PERBANDINGAN IDENTIFIKASI PEMBICARA MENGGUNAKAN MFCC DAN SBC DENGAN CIRI PENCOCOKAN LBG-VQ

Purwono Prasetyawan
Program Studi Teknik Informatika, STMIK Teknokrat
Jl. H. Zaenal Abidin Pagaralam 9-11 Labuhanratu, Bandarlampung 35142
Telp. (0721) 774061
E-mail: purwono.prasetyawan@teknokrat.ac.id

ABSTRAKS
Biometrik merupakan teknologi untuk menganalisa fisik dan prilaku manusia yang digunakan dalam autentifikasi. Salah satu karakteristik perilaku yang terkait dengan seseorang adalah suara. Suara seseorang dapat dikenali berdasarkan karakteristik sinyal suara orang tersebut. Ada beberapa metode dalam mengenali suara pembicara, diantaranya yaitu dengan Mel Frequency Cepstrum Calculation (MFCC) dan Subband Based Cepstral (SBC). Penelitian ini mencari efektifitas penggunaan ekstrasi ciri MFCC dan SBC dengan ciri pencocokan LBG Vector Quantization. Metode ekstrasi ciri yang efektif akan diuji-cobakan untuk identifikasi pembicara secara realtime. Hasil yang didapatkan dari penelitian ini adalah dengan nilai koefisien MFCC 32 lebih efektif dari pada SBC secara akurasi dan kecepatan proses identifikasi pembicara baik text-dependent maupun text-independent. Adapun hasil pengujian identifikasi pembicara secara realtime menggunakan MFCC masih belum memuaskan karena akurasi pengenalannya masih dibawah 70%.

Kata Kunci: identifikasi pembicara, MFCC, SBC, LBG-VQ

1. PENDAHULUAN

1.1 Latarbelakang

Biometrik merupakan suatu teknologi untuk menganalisa fisik dan prilaku manusia yang digunakan dalam autentifikasi. Autentifikasi dengan biometrik sangat khas, karakteristik yang terukur digunakan untuk mengenali individu. Dua kategori pengenalan biometrik yaitu karakteristik fisiologis dan perilaku. Karakteristik fisiologis berhubungan dengan bentuk fisik tubuh, seperti sidik jari, wajah, telapak tangan, DNA dan iris. Karakteristik perilaku terkait dengan perilaku seseorang, seperti ritme mengetik, tanda tangan, dan suara.

Suara seseorang dapat dikenali berdasarkan karakteristik sinyal suara orang tersebut. Ada 2 (dua) tahap yang harus dilakukan dalam mengidentifikasi pembicara. Tahap pertama adalah *training phase* yaitu memasukan sampel sinyal suara yang diuji untuk mendapatkan ciri karakteristik individu (*reference model*) dan tahap kedua *testing phase* yaitu sinyal suara yang masuk dicocokan dengan *reference model* sehingga didapat keputusan pengenalan/identifikasi pembicara.

Identifikasi pembicara mempunyai banyak metode yang digunakan, baik metode dalam ekstraksi ciri dan dalam membangun ciri pencocokan. MFCC merupakan metode yang umum dalam ekstraksi ciri/karakteristik suara dan menurut Sarikaya (1998) SBC lebih baik dari pada MFCC dengan data suara dari TIMIT. Adapun menurut Hosan (2011) Vector Quantization merupakan metode membangun ciri pencocokan yang efisien dalam komputasi dan sederhana, tidak komplek seperti pada Gaussian Mixture Model (GMM).

1.2 Rumusan masalah

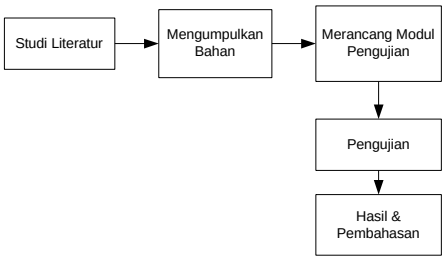
Rumusan masalah dalam penelitian ini adalah mencari efektifitas identifikasi pembicara menggunakan ekstrasi ciri MFCC apakah lebih baik daripada SBC dengan merubah koefisien masing-masing dan merubah jumlah centroid pada ciri pencocokan LBG-VQ?

1.3 Tujuan

Tujuan dalam penelitian ini adalah menguji metode ekstrasi ciri yang efektif antara MFCC dan SBC pada identifikasi pembicara secara *realtime*.

1.4 Desain Penelitian

Desain penelitian ini menggambarkan alur kerja dalam melakukan penelitian yang dapat ditunjukan pada Gambar 1 berikut.



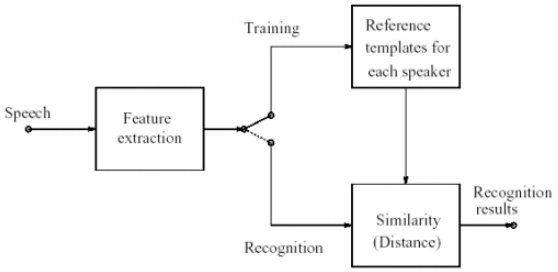
Gambar 1. Desain penelitian

Alur kerja dalam penelitian ini bersifat linier dimulai dari studi literatur, mengumpulkan bahan, merancang modul pengujian untuk mencari efektifitas ekstrasi ciri dari MFCC dan SBC serta pengujian secara *realtime* sehingga didapat hasil dan pembahasan.

1.5 Landasan Teori

1.5.1 Identifikasi Pembicara

Menurut Hosaan (2011) identifikasi pembicara *(Speaker identification)* dapat didefinisikan sebagai proses memilih pembicara yang mempunyai karakteristik ciri suara yang mendekati sama dengan suara masukan. Suara masukan diekstrasi cirinya untuk dibandingkan dengan beberapa referensi model pembicara yang ada dan dicari yang mana yang mendekati sama untuk diputuskan sebagai pembicara suara masukan tadi. Proses ini membandingkan 1:N model referensi pembicara, dan umumnya terdiri dari 3 (tiga) bagian utama, seperti terlihat pada Gambar 2.



Gambar 2. Proses identifikasi pembicara
(Singh, 2003)

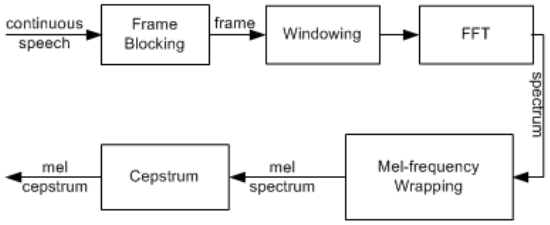
Bagian pertama yaitu *Feature extraction*, disini sinyal suara masukan (*input speech*) diekstrasi ciri/karakteristik sinyalnya dengan menggunakan salah satu metode ekstrasi ciri seperti LPC, MFCC, LFCC, SBC dan lainnya. Bagian kedua adalah bagian *Training phase*, pada bagian ini ciri sinyal suara masukan tadi yang berupa deretan vektor akustik, diklasifikasikan kemudian disimpan sebagai model referensi pembicara, dalam pengklasifikasian model referensi pembicara ini dapat menggunakan kuantisasi vektor LBG-VQ. Bagian ketiga adalah *Recognition* atau *Testing phase*, mencari kemiripan ciri suara dengan cara mengukur jarak (*distortion*) antara beberapa referensi model pembicara yang ada dengan sinyal suara masukan yang akan dikenali. Jarak yang paling dekat atau kecil yang diputuskan sebagai pembicara dari sinyal suara masukan tadi. Bila diterapkan pada verifikasi pembicara (*Speaker verification*) maka mengukur jarak hanya antara sinyal masukan dengan model referensi pembicara yang diakui. Jika jarak lebih kecil dari ambang batas (*threshold*) yang telah ditentukan maka pembicara tersebut diterima bila lebih besar ditolak.

Proses mengenali pembicara dengan masukan suara yang menggunakan teks bebas (*random*) dan tidak sesuai dengan suara teks yang di-*enrollment* atau di-*training* sebelumnya, dikenal dengan istilah *text-independent speaker identification*. Pengenalan pembicara dengan menggunakan teks suara yang sama antara *training phase* dan *testing phase* disebut dengan *text-dependent speaker identification*. Dalam hal ini dapat diimplementasikan untuk autentifikasi dengan menggunakan *password*, no kartu, kode PIN dan lainnya yang dapat memberikan hak akses ke suatu sistem bilamana suara pembicara diterima dengan teks yang sama.

1.5.2 Ekstrasi Ciri dengan MFCC

Sinyal suara merupakan sinyal *quasi-stationary*, yaitu ketika diperiksa selama periode yang cukup singkat (5-100 milidetik) karakteristiknya cukup stationer. Namun selama jangka waktu yang lama (di urutan 1/5 detik atau lebih) karakteristik sinyal berubah yang mencerminkan perbedaan suara masukan yang diucapkan. Oleh karena itu, menurut Do (2013) waktu singkat analisis spektral merupakan cara yang paling umum dan tepat untuk mengekstrasi ciri suara masukan.

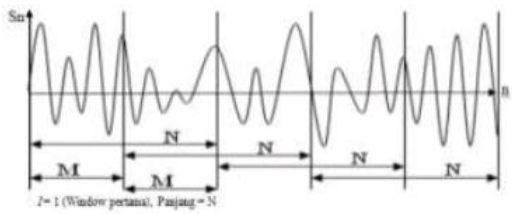
Ekstrasi ciri sinyal suara menggunakan MFCC didasarkan atas variasi bandwith terhadap frekuensi pada telinga manusia yang merupakan filter, yang bekerja secara linier pada frekuensi rendah dan bekerja secara logaritmik pada frekuensi tinggi. Filter ini digunakan untuk menangkap karakteristik fonetis penting dari sinyal suara masukan atau ucapan. Karakteristik ini digambarkan dalam skala *mel*-frekuensi, yang merupakan frekuensi linier di bawah 1000 Hz dan frekuensi logaritmik di atas 1000 Hz. Diagram proses MFCC dapat dilihat pada Gambar 3.



Gambar 3. Diagram proses MFCC

Frame Blocking merupakan proses pembagian suara menjadi beberapa *frame* dan satu *frame* terdiri dari beberapa sampel. Proses ini diperlukan untuk membentuk sinyal suara yang *non-stationary* menjadi sinyal suara yang *quasi-stationary* sehingga dapat diubah dari domain waktu ke domain frekuensi dengan transformasi Fourier. Dalam langkah ini sinyal suara terus-menerus dibagi menjadi beberapa *frame* yang berisi N-sampel, dengan *frame* yang berdekatan dipisahkan oleh M (M<N). *Frame* pertama berisi sampel N pertama. *Frame* kedua dimulai M sampel setelah permulaan *frame* pertama, sehingga *frame* kedua ini *overlap* terhadap *frame* pertama sebanyak N-M sampel dan seterusnya, *frame* ketiga dimulai M sampel setelah *frame* kedua (juga overlap sebanyak N-M sampel terhadap *frame* kedua). Proses ini berlanjut sampai seluruh suara ucapan tercakup dalam satu atau lebih *frame.* Proses ini tampak pada Gambar 4, *Sn* adalah

nilai sampel yang dihasilkan dan *n* menunjukan urutan sampel yang akan diproses.



Gambar 4. Proses *frame blocking*
(Setiawan, 2011)

Menurut Putra dan Resmawan (2011) panjang daerah *overlap* yang umum digunakan adalah kurang lebih 30% sampai 50% dari panjang *frame*. *Overlaping* dilakukan untuk menghindari hilangnya karakteristik suara pada perbatasan perpotongan *frame*.

Langkah selanjutnya adalah *windowing* setiap *frame*. Hal ini dilakukan untuk meminimalisasi diskontinuitas sinyal pada permulaan dan akhir setiap *frame*. Konsepnya adalah meruncingkan sinyal ke angka nol pada permulaan dan akhir setiap *frame*. Dengan mendefinisikan window sebagai $\omega(n), 0 \leq n \leq N$, dimana N adalah jumlah sampel dalam setiap *frame*, maka hasil *windowing* sinyal:

$$y_l = x_l(n)\,\omega(n), 0 \leq n \leq N \quad (1)$$

dengan;
y_l adalah nilai sampel sinyal hasil windwing,
$x_l(n)$ adalah nilai sampel dari *frame* sinyal ke-i,
ω merupakan fungsi window
N merupakan panjang *frame*.

Ada banyak fungsi window, namun yang sering digunakan dalam *Speaker identification* adalah *Hamming window*. Fungsi window ini menghasilkan *sidelobe level* yang tidak terlalu tinggi (kurang lebih -43 dB), selain itu *noise* yang dihasilkan pun tidak terlalu besar (Putra & Resmawan, 2011).

Fungsi hamming window dengan N merupakan panjang *frame* adalah sebagai berikut:

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

Pengolahan selanjutnya menggunakan *Fast Fourier Transform* (FFT), yang mengubah setiap *frame* N sampel dari domain waktu ke domain frekuensi. FFT adalah algoritma cepat untuk menerapkan Descrete Fourier Transform (DFT) yang didefinisikan pada set N sampel xn, sebagai berikut.

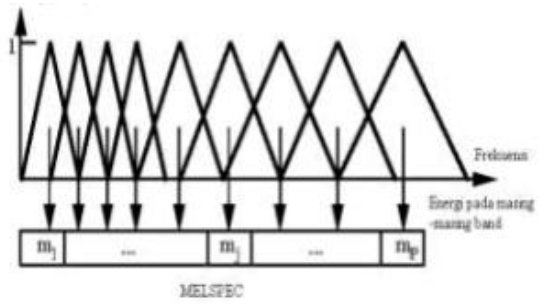
$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2j\pi kn}{N}}, k = 0,1,2,\ldots N-1 \quad (3)$$

Umumnya merupakan bilangan komplek dengan nilai absolute (frekuensi magnitude). Deret hasil diperlihatkan sebagai berikut: jika frekuensi positif $0 \leq f < F_s$ maka bernilai sama dengan $0 \leq n \leq \frac{N}{2}$, sedangkan bila frekuensi negatif $-F_s/2 < f < 0$ maka bernilai $\frac{N}{2} + 1 \leq n \leq N$. merupakan frekuensi sampling. Hasil dari langkah ini sering disebut sebagai *spectrum* atau *peridogram.*

Suatu sinyal suara dalam domain waktu dapat dicari frekuensi pembentuknya. Inilah tujuan dari penggunaan analisa fourier pada data suara, yaitu untuk merubah data dari domain waktu menjadi data spektrum di domain frekuensi. Hal ini sangat menguntungkan karena data pada domain frekeuensi dapat diproses lebih mudah dibandingkan pada domain waktu, karena pada domain frekuensi, keras lemahnya suara tidak seberapa berpengaruh (Putra & Resmawan, 2011).

Studi psikofisik telah menunjukkan bahwa persepsi manusia tentang frekuensi suara untuk sinyal ucapan tidak mengikuti skala linier. Jadi untuk setiap nada dengan frekuensi aktual *f* (diukur dalam Hz), secara subjektif diukur pada skala yang disebut skala '*mel*'. Skala *mel*-frekuensi adalah frekuensi linier di bawah 1000 Hz dan logaritmik di atas 1000 Hz. Sebuah pendekatan untuk simulasi spektrum dalam skala mel adalah dengan menggunakan *filterbank*. *Filterbank* adalah salah satu bentuk filter yang dilakukan dengan tujuan untuk mengetahui ukuran energi dari frekuensi band tertentu dalam sinyal suara.

Dalam *mel-frequency wrapping*, sinyal hasil FFT dikelompokkan ke dalam berkas *filter triangular* ini. Maksud pengelompokkan ini adalah setiap nilai FFT dikalikan terhadap *gain filter* yang bersesuaian dan hasilnya dijumlahkan. Maka setiap kelompok mengandung sejumlah bobot energi sinyal sebagaimana dinyatakan sebagai $m_1 \ldots m_p$ seperti pada Gambar 5.



Gambar 5. Contoh *mel-spaced filterbank*
(Setiawan, 2011)

Langkah terakhir dalam proses MFCC ini mengubah spektrum log *mel* kembali ke waktu. Hasilnya disebut MFCC (Mel Frequency Cepstrum Coeficients). Cepstrum adalah sebutan kebalikan untuk spectrum. Cepstrum biasa digunakan untuk mendapatkan informasi dari suatu sinyal suara yang diucapkan oleh manusia. Karena koefisien spektrum *mel* adalah bilangan real, maka konversinya ke domain waktu menggunakan Discrete Cosine

Transform (DCT). Formula untuk menghitung koefisien MFCC itu adalah:

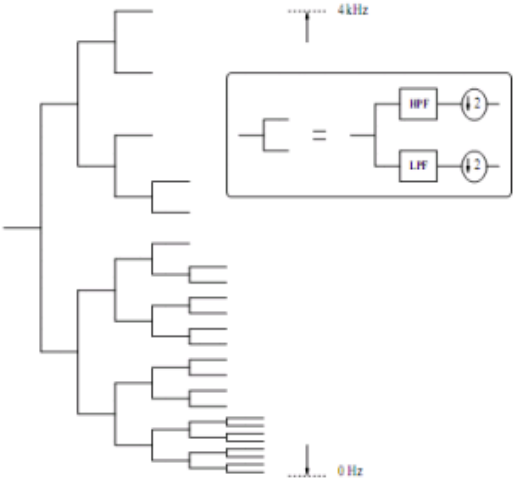
$$C_n = \sum_{k=1}^{K} (\log S_k) \cos\left[\frac{n\left(k-\frac{1}{2}\right)\pi}{K}\right],$$
$$n = 1,2\ldots,K \quad (4)$$

dengan S_k merupakan keluaran dari proses *filterbank* dan jumlah koefisien yang diharapkan.

1.5.3 Ekstrasi Ciri dengan SBC

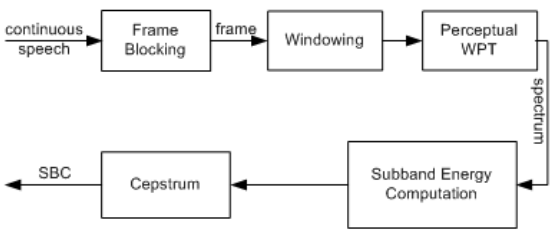
Sebuah wavelet merupakan gelombang singkat (*small wave*) yang energinya terkonsentrasi pada suatu selang waktu untuk memberikan kemampuan analisis transien, ketidakstasioneran, atau fenomena berubah terhadap waktu (*time varying*), seperti pada suara manusia.

Paket wavelet didasarkan pada iterasi dari 2 kanal *filterbank low pass* dan *high pass*. Pada penelitian ini menggunakan 24 *subband wavelet packet tree* yang mana mendekati *mel-scale frequency* seperti ditunjukkan pada Gambar 6. *Wavelet Packet Tree* dibangun dari mendekomposisi 2 kanal standar *filterbank* menjadi beberapa level.



Gambar 6. Ilustrasi WPT *filterbank*
(Sarikaya, 1998)

Proses SBC sebenarnya hampir sama dengan proses MFCC, hanya saja bila pada MFCC sinyal mengalami proses FFT, tapi pada SBC digunakan transformasi paket wavelet untuk menggantikan transformasi fourer FFT seperti ditunjukan pada Gambar 7.



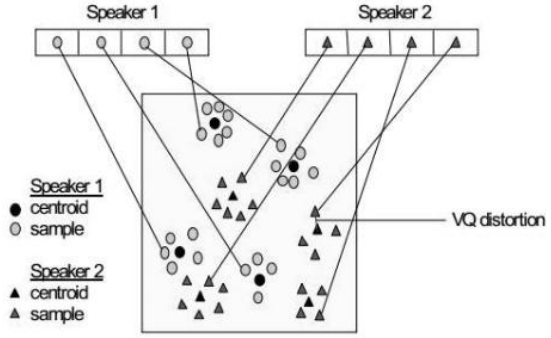
Gambar 7. Diagram proses SBC

Komputasi sinyal dalam domain frekuensi dengan *Wavelet Packet Transform* (WPT) dilakukan setelah *Frame Blocking* dan *windowing*. Spektrum sinyal hasil WPT adalah *filtering magnitude* yang merupakan representasi dari energi tiap band pada *filterbank*. Kemudian energi dari tiap band keluaran dari tiap filter dihitung secara logaritmik. Proses logaritmik sinyal digunakan untuk mengadaptasi sistem telinga manusia, karena sinyal suara yang berada dibawah frekuensi 1 kHz akan terdengar linier namun bila lebih dari 1 kHz grafiknya akan menjadi logaritmis. Proses terakhir untuk menghasilkan SBC yaitu dengan *Descrette Cossine Transform* (DCT).

1.5.4 Ciri Pencocokan dengan LBG-VQ

Permasalahan dalam *speaker identification* yang sering dibahas dalam ilmu pengetahuan dan rekayasa salah satunya adalah *pattern recognition*. Tujuan *pattern recognition* yaitu mengklasifikasikan objek menjadi sejumlah kategori atau kelas. Objek-objek yang memiliki kesamaan secara umum disebut *patterns* dan dalam hal ini *patterns* tersebut adalah sederetan vektor akustik yang diekstrasi dari sinyal suara masukan dengan menggunakan metode MFCC atau SBC. *Patterns* tersebut dapat digunakan sebagai ciri pencocokan (*feature matching*).

Vector Quantitation (VQ) merupakan teknik dalam membangun ciri pencocokan (*feature matching*) pada *speaker identification*. Keuntungan dari menggunakan VQ adalah sederhana, mudah diimplementasikan dan akurasi tinggi. VQ adalah proses *mapping* vektor dari ruang vektor yang luas menjadi sejumlah daerah terhingga di ruangan itu. Masing-masing daerah disebut *cluster* dan diwakili pusatnya disebut *codeword*. Kumpulan *codeword* disebut *codebook*. Jarak dari vektor akustik ke *codeword* terdekat pada *codebook* disebut *VQ-distortion*. Diagram yang menggambarkan ciri pencocokan dengan vektor kuantisasi dapat dilihat pada Gambar 8.



Gambar 8. Konseptual vektor kuantisasi
(Do, 2013)

Gambar 8 tersebut mengilustrasikan ada 2 (dua) pembicara dengan 2 (dua) dimensi ruang akustik yang ditampilkan. Bentuk lingkaran mengacu pada vektor akustik dari pembicara ke-1 sedangkan

segitiga vektor akustik dari pembicara ke-2. *VQ-Codebook* dihasilkan untuk setiap pembicara dengan cara mengelompokkan vektor akustik pelatihan nya. Hasil *codeword* (centroid) ditunjukkan oleh lingkaran hitam untuk pembicara ke-1 dan segitiga hitam untuk pembicara ke-2. Jarak dari vektor akustik ke *codeword* terdekat pada *codebook* disebut *VQ-distortion*. Pada *testing phase*, ucapan suara yang merupakan vektor akustik ditelusuri kesamaan (*similarity*) dengan *VQ-codebook* yang dibangun pada saat *training phase*, dengan cara menghitung *VQ-distortion*. Pembicara dengan *VQ-codebook* yang memiliki distorsi jarak terkecil dengan vektor akustik hasil ekstrasi ciri sinyal ucapan, diidentifikasi sebagai pembicara ucapan tersebut.

Perhitungan *VQ-distortion* menggunakan Euclidean *distance*. Euclidean merupakan metode statistika yang digunakan untuk mencari data antara parameter data referensi dengan parameter data baru. Formula euclidean *distance* dituliskan sebagai berikut

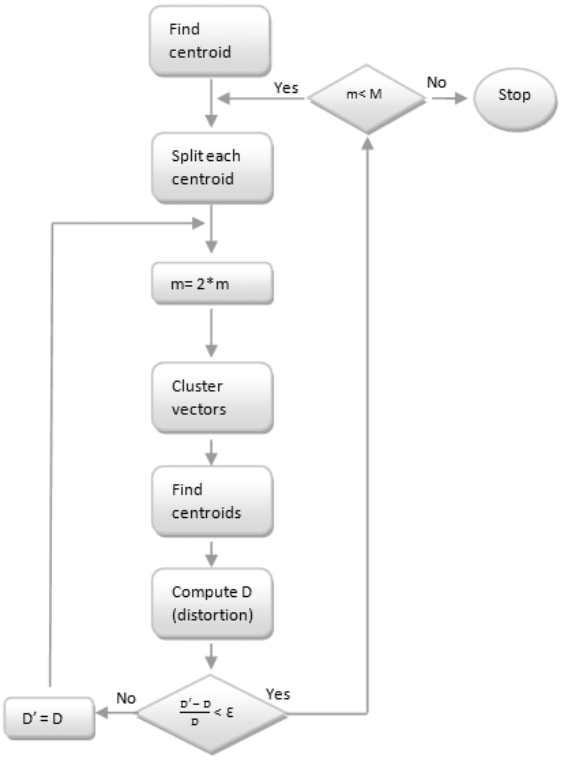
$$D_i = \sqrt{\sum_{i=0}^{N} (x_1 - x_2)^2} \qquad (5)$$

dengan:
Di = jarak terhadap tekstur i yang terkecil pada
x_1 = ciri dari tekstur yang diklasifikasikan
x_2 = ciri dari tekstur yang terdapat pada.
Tekstur akan diklasifikasikan sebagai tekstur *i* apabila *Di* merupakan jarak terkecil dibandingkan dengan jarak yang lainnya.

Algoritma yang terkenal untuk membangun *VQ-codebook*, yaitu algoritma LBG. Algoritma tersebut dilaksanakan oleh prosedur rekrusif dengan langkah berikut ini:

1. Desain sebuah *codebook* vektor-1, yang merupakan centroid dari seluruh himpunan vektor *training* (tidak ada iterasi yang diperlukan disini).
2. Gandakan ukuran *codebook* dengan memisahkan setiap *current codebook* y_n sesuai dengan aturan
   $y_n^+ = y_n(1+\varepsilon)$
   $y_n^- = y_n(1-\varepsilon)$
   dengan n bervariasi dari 1 sampai ukuran *current codebook* dan ε adalah parameter pemisahan (ε = 0,01).
3. *Nearest-Neighbor Search*: untuk setiap vektor *training*, temukan *codeword* pada *codebook* saat yang jaraknya terdekat (dalam hal pengukuran kesamaan) dan menetapkan vektor tersebut ke *cell* yang sesuai (terkait dengan *codeword* terdekat).
4. Perbarui centroid, memperbarui *codeword* dalam setiap cell menggunakan centroid dari vektor training yang ditetapkan ke cell tersebut.
5. Iterasi 1: ulangi langkah 3 dan 4 sampai jarak (*distance*) rata-rata turun dibawah ambang (*preset threshold*).
6. Iterasi 2: ulangi langkah 2, 3 dan 4 sampai ukuran *codebook* dirancang.

Secara intuitif algoritma LBG mendesain sebuah *codebook* M-vektor secara bertahap. Dimulai pertama dengan merancang suatu 1-vektor *codebook*, kemudian menggunakan teknik *splitting* pada *codeword* untuk menginisialisasi mencari 2-vektor *codebook*, dan terus proses pemisahan sampai diinginkan M-vektor *codebook* diperoleh. Gambar 9 menunjukkan diagram alir, langkah-langkah rinci dari algoritma LBG.



Gambar 9. Diagram alir algoritma LBG

"*Vektor Cluster*" adalah prosedur pencarian terdekat-tetangga yang memberikan tiap vektor pelatihan untuk *cluster* terkait dengan *codeword* terdekat. "*Find centroid*" adalah prosedur pembaruan massa. "*Compute D*" merangkum jarak dari semua vektor pelatihan dalam pencarian terdekat-tetangga sehingga menentukan apakah prosedur telah berkumpul.

2. PEMBAHASAN

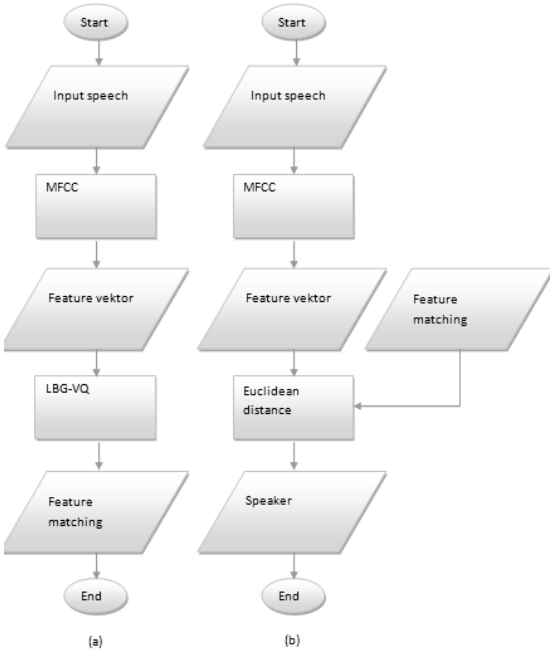
2.1 Bahan Penelitian

Sampel suara yang digunakan sebagai bahan penelitian diambil dari merekam sejumlah 50 peserta yang mengucapkan "*Selamat pagi, saya [nama partisipan]*" untuk bahan *training phase* dan "*Selamat pagi, saya [nama partisipan]*" untuk bahan *testing phase*. Dari kedua kalimat yang diucapkan tadi masing-masing dipotong per-kata untuk

dicobakan dalam pengujian berdasarkan: *text-dependent* dengan menggunakan 2 (dua) kata yang sama yaitu "*Selamat*" dan *text-independent* dengan menggunakan 2 kata yang berbeda, yaitu "*pagi*" dan "*siang*".

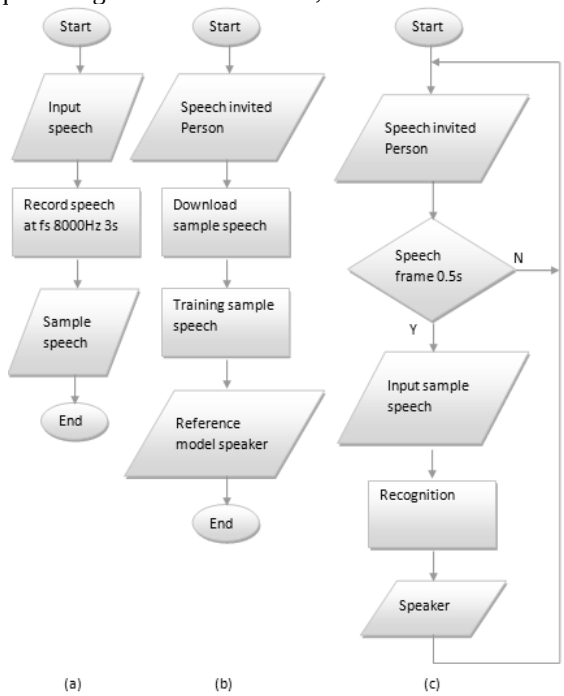
2.2 Perancangan Modul Pengujian

Pengujian ekstrasi ciri identifikasi pembicara dengan metode yang diperbandingkan ini, dirancang dengan menggunakan MATLAB R2011a. Modul ini terdiri dari 2 bagian yaitu bagian *enrolment (training phase)* dan bagian identifikasi *(testing phase)*.



Gambar 10. Diagram alir (a) *training phase* (b) *testing phase*

Gambar 10 menunjukkan diagram alir dari modul pengujian identifikasi pembicara dengan metode ekstrasi ciri yang diperbandingkan. Pada Gambar 10 tersebut terlihat metode yang digunakan dalam ekstrasi ciri adalah MFCC, bila ingin melihat kinerja yang SBC cukup dengan mengganti proses ekstrasi ciri tersebut dengan SBC. Setelah mendapatkan ekstrasi ciri yang lebih efektif antara MFCC dan SBC, maka diuji secara *realtime*. Adapun modul identifikasi pembicara secara *realtime* terdiri dari beberapa bagian, yaitu *enrollment phase*, *training phase* dan *testing phase*. *Enrollment phase*, merupakan bagian pendaftaran suara, akuisisi suara peserta untuk dijadikan sampel suara pada training phase. *Training phase* adalah bagian mengklasifikasikan model pembicara dengan proses vektor kuantisasi dari vektor akustik hasil ekstrasi ciri dari sinyal *sample speech*. *Testing/ recognition phase*, merupakan bagian pengenalan suara *realtime*, suara peserta dikenali dengan cara mengambil sinyal suara masukan tiap *frame* dengan ukuran panjang *frame* adalah 500 milidetik. Modul identifikasi pembicara secara *realtime* dapat dilihat pada diagram alir Gambar 11, berikut ini.



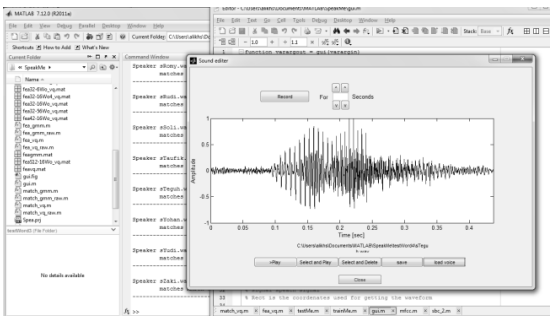
Gambar 11. (a) *Enrollment phase* (b) *training phase* (c) *testing phase*

Peserta yang diundang terlebih dahulu sudah melakukan pendaftaran (*enrollment*) dengan merekam atau mengakuisisi suaranya sebagai *speech sample* (suara yang akan di-*training*) yang disimpan dalam basisdata komputer server. Selanjutnya proses ekstrasi ciri dilakukan di komputer lokal dengan terlebih dahulu men-*download speech sample* untuk masng-masing peserta yang diundang/didaftarkan. *Feature*/*template* hasil ekstrasi ciri itu dijadikan *reference model* dan digunakan sebagai pembanding saat proses identifikasi pembicara *(recognition*).

2.3 Hasil Pengujian

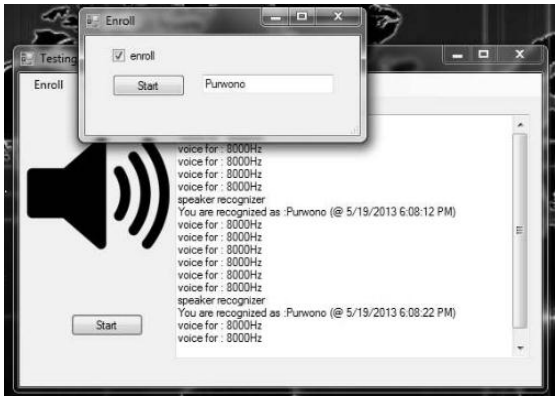
2.3.1 Modul Pengujian

Modul pengujian dari perancangan identifikasi pembicara yang digunakan dalam membandingkan ekstrasi ciri MFCC dan SBC dapat dilihat pada Gambar 12 berikut ini.



Gambar 12. Modul pengujian MFCC dan SBC dengan menggunakan matlab

Hasil dari perancangan Modul pengujian identifikasi pembicara secara *realtime* dapat dilihat pada Gambar 13 berikut ini. Modul ini dibangun dengan menggunakan C# Visual Studio 2010.



Gambar 13. Modul pengujian secara *realtime*

2.3.2 Hasil Pengujian MFCC dan SBC

Berikut merupakan hasil dari pengujian dengan mengubah nilai beberapa koefisien SBC dan MFCC serta jumlah centroid pada Vektor Kuantisasi. Diujikan terhadap 50 peserta dengan mengucapkan kalimat;

1. "*Selamat pagi, saya [nama partisipan]*",
2. "*Selamat siang, saya [nama partisipan]*".

Kemudian diambil kata "*Selamat*" pada kalimat 1 dan 2, untuk pengujian *text-dependent* dan kata "*pagi*" dan "*siang*" untuk pengujian *text-independent.* Hasil pengujian dengan menggunakan ekstrasi ciri SBC dapat dilihat pada Tabel 1 berikut.

Tabel 1. Hasil metode SBC

Koef sbc	*Text Dependent*				*Text Independent*			
	R	*F*	*A*	*T*	*R*	*F*	*A*	*T*
12	25	25	50%	24"	12	38	24%	24"
22	24	26	48%	28"	17	33	34%	28"
32	24	26	48%	30"	18	32	36%	30"

keterangan:

R = Right, benar dalam mengidentifikasi
F = False, salah dalam mengidentifikasi
A = Accurate, prosentase benar-salah
T = Time, waktu respon identifikasi

Mengubah atau menaikkan nilai koefisien SBC tidak signifikan mempengaruhi akurasi, dan jadi memperlambat proses identifikasi. Hasil pengujian dengan menggunakan ekstrasi ciri MFCC dapat dilihat pada Tabel 2 berikut ini.

Tabel 2. Hasil metode MFCC

Koef mfcc	*Text Dependent*				*Text Independent*			
	R	*F*	*A*	*T*	*R*	*F*	*A*	*T*
12	25	25	50%	24"	12	38	24%	24"
22	24	26	48%	28"	17	33	34%	28"
32	24	26	48%	30"	18	32	36%	30"

keterangan:

R = Right, benar dalam mengidentifikasi
F = False, salah dalam mengidentifikasi
A = Accurate, prosentase benar-salah
T = Time, waktu respon identifikasi

Mengubah nilai koefisien MFCC, didapat koefisien 32 (tigapuluh dua) lebih baik untuk identifikasi dan prosesnya lebih cepat dari SBC. Kemudian mencoba variasi jumlah centroid VQ untuk MFCC dengan koefisien 32, untuk melihat adakah pengaruhnya dalam mencari efektifitas identifikasi pembicara, hasilnya seperti pada Tabel 3 berikut ini.

Tabel 3. MFCC dengan varian jumlah centroid

Jml Cent	*Text Dependent*				*Text Independent*			
	R	*F*	*A*	*T*	*R*	*F*	*A*	*T*
6	33	17	66%	2"	16	34	32%	2"
16	37	13	74%	8"	25	25	50%	8"
26	33	17	66%	10"	20	30	40%	10"

keterangan:

R = Right, benar dalam mengidentifikasi
F = False, salah dalam mengidentifikasi
A = Accurate, prosentase benar-salah
T = Time, waktu respon identifikasi

Mengubah jumlah centroid tidak meningkatkan nilai akurasi. Semakin kecil jumlah centroid, maka semakin cepat prosesnya dan sebaliknya semakin besar semakin lama.

2.3.3 Hasil Pengujian secara *Realtime*

Hasil dari pengujian MFCC dan SBC mendapatkan metode ekstrasi ciri yang lebih efektif dan cepat, yaitu MFCC koefisien 32 dan ciri pencocokan LBG-VQ dengan jumlah centroid 16, kemudian metode tersebut diujikan secara *realtime*. Pengujian ini dilakukan dengan 2 peserta, dimana masing-masing mengucapkan kata "*Selamat pagi saya [nama peserta]"* sebagai *training phase*, kemudian saat *realtime* peserta mengucapkan kalimat yang berbeda, yaitu: "*ketuhanan yang maha esa"* dan kalimat sama dengan *training phase*. Ini dimaksudkan untuk pengujian yang *text-independent* dan *text-dependent* dan hasil pengujian ini dapat dilihat pada Tabel 4 dan Tabel 5.

Tabel 4. Hasil uji *realtime text-independent*

Pembicara	*Dikenali sebagai*		*Jml Frame*	*Akurasi (%)*
	P1	*P2*		
P1	5	3	8	62.5
P2	2	3	5	60
			Rata-rata akurasi	61.25

keterangan;

P1 = pembicara ke-1, P2 = pembicara ke-2

Tabel 5. Hasil uji *realtime text-dependent*

Pembicara	*Dikenali sebagai*		*Jml Frame*	*Akurasi (%)*
	P1	*P2*		
P1	5	2	7	71.43
P2	1	4	5	80.00
			Rata-rata akurasi	75.71

keterangan;

P1 = pembicara ke-1, P2 = pembicara ke-2

Dalam pengujian identifikasi pembicara, suara ucapan yang diujikan dikenali dalam setiap *frame* (1 *frame* = 500 milidetik). Masing-masing pembicara menghasilkan jumlah *frame* yang berbeda dalam mengucapkan suara/kalimat yang diujikan. Rata-rata dikenali sebagai pembicara untuk kalimat yang beda 61.25 % dan pada kalimat yang sama 75,71% untuk 2 (dua) peserta. Secara keseluruhan akurasi untuk identifikasi pembicara *realtime* ini masih belum memuaskan, karena dengan hanya 2 (dua) peserta akurasi pengenalannya masih dibawah 70% untuk kalimat yang beda.

2.3.4 Pembahasan Hasil Uji

Efektifitas metode, dengan merubah nilai koefisien masing masing parameter SBC dan MFCC dapat diketahui nilai yang baik untuk meningkatkan akurasi yaitu jika pada MFCC koefisien diisi nilai 32 dan namun pada SBC koefisien ini tidak berpengaruh signifikan. Kemudain rata-rata keseluruahan waktu proses MFCC lebih cepat dibandingkan SBC. Jumlah centroid, pada klasifikasi VQ bila diturunkan jumlahnya maka akan lebih cepat tapi kurang akurasi pencocokannya.

3. KESIMPULAN

Kesimpulan dari pembahasan dalam penelitian ini, adalah;

1. pada klasifikasi model identifikasi pembicara dengan LBG-VQ penggunaan ekstrasi ciri MFCC lebih tinggi nilai akurasinya dan lebih cepat prosesnya dibandingkan dengan menggunakan SBC baik untuk identifikasi pembicara berdasarkan *text-independent* maupun *text-dependent*,
2. identifikasi pembicara menggunakan ekstrasi ciri MFCC dengan ciri pencocokan LBG-VQ dapat diimplementasikan pada *speaker identification* secara *realtime*, tetapi masih belum cukup memuaskan akurasinya, masih dibawah 70%.

Saran yang dapat dilanjutkan pada penelitian berikutnya, adalah;

1. diperlukan akuisisi suara yang lebih baik dengan menerapkan pra-pemrosesan sinyal suara menggunakan *Pre-emphasize Filtering* sehingga meminimalisasi *noise* pada sinyal masukan. *Pre-emphasize Filtering* adalah salah satu jenis filter yang sering digunakan sebelum sebuah signal diproses lebih lanjut. Filter ini mempertahankan frekuensi-frekuensi tinggi pada sebuah spektrum, yang umumnya tereleminasi pada saat proses produksi suara (Putra & Resman 2011)
2. mengganti metode *feature matching* LBQ-VQ dalam membangun model referensi pembicara dengan *Information Theoretic Vector Quantization* (ITVQ). Menurut Memon (2010) menggunakan ITVQ lebih baik dari vektor kuantisasi dengan K-*means* dan LBG-VQ.

PUSTAKA

Do**,** N. Minh. 2013. *DSP Mini-Project: An Automatic Speaker Recognition System*. Electrical and Computer Engineering, University of Illinois (Online), (http://www.ifp.illinois.edu/~minhdo/teaching/speaker_recognition/speaker_recognition.html, diakses 16 Juni 2013).

Hosaan, A. Md. 2011. *Autometic Speaker Recognition Dynamic Feature Identification and Clasification using Distributed Discrete Cosine Transform Based Mel Frequency Cepstral Coefficients and Fuzzy Vector Quantization*. Thesis Master, RMIT University, (Online), (https://researchbank.rmit.edu.au/eserv/rmit:12308/Hossan.pdf, diakses 21 Februari 2016).

Memon, S. 2010. *Automatic Speaker Recognition: Modeling, Feature Extraction, and Efect of Clinical Environment*. Thesis Doctor, RMIT University, (Online), (https://researchbank.rmit.edu.au/eserv/rmit:12426/Memon.pdf, diakses 21 Februari 2016).

Putra, D., Resmawan, A. 2011. Verifikasi Biometrika Suara Menggunakan Metode MFCC dan DTW. *LONTAR KOMPUTER*, Vol.2 No.1, hlm. 8-21, (Online), (http://ojs.unud.ac.id/index.php/lontar/article/download/3711/2734, diakses diakses 21 Februari 2016).

Sarikaya, R., Pellom, L. B., Hansen, H.L.J. 1998. *Wavelet Packet Transform Feature with Aplication to Speaker Identification*. Center Robust Speech Processing, University of Texas, (Online), (http://crss.utdallas.edu/Publications/Sarikaya1998.pdf, diakses 24 Februari 2016).

Setiawan, A., Hidayatno, A., Isnanto, R. R. 2011. Aplikasi Pengenalan Ucapan dengan Ekstrasi Mel-Frequency Cepstrum Coefficients (MFCC) Melalui Jaringan Syaraf Tiruan Learning Vector Quantization (LVQ) untuk Mengoperasikan Kursor Komputer. *TRANSMISI*, Vol.13-3, hlm. 82-86, (Online), (http://ejournal.undip.ac.id/index.php/transmisi, diakses 21 Februari 2016).

Singh, K. S. 2003. *Feature And Techniques for Speaker Recognition*. M. Tech. Credit Seminar Report, Electronic Systems Group, EE Dept, IIT Bombay, (Online), (https://www.ee.iitb.ac.in/~esgroup/es_mtech03_sem/sem03_paper_03307409.pdf, diakses 21 Februari 2016).