terlalu memberikan perbedaan yang berarti, seperti yang ditunjukkan pada Bab 4, yaitu berkisar dari 98.8% hingga 100% untuk data asli.

6.3.1. Parameter Transformasi WC

Pada bagian ini akan dilakukan serangkaian percobaan untuk mengkaji lebih lanjut mengenai faktor-faktor atau parameter-parameter pada transformasi WC, juga pada jenis peubah yang akan dikenakan *wrapping*. Untuk memberikan ilustrasi mengenai faktor-faktor tersebut, perhatikan **Gambar 6.24.** yang menyajikan sebuah filter beserta parameter-parameternya.



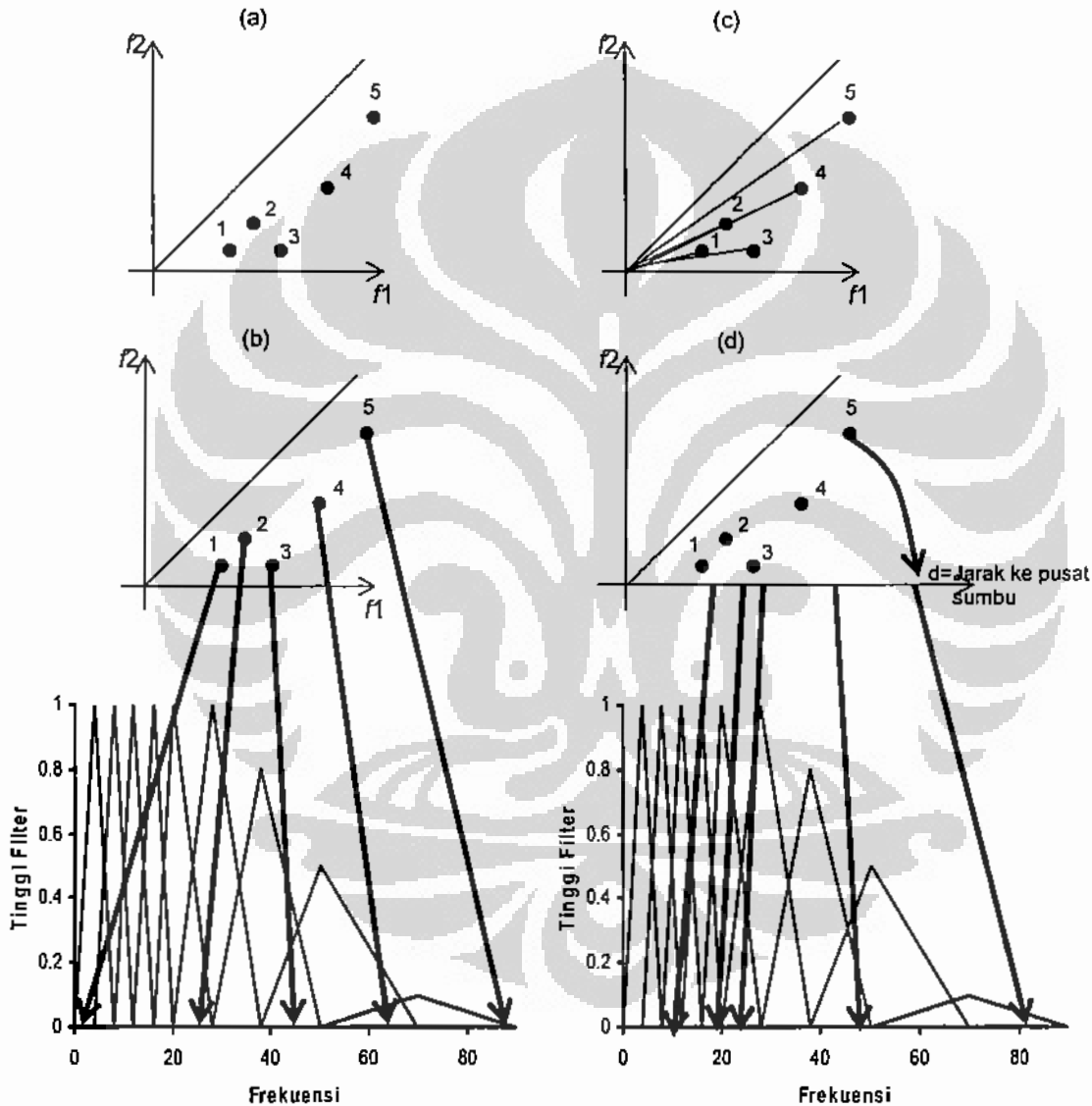
Gambar 6.24. Filter pada Proses *Wrapping* Beserta Parameternya

Untuk memudahkan dalam penyebutan, berikut adalah notasi yang dipakai serta level untuk setiap faktor.

- W : Jenis peubah yang dikenakan proses *wrapping*, ada 2 level (w1:posisi channel, w2:jarak titik channel ke pusat sumbu)
- D : Lebar filter linear, ada 3 level (d1:25, d2:50, dan d3:75)
- G : lebar filter logaritma, ada 3 level (g1:1.06, g2:1.07, dan g3:1.08)
- R : Proporsi jumlah filter linear dan filter logaritma, ada 3 level (r1:10/40, r2:20/30, dan r3:30/20)

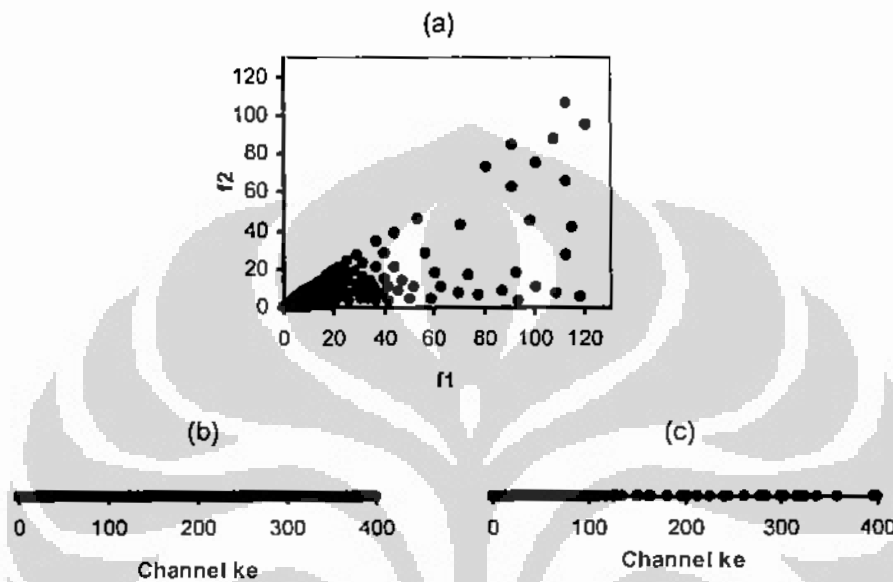
Peubah yang Dikenakan Proses *Wrapping* (W)

Oleh karena titik channel berada pada bidang (f_1, f_2) , maka jenis peubah yang dikenakan proses *wrapping* yang dicobakan adalah posisi titik channel dan yang kedua adalah jarak titik channel terhadap pusat sumbu. **Gambar 6.25.** memberikan ilustrasi mengenai peubah yang dikenakan proses *wrapping* dikaitkan dengan filter yang dipakai.



Gambar 6.25. Ilustrasi Peubah yang Dikenakan Proses *Wrapping*, (a) 5 titik channel, (b) *wrapping* pada posisi channel, w_1 , (c) 5 titik channel beserta jaraknya terhadap pusat sumbu, dan (d) *wrapping* pada jarak channel ke pusat sumbu, w_2

Pada w_1 , pusat channel dipetakan ke sumbu frekuensi secara merata sesuai urutan jaraknya terhadap pusat. Sedangkan pada w_2 , pusat channel dipetakan ke sumbu frekuensi sesuai jaraknya terhadap pusat sumbu. Berdasarkan proses tersebut, maka tebaran titik dari 400 pusat channel pada proses wrapping adalah seperti disajikan pada **Gambar 6.26**.



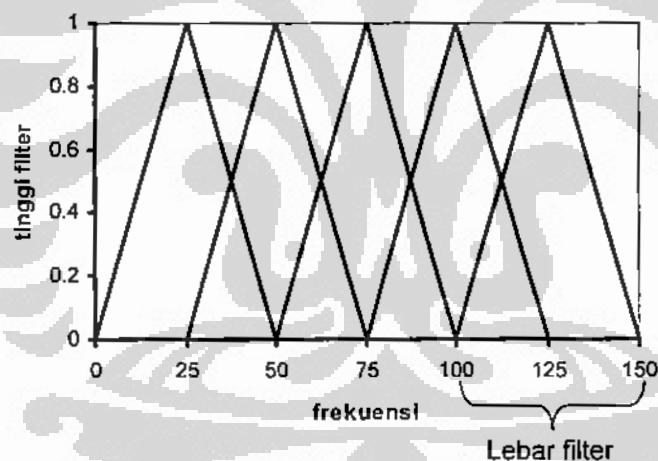
Gambar 6.26. Perbedaan Posisi Titik Peubah yang Dikenakan *Wrapping* pada Sumbu Frekuensi pada Kuantisasi Vektor dengan 400 Channel, (a) 400 titik channel, (b) posisi titik yang di *wrapping* sesuai urutan channel dan (c) posisi titik yang di-*wrapping* sesuai jarak

Dengan *wrapping* sesuai urutan akan menata channel pada satu sumbu dengan jarak yang sama (**Gambar 6.26b**). Sedangkan *wrapping* sesuai jarak akan menata titik channel sesuai kerapatannya (**Gambar 6.26c**).

Peubah Lebar Filter Linear (D)

Jumlah filter yang akan dipakai pada percobaan ini adalah 50 yang terdiri dari filter linear dan filter logaritma. Yang dimaksud dengan filter linear adalah filter dengan lebar yang sama. Sedangkan filter logaritma adalah filter dengan lebar filter pada sumbu logaritma adalah sama. Pemilihan jumlah 50 ini didasarkan dari hasil empiris pada Bab 4 yang menggunakan filter sebanyak 40 dan memberikan akurasi di

atas 99% untuk sinyal asli. Jumlah filter 40 tersebut terdiri dari 13 filter linear dan 27 filter logaritma. Filter inilah yang dipakai juga pada percobaan dengan kuantisasi skalar dan vektor sebelumnya. Oleh karena hasil maksimum untuk sinyal asli pada percobaan sebelumnya dengan kuantisasi vektor belum mencapai di atas 90 %, maka dengan alasan inilah pada bagian ini dicobakan jumlah filter sebanyak 50. Dari 50 filter ini, pada bagian frekuensi yang kecil digunakan filter linear dan pada frekuensi yang besar digunakan filter logaritma. Lebar filter linear akan menentukan seberapa banyak titik-titik channel yang akan tercakup dalam filter linear ini, yang pada akhirnya akan menentukan koefisien akhir dari proses ekstraksi ciri yang dilakukan. Nilai lebar filter linear yang dipilih adalah 25, 50 dan 75. Pemilihan 25 ini didasarkan bahwa dengan lebar 25, maka akan melingkupi frekuensi kurang dari 1000 Hz. Sedangkan nilai 75 akan melingkupi sebagian besar nilai frekuensi di atas 1000 Hz, untuk jumlah filter 10 sampai dengan 40. **Gambar 6.27.** berikut adalah contoh 5 filter linear dengan lebar 50.



Gambar 6.27. Contoh Lima Filter Linear dengan Lebar 50

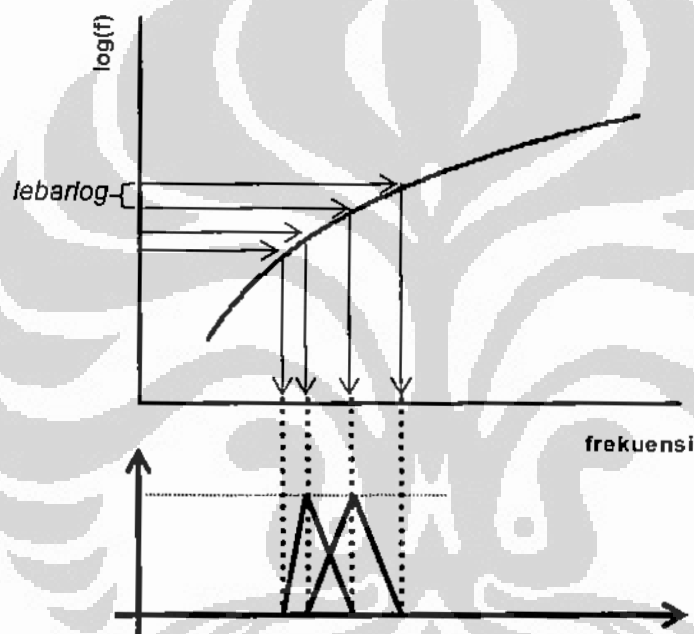
Peubah Lebar Filter Logaritma (G)

Filter logaritma dibentuk sedemikian sehingga lebar filter pada domain frekuensi jika ditransformasi ke logaritma akan memiliki nilai yang tetap. Nilai yang tetap pada domain logaritma ini dapat diperoleh dengan menetapkan nilai tertentu yang akan dikonversikan sebagai pusat filter logaritma pada domain frekuensi. Nilai

tertentu ini berikutnya disebut sebagai *lebarlog*, dan pada percobaan ini dipilih sebesar 1.06, 1.07 dan 1.08. Pemilihan nilai ini didasarkan pada argumen yang disajikan pada [Gan05], yang mendapatkan nilai terbaik sekitar 1.071. Tahapan untuk memperoleh pusat-pusat filter logaritma adalah sebagai berikut :

1. $F1 = \text{lebar filter linear} \times \text{jumlah filter linear}$
2. $A = \log(F1)$
3. untuk $i = 1 : \text{jumlah filter logaritma}$
 - 1.1. $K = A + i \times \text{lebarlog}$
 - 1.2. pusat filter ke i adalah $\text{antilog}(K)$

Gambar 6.28. memberikan ilustrasi penentuan filter logaritma.



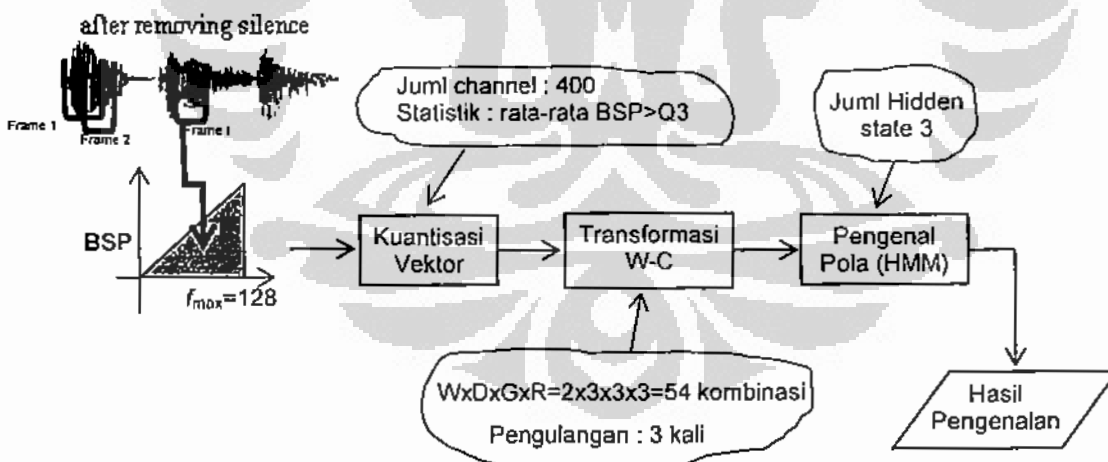
Gambar 6.28. Ilustrasi Penentuan Filter Logaritma

Tinggi filter ditentukan sedemikian sehingga luas filter adalah 1, [Gan05], yaitu sebesar 2 dibagi lebar filter, atau $2/(\text{batas kanan}-\text{batas kiri})$.

Selain ke tiga faktor di atas, faktor lain yang diperhatikan adalah jumlah filter linear dan jumlah filter logaritma. Seperti telah disebutkan di atas bahwa jumlah total filter adalah 50. Proporsi jumlah filter linear dengan filter logaritma yang dipilih adalah 10:40, 20:30 dan 30:20.

6.3.2. Hasil Percobaan

Untuk memperoleh gambaran yang lebih lengkap, maka akan digunakan rancangan percobaan faktorial dan hasilnya dianalisis menggunakan tabel sidik ragam. Percobaan yang dilakukan melibatkan empat faktor yang akan diamati, yaitu jenis peubah yang dikenakan proses *wrapping*, W (2 level); lebar filter linear, D (3 level); lebar filter logaritma, G (3 level); dan proporsi jumlah filter linear dan filter logaritma, R (ada 3 level). Oleh karena itu jumlah perlakuan yang dicobakan adalah $2 \times 3 \times 3 \times 3$, yaitu 54. Setiap perlakuan diulang 3 kali, sehingga banyaknya percobaan yang dilakukan adalah 54×3 , yaitu 162 kali. Pengamatan dilakukan terhadap akurasi sistem yang dihasilkan yang menggunakan HMM sebagai pengenalan pola dengan jumlah hidden state sebanyak 3, dilakukan pada sinyal asli tanpa penambahan *noise*, dan sinyal asli dengan penambahan *noise* sebesar 20 dB, 10 dB, 5 dB, dan 0 dB. Sesuai dengan percobaan-percobaan pada bab-bab sebelumnya, identifikasi yang dilakukan melibatkan 10 pembicara yang masing-masing mengucapkan kata "PUDESHA" sebanyak 80 kali. Dari 80 kali pengucapan ini, 60 ucapan sebagai data training dan 20 ucapan sebagai data uji. Gambar 6.29. menyajikan blok diagram percobaan yang akan dilakukan.



Gambar 6.29. Blok Diagram Rancangan Percobaan untuk Identifikasi Pembicara

Sesuai dengan hasil sebelumnya, maka pada bagian ini percobaan akan menggunakan kuantisasi vektor sebagai ekstraksi ciri, dengan jumlah channel

sebanyak 400. Untuk menghitung nilai bispektrum pada setiap titik channel, digunakan statistik rata-rata nilai bispektrum setelah persentil 75%.

Hasil percobaan secara lengkap disajikan pada Lampiran 7. Bagian berikut akan membahas mulai dari sinyal asli dan diikuti pembahasan pada sinyal ber-*noise* 20 dB dan sinyal ber-*noise* 10 dB, dan ditutup dengan perbandingan akurasi pada seluruh jenis sinyal (asli, serta sinyal ber-*noise* 20 dB, 10 dB, 5 dB dan 0 dB). Pemisahan analisis sesuai level *noise* ini didasarkan fakta bahwa pengaruh faktor-faktor dalam percobaan adalah berbeda untuk *noise* yang berbeda.

Sinyal Asli

Hasil analisis ragam untuk sinyal asli disajikan pada Tabel 6.2.

Tabel 6.2. Tabel Analisis Ragam untuk Data Sinyal Asli

Sumber	Jumlah Kuadrat	Derajat Bebas	Rataan Jumlah kuadrat	F	Sig.
Perlakuan	10350.823(a)	53	195.299	24.028	.000
Rataan umum	1301332.594	1	1301332.594	160103.194	.000
W	8587.779	1	8587.779	1056.556	.000
D	938.818	2	469.409	57.751	.000
G	217.642	2	108.821	13.388	.000
R	62.633	2	31.316	3.853	.024
W * D	268.355	2	134.177	16.508	.000
W * G	156.753	2	78.377	9.643	.000
D * G	3.071	4	.768	.094	.984
W * D * G	28.127	4	7.032	.865	.487
W * R	25.948	2	12.974	1.596	.207
D * R	6.469	4	1.617	.199	.938
W * D * R	7.432	4	1.858	.229	.922
G * R	7.201	4	1.800	.221	.926
W * G * R	23.201	4	5.800	.714	.584
D * G * R	5.364	8	.671	.082	1.000
W * D * G * R	12.031	8	1.504	.185	.993
Galat	877.833	108	8.128		
Total Terkoreksi	11228.656	161			

(a)R kuadrat = .922 (Adjusted R kuadrat = .883)

Dari tabel tersebut terlihat ada perbedaan pengaruh yang nyata dari kombinasi level setiap faktor terhadap akurasi sistem. Hal ini selain ditunjukkan oleh *level of*

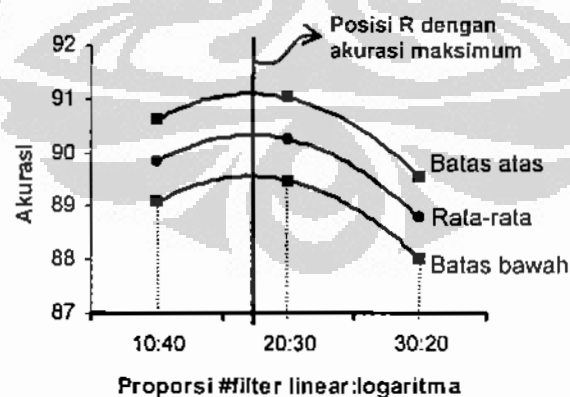
significancy yang kecil (Sig.<5%), juga pada nilai R kuadrat yang cukup tinggi (0.922). Terlihat ada interaksi antara W dengan D, serta W dengan G. Oleh karena itu, ketiga peubah tersebut tidak bisa dianalisis secara terpisah, namun harus secara bersama. Sedangkan peubah R, proporsi jumlah filter linear dengan filter logaritma, tidak berinteraksi dengan peubah lainnya, sehingga peubah ini bisa dilihat secara terpisah. Untuk itu, analisis untuk faktor R dapat dilakukan secara terpisah.

Pengaruh proporsi jumlah filter linear dan jumlah filter logaritmik terhadap akurasi sistem diperlihatkan pada Tabel 6.3. dan Gambar 6.30. Tabel 6.3. menyajikan rata-rata akurasi sistem untuk setiap nilai proporsi filter linear dengan filter logaritma.

Tabel 6.3. Rata-rata Akurasi untuk Berbagai Nilai R

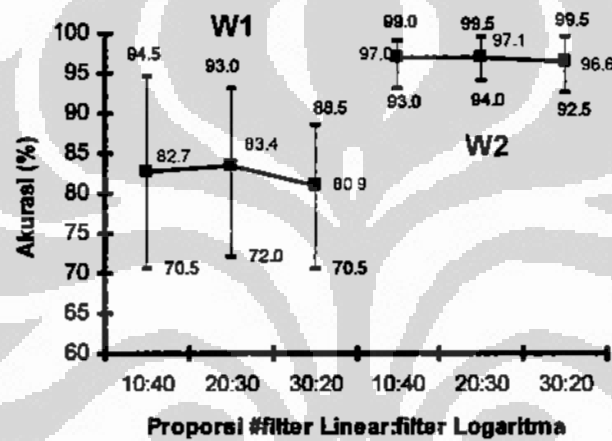
R	Rata-rata akurasi (%)	Selang kepercayaan 95%	
		Batas bawah	Batas atas
10:40	89.852	89.083	90.621
20:30	90.250	89.481	91.019
30:20	88.778	88.009	89.547

Dari tabel tersebut terlihat bahwa nilai proporsi yang paling memberikan hasil terbaik adalah 20:30. Akurasi untuk proporsi 10:40, 20:30, dan 30:20, secara berturut-turut berkisar 89% hingga 90%, 89% hingga 91% dan 88% hingga 89%, pada level taraf nyata 95%.



Gambar 6.30. Rata-rata Akurasi untuk Sinyal Asli pada Berbagai Nilai Peubah R

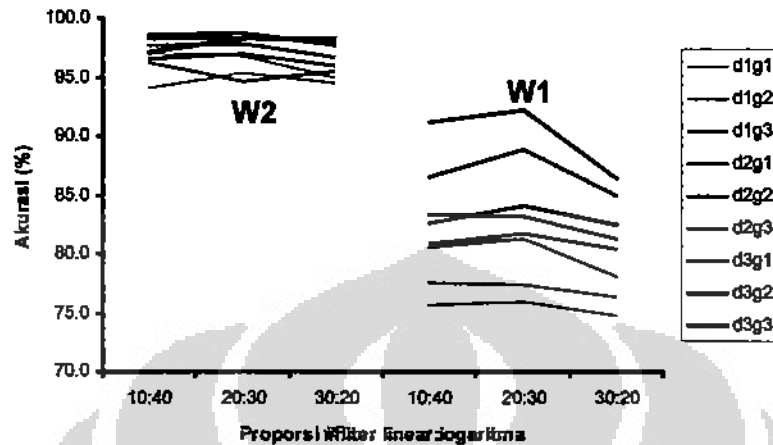
Hal ini menunjukkan bahwa kuantisasi vektor dengan transformasi WC di atas akan memberikan akurasi rata-rata sekitar 90% untuk sinyal asli tanpa penambahan *noise*. Pola pengaruh tersebut dapat dilihat secara visual pada **Gambar 6.30**. Dari gambar tersebut, sekilas terlihat bahwa pola pengaruh R bersifat kuadratik, dan mencapai nilai terbaik pada jumlah filter linear lebih besar dari 10, namun kurang dari 20. Meskipun nilai tersebut kurang begitu tinggi secara keseluruhan, namun kalau dipisahkan untuk kondisi w_1 dan w_2 secara tersendiri, terlihat bahwa untuk w_1 (wrapping sesuai posisi titik channel) akan memberikan akurasi rata-rata 83%, dan sekitar 87% untuk w_2 , seperti ditunjukkan oleh **Gambar 6.31**.



Gambar 6.31. Rata-rata Akurasi untuk Sinyal Asli pada Berbagai Nilai Peubah R untuk Kondisi w_1 dan w_2

Gambar 6.31. secara jelas memperlihatkan sistem dengan w_2 memberikan akurasi yang jauh lebih tinggi dibanding sistem dengan w_1 untuk sinyal asli (dengan selisih sekitar 14%). Kalau diperhatikan lagi mengenai pola pengaruh, terlihat adanya pola kuadratik seperti yang sudah disebutkan disebutkan sebelumnya. Dari bentuk kuadratik ini, secara visual bahwa nilai proporsi terbaik adalah dengan jumlah filter linear lebih dari 10 dan kurang dari 20. Dengan pemilihan D dan G yang tepat, terlihat bahwa akurasi sistem masih bisa dioptimalkan dengan nilai hingga 95% untuk W1 dan 100% untuk W2, seperti ditunjukkan pada **Gambar 6.32**. Dari gambar tersebut terlihat bahwa baik untuk w_1 maupun w_2 , nilai proporsi jumlah filter linear dengan filter logaritma yang sesuai adalah di atas 10:40 dan kurang dari 20:30, seperti

yang sudah ditunjukkan sebelumnya. Hal ini didukung adanya fakta bahwa nilai akurasi pada w1 maupun w2 meningkat setelah 10:40 dan menurun setelah 20:30.



Gambar 6.32. Perbandingan Akurasi pada Sinyal Asli untuk Berbagai Kondisi R, W, D dan G

Hasil tertinggi untuk w2 maupun w1 pada proporsi filter linear dengan logaritma sebesar 20:30 adalah pada kondisi d3 (lebar filter linear 75) dan g3 (lebar filter linear 1.08). Pada kondisi ini, terutama pada w1, kurva menurun lebih tajam di bagian kanan dari proporsi 20:30 dibanding di sebelah kirinya. Ini sekali lagi menunjukkan bahwa jumlah filter linear dibanding logaritma yang terbaik adalah kurang dari 20:30, namun di atas 10:40.

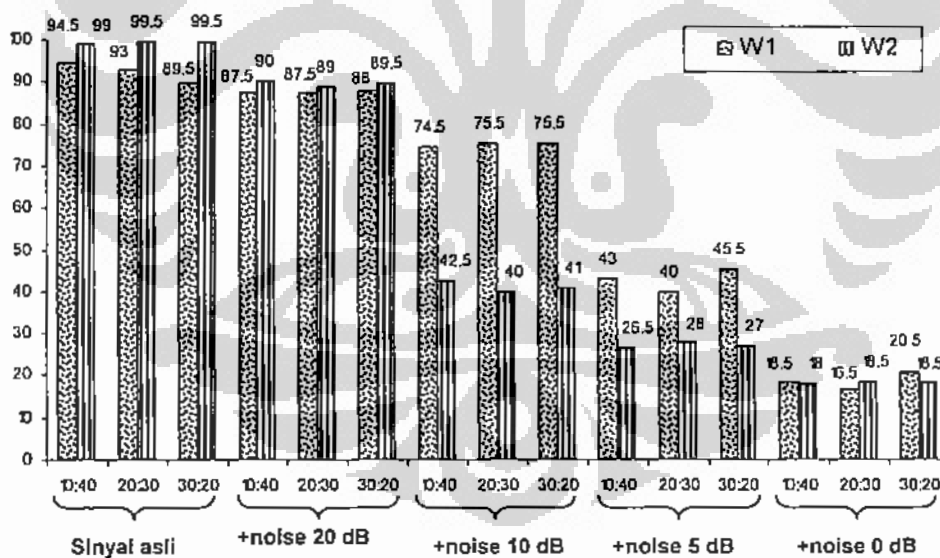
Sinyal Asli Bernoise 20 dB dan 10 dB

Hasil analisis ragam untuk sinyal ini disajikan pada Lampiran 8a dan 8b, yang menunjukkan bahwa interaksi W, D, G, dan R adalah nyata, baik pada penambahan *noise* 20 dB maupun 10 dB. Ini menunjukkan bahwa pola pengaruh setiap peubah/faktor tersebut tidak konsisten pada nilai yang berbeda dari peubah lainnya. Oleh karena itu, perlu dilihat lebih jauh lagi pada setiap kondisi kombinasi nilai dari ke empat peubah tersebut, seperti yang disajikan pada Tabel 6.4. Dari tabel tersebut terlihat bahwa nilai parameter D dan G terbaik (pada *noise* 20 dB dan *noise* 10 dB) untuk kondisi w1 dengan kondisi w2 tidaklah sama. Untuk w1, nilai D terbaik adalah d3 (lebar filter linear 75) dan lebar filter logaritma g1 (1.06).

Tabel 6.4. Hasil Akurasi untuk Sinyal Ber-noise 20 dB dan 10 dB

Nilai	D	G	w1						w2					
			Asli+noise 20 dB			Asli + noise 10 dB			Asli+noise 20 dB			Asli + noise 10 dB		
			r1	r2	r3	r1	r2	r3	r1	r2	r3	r1	r2	r3
Rata-rata	d1	g1	70.0	71.0	71.5	40.3	51.2	47.3	88.3	84.2	86.7	37.7	33.7	35.5
		g2	72.8	74.3	72.5	55.5	57.5	54.7	77.7	76.8	82.2	30.3	26	30
		g3	76.7	76.7	74	62.8	62.2	55	53	61.8	62	24.7	25.7	26.3
	d2	g1	77.7	77.7	77	61.8	63.0	61.8	61.8	53.2	59	28	28.2	28
		g2	70.2	70.5	76.8	71.5	70.2	65.3	64.3	60.5	50.8	22.7	13.7	29.3
		g3	76.7	76.7	74	62.8	62.2	55	53	61.8	62	24.7	25.7	26.3
	d3	g1	80	82.3	80.8	73	69.3	72.3	57.8	40.8	64.5	24	13.7	18.5
		g2	83.2	80	82.5	27.8	27.5	63.8	41.3	40.5	39.5	15.8	12.8	13.7
		g3	79.2	83.7	80.7	29	29.3	29	38.7	40.2	39.5	16.5	11.7	14.8
Maksimum	d1	g1	76.5	77	77	51	52.5	48	90	85	89	42.5	40	41
		g2	78	78.5	77	57	59	56.5	84	88.5	83.5	32	30.5	33.5
		g3	78.5	79	76.5	66.5	84.5	55.5	62.5	66	83	28.5	31	33.5
	d2	g1	83.5	82.5	81.5	62	60	64.5	65.5	57.5	62.5	30	29	30
		g2	84	82	81	72.5	74	67.5	65.5	67	54.5	25	16.5	30
		g3	85.5	83.5	84.5	43	43.5	75	44.5	46	71.5	19.5	26	22.5
	d3	g1	84.5	83.5	85.5	74.5	72	75.5	64	46	68.5	28.5	15.5	22.5
		g2	86.5	85	86.5	35	29.5	71.5	49.5	47	48.5	23	16	16.5
		g3	84.5	85	84	35.5	35	32.5	39	45	47.5	19.5	15	21

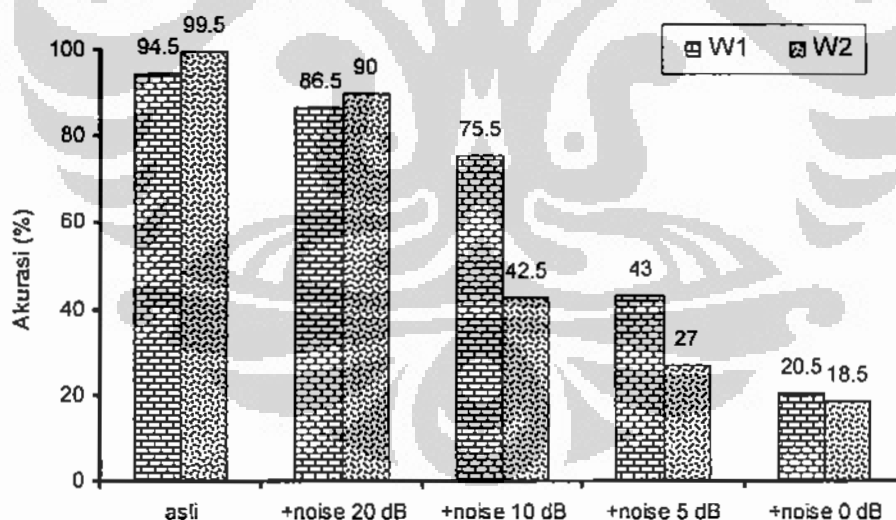
Sedangkan untuk w2, nilai D dan G yang sesuai adalah d1 (lebar filter linear 25) dan g1 (lebar filter logaritmik 1.06), seperti yang tercetak tebal pada tabel di atas. Untuk memperjelas perbandingan ini, perhatikan Gambar 6.33.



Gambar 6.33. Perbandingan Akurasi antara w1 dan w2 pada Sinyal Asli dan Berbagai Noise

Dari **Gambar 6.33**, terlihat bahwa untuk sinyal asli, w1 memberikan akurasi tertingginya sekitar 5% di bawah w2. Sedangkan untuk *noise* 20 dB, secara relatif teknik w1 sekitar 1 sampai 2 % di bawah w2. Namun dengan meningkatkan *noise* menjadi 10 dB, 5 dB dan 0 dB, hal sebaliknya terjadi. Sistem dengan teknik w1 memberikan akurasi jauh di atas yang diberikan oleh w2. Ini menunjukkan bahwa tingkat *robustness* w1 terhadap *noise* lebih tinggi dibandingkan dengan w2. Teknik w2 mampu bertahan hingga *noise* 20 dB. Sedangkan w1, untuk untuk *noise* 10 dB masih mampu memberikan akurasi hingga 75%. Nilai ini jauh di atas akurasi pada w2 yang hanya mencapai nilai tertinggi 42%.

Seperti telah disebutkan sebelumnya, bahwa untuk kondisi w1, nilai D dan G yang cocok adalah d3, yaitu 75 dan g1, yaitu 1.06. Sedangkan untuk w2, nilai D dan G yang sesuai adalah d1 dan g1. Sedangkan untuk nilai R, dapat dipilih r1 (10:40). Namun untuk kondisi sinyal tanpa penambahan *noise*, maka nilai parameter terbaiknya berbeda dengan pada kondisi sinyal dengan penambahan *noise*. Oleh karena itu, dapat disebutkan bahwa nilai parameter yang terbaik tergantung dari kondisi sinyal.



Gambar 6.34. Perbandingan Akurasi Terbaik antara Kondisi w1 dengan w2 pada Sinyal Asli dan yang Telah Memperoleh Penambahan *Noise*

Gambar 6.34. menyajikan perbandingan akurasi antara w_1 dengan w_2 untuk sinyal asli hingga *noise* 0 dB. Gambar di atas sekali lagi mendukung pernyataan sebelumnya yang menyebutkan bahwa sistem dengan *wrapping* pada posisi channel akan bersifat lebih *robust* terhadap *noise* dibanding *wrapping* sesuai jarak channel.

6.3.3. Beberapa Temuan dan Tahapan Selanjutnya

Beberapa hal yang bisa diutarakan berdasar hasil percobaan untuk mengkaji lanjut terhadap parameter kuantisasi vektor yang telah dilakukan adalah :

1. Jenis peubah yang dikenakan *wrapping*, lebar filter linear dan logaritma, proporsi jumlah filter linear dan filter logaritma secara nyata memberikan pengaruh terhadap hasil akurasi sistem.
2. Pada sinyal asli, teknik menggunakan jarak channel yang dikenakan *wrapping* (w_2) memberikan hasil dengan akurasi rata-rata 97% (maksimum 99.5%) pada lebar filter linear 75 dan filter logaritma 1.08 dengan proporsi jumlah filter linear dibanding logaritma 20:30. Pada kondisi yang sama, teknik *wrapping* yang menggunakan posisi channel memberikan akurasi rata-rata 83% dengan nilai maksimum 94.5%. Pada kondisi sinyal asli ini, nilai proporsi jumlah filter linear dengan filter logaritma yang optimum terletak di antara 10:40 dengan 20:30.
3. Dengan bertambahnya *noise*, baik teknik w_1 maupun w_2 mengalami penurunan, dengan nilai akurasi maksimum untuk w_1 dan w_2 pada sinyal asli, *noise* 20 dB, 10 dB, 5 dB dan 0 dB berturut-turut adalah 94.5%, 86.5%, 75.5%, 43%, 20.5% dan 99.5%, 90%, 42.5%, 27%, 18.5%. Nilai maksimum ini dicapai dengan kondisi nilai parameter yang berbeda-beda pada sinyal asli dan sinyal dengan *noise* 20 dB, 10 dB, 5 dB, dan 0 dB.

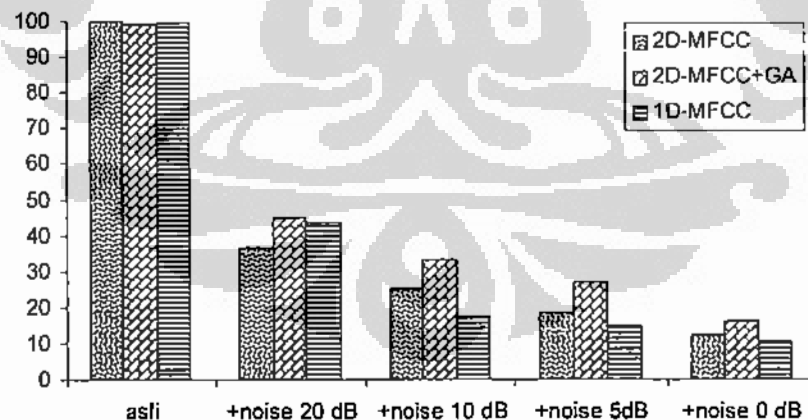
Setelah melakukan serangkaian percobaan dan diperoleh fakta-fakta empiris, maka pada bagian selanjutnya akan dilakukan analisis perbandingan dari seluruh teknik yang sudah dicobakan. Hal ini diperlukan untuk mengformulasikan sistem identifikasi pembicara yang relatif lebih *robust* terhadap *noise*.

BAB 7 ANALISIS PERBANDINGAN KINERJA DAN RANCANGAN SISTEM IDENTIFIKASI PEMBICARA

Sejauh ini telah disajikan hasil percobaan yang meliputi 4 bagian, yaitu percobaan berbasis power spektrum (Bab 4), percobaan berbasis *higher order statistics* (Bab 5 dan 6). Bagian ini akan membahas perbandingan akurasi dari semua percobaan yang telah dilakukan. Pembahasan akan dimulai dengan membandingkan antara akurasi sistem berbasis power spektrum pada Bab 4 dengan akurasi sistem berbasis bispektrum dengan ekstraksi ciri menggunakan 2D-MFCC yang telah disajikan pada Bab 5. Berikutnya adalah perbandingan akurasi sistem berbasis bispektrum dengan pengekstraksi ciri yang berbeda, yaitu 2D-MFCC, kuantisasi skalar dan kuantisasi vektor. Bagian terakhir akan menyajikan perbandingan secara keseluruhan, sehingga secara komprehensif bisa diketahui efektifitas sistem dalam melakukan pengenalan pembicara berdasar suara.

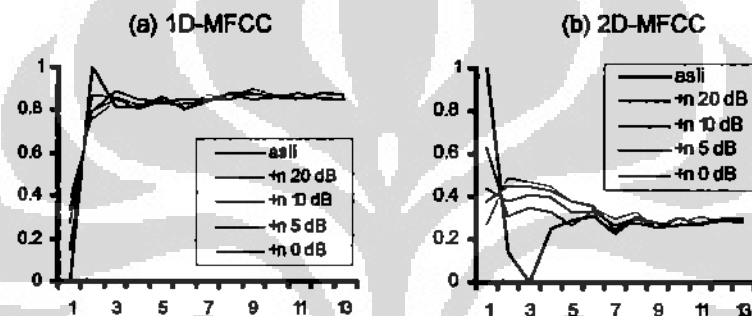
7.1. Sistem Berbasis Power Spektrum vs Bispektrum

Perbandingan akurasi sistem yang berbasis power spektrum (1D-MFCC) dengan yang berbasis bispektrum (2D-MFCC) disajikan pada Gambar 7.1.



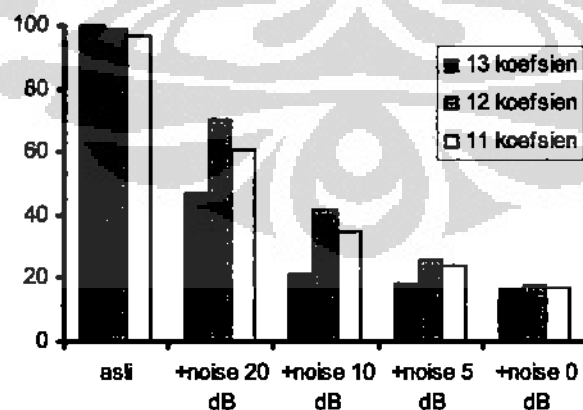
Gambar 7.1. Perbandingan Akurasi Sistem yang Berbasis Power Spektrum dengan yang Berbasis Bispektrum

Dari gambar tersebut terlihat bahwa untuk sinyal asli, ke tiga sistem memberikan akurasi yang relatif baik, yaitu ketiganya sekitar 99%. Hal ini mengindikasikan bahwa jika sinyal yang masuk tidak terkontaminasi *noise*, atau *noise* yang ada cukup kecil, maka sebaiknya dipergunakan teknik 1D-MFCC. Teknik ini dengan akurasi yang tinggi, dapat dilakukan komputasi dengan kompleksitas yang lebih kecil dibanding dengan 2D-MFCC. Namun demikian untuk sinyal dengan penambahan *noise*, ketiga teknik tersebut gagal bekerja dengan baik. Setelah ditelusuri lebih lanjut terhadap komponen vektor ciri, terlihat bahwa komponen pertama dan kedua bersifat sangat sensitif terhadap *noise*, seperti ditunjukkan pada Gambar 7.2.



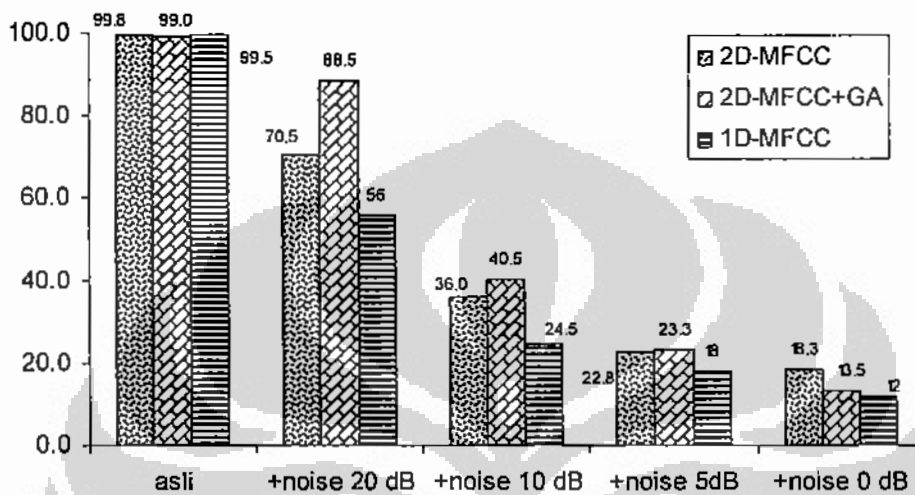
Gambar 7.2. Pola 13 Koefisien Vektor Ciri (1D-MFCC dan 2D-MFCC)

Mengacu pada Gambar 7.2., dilakukan percobaan identifikasi, dengan mengabaikan secara berturut-turut koefisien pertama dan kedua, dan hasilnya disajikan pada Gambar 7.3.



Gambar 7.3. Perbandingan Akurasi Sistem dengan Teknik 2D-MFCC tanpa GA pada Berbagai Jumlah Koefisien

Dari gambar tersebut terlihat bahwa pemilihan jumlah koefisien 12 lebih sesuai dibanding 13 ataupun 11 koefisien. Untuk *noise* 20 dB, dengan 12 koefisien, akurasi mencapai 70.5% untuk 2D-MFCC. Untuk melihat perbandingan secara komprehensif antara 2D-MFCC tanpa GA, 2D-MFCC dengan GA, dan 1D-MFCC, perhatikan **Gambar 7.4**.



Gambar 7.4. Perbandingan Akurasi Sistem dengan Teknik 2D-MFCC (tanpa dan dengan GA) serta 1D-MFCC untuk Jumlah Koefisien 12

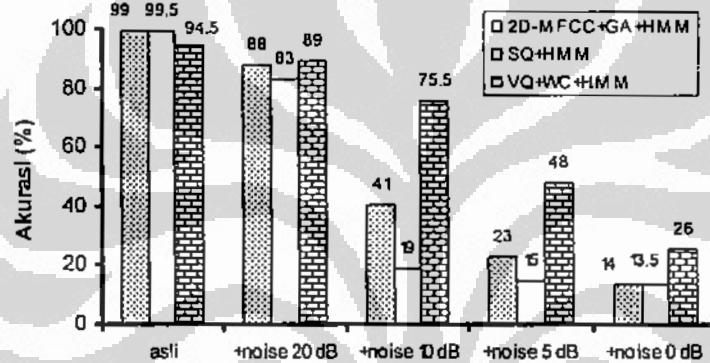
Terlihat bahwa dengan mengabaikan koefisien pertama, untuk sinyal asli ditambah *noise* 20 dB, ternyata teknik 2D-MFCC tanpa GA mampu meningkatkan akurasi menjadi 70.5% dari sebelumnya hanya sekitar 43%. Sedangkan 1D-MFCC hanya meningkat sekitar 16% dari sekitar 40% menjadi 56%. Dengan menggunakan algoritme genetika, ternyata akurasi sistem dengan teknik 2D-MFCC meningkat menjadi 88.5%. Namun demikian, untuk *noise* yang lebih tinggi, ternyata sistem masih gagal melakukan pengenalan, yaitu dengan akurasi di bawah 50%.

7.2. 2D-MFCC vs Kuantisasi Skalar dan Vektor

Pada bagian sebelumnya telah ditunjukkan bahwa nilai bispektrum sinyal yang telah terkontaminasi *noise* secara visual masih memiliki kemiripan dengan sinyal asli tanpa penambahan *noise*. Kecuali pada penambahan *noise* 0 dB, bentuk bispektrum telah mengalami perbedaan yang berarti dari bispektrum pada sinyal asli.

Hal ini menunjukkan bahwa bispektrum sebagai penentu ciri sinyal bersifat *robust* terhadap *noise*. Juga terbukti bahwa teknik 2D-MFCC mampu mengekstraksi ciri dengan baik, namun bersifat sensitif terhadap penambahan *noise*. Oleh karena itu, pada Bab 6 dikembangkan teknik kuantisasi terhadap nilai bispektrum sebelum masuk ke tahap *wrapping* dan transformasi kosinus. Sub bab ini akan melakukan perbandingan akurasi antara sistem berbasis bispektrum dengan ekstraksi ciri menggunakan 2D-MFCC, kuantisasi skalar dan kuantisasi vektor.

Gambar 7.5. menyajikan perbandingan akurasi antara sistem berbasis bispektrum dengan teknik ekstraksi ciri menggunakan 2D-MFCC, kuantisasi skalar dan kuantisasi vektor.

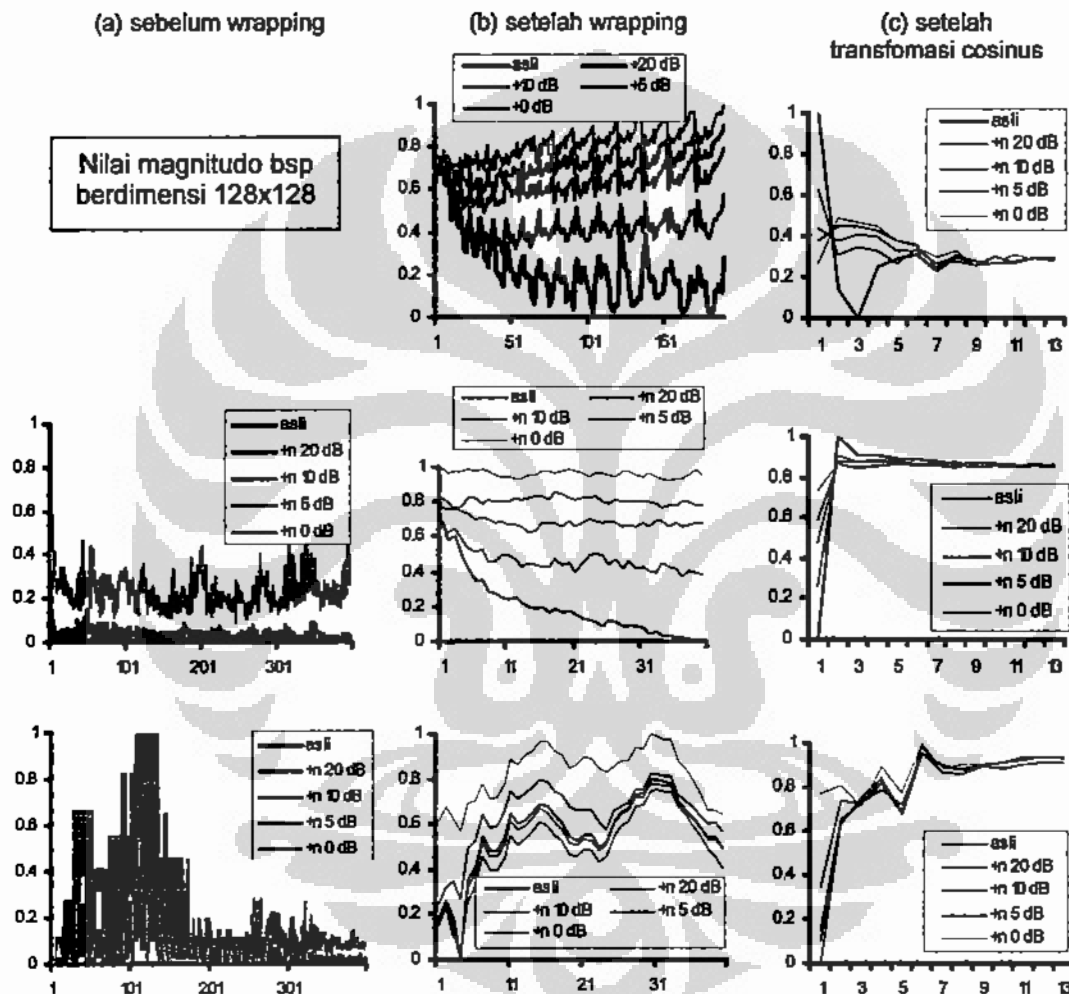


Gambar 7.5. Perbandingan Akurasi Sistem Berbasis Bispektrum dengan Berbagai Teknik Ekstraksi Ciri pada Berbagai Level *Noise*

Dari gambar tersebut terlihat bahwa penurunan akurasi untuk teknik kuantisasi vektor bersifat gradual. Sementara itu, akurasi dua teknik lainnya turun secara drastis untuk level *noise* yang makin besar. Hal ini menunjukkan bahwa teknik kuantisasi vektor bersifat lebih *robust* terhadap *noise* dibanding lainnya. Untuk sinyal asli, semua teknik ekstraksi ciri memberikan hasil akurasi di atas 95%, kecuali teknik kuantisasi vektor dengan proses *wrapping* dan kosinus yang hanya 94.5%. Untuk *noise* 20 dB, ketiga teknik masih bertahan dengan akurasi sedikit di bawah 90%. Namun untuk *noise* yang lebih besar, yaitu 10 dB, 2D-MFCC dan kuantisasi skalar telah gagal bekerja dengan baik. Sementara itu, kuantisasi vektor masih memberikan akurasi sekitar 75%. Untuk *noise* yang lebih besar lagi, yaitu 5 dB dan 0 dB, akurasi yang

diberikan oleh kuantisasi vektor adalah 48% dan 26%. Nilai ini hampir dua kali lipat dari akurasi yang diberikan oleh 2D-MFCC dan kuantisasi skalar.

Fakta diatas dapat dijelaskan dengan **Gambar 7.6**, yang menyajikan pola data hasil pengolahan sinyal suara hingga vektor ciri menggunakan teknik 2D-MFCC (baris pertama), kuantisasi skalar (baris kedua), dan kuantisasi vektor (baris ketiga).



Gambar 7.6. Pola Data Hasil Pengolahan Sinyal Menjadi Vektor Ciri

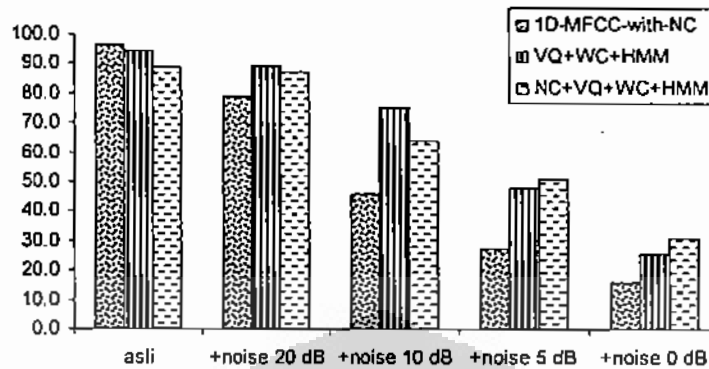
Baris pertama pada gambar di atas adalah data yang dihasilkan pada proses ekstraksi ciri menggunakan 2D-MFCC. Dari sebuah frame pada sinyal suara, dihitung nilai bispektrumnya, dan dilanjutkan dengan proses *wrapping*. Keluaran dari proses

wrapping, menjadi masukan untuk dilakukan transformasi kosinus dan diperoleh vektor ciri dari frame masukan tersebut. Pada baris pertama terlihat bahwa output dari *wrapping* ini untuk sinyal asli dengan sinyal ber-*noise* telah mengalami perubahan relatif besar. Hal inilah yang menyebabkan teknik 2D-MFCC sangat sensitif terhadap *noise*.

Baris kedua dimulai dengan melakukan kuantisasi skalar terhadap bispektrum. Hasil kuantisasi skalar ini ditunjukkan pada kolom pertama baris kedua. Terlihat bahwa nilai kuantisasi skalar bispektrum relatif mirip antara sinyal asli dengan sinyal ber-*noise*, kecuali untuk *noise* 0 dB yang sangat berbeda jauh. Namun demikian, setelah memasuki proses *wrapping*, terjadi perubahan yang nyata antara sinyal asli dengan sinyal ber-*noise*. Hal inilah yang menyebabkan vektor ciri yang dihasilkan berbeda jauh dengan vektor ciri untuk sinyal aslinya, yang pada akhirnya berakibat pada rendahnya akurasi untuk sinyal ber-*noise*. Meskipun demikian, untuk *noise* 20 dB, sistem yang dihasilkan masih mampu memberikan akurasi yang cukup kompetitif, yaitu sebesar 83% (diatas teknik 2D-MFCC tanpa GA yang hanya sekitar 70%).

Seperti pada kuantisasi skalar, bispektrum dari sebuah frame suara dikuantisasi vektor dan output dari sini (kolom pertama) masuk ke proses *wrapping*. Output proses *wrapping* ini (kolom ke dua) selanjutnya memasuki proses tranformasi kosinus dan dihasilkan vektor ciri berdimensi 13. Dari baris ke tiga tersebut terlihat bahwa hasil kuantisasi vektor untuk sinyal dengan sinyal yang sudah ditambah *noise* relatif lebih mirip (dibandingkan dengan dua teknik sebelumnya). Hal inilah yang menyebabkan keluaran dari proses *wrapping* relatif mirip antara sinyal asli dengan sinyal ber-*noise*, khususnya untuk *noise* 20 dB dan 10 dB. Fakta ini juga ditunjukkan pada kolom ke tiga yang menunjukkan vektor ciri yang dihasilkan memiliki kemiripan antara sinyal asli dengan sinyal ber-*noise* (khususnya 20 dB dan 10 dB). Akibat dari sini adalah dihasilkannya akurasi yang relatif tinggi hingga *noise* 10 dB, yaitu sebesar 75.5%. Sementara itu, teknik lainnya menghasilkan akurasi sekitar 40%, (**Gambar 7.5.**)

Gambar 7.7. menyajikan perbandingan akurasi sistem antara 1D-MFCC dengan NC dengan kuantisasi vektor dengan dan tanpa NC.

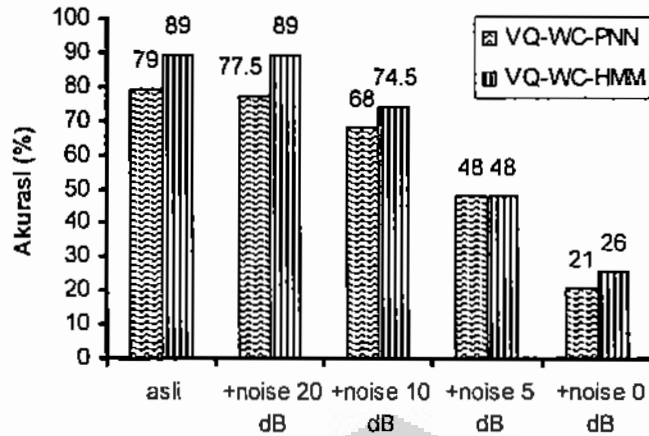


Gambar 7.7. Perbandingan Akurasi antara 1D-MFCC dengan NC dengan Kuantisasi Vektor dengan dan tanpa NC

Dari gambar tersebut terlihat bahwa kuantisasi vektor memberikan akurasi yang lebih baik untuk sinyal yang sudah ditambah *noise* dibandingkan dengan 1D-MFCC dengan *noise canceling*. Teknik *noise canceling* yang dipadukan dengan kuantisasi vektor tidak memberikan pengaruh yang berarti, khususnya untuk sinyal asli, sinyal asli ditambah *noise* 20 dB dan 10 dB. Untuk penambahan *noise* 5 dB dan 0 dB, *noise canceling* mampu meningkatkan sedikit akurasi.

7.3. HMM vs PNN

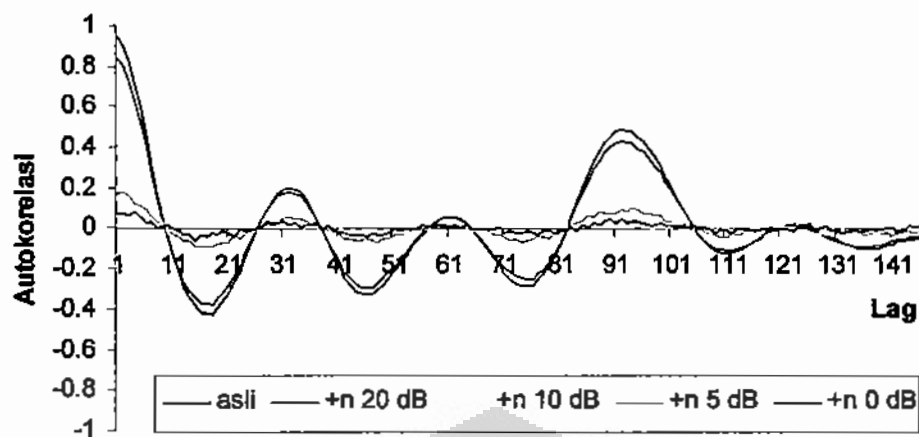
Pada bagian sebelumnya disajikan perbandingan akurasi untuk beberapa teknik ekstraksi ciri. Pada bagian ini disajikan perbandingan akurasi antara teknik HMM dengan PNN sebagai pengenalan pola, dengan menggunakan kuantisasi vektor dan transformasi *wrapping* dan kosinus sebagai pengekstraksi ciri. **Gambar 7.8.** menyajikan perbandingan akurasi antara teknik HMM dengan PNN sebagai pengenalan pola. Terlihat bahwa teknik HMM relatif lebih baik dibanding PNN. Hal ini disebabkan karena pada HMM, proses pengenalan melibatkan setiap frame suara dari setiap unit waktu. Sementara pada PNN, nilai ekstraksi ciri diperoleh dengan meratakan seluruh frame yang ada pada suara. Oleh karena itu, HMM mengakomodasi perubahan atau variasi antar unit waktu dari satu suara (hal ini tidak terjadi pada PNN).



Gambar 7.8. Perbandingan Akurasi antara Teknik HMM dengan PNN sebagai Pengenal Pola

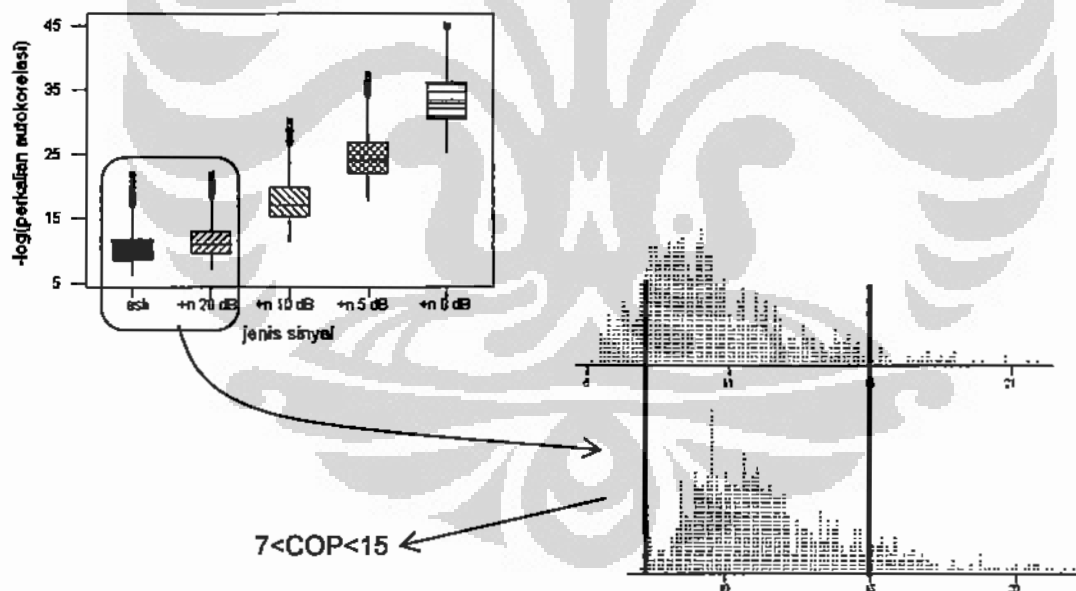
7.4. Rancangan Sistem Identifikasi Pembicara

Setelah memperhatikan hasil-hasil percobaan di atas, maka berikut ini akan disajikan rancangan sistem identifikasi pembicara yang diimplementasikan dengan Matlab versi 7.1. Berdasar fakta-fakta empiris di atas, maka teknik yang akan dipergunakan untuk identifikasi pembicara akan berbeda sesuai kualitas sinyal masukkan. Untuk sinyal asli (dengan kualitas baik), maka teknik 1D-MFCC akan menjadi pilihan. Teknik ini mampu memberikan akurasi sekitar 99% dan mempunyai kompleksitas yang lebih kecil. Sedangkan untuk sinyal dengan kualitas lebih jelek (terdistorsi *noise* cukup besar), teknik yang sesuai adalah bispektrum yang direpresentasikan dengan kuantisasi vektor. **Gambar 7.9.** menyajikan perbandingan plot autokorelasi antar sinyal asli dengan sinyal asli yang sudah ditambah *noise*. Gambar tersebut memperlihatkan bahwa penambahan *noise* akan mengakibatkan nilai autokorelasinya mengalami penurunan (pada gambar ditunjukkan autokorelasi dengan penambahan 5 dB dan 0 dB, kurva lebih dekat ke korelasi nol). Ini berarti bahwa perkalian nilai mutlak autokorelasi hingga lag 21 untuk sinyal dengan penambahan *noise* akan makin kecil dengan makin besarnya *noise*. Dengan kata lain nilai ini diharapkan dapat dipergunakan untuk membedakan sinyal sesuai kualitasnya. Hal yang diperlukan adalah suatu titik *cut of point* untuk memutuskan teknik identifikasi yang dipergunakan.



Gambar 7.9. Pola Autokorelasi dari Sinyal Asli dan Sinyal Asli yang telah Mengalami Penambahan *Noise*

Gambar 7.10. menyajikan tahapan penentuan *cut off point* (COP) tersebut dengan menggunakan analisis eksploratif dari negatif logaritma perkalian autokorelasi hingga lag 21.



Gambar 7.10. Tahapan Proses Penentuan *Cut of Point*

Dari gambar tersebut terlihat bahwa yang sulit dibedakan adalah antara sinyal asli dengan sinyal yang sudah ditambah noise 20 dB. Pada kedua jenis sinyal ini overlap terjadi untuk rentang nilai negatif logaritma perkalian autokorelasi hingga

lag 21 dari 7 hingga 15. Untuk mengetahui nilai COP optimum, maka dilakukan uji coba untuk beberapa nilai COP (7.5, 10, 12.5, dan 15) dengan melibatkan sinyal asli maupun sinyal asli yang sudah ditambah *noise* 20 dB dan 10 dB (sehingga ada $4 \times 3 = 12$ kombinasi). Percobaan dilakukan pada 10 pembicara dengan masing-masing menggunakan 20 data suara uji yang dipilih secara acak yang tidak termasuk dalam data pelatihan. Hasil percobaan disajikan pada **Tabel 7.1**.

Tabel 7.1. Perbandingan Persentasi Akurasi Sistem untuk Berbagai nilai COP

Nilai COP	Sinyal Asli	+noise 20 dB	+noise 10 dB	rata-rata
7.5	85.5	85.0	75.5	82.0
10.0	91.5	71.5	75.5	79.5
12.5	95.5	57.0	75.5	76.0
15.0	98.0	46.5	73.5	72.7

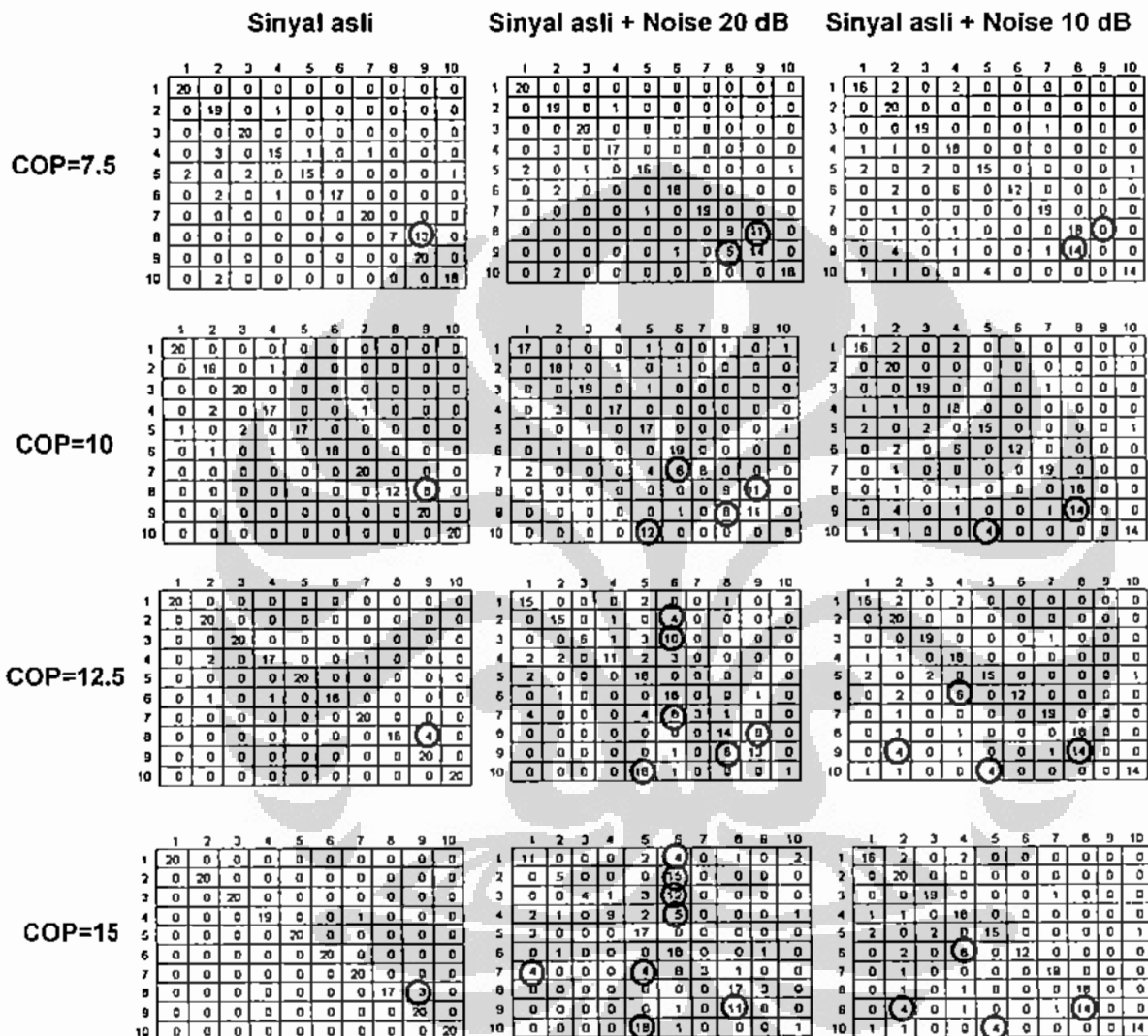
Dari tabel tersebut terlihat bahwa pemilihan nilai COP sangat tergantung dari kondisi input sinyal yang masuk. Jika sinyal yang masuk mempunyai komposisi yang sama antara yang berkualitas baik dengan yang jelek, maka nilai COP yang layak adalah 7.5.

Tabel 7.2. Jarak Kullback-Leibler antar 10 Model HMM, [DD96].

	HMM1	HMM2	HMM3	HMM4	HMM5	HMM6	HMM7	HMM8	HMM9	HMM10
HMM1	0.00	5.09	5.61	2.72	3.44	5.36	4.39	18.84	14.59	3.37
HMM2	5.09	0.00	6.91	1.07	5.39	5.14	8.78	14.84	11.30	5.89
HMM3	5.61	6.91	0.00	6.13	2.96	6.26	5.46	23.03	21.10	5.61
HMM4	2.72	1.07	6.13	0.00	2.06	2.79	4.39	13.45	12.65	3.09
HMM5	3.44	5.39	2.96	2.06	0.00	5.28	3.36	20.12	19.35	1.62
HMM6	5.36	5.14	6.26	2.79	5.28	0.00	6.21	10.53	9.28	8.44
HMM7	4.39	8.78	5.46	4.39	3.36	6.21	0.00	19.72	17.28	6.28
HMM8	18.84	14.84	23.03	13.45	20.12	10.53	19.72	0.00	1.10	30.41
HMM9	14.59	11.30	21.10	12.65	19.35	9.28	17.28	1.10	0.00	27.13
HMM10	3.37	5.89	5.61	3.09	1.62	8.44	6.28	30.41	27.13	0.00

Sedangkan jika sebagian besar kualitas sinyal yang masuk adalah baik, maka nilai COP yang layak adalah 15. Untuk mengetahui lebih detail lagi mengenai hasil pengenalan, pada **Gambar 7.11**. disajikan hasil klasifikasi dari 12 kombinasi tersebut. Kalau diperhatikan secara cermat pada **Gambar 7.11**. terlihat bahwa kesalahan secara konsisten terjadi antara speaker 8 dengan 9 (baik yang 8 masuk ke 9

ataupun 9 masuk ke 8). Hal ini wajar terjadi karena kedua pembicara tersebut mempunyai karakteristik yang mirip seperti telah disebutkan pada Tabel 1.1. di Bab 1. Kemiripan kedua pembicara tersebut juga ditunjukkan oleh jarak antara model HMM masing-masing yang kecil, seperti ditunjukkan pada Tabel 7.2.

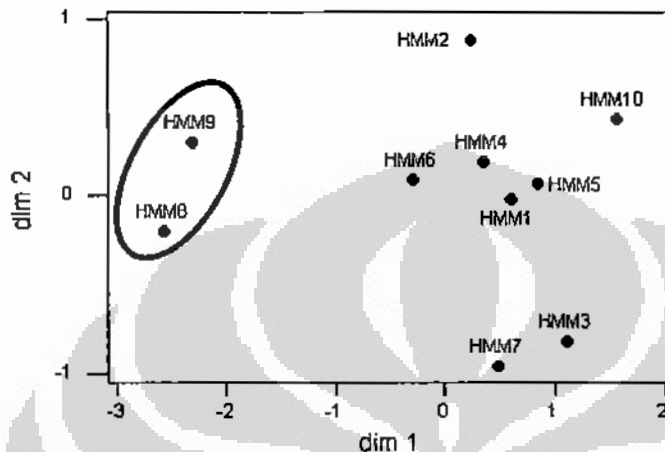


Catatan : nilai a_{ij} artinya jumlah suara pembicara ke i yang terklasifikasikan ke pembicara ke j

Gambar 7.11. Hasil Klasifikasi per Kombinasi COP-Sinyal Input dari 10 Pembicara (masing-masing 20 suara)

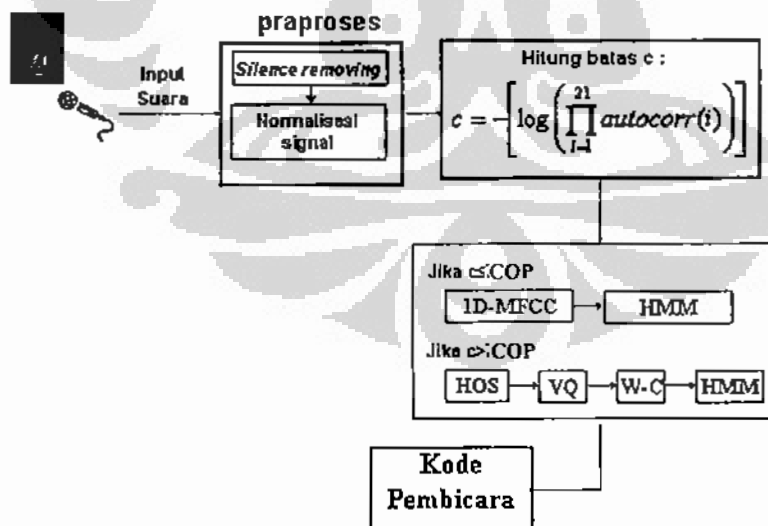
Untuk mengetahui posisi relatif dari 10 model tersebut pada bidang dimensi dua, digunakan teknik Penskalaan Dimensi Ganda, dan hasilnya disajikan pada

Gambar 7.12. Dari **Gambar 7.12.** terlihat bahwa HMM untuk pembicara 8 dan 9 berada pada daerah yang memisah dari HMM untuk pembicara lainnya. Hal ini menunjukkan bahwa adanya pemisahan model HMM antara pembicara laki-laki dengan perempuan. Dari fakta ini, terlihat jelas bahwa HMM dapat dipergunakan untuk membedakan antara pembicara laki-laki dengan pembicara perempuan.



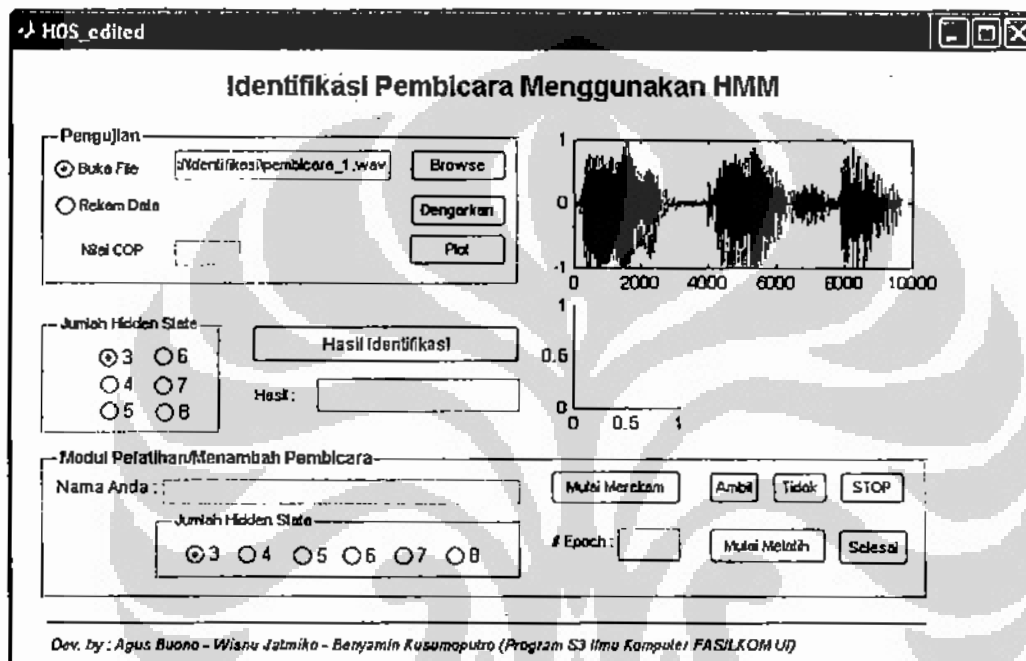
Gambar 7.12. Posisi Relatif antar Model HMM Hasil Penskalaan Dimensi Ganda

Kembali pada masalah rancangan teknik ekstraksi ciri yang efektif, maka setelah menetapkan cut of point bagi kualitas sinyal masukan, dilanjutkan dengan menentukan teknik pengenalan yang sesuai.



Gambar 7.13. Rancangan Sistem Identifikasi Pembicara

Jika nilai negatif logaritma dari perkalian autokorelasi hingga lag 21 kurang dari COP, ini menunjukkan bahwa kualitas sinyal masukan relatif baik, sehingga dipergunakan teknik 1D-MFCC sebagai ekstraksi ciri. Sedangkan jika lebih besar, maka digunakan teknik ekstraksi dengan kuantisasi vektor terhadap nilai bispektrum dan dilanjutkan dengan transformasi *wrapping* dan kosinus. Sedangkan antar muka sistem yang diimplementasikan dengan Matlab ini disajikan pada **Gambar7.14**.



Gambar7.14. Antar Muka Sistem Identifikasi Pembicara

Terlihat bahwa input suara yang diberikan bisa berasal dari suara yang sudah berupa file atau suara yang dimasukkan saat sistem dijalankan. Sebelum melakukan identifikasi, pengguna bisa memilih nilai COP yang diinginkan, juga jumlah hidden state pada model HMM. Dalam hal ini terdapat pilihan jumlah hidden state yang bisa dipergunakan, yaitu dari 3 hingga 7. Untuk memperbaiki model, juga disediakan modul untuk pelatihan sesuai dengan jumlah hidden state yang diinginkan.

7.5. Beberapa Temuan

Beberapa temuan yang bisa disampaikan pada bab ini adalah :

1. Bispektrum lebih mampu sebagai penentu ciri sinyal suara dibandingkan dengan power spektrum. Tingkat perbedaan sinyal asli dengan sinyal ber-*noise* ini masih terlihat untuk *noise* di atas 0 dB. Untuk *noise* 0 dB, bentuk bispektrum sudah sulit dibedakan secara visual dengan bispektrum dari sinyal asli.
2. Teknik ekstraksi ciri 1D-MFCC maupun 2D-MFCC bersifat sensitif terhadap adanya *noise*. Tahapan yang paling bersifat sensitif ini adalah pada bagian proses *wrapping* yang menggunakan satu set filter, dan nilai vektor ciri pada elemen pertama adalah yang paling tidak stabil dengan adanya *noise*. Dengan algoritme genetika dan pengabaian elemen pertama, mampu dihasilkan sistem dengan akurasi mencapai 88%, yang meningkat lebih dari 40% dari sebelumnya.
3. Teknik penghapusan *noise* (*noise canceling*, NC) hanya mampu bekerja pada sinyal ber-*noise* 20 dB dan dengan akurasi mencapai 78.8%.
4. Teknik kuantisasi skalar mampu menghasilkan sistem dengan akurasi yang baik untuk sinyal asli (akurasi > 98%) dan untuk sinyal ber-*noise* 20 dB mencapai 80.5%. Namun gagal untuk *noise* yang lebih besar.
5. Teknik kuantisasi vektor memberikan hasil yang relatif lebih baik untuk semua level *noise* dari 20 dB hingga 0 dB. Hasil yang memadai dicapai sampai pada level *noise* 10 dB dengan akurasi sekitar 75%. Namun demikian untuk sinyal tanpa penambahan *noise* hanya mampu memberikan akurasi tertingginya 94.5%.
6. Penggunaan jarak channel pada kuantisasi vektor sebagai peubah yang dikenakan proses *wrapping* mampu menghasilkan sistem dengan akurasi di atas 98% untuk sinyal asli dan 90% untuk sinyal ber-*noise* 20 dB. Namun untuk sinyal ber-*noise* lebih besar dari 20 dB, sistem gagal melakukan pengenalan dengan baik.

7. Bagian yang menentukan pada proses ekstraksi ciri adalah proses *wrapping*. Pemilihan parameter yang sesuai akan menentukan tingkat *robustness* sistem yang dihasilkan. Empat parameter utama pada proses *wrapping* adalah jenis peubah yang dikenakan *wrapping*, proporsi jumlah filter linear dengan filter logaritma, lebar filter linear dan lebar filter logaritma.
8. Teknik HMM sebagai pengenalan pola sinyal suara memberikan hasil yang lebih baik pada semua level *noise* dibandingkan dengan PNN. Namun demikian, teknik HMM ini hanya bekerja dengan baik untuk data yang pengaruh *noise*-nya telah diminimalkan.
9. Teknik ekstraksi ciri yang optimum tergantung dari kualitas sinyal masukan. Kualitas sinyal masukan dapat diidentifikasi berdasar negatif dari logaritma perkalian autokorelasi hingga lag 21. Makin besar *noise* maka nilai ini juga makin besar secara kuadratik.
10. Titik batas untuk memutuskan teknik ekstraksi ciri yang optimum dengan nilai negatif logaritma perkalian autokorelasi hingga lag 21 adalah berkisar dari 7 hingga 15. Pemilihan nilai yang sesuai sangat tergantung dari kondisi sinyal masukan.

BAB 8 KESIMPULAN DAN SARAN

Beberapa hal yang bisa disebutkan sebagai kesimpulan akhir adalah :

1. Sistem identifikasi pembicara konvensional yang berbasis pada power spektrum mampu memberikan hasil dengan akurasi >98% untuk sinyal asli tanpa penambahan *noise*, namun gagal untuk sinyal dengan penambahan *noise*, meskipun hanya pada level 20 dB. Teknik penghapusan *noise* hanya mampu menghasilkan sistem dengan akurasi memadai (78.8%) hingga level *noise* 20 dB. Untuk *noise* yang lebih besar, teknik ini tidak bisa bekerja dengan baik.
2. Bispektrum mampu menangkap ciri sinyal suara baik tanpa penambahan *noise* maupun dengan penambahan *noise*, dan secara visual masih terlihat hingga level di atas 0 dB. Untuk level *noise* 0 dB, bentuk bispektrum secara visual telah mengalami perubahan secara nyata dibanding dengan sinyal aslinya.
3. Teknik 2D-MFCC sebagai pengembangan 1D-MFCC untuk memproses lanjut nilai bispektrum yang berdimensi dua memberikan hasil lebih tinggi dibanding dengan akurasi sistem dengan teknik 1D-MFCC yang berbasis power spektrum. Dengan algoritme genetika, akurasi sistem dengan teknik 2D-MFCC meningkat lebih dari 45%, menjadi 88.5% pada level *noise* 20 dB. Namun untuk *noise* yang lebih besar, teknik ini tidak bisa bekerja dengan baik.
4. Kuantisasi skalar dan vektor memperbaiki sistem dari aspek cara membaca nilai bispektrum yang dikelompokkan pada beberapa channel. Nilai statistik yang optimum untuk setiap channel adalah rata-rata nilai bispektrum di atas persentil 75%. Dengan nilai ini, kuantisasi skalar mampu menghasilkan sistem dengan akurasi yang baik (>98% untuk

- sinyal asli), dan memadai (80.5% untuk sinyal ber-*noise* 20 dB). Untuk *noise* yang lebih tinggi kuantisasi skalar gagal bekerja dengan baik.
5. Pada kuantisasi vektor, nilai bispektrum dikelompokkan pada beberapa channel yang dibentuk dengan mengikuti distribusi data bispektrum. Selanjutnya dilakukan tahapan ekstraksi ciri melalui proses *wrapping* dan transformasi kosinus. Teknik ini mampu memberikan akurasi terbaik dibandingkan dengan teknik sebelumnya, yaitu 1D-MFCC, 2D-MFCC tanpa GA, 2D-MFCC dengan GA, maupun kuantisasi skalar, untuk semua level *noise*, kecuali pada sinyal asli. Akurasi sistem dengan kuantisasi vektor pada sinyal asli, sinyal asli yang ditambah *noise* 20 dB, 10 dB, 5 dB dan 0 dB berturut-turut adalah 94.5%, 88.5%, 75.5%, 48% dan 26%.
 6. Dengan menggunakan bispektrum sebagai penciri sinyal dan kuantisasi vektor untuk merepresentasikan nilai bispektrum, maka bagian yang menentukan pada proses ekstraksi ciri adalah proses *wrapping*. Pemilihan parameter yang sesuai akan menentukan tingkat *robustness* sistem yang dihasilkan. Empat parameter utama pada proses *wrapping* adalah jenis peubah yang dikenakan *wrapping*, proporsi jumlah filter linear dengan filter logaritma, lebar filter linear dan lebar filter logaritma.
 7. Kualitas sinyal dapat dibedakan dengan besaran negatif dari logaritma perkalian absolut nilai autokorelasi hingga lag 21. Nilai besaran tersebut akan kecil untuk sinyal dengan kualitas baik dan makin besar dengan berkurangnya kualitas sinyal. Dari studi empiris, terlihat nilai batas besaran tersebut untuk mengatakan sinyal dengan kualitas baik (sedikit sekali terdistorsi oleh *noise*) atau kurang baik (banyak terdistorsi oleh *noise*) adalah berkisar dari 7 hingga 15.
 8. Untuk sinyal dengan kualitas relatif baik, sebaiknya digunakan teknik 1D-MFCC sebagai pengekstraksi ciri. Teknik ini selain memberikan akurasi yang baik, sekitar 99%, juga kompleksitasnya kecil. Sedangkan untuk sinyal dengan kualitas kurang baik (banyak terdistorsi oleh *noise*) yang

ditunjukkan dengan nilai negatif logaritma perkalian absolut autokorelasi hingga lag 21 yang besar (>15), sebaiknya digunakan teknik dengan bispektrum sebagai penciri sinyal, yang dikuantisasi vektor dan dilanjutkan dengan transformasi *wrapping* dan kosinus. Teknik ini relatif lebih *robust* dibanding teknik lainnya.

Beberapa saran untuk penelitian selanjutnya adalah :

1. Teknik 2D-MFCC (baik dengan atau tanpa GA) perlu dikaji lanjut dengan menggunakan bentuk-bentuk filter lainnya, misalnya yang dikembangkan oleh Slaney (filter mempunyai luasan yang konstan, sehingga tinggi filter tidak tetap, tetapi mengikuti lebarnya, makin lebar, maka makin kecil), juga dari aspek jumlah filternya (linear dan logaritma).
2. Perlu dilakukan kaji lanjut untuk menerapkan teknik 2D-MFCC dengan hasil ekstraksi nilai bispektrum yang direpresentasikan dengan kuantisasi vektor sebagai inputnya.
3. Perlu dilakukan studi mengenai sistem pemrosesan sinyal yang mengintegrasikan : proses penghapusan *noise*, ekstraksi ciri berbasis bispektrum dan berbagai teknik pengenalan pola terbaru, seperti seperti Fuzzy HMM, IO-HMM, 2D-HMM serta HMM/NN.
4. Perlu dilakukan riset untuk menerapkan berbagai teknik *grid computing* sehingga algoritme pemrosesan suara dengan statistik orde tinggi yang mempunyai kompleksitas yang besar menjadi layak diaplikasikan.
5. Perlu dilakukan riset untuk menerapkan teknik yang dikaji pada beberapa bidang terapan seperti : sistem keamanan berbasis suara, konversi suara ke teks, *information retrieval* (*searching* informasi berbasis suara, baik pada data teks maupun data audio), penerapan pada bidang kesehatan, juga dikembangkan pada analisis citra.

DAFTAR REFERENSI

- [JM00] Jurafsky dan J. H. Martin. *Speech and Language Processing : An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall Inc., New Jersey, 2000.
- [Den08] Deng, Li. Expanding the Scope of Signal Processing. *IEEE Signal Processing Magazine*, Vol. 25, No. 3, hal 2-4, Mei 2008.
- [Cam97] Joseph P Camphell. Speaker Recognition : A Tutorial. *Proceeding of the IEEE*, Vol. 85, No. 9, hal 1437 - 1460, September 1997.
- [Car95] J.S. Carmona. *A Hybrid System with Symbolic AI and Statistical Methods for Speech Recognition*. Thesis, University of Washington, Washington. 1995.
- [Rey02] Reynolds, D. Automatic Speaker Recognition Acoustics and Beyond. Tutorial note, MIT Lincoln Laboratory, 2002 .
- [BB08] Buono, A. dan B. Kusumoputro. A Problem in Data Variability on Speaker Identification System Using Hidden Markov Model. *Proceeding IASTED*, Insbruck, Pebruari 2008
- [BJB08]^a Buono, A., W. Jatmiko, and B. Kusumoputro. Development of 2D Mel-Frequency Cepstrum Coefficients Method for Processing Bispectrum Data as Feature Extraction Technique in Speaker Identification System. *Prociding of the International Conference on Artificial Intelegence and Its Applications (ICACIA)*. Depok, September 2008
- [BJB08]^b Buono, A., W. Jatmiko, dan B. Kusumoputro. Sistem Identifikasi Pembicara dengan MFCC sebagai Pengekstraksi Ciri dengan Hidden Markov Model sebagai Classifier. Laporan Teknis Penelitian, Lab. Kecerdasan Komputasional.
- [NP93] C.L. Nikeas dan A.P. Petropulu. *Higher Order Spectra Analysis : A Nonlinear Signal Processing Framework*. Prentice-Hall, Inc. New Jersey, 1993.
- [RN85] Raghuv eer, M. R. dan C. L. Nikeas. Bispectrum Estimation : A Parametric Approach. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. ASSP-33, No. 4, hal 1213-1230. Oktober 1985.
- [NM87] C.L. Nikias dan Mysore R.R. Bispektrum Estimation : A Digital Signal Processing Framework. *Proceeding of the IEEE*, Vol. 75, No. 7, hal. 869-890. Juli 1987.
- [AFCC05] Albera, L, A. Ferreol, P. Chevalier, dan P. Comon. ICAR: A Tool for Blind Source Separation Using Fourth-Order Statistics Only. *IEEE*

- Transactions on Signal Processing*, Vol. 53, No. 10, hal 3633-3643. Oktober 2005.
- [FB98] Fanany, M.I. dan B. Kusumoputro. 1998. *Bispectrum Pattern Analysis and Quantization to Speaker Identification*. Thesis Master Ilmu Komputer, Fasilkom Universitas Indonesia. 1998.
- [HB99] Hidayat, N. dan B. Kusumoputro. *Pengembangan Sistem Pengenal Suara Menggunakan Estimasi Trispektrum dan Kuantisasi Skalar*. Thesis Master Ilmu Komputer Fasilkom Universitas Indonesia. 1999.
- [TB00] Triyanto, A. dan B. Kusumoputro. Ekstraksi Ciri Pada Data Suara Menggunakan Spektra Orde Tinggi dan Kuantisasi Vektor untuk Identifikasi Pembicara Menggunakan Jaringan Neural Buatan. Thesis Program Master Ilmu Komputer, Fasilkom Universitas Indonesia. 2000.
- [Aka07] Al-Akaidi, M. *Fractal Speech Processing*. Cambridge University Press, 2007.
- [PM96] Proakis, J.G., dan D.G. Manolakis. *Digital Signal Processing : Principles, Algorithm, and Applications*. Edisi ke tiga, Prentice Hall, New Jersey, 1996.
- [NE02] M. Nilsson dan M. Ejnarsson. *Speech Recognition using Hidden Markov Model : Performance Evaluation in Noisy Environment*. Master Thesis, Departement of Telecommunications and Signal Processing, Blekinge Institute of Technology, Maret 2002.
- [Gan05] Todor D. Ganchev. *Speaker Recognition*. PhD Dissertation, Wire Communications Laboratory, Department of Computer and Electrical Engineering, University of Patras Greece.
- [Sla98] Slaney, M. Auditory Toolbox. Technical Report, Interval Research Corporation, versi ke 2. 1998.
- [TK84] H.M. Taylor dan S. Karlin. *An Introduction to Stochastic Modeling*. Academic Press, Inc., Florida, 1984.
- [DD96] Dugad, R. Dan U.B. Desai. A Tutorial on Hidden Markov Model. Technical Report, Departement of Electrical Engineering, Indian Institute of Technology, Bombay, 1996.
- [Rab89] L. Rabiner. A Tutorial on Hidden Markov Model and Selected Applications in Speech Recognition. *Proceeding IEEE*, Vol 77 No. 2. Pebruari 1989.
- [DHS01] Duda, R.O., P.E. Hart, dan D.G. Stork. *Pattern Classification*. Edisi Ke dua, John Wiley & Sons, INC., 2001.
- [GF08] Gilbert, M., dan J. Feng. Speech and Language Processing over the Web : Changing the Way People Communicate and Access Information. *IEEE Signal Processing Magazine*, Vol. 25, No. 3, hal 18-28, Mei 2008.

- [Fur97] Furui, S. Recent Advances in Speaker Recognition. *Pattern Recognition Letters 18, Elsevier*, 1997.
- [Pie03] Pieczynski, W. Pairwise Markov. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 25, No. 5. May 2003.
- [DP04] Derrode, S. Dan W. Pieczynski. Signal and Image Segmentation Using Pairwise Markov Chains. *IEEE Transactions on Signal Processing*, Vol. 52, No. 9, hal 2477-2489. September 2004.
- [PKM05] Pawar, R. V., P. P. Kajane, dan S. N. Mali. Speaker Identification Using Neural Networks. *Transactions on Engineering, Computing and Technology*. V7, ISSN 1305-5313, hal 429-433. Agustus 2005.
- [MG00]^a Mohamed, M. A. dan P. Gader. Generalized Hidden Markov Models-Part I: Theoretical Frameworks. *IEEE Transactions on Fuzzy Systems*, Vol. 8, No. 1, hal 67-81. Februari 2000.
- [MG00]^b Mohamed, M. A. dan P. Gader. Generalized Hidden Markov Models-Part II: Application to Handwritten Word Recognition. *IEEE Transactions on Fuzzy Systems*, Vol. 8, No. 1, hal 82-94. Februari 2000.
- [HT05] Hosseyndoost, F. dan M. Teshnehlab. Phoneme Classification and Phonetic Transcription Using a New Fuzzy Hidden Markov Model. *WSEAS Transaction on Computers*, Issue 6, Vol. 4, 541-547. Juni 2005.
- [PLL04] Pan, H., S. E. Levinson, dan T. Z. Liang. A Fused Hidden Markov Model with Application to Bimodal Speech Processing. *IEEE Transactions on Signal Processing*, Vol. 52, No. 3, hal 537-581. Maret 2004.
- [OA03] Othman, H. dan T. Aboulnasr. A Separable Low Complexity 2D HMM with Application to Face Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 25, No. 10. Oktober 2003.
- [CH03] Cornaz, C. dan U. Hunkeler. An Automatic Speaker Recognition System. Mini-Project.
http://www.ifp.uiuc.edu/~minhdo/teaching/speaker_recognition, akses : 15 Agustus 2005.
- [XG06] Xue, H. dan V. Govindaraju. Hidden Markov Models Combining Discrete Symbols and Continuous Attributes in Handwriting Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 28, No. 3. Maret 2006.
- [NPH05] Nel, E. M., J. A. D. Preez, dan B. M. Herbst. Estimating the Pen Trajectories of Static Signatures Using Hidden Markov Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 27, No. 11. November 2005.

- [VGS02] Valsan, Z., I. Gavut, dan B. Sabac. Statistical and Hybrid Methods for Speech Recognition in Romanian. *International Journal of Speech Technology*. Juli 2002.
- [GHP04] Ganapathiraju, A. J. E. Hamaker, dan J. Picone. Applications of Support Vector Machines to Speech Recognition. *IEEE Transactions on Signal Processing*, Vol. 52, No. 9, hal 2348-1362. September 2004.
- [Fau94] Laurene Fauset. *Fundamentals of Neural Networks*. Prentice Hall, Englewood Cliffs, 1994.
- [Bol79] Boll, S. F. Suppression of Acoustic Noise in Speech Using Spectral Subtraction. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. ASSP-27, No. 2, hal 113-120. April 1979.
- [Wid75] Widrow, B. et. al. Adaptive Noise Cancelling: Principles and Applications. *Proceeding of the IEEE*, Vol. 63. No. 12. hal 1691-1716. 1975.
- [BRHC04] Blanco, D., D. P. Ruiz, E. A. Hernandez dan M. C. Carrion. A Fourth-Order Stationarity and Ergodicity Condition for Harmonic Processes. *IEEE Transactions on Signal Processing*, Vol. 52, No. 6, hal 1641-1649. June 2004.
- [SMN01] Swami, A., J.M. Mendel dan C. L. Nikeas. *Higher-Order Spectral Analysis Toolbox : for use eith Matlab*. United Signals & Systems, Inc. 2001.
- [Zbi96] Zbigniew, M. *Genetic Algorithms + Data Structures = Evolution Programs*. Edisi ke tiga. Springer. 1996.



LAMPIRAN

Lampiran 1. Algoritma Forward [Rab89]

Algoritma ini untuk menduga peluang $P(O|\lambda)$, yaitu peluang munculnya barisan $O = O_1, O_2, O_3, \dots, O_T$, dari suatu HMM $\lambda = (A, B, \Pi)$ tertentu. Pada algoritma ini didefinisikan suatu simbol baru, $\alpha_t(i)$, yaitu peluang barisan observasi $O_1, O_2, O_3, \dots, O_t$, dan state S_i pada waktu t , yang dikenal sebagai variabel forward dan dirumuskan :

$$\alpha_t(i) = P(O_1, O_2, \dots, O_t, q_t = S_i | \lambda)$$

tahap 1 : inisialisasi

$$\alpha_1(i) = \pi_i b_i(O_1) \quad \text{untuk } 1 \leq i \leq N$$

tahap 2 : Induksi

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^N \alpha_t(i) a_{ij} \right] b_j(O_{t+1}) \quad \text{untuk } 1 \leq t \leq T-1 \text{ dan } 1 \leq j \leq N$$

tahap 3 : Terminasi

$$P(O | \lambda) = \sum_{i=1}^N \alpha_T(i)$$

Lampiran 2. Algoritma Backward [Rab89]

Pada algoritma ini didefinisikan suatu simbol baru, $\beta_t(i)$, yaitu peluang barisan observasi $O_{t+1}, O_{t+2}, O_{t+3}, \dots, O_T$, jika state pada periode t adalah S_i . Peubah backward ini dirumuskan sebagai :

$$\beta_t(i) = P(O_{t+1}, O_{t+2}, \dots, O_T | q_t = S_i, \lambda)$$

tahap 1 : Inisialisasi

$$\beta_T(i) = 1 \quad \text{untuk } 1 \leq i \leq N$$

tahap 2 : Induksi

$$\beta_t(i) = \sum_{j=1}^N a_{ij} b_j(O_{t+1}) \beta_{t+1}(j) \quad \text{untuk } t=T-1, T-2, \dots, 1 \text{ dan } 1 \leq i \leq N$$

tahap 3 : Terminasi untuk sembarang t , $1 \leq t \leq T-1$

$$P(O | \lambda) = \sum_{i=1}^N \sum_{j=1}^N \alpha_1(i) a_{ij} b_j(O_2) \beta_2(j)$$

Lampiran 3. Algoritma Viterbi [Rab89]

Algoritma ini ingin menemukan barisan state yang optimum, $Q = \{q_1, q_2, q_3, \dots, q_T\}$, kalau diberikan barisan observasi $O = \{O_1, O_2, O_3, \dots, O_T\}$. Pada algoritma ini didefinisikan peubah baru, $\delta_t(i)$, sebagai skor terbaik (peluang tertinggi) yang dihitung pada path dari state periode pertama hingga state S_i pada periode t .

$$\delta_t(i) = \max_{q_1, q_2, \dots, q_{t-1}} P(q_1, q_2, \dots, q_t = i, O_1, O_2, \dots, O_t | \lambda)$$

Oleh karena itu, nilai tersebut secara induktif dapat dirumuskan sebagai :

$$\delta_{t+1}(j) = \left[\max_i \delta_t(i) a_{ij} \right] \cdot b_j(O_{t+1})$$

Prosedur lengkap algoritma Viterbi adalah :

1. Inisialisasi, untuk $1 \leq i \leq N$:

$$\delta_1(i) = \pi_i b_i(O_1) \text{ dan } \psi_1(i) = 0$$

2. Proses rekursi :

$$\delta_t(j) = \left[\max_{1 \leq i \leq N} \delta_{t-1}(i) a_{ij} \right] \cdot b_j(O_t) \text{ untuk } 2 \leq t \leq T \text{ dan } 1 \leq j \leq N$$

$$\psi_t(j) = \operatorname{argmax}_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}] \text{ untuk } 2 \leq t \leq T \text{ dan } 1 \leq j \leq N$$

3. Terminasi :

$$P^* = \max_{1 \leq i \leq N} \delta_T(i) \text{ dan } q_T^* = \operatorname{argmax}_{1 \leq i \leq N} \delta_T(i)$$

4. Backtracking :

$$q_t^* = \psi_{t+1}(q_{t+1}^*) \text{ untuk } t = T-1, T-2, \dots, 1.$$

Lampiran 4. Algoritma K-Means [DD99]

Pada metode ini, parameter dari model HMM $\lambda=(A,B,\Pi)$ disesuaikan untuk memaksimumkan $P(O,Q|\lambda)$ dengan Q adalah barisan state yang optimum hasil dari solusi problem 2.

Metode ini menduga parameter λ^{k+1} dari λ^k , sedemikian sehingga $P(O,Q_k^*|\lambda^k) \leq P(O,Q_{k+1}^*|\lambda^{k+1})$, dan Q_k^* adalah barisan hidden state yang optimum dengan barisan observasi $O=O_1, O_2, O_3, \dots, O_T$. O_i merupakan vektor pengamatan berdimensi d . Oleh karena itu, kriteria yang dipakai adalah fungsi :

$$P(O,Q^*|\lambda) = \max_{vQ} P(O,Q|\lambda)$$

Algoritma segmental K-means adalah sebagai berikut :

Tahap 1 : secara acak dipilih N observasi, dan kelompokkan semua observasi data training (ada ωT observasi) ke salah satu dari N observasi yang dipilih secara acak tersebut, dengan kriteria jarak Euclid minimum. Dari sini terbentuk N kluster, yang masing-masing sebagai sebuah state.

Tahap 2 : Hitung nilai peluang awal dan peluang transisi :

$$\text{untuk } 1 \leq i \leq N : \hat{\pi}_i = \frac{\text{banyaknya kejadian } \{O_t \in i\}}{\text{total kejadian } O_t (\omega)}$$

untuk $1 \leq i \leq N$ dan $1 \leq j \leq N$:

$$\hat{a}_{ij} = \frac{\text{banyaknya kejadian } \{O_t \in i, O_{t+1} \in j\} \text{ untuk semua } t}{\text{banyaknya kejadian } \{O_t \in i\} \text{ untuk semua } t}$$

Tahap 3 : Hitung vektor rata-rata dan matriks kovarian untuk setiap $1 \leq i \leq N$:

$$\hat{\mu}_i = \frac{1}{N_i} \sum_{O_t \in i} O_t \quad \text{dan} \quad \hat{V}_i = \frac{1}{N_i} \sum_{O_t \in i} (O_t - \hat{\mu}_i)^T (O_t - \hat{\mu}_i)$$

Tahap 4 : Hitung matriks kepadatan (kekuatan) observasi dari setiap data training untuk setiap state (dalam hal ini diasumsikan berdistribusi Normal), $1 \leq i \leq N$:

$$\hat{b}_i(O_t) = \frac{1}{(2\pi)^{d/2} |\hat{V}_i|^{1/2}} \exp\left[-1/2(O_t - \hat{\mu}_i)^T \hat{V}_i^{-1} (O_t - \hat{\mu}_i)\right]$$

Tahap 5 : Temukan barisan state (path) optimal, Q^* , dengan algoritma pada Viterbi, untuk setiap data training dari HMM $\hat{\lambda}_i = (\hat{A}_i, \hat{B}_i, \hat{\pi}_i)$ yang dihitung dari tahap 2 s/d 4 di atas. Jika suatu vektor observasi ke r , O_r , pada training ke k , diklasterkan ke state j , sedangkan hidden state ke r pada barisan state optimal yang diperoleh dari training ke k adalah state p ($p \neq j$), maka O_r di-reassign ke state p dari state j .

Tahap 6 : Jika tidak ada observasi yang di-reassign ke state baru, maka selesai. Jika tidak, maka ulangi langkah 2 s/d 6.

Lampiran 5. Formula Baum-Welch reestimation [Rab89]

Pada metode ini, parameter dari model HMM $\lambda=(A,B,\Pi)$ disesuaikan untuk meningkatkan $P(O|\lambda)$ hingga maksimum dicapai. Pada persamaan (2.4.9.) terlihat bahwa penghitungan $P(O|\lambda)$ mencakup penjumlahan nilai peluang $P(O,Q|\lambda)$ untuk semua kemungkinan barisan Q .

Algoritma Baum-Welch reestimation [DHS01]

Parameter HMM yang diduga melalui algoritma ini akan memaksimumkan nilai $P(O|\lambda)$. Beberapa notasi yang dipakai dalam algoritma ini adalah :

a. forward dan backward : $\alpha_t(i)$ dan $\beta_t(i)$

$$c. \gamma_t(i) = P(q_t = S_i | O, \lambda) = \frac{\alpha_t(i)\beta_t(i)}{P(O|\lambda)} = \frac{\alpha_t(i)\beta_t(i)}{\sum_{i=1}^N \alpha_t(i)\beta_t(i)}$$

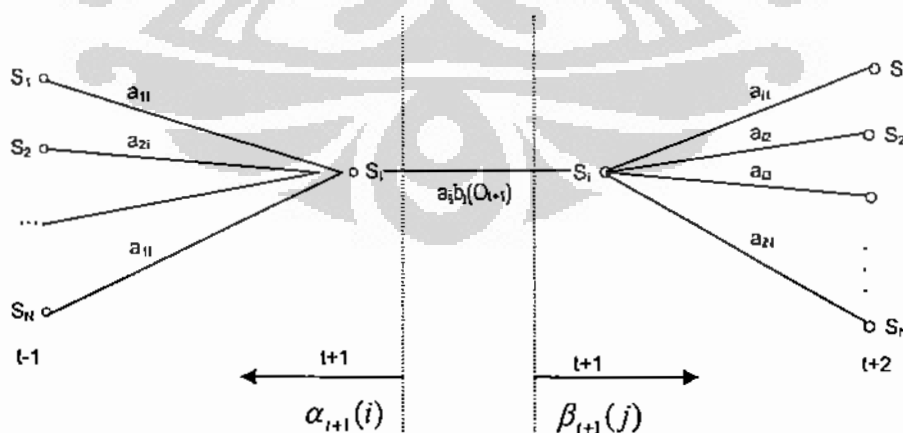
d. $\xi_t(i,j) = P(q_t=S_i, q_{t+1}=S_j|O,\lambda)$, yaitu peluang berada pada state i pada periode t dan state j pada $t+1$, yang dirumuskan sebagai :

$$\xi_t(i,j) = \frac{\alpha_t(i)a_{ij}b_j(O_{t+1})\beta_{t+1}(j)}{P(O|\lambda)} = \frac{\alpha_t(i)a_{ij}b_j(O_{t+1})\beta_{t+1}(j)}{\sum_{i=1}^N \sum_{j=1}^N \alpha_t(i)a_{ij}b_j(O_{t+1})\beta_{t+1}(j)}$$

Dalam hal ini :

$\sum_{i=1}^{T-1} \xi_t(i,j)$ adalah banyaknya transisi yang terjadi dari state S_i ke S_j

$\sum_{i=1}^{T-1} \gamma_t(i)$ adalah banyaknya transisi yang terjadi dari state S_i



Pendugaan parameter HMM adalah sebagai berikut :

a. Inisialisasi : $z=0$, a_{ij} , b_{jk} , dan data training V^T . (pemberian nilai awal parameter HMM dapat dilakukan secara acak, atau bisa juga menggunakan tahapan 1 s/d 5 pada algoritma K-means)

b. Selama kriteria pemberhentian belum dipenuhi, lakukan :

1. $z=z+1$

2. $\bar{\pi}_i(z) = \gamma_i(i)$

$$3. \bar{a}_{ij}(z) = \frac{\sum_{t=1}^{T-1} \xi_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)}$$

4a. Untuk observasi bersifat diskret : $\bar{b}_{jk}(z) = \frac{\sum_{t=1, O_t=v_k}^T \gamma_t(j)}{\sum_{t=1}^T \gamma_t(j)}$

4b. Untuk observasi bersifat kontinyu :

Observasi $B = \{b_j(\cdot)\}$ berdistribusi campuran (Mixture Distribution) dari c komponen yang dirumuskan sebagai berikut :

$$b_j(O_t) = \sum_{m=1}^c \omega_{jm} f_{jm}(O_t)$$

Dalam hal ini $\omega_{jm} > 0$ dan $\sum_{m=1}^c \omega_{jm} = 1$ untuk $1 \leq j \leq N$, adalah proporsi komponen ke m pada state j . Untuk $f_{jm}(O_t) = N(O_t, \mu_{jm}, V_{jm})$, maka diperlukan nilai inisial untuk proporsi, rata-rata dan matriks kovarian untuk menghitung nilai $b_j(O_t)$ untuk $1 \leq j \leq N$ dan $1 \leq t \leq T$. Formula pendugaan kembali adalah sebagai berikut :

$$\omega_{jm} = \frac{\sum_{t=1}^T \gamma_t(j, m)}{\sum_{k=1}^c \sum_{t=1}^T \gamma_t(j, k)}, \quad \mu_{jm} = \frac{\sum_{t=1}^T \gamma_t(j, m) O_t}{\sum_{t=1}^T \gamma_t(j, m)}$$

$$V_{jm} = \frac{\sum_{t=1}^T \gamma_t(j, m) (O_t - \mu_{jm})(O_t - \mu_{jm})'}{\sum_{t=1}^T \gamma_t(j, m)}$$

5. Kriteria henti :

$$\max_{i,j,k} [(a_{ij}(z) - a_{ij}(z-1)), (b_{jk}(z) - b_{jk}(z-1))] < \theta$$

a. terminasi : $\pi_i = \pi_i(z)$, $a_{ij} = a_{ij}(z)$, dan $b_{ij} = b_{ij}(z)$.

Lampiran 6. Tahapan Proses Kuantisasi Vektor

Tahap 1. Pembentukan bispektrum yang akan disampling

- 1.1. Ambil satu suara sembarang dari setiap pembicara
- 1.2. Baca frame demi frame untuk setiap suara tersebut dan hitung bispektrum frame tersebut dan normalisasikan dengan nilai maksimumnya
- 1.3. Rata-ratakan bispektrum dari T frame untuk setiap pembicara, sehingga diperoleh N bispektrum rata-rata
- 1.4. Rata-rata bispektrum dari N pembicara yang tersebut untuk memperoleh satu bispektrum rata-rata yang berdimensi 128×128 .

Tahap 2. Melakukan sampling titik (f_1, f_2) berdasar nilai bispektrum sampai diperoleh K titik pada bidang $F_1 \times F_2$

Selama jumlah titik masih kurang dari K :

- 2.1. bangkitkan bilangan random r_1 ($0 \leq r_1 \leq 2\pi$) dan r_2 ($0 \leq r_2 \leq 1$)
- 2.2. bangkitkan bilangan random $r_3 \in [0, 1]$
- 2.3. tentukan titik (f_1, f_2) yang terdekat dengan (r_1, r_2) , misalnya (f_{1i}, f_{2i})
- 2.4. jika $r_3 < \text{bsp}(f_{1i}, f_{2i})$, maka titik (r_1, r_2) diambil, dan dimasukkan ke dalam set B

Tahap 3. Menentukan P channel

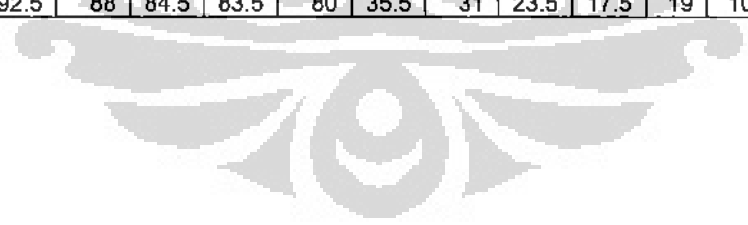
- 3.1. lakukan proses k-means clustering dengan $k=P$.
- 3.2. penamaan klaster diurut sesuai dengan jaraknya terhadap pusat sumbu

Tahap 4. Menentukan nilai hasil kuantisasi

- 4.1. nilai channel ke i ($i=1, 2, 3, \dots, P$) dihitung berdasar bispektrum titik pada bidang $F_1 \times F_2$ yang masuk pada klaster ke I, sesuai staitik yang diinginkan (rata-rata, median, rata-rata bispektrum di atas persentil 75 atau rata-rata di atas persentil 90).

Lampiran 7a. Hasil Akurasi Sistem dengan Kuantisasi Vektor dengan Wrapping pada Posisi Urutan Channel dan Transformasi Kosinus

D	G	ul.	asli			+noise 20			+noise 10			+noise 5			+noise 0		
			r1	r2	r3	r1	r2	r3	r1	r2	r3	r1	r2	r3	r1	r2	r3
d1	g1	1	70.5	72	70.5	67	68.5	68.5	46.5	52.5	48	22	27	27	12	11.5	12.5
		2	76	76.5	74.5	69	70	69	47.5	49	46.5	25	23	25	11	11	10.5
		3	80.5	79.5	79.5	76.5	77	77	51	52	47.5	26	24	26	12.5	12.5	15
	g2	1	72.5	72	72.5	70	70.5	68.5	55.5	56.5	53.5	29.5	31	27	18.5	15.5	13
		2	77.5	78	75.5	70.5	74	72	54	59	54	22.5	25	23	10	12	10.5
		3	82.5	82	81	78	78.5	77	57	57	56.5	30	28	29	15.5	15	20.5
	g3	1	77	77.5	75	74	74	72	63	61.5	55.5	34	34	28	14	15.5	13.5
		2	81	81.5	77	76.5	77	73.5	66.5	64.5	54.5	28.5	26	23	10.5	11	10
		3	85	85	82.5	79.5	79	76.5	59	60.5	55	34.5	30	31	16.5	14.5	15
d2	g1	1	76	76.5	75	71.5	73	72.5	62	63.5	60.5	33	37	31	12.5	11	11.5
		2	81.5	83.5	83.5	78	77.5	77	61.5	62	60.5	28	33	29	10.5	11	10.5
		3	85	85	83	83.5	82.5	81.5	62	66	64.5	31	33	31	11.5	12	11.5
	g2	1	80	80	77.5	76	77.5	72.5	70	66	65	33.5	33	35	10.5	11	11.5
		2	84	85	82	77.5	79	77	72.5	74	63.5	38.5	40	33	15	12	11
		3	86	84.5	84.5	84	82	81	72	70.5	67.5	35.5	35	33	13.5	11	12
	g3	1	82	80.5	80.5	77	76.5	76.5	33.5	35.5	68	20.5	23	31	13.5	15.5	11
		2	89	89	84	85.5	83.5	79	43	43.5	75	21.5	24	31	10	10	12
		3	87	87.5	86	82.5	83	84.5	24	27	70	10	11	35	10	10	12.5
d3	g1	1	79.5	80.5	79.5	77	81	77.5	70	66.5	69.5	43	36	43	15.5	9.5	13
		2	83	86.5	82.5	78.5	82.5	79.5	74.5	72	72	31.5	36	35	15.5	11	12.5
		3	85.5	85	85.5	84.5	83.5	85.5	74.5	69.5	75.5	36	28	39	12	10.5	11.5
	g2	1	82	84.5	83	80.5	78.5	79.5	35	29.5	58.5	21	16	24	11	10	10
		2	90	93	86.5	86.5	85	81.5	31	26	71.5	10	11	38	10	10	13.5
		3	87.5	89	85.5	82.5	76.5	86.5	17.5	27	61.5	10.5	15	23	10	10	10
	g3	1	94.5	91	82.5	73	82.5	78	30	22	32.5	25	16	18	13	10	10
		2	87	93	88.5	80	85	84	21.5	35	31	12.5	17	11	10	10	10
		3	92	92.5	88	84.5	83.5	80	35.5	31	23.5	17.5	19	10	14.5	11	10



Lampiran 7b. Hasil Akurasi Sistem dengan Kuantisasi Vektor dan Wrapping pada Jarak Channel dan Transformasi Kosinus

D	G	ul.	asli			+noise 20			+noise 10			+noise 5			+noise 0		
			r1	r2	r3	r1	r2	r3	r1	r2	r3	r1	r2	r3	r1	r2	r3
d1	g1	1	95	95	93.5	88	84.5	84.5	42.5	34.5	33.5	25	20	21	12	11.5	14
		2	93	96	96.5	87	85	89	29.5	40	41	18	26	25	12	12.5	13
		3	94	95	93.5	90	83	86.5	41	26.5	32	26.5	16	18	18	15	11
	g2	1	97.5	95	92.5	79	88.5	83.5	32	27	31.5	21.5	22	17	11.5	13	11.5
		2	98	98.5	96.5	70	70.5	80.5	31	30.5	33.5	21	24	24	12.5	15.5	13
		3	95.5	97	96	84	71.5	82.5	28	20.5	25	10	13	16	10	10	13.5
	g3	1	95.5	94.5	96	62.5	66	81.5	21	19	33.5	12.5	10	22	11	9.5	12
		2	95.5	95.5	95	48	62.5	81.5	28.5	27	23.5	23	15	18	15	10.5	16
		3	97.5	94	95.5	48.5	57	83	24.5	31	28	13	24	16	8.5	15	11
d2	g1	1	96	95.5	95	65.5	49.5	59	28	28.5	26	23.5	26	15	13.5	18.5	10
		2	96.5	97	95	62	57.5	62.5	30	29	30	21	23	20	11.5	10	16
		3	97	98.5	98	58	52.5	55.5	26	27	28	21	26	20	14.5	10.5	12.5
	g2	1	98	97	95.5	65	60	47	18.5	16.5	29.5	13.5	16	27	11	12.5	14.5
		2	97	98	96	65.5	67	51	25	12	28.5	19	11	24	10.5	11	10
		3	98	98.5	98.5	62.5	54.5	54.5	24.5	12.5	30	16	11	19	10	10	10
	g3	1	97.5	96	96.5	44.5	36	65	10	26	20.5	10	20	19	10	10.5	12.5
		2	97.5	97.5	97	32.5	46	71.5	13	10	18.5	11.5	10	19	10	10	11
		3	97	98.5	99.5	36	34	56	19.5	14	22.5	18.5	13	19	10	11	10.5
d3	g1	1	99	98.5	97	54	40.5	64	28.5	15.5	18.5	20.5	13	10	10.5	13	9.5
		2	97.5	98	96.5	55.5	46	68.5	24	10	14.5	18	10	14	11	10	16
		3	99	99.5	99.5	64	36	61	19.5	15.5	22.5	16.5	12	19	16	10	12.5
	g2	1	98	97.5	97.5	49.5	41	35.5	10	16	16.5	10	20	11	10	13.5	10
		2	99	98.5	98	38	47	48.5	14.5	10	10	12.5	10	10	11.5	10	10
		3	98	99	99	36.5	33.5	34.5	23	12.5	14.5	17.5	11	12	10	10	10
	g3	1	98.5	99	98	39	37	47.5	10.5	15	10	10	11	10	10	10	10
		2	96.5	97.5	99.5	38.5	38.5	35	19.5	10	13.5	19.5	10	12	11.5	10	10
		3	96.5	98.5	97.5	38.5	45	36	19.5	10	21	19.5	10	21	11.5	10	11



**Lampiran 8a. Analisis Ragam untuk Akurasi Sistem dengan Teknik
Kuantisasi Vektor pada Data Sinyal Asli + Noise 20 dB**

Sumber	Jumlah Kuadrat	Derajat Bebas	Rataan Jumlah kuadrat	F	Sig.
Perlakuan	38893.660(a)	53	733.843	32.548	.000
Rataan umum	756313.340	1	756313.340	33544.904	.000
W	14926.080	1	14926.080	662.019	.000
D	4198.077	2	2099.039	93.099	.000
G	1142.818	2	571.409	25.344	.000
R	321.244	2	160.622	7.124	.001
W * D	11425.466	2	5712.733	253.378	.000
W * G	2262.892	2	1131.446	50.183	.000
D * G	297.043	4	74.261	3.294	.014
W * D * G	480.840	4	120.210	5.332	.001
W * R	586.688	2	293.344	13.011	.000
D * R	100.756	4	25.189	1.117	.352
W * D * R	123.293	4	30.823	1.367	.250
G * R	752.821	4	188.205	8.348	.000
W * G * R	847.173	4	211.793	9.394	.000
D * G * R	681.068	8	85.133	3.776	.001
W * D * G * R	747.401	8	93.425	4.144	.000
Galat	2435.000	108	22.546		
Total Terkoreksi	41328.660	161			

(a) R Kuadrat = .941 (Adjusted R Kuadrat = .912)

Lampiran 8b. Analisis Ragam untuk Akurasi Sistem dengan Teknik Kuantisasi Vektor pada Data Sinyal Asli + Noise 10 dB

Sumber	Jumlah Kuadrat	Derajat Bebas	Rataan Jumlah kuadrat	F	Sig.
Perlakuan	62901.236(a)	53	1186.816	60.394	.000
Rataan umum	236786.681	1	236786.681	12049.456	.000
W	38781.125	1	38781.125	1973.470	.000
D	4016.787	2	2008.394	102.202	.000
G	3888.000	2	1944.000	98.925	.000
R	715.287	2	357.644	18.200	.000
W * D	1017.861	2	508.931	25.898	.000
W * G	491.815	2	245.907	12.514	.000
D * G	3751.296	4	937.824	47.723	.000
W * D * G	4689.074	4	1172.269	59.654	.000
W * R	235.287	2	117.644	5.987	.003
D * R	593.315	4	148.329	7.548	.000
W * D * R	492.741	4	123.185	6.269	.000
G * R	437.769	4	109.442	5.569	.000
W * G * R	118.843	4	29.711	1.512	.204
D * G * R	1583.157	8	197.895	10.070	.000
W * D * G * R	2088.880	8	261.110	13.287	.000
Galat	2122.333	108	19.651		
Total Terkoreksi	65023.569	161			

(a) R Kuadrat = .967 (Adjusted R Kuadrat = .951)

Lampiran 9. Daftar Paper yang Dipublikasikan

1. Buono, A. dan B. Kusumoputro. Pengembangan Model HMM Berbasis Maksimum Lokal Menggunakan Jarak Euclid untuk Sistem Identifikasi Pembicara. *Prosiding pada workshop NACSIIT*, Depok, Januari 2007.
2. Buono, A. and B. Kusumoputro. Speaker Identification System Using Hidden Markov Model Based on Local Distribution Membership. *Prociding of the ICSIIT workshop*, Bali, July 2007
3. Buono, A. and B. Kusumoputro. A Problem in Data Variability on Speaker Identification System Using Hidden Markov Model. *Prociding of the Conference on Artificial Intelligence and Application (AIA)*, IASTED , Innsbruck, Pebraury 2008.
4. Buono, A., W. Jatmiko, and B. Kusumoputro. Genetics Algorithm for 2D-MFCC Filter Development in Speaker Identification System Using HMM. *Prociding of the International Conference on Artificial Intelegence and Its Applications (ICACIA)*. Depok, September 2008.
5. Buono, A., W. Jatmiko, and B. Kusumoputro. Development of 2D Mel-Frequency Cepstrum Coefficients Method for Processing Bispectrum Data as Feature Extraction Technique in Speaker Identification System. *Prociding of the International Conference on Artificial Intelegence and Its Applications (ICACIA)*. Depok, September 2008.
6. Buono, A., W. Jatmiko, and B. Kusumoputro. Perluasan Metode MFCC 1D ke 2D Sebagai Ekstraksi Ciri Pada Sistem Identifikasi Pembicara Menggunakan HMM. *Jurnal Makara, Sains*, Vol. 13, No. 1, April 2009:85-95, Depok, Universitas Indonesia.