

# Pengenalan Pengucap Tak Bergantung Teks dengan Metode *Vector Quantization* (VQ) Melalui Ekstraksi *Linear Predictive Coding* (LPC)

Fachrudin AN (L2F099604)  
Jurusan Teknik Elektro Fakultas Teknik  
Universitas Diponegoro

## Abstrak

Penelitian yang intensif dalam bidang pengolahan sinyal menyebabkan teknologi komunikasi berkembang dengan pesat, salah satunya adalah pengenalan pengucap (*speaker identification*). Pengenalan pengucap merupakan cara yang digunakan untuk mengetahui identitas seseorang yang mengucapkan sinyal informasi. Tugas akhir ini akan membahas mengenai pengenalan pengucap tak bergantung teks, dimana akan dibuat sebuah sistem komputer yang dapat mendengar dan mengenali pengucap yang sedang berbicara.

Dalam proses pengenalan tersebut akan digunakan algoritma ekstraksi yang disebut *Linear Predictive Coding* (LPC), sedang proses pencocokan menggunakan metode *Vector Quantization* (VQ). LPC merupakan salah satu metode analisis sinyal suara yang merepresentasikan sinyal suara dari pengucap dalam bentuk koefisien-koefisien LPC. VQ akan mengelompokkan vektor-vektor hasil ekstraksi LPC kedalam suatu *codebook* dengan ukuran tertentu, dimana tiap *codebook* mewakili beberapa vektor hasil ekstraksi LPC yang merupakan ciri khas dari masing-masing pengucap.

Hasil pengujian menunjukkan persentase pengenalan tertinggi untuk data rekaman diperoleh pada kombinasi parameter orde LPC 12, panjang frame 20 mS, ukuran *codebook* 16 dan orde LPC 12 panjang frame 20 mS, ukuran *codebook* 32. Sedangkan kombinasi parameter orde LPC 8, panjang frame 10 mS dan ukuran *codebook* 64 memberikan persentase pengenalan tertinggi untuk pengujian *realtime*.

## I. PENDAHULUAN

### 1.1 Latar Belakang

Dalam ilmu pengetahuan dan teknologi khususnya bidang komunikasi, pengolahan sinyal memegang peranan yang penting. Penelitian yang intensif dalam bidang pengolahan sinyal menyebabkan teknologi komunikasi berkembang dengan pesat. Salah satunya adalah pengenalan pengucap. Pengenalan pengucap merupakan cara yang digunakan untuk mengetahui identitas seseorang yang mengucapkan sinyal informasi. Hal ini bisa dilakukan karena masing-masing individu memiliki karakteristik-karakteristik sinyal ucapan yang spesifik.

Karakteristik ucapan dapat dibedakan melalui ekstraksi dengan suatu teknik pengkodean. Teknik pengkodean yang umum digunakan dalam pegekstraksian sinyal ucapan adalah LPC (*Linear Predictive Coding*). Analisa LPC menghasilkan suatu estimasi parameter ucapan dasar, antara lain *pitch*, *formant*, persamaan area jalur vokal, dan untuk memampatkan (kompresi) sinyal ucapan agar didapat *bit-rate* rendah untuk keperluan transmisi atau penyimpanan.

Berdasarkan parameter sinyal ucapan, dapat dibuat suatu sistem pengenalan (identifikasi) pengucap. Sistem pengenalan pengucap pada dasarnya adalah perbandingan suatu parameter sinyal ucapan yang ingin dikenali dengan parameter sinyal ucapan lain. Pada proses pencocokan (*feature matching*), digunakan metode VQ (*Vector Quantization*), yang mengubah hasil ekstraksi analisis LPC dari masing-masing pengucap menjadi sekumpulan *codebook*. *Codebook* tersebut kemudian dibandingkan dengan hasil ekstraksi koefisien LPC dari sinyal masukan yang akan dikenali. Identifikasi

ditentukan dengan menghitung jarak penyimpangan. *Codebook* dengan jarak penyimpangan terkecil merupakan identitas dari sinyal tersebut.

### 1.2 Tujuan

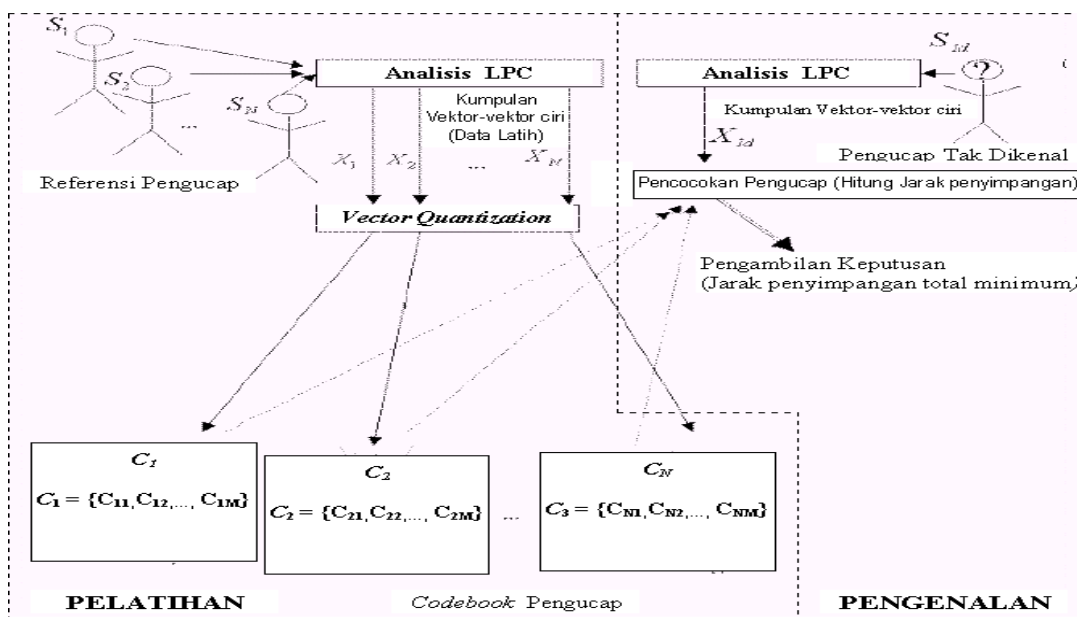
Tujuan dari tugas akhir ini adalah membuat suatu program pengenalan pengucap tak bergantung teks dengan menggunakan program bantu MATLAB 6.5..

### 1.3 Batasan Masalah

1. Data masukan (pada basisdata) berupa sinyal suara yang diambil dari 10 orang responden (5 pria dan 5 wanita).
2. Pengenalan bersifat *text independent*, artinya ucapan pada basisdata tidak harus sama dengan ucapan yang akan dikenali. Suara berupa ucapan dari suatu teks Bahasa Indonesia berdurasi  $\pm 6$  detik sebagai basisdata dan teks Bahasa Indonesia berdurasi  $\pm 1,5$  detik sebagai sinyal uji.
3. Derau (*noise*) yang turut terekam pada proses perekaman diabaikan.
4. Ciri dari ucapan akan diambil dengan menggunakan *Linear Predictive Coding* (LPC)
5. Proses pencocokan ciri dilakukan dengan menggunakan metode *Vector Quantization* (VQ).
6. Parameter-parameter yang digunakan adalah panjang *frame*, orde LPC, dan ukuran *codebook*.

## II. LANDASAN TEORI

Ucapan berisi beberapa karakteristik yang spesifik untuk setiap individu, yang beberapa diantaranya tidak dipengaruhi oleh pesan linguistik yang terkandung dalam



Gambar 1 Struktur identifikasi pengucap dengan analisis LPC dan berdasarkan metode VQ

suatu ucapan<sup>[3]</sup>. Perbedaan karakteristik ucapan itulah yang menjadi dasar pengenalan pengucap melalui ucapannya. Proses pengenalan pengucap (speaker recognition) dapat diklasifikasikan menjadi dua proses, yaitu proses identifikasi dan verifikasi<sup>[2]</sup>. Identifikasi merupakan proses menjelaskan ucapan yang diberikan pengucap yang telah terdaftar dalam basisdata, sedangkan speaker verification merupakan proses diterima atau ditolaknya identitas dari pengucap. Gambar 1 menunjukkan struktur identifikasi pengucap dengan analisis LPC dan berdasarkan metode VQ

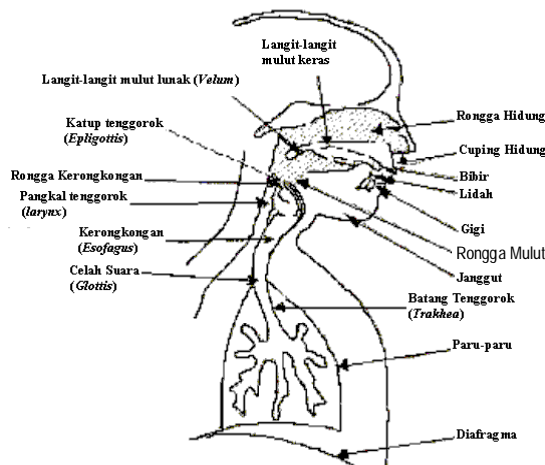
## 2.1 Teori Dasar Ucapan

Ucapan berisi gelombang suara kompleks yang berubah secara kontinu menghubungkan mulut dengan telinga pengucap. Ucapan dihasilkan melalui organ-organ vokal manusia seperti pada Gambar 2. Jalur vokal dimulai dari lubang *cord* vokal atau *glottis*, dan berujung pada bibir. Jalur vokal terdiri dari *pharynx* (penghubung antara kerongkongan (*esophagus*) dengan mulut) dan mulut, atau lubang mulut (*oral cavity*). Pada pria dewasa, panjang total jalur vokal adalah sekitar 17 cm. Daerah melintang dari jalur vokal meliputi lidah, bibir, rahang dan *velum* (langit-langit mulut), lebarnya sekitar 20 cm<sup>2</sup>. Jalur *nasal* (nafas), dimulai dari *velum* dan berujung pada *nostril* (cuping hidung). Ketika *velum* merendah/menutup, jalur *nasal* terhubung dengan jalur vokal secara akustik untuk membentuk bunyi ucapan *nasal* (sengau).

Udara masuk kedalam paru-paru melalui mekanisme pernafasan biasa. Udara kemudian disalurkan ke batang tenggorok (*trachea*), yang menyebabkan *cord* vokal yang tertekan dalam pangkal tenggorok (*larynx*) bergetar. Aliran udara dipotong-potong sehingga menjadi pulsa-pulsa kuasi-periodik yang akan memodulasi frekuensi pada saat melewati tekak (*pharynx*), lubang mulut, dan cuping hidung. Berdasarkan posisi dari

artikulator (misalnya rahang, lidah, *velum*, bibir, atau mulut), akan dihasilkan bermacam-macam bunyi yang berbeda.

Bermacam-macam bunyi ucapan (fonem) yang berbeda merupakan cara yang paling mudah untuk membedakan antara pengucap yang satu dengan yang lain. Fonem yang berbeda tersebut antara lain vokal, *diphthong*, semivokal, *unvoiced fricative*, *voiced fricative*, *voice* dan *unvoiced stop*.



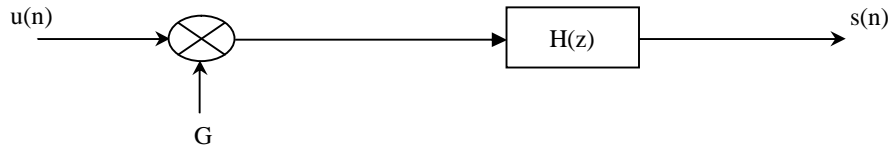
Gambar 2 Skema organ vokal manusia

## 2.2 Linear Predictive Coding (LPC)

Prinsip dasar dari pemodelan sinyal dengan menggunakan LPC adalah bahwa contoh sinyal ucapan  $s(n)$  pada waktu ke- $n$  dapat diperkirakan sebagai kombinasi linier dari  $p$  sampel sinyal ucapan sebelumnya yaitu :

$$s(n) \approx a_1 s(n-1) + a_2 s(n-2) + \dots + a_p s(n-p) \quad (1)$$

dengan koefisien  $a_1, a_2, \dots, a_p$  diasumsikan bernilai konstan selama *frame* analisis ucapan.



Gambar 3 Model LPC

Persamaan 1 dapat diubah menjadi suatu persamaan mencakup masa pembangkitan,  $G u(n)$ , menghasilkan :

$$s(n) = \sum_{k=1}^p a_k s(n-k) + G \cdot u(n) \quad (2)$$

persamaan 2 dalam daerah  $z$ , kita peroleh hubungan

$$S(z) = \sum_{i=1}^p a_i \cdot z^{-i} S(z) + G \cdot U(z) \quad (3)$$

membawa kita pada fungsi alih

$$H(z) = \frac{S(z)}{G \cdot U(z)} = \frac{1}{\left(1 - \sum_{i=1}^p a_i \cdot z^{-i}\right)} = \frac{1}{A(z)} \quad (4)$$

Interpretasi dari persamaan 4 diberikan pada gambar 3.

Dengan mempertimbangkan kombinasi linier dari sampel ucapan sebelumnya sebagai peramalan  $\tilde{s}(n)$ , dinyatakan sebagai

$$\tilde{s}(n) = \sum_{k=1}^p a_k s(n-k) \quad (5)$$

Kemudian dapat dibentuk prediksi kesalahan (error),  $e(n)$ , yang dinyatakan sebagai

$$e(n) = s(n) - \tilde{s}(n) = s(n) - \sum_{k=1}^p a_k s(n-k) \quad (6)$$

dengan fungsi alih kesalahan

$$A(z) = \frac{E(z)}{S(z)} = 1 - \sum_{k=1}^p a_k z^{-k} \quad (7)$$

Tujuan utama dari analisa peramalan linear adalah untuk mendapatkan serangkaian koefisien peramalan ( $\alpha_k$ ) langsung dari sinyal ucapan. Koefisien peramalan harus diestimasi dari segmen-segmen pendek sinyal ucapan karena keadaan dasar sinyal ucapan yang bervariasi terhadap waktu. Pendekatan ini adalah untuk mendapatkan koefisien peramalan yang meminimalisir kuadrat rata-rata kesalahan peramalan pada satu segmen pendek gelombang ucapan. Untuk menganalisa sinyal ucapan pada segmen terbatas, diasumsikan bahwa besarnya sinyal diluar segmen atau  $0 \leq m \leq N-1$  besarnya identik dengan nol, untuk mendapatkan keadaan seperti ini, maka sinyal ucapan  $s(n+m)$  dikalikan dengan jendela (*window*),  $w(n)$ , yang panjangnya terbatas. Dari hasil perkalian tersebut didapat sampel sinyal ucapan sebagai berikut :

$$s_n(m) = \begin{cases} s(m+n) \cdot w(m), & 0 \leq m \leq N-1 \\ 0, & \text{lainnya} \end{cases} \quad (8)$$

Dari sinyal yang telah dijendelakan diatas, kesalahan kuadrat rata-ratanya menjadi :

$$E_n = \sum_{m=0}^{N-1+p} e_n^2(m) \quad (9)$$

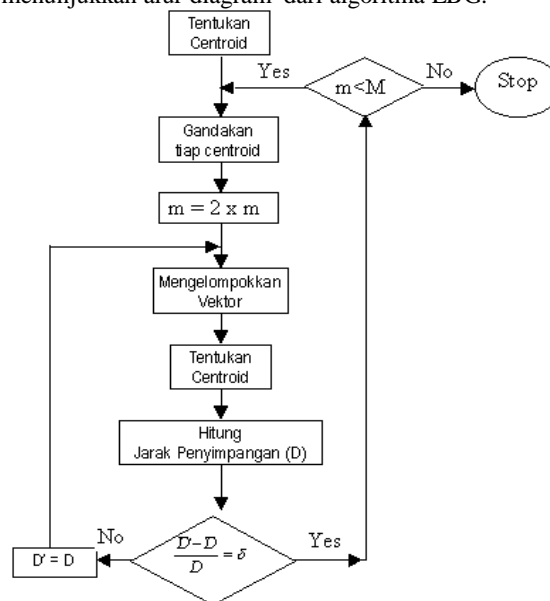
Untuk mendapatkan koefisien LPC digunakan metode autokorelasi. Autokorelasi dari  $s(n)$  didefinisikan sebagai  $r_n(k)$  ditunjukkan oleh persamaan 10. koefisien LPC dapat diperoleh dengan menyelesaikan matrik seperti pada persamaan 11.

$$r_s(k) = \sum_{n=0}^{N-1-k} s(n)s(n-k) \quad (10)$$

$$\begin{bmatrix} r_n(0) & r_n(1) & \cdots & r_n(p-1) \\ r_n(1) & r_n(0) & \cdots & r_n(p-2) \\ \vdots & \vdots & \ddots & \vdots \\ r_n(p-1) & r_n(p-2) & \cdots & r_n(0) \end{bmatrix} \begin{bmatrix} \hat{a}_1 \\ \hat{a}_2 \\ \vdots \\ \hat{a}_p \end{bmatrix} = \begin{bmatrix} r_n(1) \\ r_n(2) \\ \vdots \\ r_n(p) \end{bmatrix} \quad (11)$$

### 2.3 Vector Quantization(VQ)

*Vector Quantization* (VQ) merupakan salah satu metode *template matching* (pencocokan template). VQ melakukan proses pemetaan vektor dari vektor yang berjumlah banyak menjadi vektor dengan jumlah tertentu. Pada pengenalan pengucap, vektor berupa vektor ciri dari masing-masing pengucap, yang diperoleh dari proses ekstraksi ciri. Dengan proses VQ, akan diperoleh representasi dari vektor ciri masing-masing pengucap dengan jumlah vektor yang lebih sedikit, vektor itu disebut sebagai *codebook* dari tiap-tiap pengucap. Algoritma yang dipakai untuk membentuk *codebook* adalah algoritma LBG (*Linde Buzo Gray Algorithm*). Gambar 4 menunjukkan alur diagram dari algoritma LBG.



Gambar 4. Alur diagram algoritma LBG

Algoritma tersebut diimplementasikan dalam prosedur berulang sebagai berikut :

1. Menentukan vektor *codebook* pertama, yang merupakan *centroid* dari keseluruhan vektor ciri. (tidak ada pengulangan pada tahap ini)
2. Menggandakan jumlah *codebook* dengan membagi tiap vektor  $y_n$  dari *codebook* yang telah terbentuk, dengan

$$\begin{aligned} \mathbf{y}_n^+ &= \mathbf{y}_n(1 + \varepsilon) \\ \mathbf{y}_n^- &= \mathbf{y}_n(1 - \varepsilon) \end{aligned}$$

dimana  $n$  memiliki nilai antara 1 sampai dengan  $M$  (ukuran dari *codebook* yang diinginkan), dan  $\varepsilon$  merupakan parameter pembagi (dimana  $\varepsilon = 0,01$ )

3. Pencarian *Nearest-Neighbour* : untuk tiap vektor ciri, temukan *codeword* di dalam *codebook* tersebut (*codebook* saat itu) yang paling dekat (jarak penyimpangannya paling kecil), dan tempatkan vektor tersebut dalam kelompok *codeword* tersebut.
4. Pembaharuan *centroid* : memperbaharui *codeword* pada tiap kelompok dengan menggunakan *centroid* dari vektor ciri terletak di sel tersebut.
5. Iterasi I : mengulang langkah 3 dan 4 sampai diperoleh jarak penyimpangan rata rata ( $D$ ) yang besarnya dibawah batasan yang telah ditentukan ( $\delta$ ).  $D'$  merupakan nilai distorsi awal yang nilainya ditentukan pada saat inialisasi pada awal program.
6. Iterasi II : mengulang langkah 2,3 dan 4 sampai diperoleh *codebook* dengan ukuran  $M$ .

#### 2.4 Perhitungan Jarak Penyimpangan<sup>[9]</sup>

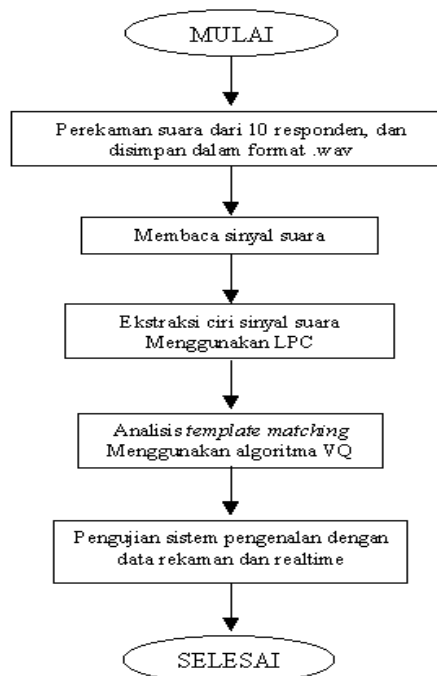
Untuk menghitung jarak penyimpangan antara dua vektor maka digunakan *Euclidean distance* (jarak Euclidean). Persamaan untuk menghitung jarak Euclidean ditunjukkan oleh persamaan 9.

$$d_E(\mathbf{x}, \mathbf{c}) = \sqrt{\sum_{i=1}^{\dim} (x_i - c_i)^2} \quad (9)$$

Dimana  $x$  merupakan vektor ciri dan  $c$  merupakan vektor dari suatu *codebook*. Pada pengenalan pengucap dengan menggunakan metode VQ, jarak penyimpangan Euclidean ini digunakan untuk menghitung jarak penyimpangan antara masing-masing vektor ciri dengan *codeword* pada tiap-tiap *codebook*, sehingga dapat diketahui *codeword* mana yang memiliki jarak penyimpangan terdekat dengan vektor ciri.

### III. PERANCANGAN PROGRAM

Sistem pengenalan pengucap dapat diwujudkan kedalam suatu perangkat lunak(program) menggunakan bahasa pemrograman Matlab 6.5. Dalam pembuatan program simulasi pengenalan pengucap dibagi menjadi 5 tahap, tahap pertama adalah perekaman suara, tahap kedua adalah proses pembacaan dan normalisasi sinyal suara, tahap ketiga adalah proses pengambilan ciri sinyal suara, tahap keempat adalah pembentukan template dan tahap yang kelima adalah pengujian. Alur program pengenalan pengucap dapat dilihat pada Gambar 5



Gambar 5. Alur program pengenalan pengucap

Sistem pengenalan pengucap diawali dengan perekaman sinyal ucapan dari ke-10 responden yang akan disimpan dalam basisdata. Hasil perekaman kemudian dibaca untuk mendapatkan bentuk diskrit dari sinyal ucapan. Pembacaan data suara diikuti dengan normalisasi sinyal ucapan. Sinyal ucapan yang telah dinormalisasi kemudian dianalisis dengan menggunakan metode LPC. Hasil dari analisis LPC yang berupa koefisien LPC dibuat *template*-nya dengan menggunakan metode VQ, sehingga diperoleh representasi dari koefisien LPC berupa *codebook-codebook*. Pengujian program pengenalan dengan menggunakan data rekaman dan data *realtime* dilakukan setelah *codebook* dari masing-masing pengucap terbentuk.

#### 3.1 Perekaman dan Pembacaan Sinyal Suara

Sinyal ucapan dari masing-masing pengucap yang akan dijadikan objek dalam pembuatan program direkam untuk dijadikan sebagai data masukan bagi proses pelatihan dan pengujian. Ucapan masukan yang digunakan untuk pelatihan berupa kalimat dengan durasi waktu antara 4 s sampai 6,5 s, sedangkan ucapan untuk data uji berupa teks dengan durasi waktu antara 0,5 s sampai 1,5 s.

Proses perekaman ini dilakukan dengan menggunakan perintah `wavrecord`, instruksinya adalah sebagai berikut:

```
y = wavrecord(durasi, frek.cuplik, jumlah bit);
```

dimana  $y$  adalah data hasil perekaman,  $durasi$  adalah rentang waktu perekaman dalam detik, sedangkan  $frek.cuplik$  adalah besarnya frekuensi sampel, besar frekuensi sampel yang digunakan adalah 8 KHz, sedang yang terakhir adalah jumlah bit, dimana jumlah bit yang digunakan adalah sebesar 16 bit. Setelah direkam proses selanjutnya adalah menyimpan data hasil perekaman dengan menggunakan cara :

```
wavwrite(y, fs, NamaFile);
```

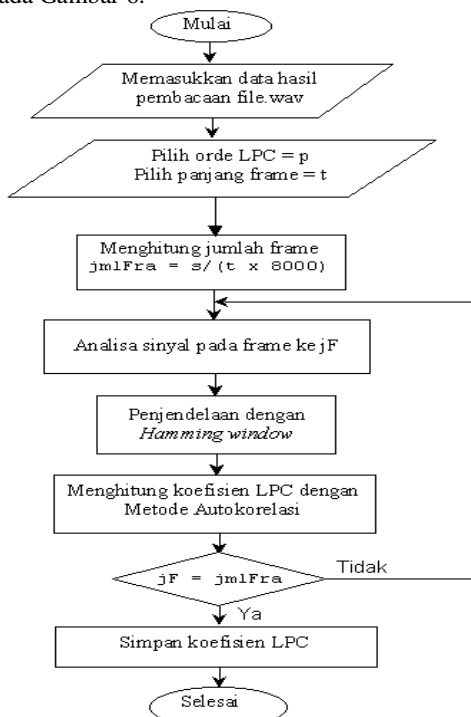
dimana  $y$  adalah sinyal hasil perekaman,  $f_s$  adalah frekuensi sampel dan  $\text{NamaFile}$  adalah nama file dari sinyal yang akan disimpan. File yang telah disimpan dalam format .wav. Jika ingin memanggil sinyal yang telah disimpan menggunakan cara;

```
[y, fs, Nbit] = wavread('NamaFile');
```

dimana  $y$  adalah sinyal hasil pembacaan oleh Matlab,  $f_s$  adalah frekuensi sampel pada saat sinyal direkam dan  $\text{Nbit}$  adalah resolusi perekaman.

### 3.2 Metode Analisis LPC untuk Mendapatkan koefisien LPC

Setelah sinyal ucapan dibaca atau dipanggil, proses selanjutnya adalah mencari nilai koefisien-koefisien LPC sinyal tersebut. Urutan pengolahan sinyal hingga diperoleh nilai koefisien-koefisien LPC dapat dilihat pada Gambar 6.



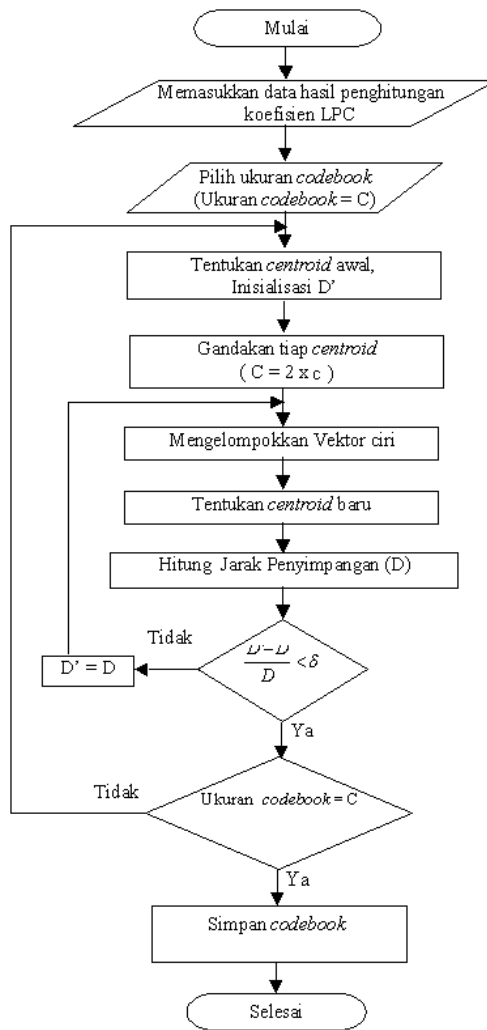
Gambar 6. Alur ekstraksi ciri sinyal bunyi

Pertama adalah dengan menentukan nilai-nilai parameter yang dibutuhkan dalam proses ekstraksi, antara lain orde sinyal yang memiliki simbol  $p$ , kemudian panjang *frame* yang memiliki simbol  $t$ , panjang *frame* akan menentukan jumlah segmen yang didapat dari pemecahan sinyal awal. Proses selanjutnya adalah melewati setiap segmen yang didapatkan dengan menggunakan *Hamming window* dan kemudian koefisien LPC didapat dengan menggunakan metode autokorelasi pada setiap segmen yang ada, jika koefisien LPC dari seluruh segmen telah diperoleh, maka koefisien-koefisien tersebut disimpan untuk digunakan dalam proses selanjutnya.

### 3.3 Pembuatan Template dengan Metode VQ

Hasil ekstraksi LPC(koefisien LPC) untuk basisdata kemudian dibuat *codebook*-nya dengan

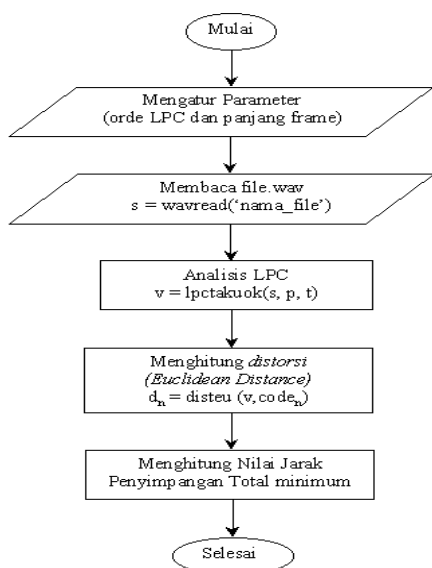
menggunakan metode VQ. dengan menggunakan metode VQ diperoleh *codebook* yang isinya merepresentasikan koefisien LPC yang telah diperoleh, yang berisi tentang informasi karakteristik dari masing-masing pengucap. Diagram alir untuk memperoleh *codebook* dengan menggunakan metode VQ ditunjukkan oleh Gambar 7



Gambar 7. Alur pembentukan *codebook* (template) dengan Metode VQ

### 3.4 Proses Pengujian

Pada tahap pengujian, sinyal ucapan yang tidak dikenal diekstraksi menggunakan LPC. Hasil ekstraksi yang berupa koefisien LPC dibandingkan dengan *codebook* dari masing-masing pengucap untuk dicari jarak penyimpangannya. Hasil identifikasi adalah jarak penyimpangan yang mempunyai nilai paling rendah. Alur program pengujian ditunjukkan oleh Gambar 8.



Gambar 8. Alur program pengujian

#### IV. PENGUJIAN DAN ANALISIS

Pengujian program ini dilakukan dalam dua tahap, tahap pertama yaitu pengujian dengan menggunakan data rekaman dan tahap kedua adalah pengujian secara *realtime*.

##### 4.1 Langkah Pengujian

Pengujian simulasi ini dilakukan dengan memberikan suatu sinyal masukan yang ingin dikenali. Simulasi pengenalan pengucap ini dapat dijalankan menggunakan tiga buah parameter ekstraksi komponen sinyal. Parameter-parameter tersebut dapat dilihat pada Tabel 1.

Tabel 1. Variasi nilai parameter

Parameter	Variasi
Orde LPC	8
	10
	12
	14
Panjang <i>Frame</i>	10 ms
	20 ms
Ukuran <i>codebook</i>	16
	32
	64

Pengujian dilakukan dengan mengkombinasikan masing-masing nilai parameter yang ada. Jumlah kombinasi didapat dengan mengalikan banyaknya variasi tiap-tiap parameter sehingga didapatkan kombinasi parameter sebanyak 24 kombinasi. Dengan melakukan pengujian menggunakan kombinasi tersebut, diharapkan akan didapat kombinasi parameter yang menghasilkan pengenalan yang paling baik.

##### 4.2 Pengujian dan Analisis

Pengujian yang pertama adalah pengujian dengan menggunakan data rekaman, yaitu ucapan yang telah direkam sebanyak 20 ucapan untuk masing-masing

responden. Ucapan ini diujikan pada seluruh kombinasi parameter. Hasil pengenalan untuk kombinasi parameter orde LPC 8, panjang *frame* 10 ms dan ukuran *codebook* 16 ditunjukkan oleh Tabel 2.

Tabel 2. Persentase pengenalan data rekaman (orde LPC 8, panjang *frame* 10ms dan ukuran *codebook* 16)

Nama Responden	Dikenali sebagai suara (%)									
	Arif B	Aji K	Ardinur	Dalma	Fachrudin	Putri	Nunung	Rani	Syafitri N	Vwwid
Arif B	55	25	5	5	0	0	10	0	0	0
Aji K	20	80	0	0	0	0	0	0	0	0
Ardinur	0	0	75	25	0	0	0	0	0	0
Dalma	20	5	5	70	0	0	0	0	0	0
Fachrudin	5	5	0	5	85	0	0	0	0	0
Putri	0	5	0	0	0	55	10	5	25	0
Nunung	0	0	0	0	0	15	35	20	30	0
Rani	0	0	0	0	0	0	10	85	5	0
Syafitri	0	20	0	0	0	0	10	5	65	0
Vwwid	0	5	0	5	0	5	15	0	15	55

Pada pengujian *realtime* program ditambah dengan batasan nilai jarak penyimpangan minimum, sehingga apabila besar jarak penyimpangan minimum yang diperoleh pada pengujian *realtime* diluar batasan itu maka ucapan yang masuk akan tidak dikenali. Pada pengujian *realtime* ini responden mengucapkan teks yang langsung direkam dan dianalisis sehingga langsung dapat dikenali atau tidak. Proses pengujian ini dilakukan terhadap 5 responden, tiap responden mengucapkan 10 teks yang berbeda untuk 24 macam kombinasi, sehingga total teks yang diucapkan oleh tiap responden adalah 240 teks. Dari pengujian yang dilakukan diperoleh hasil pengenalan untuk orde LPC 8, panjang *frame* 10 mS dan ukuran *codebook* 16 seperti pada Tabel 3.

Tabel 3. Persentase pengenalan data rekaman

(orde LPC 8, panjang *frame* 10mS dan ukuran *codebook* 16)

Nama Responden	Dikenali sebagai suara(%*)										Tidak Cocok
	Arif B	Aji K	Ardinur	Dalma	Fachrudin	Putri	Nunung	Rani	Syafitri N	Vwwid	
Arif B	30	20	10	30	0	0	0	0	0	0	10
Aji K	0	20	0	0	0	0	10	0	0	0	70
Fachrudin	0	10	0	0	60	0	0	0	0	0	30
Rani	0	0	0	0	0	0	0	80	20	0	0
Syafitri	0	40	0	0	0	10	10	0	40	0	0

Dari pengujian yang telah dilakukan dihitung rata-rata persentase pengenalan total dari ke 24 kombinasi parameter dengan cara,

$$(\%) \text{Pengenalan} = \frac{\text{JumlahPengenalanTiapPengucap}}{\text{JumlahPengucap}} \times 100\%$$

dimana jumlah pengucap pada pengujian dengan data rekaman adalah 10 dan jumlah pengucap pada pengujian *realtime* adalah 5. Rata-rata persentase pengenalan total untuk semua kombinasi parameter baik untuk pengujian dengan data rekaman maupun pengujian *realtime* ditunjukkan oleh Tabel 4.

Tabel 4. Rata-rata persentase total untuk pengenalan data rekaman dan pengenalan *realtime*.

Panjang Frame (mS)	Orde LPC	Data Rekaman			<i>Realtime</i>		
		Codebook			Codebook		
		16	32	64	16	32	64
	8	66	66,5	66,5	46	50	58
10	10	65,5	65,5	65,5	54	50	50
	12	63,5	64	63,5	56	48	54
	14	62,5	63	63,5	54	50	52
	8	60,5	61	61,5	54	44	46
20	10	67	66	67	54	50	42
	12	68,5	68,5	68	40	36	36
	14	65,5	66,5	65,5	44	42	30

Tabel 4 menunjukkan persentase pengenalan data rekaman diperoleh hasil pengenalan tertinggi terjadi pada saat kombinasi nilai parameter orde LPC 12, panjang *frame* 20 mS, ukuran *codebook* 16 dan orde LPC 12, panjang *frame* 20 ms, ukuran *codebook* 32 sebesar 68,5 %. Sedang hasil pengenalan tertinggi untuk pengenalan *realtime* terjadi saat kombinasi parameter orde LPC 8, panjang *frame* 10 mS, ukuran *codebook* 64 sebesar 58 %.

Dari Tabel 4 didapatkan bahwa untuk pengenalan *realtime* memiliki persentase pengenalan yang lebih rendah jika dibandingkan dengan pengenalan dengan data rekaman. Perbedaan antara hasil pengenalan data rekaman dan pengenalan *realtime* disebabkan sinyal ucapan data rekaman direkam langsung setelah pengambilan data latihan sedangkan pada pengenalan *realtime* beberapa hari setelah dilakukan pengambilan data latihan, hal ini mengakibatkan adanya perbedaan karakteristik sinyal ucapan, sehingga akan memberikan koefisien LPC yang berbeda pula.

Peningkatan hasil pengenalan bisa didapatkan dengan penambahan basisdata untuk setiap responden. Panjang sinyal ucapan yang digunakan pada proses pengenalan juga mempengaruhi hasil pengenalan. Semakin banyak sinyal ucapan, maka ciri-ciri akustik untuk setiap responden akan lebih banyak disarikan, sehingga hasil pengenalan akan semakin bagus.

Berdasarkan hasil analisa data pengujian, maka dalam pengenalan pengucap menggunakan analisis LPC ini perlu diperhatikan hal-hal sebagai berikut :

#### 1. Kondisi Lingkungan

Karena kondisi ruangan saat perekaman tidak kedap suara sehingga memungkinkan adanya suara-suara lain yang ikut terekam. Dengan adanya derau maka akan dihasilkan vektor ciri yang tidak mencirikan sinyal asli sehingga menghasilkan nilai jarak penyimpangan minimum pada tempat yang salah. Untuk mengatasi hal ini diperlukan ruangan perekaman yang seminimal mungkin bebas derau.

#### 2. Kondisi suara responden

Kondisi suara responden sangat mempengaruhi hasil pengenalan. Setiap sinyal suara yang diucapkan oleh seseorang selalu memiliki karakteristik yang berbeda, baik itu panjang-pendek, keras-pelan dan lain lain. Hal

ini dapat diatasi dengan megusahakan pengucapan sesuai kondisi normalnya.

#### 3. Letak mikrofon.

Karena perekaman dilakukan secara bertahap sehingga dalam peletakan mikrofon tidak sama. Jarak dan sudut mikrofon sangat mempengaruhi pada sinyal suara yang dihasilkan. Untuk mengatasi hal ini dapat dilakukan dengan mengatur jarak mikrofon dan sudut mikrofon yang sebisa mungkin sama.

#### 4. Cara perekaman sinyal suara

Perekaman sinyal suara yang tepat juga ikut mempengaruhi akurasi pengenalan. Cara perekaman yang baik adalah ucapan diucapkan sewajar mungkin dan tidak dibuat-buat, suara diucapkan tidak terlalu keras dan tidak terlalu lemah.

#### 5. Penggunaan basisdata

Basisdata sangat berpengaruh pada akurasi pengenalan, semakin banyak basisdata yang digunakan maka ciri karakteristik sinyal suara yang diperoleh juga akan semakin beragam, sehingga peluang pengenalan sinyal yang diujikan akan semakin besar. Basisdata yang digunakan sebaiknya direkam tidak paada suatu waktu, sehingga ragam cara pengucapan yang dimiliki seseorang akan terwakili, maka peluang pengenalan akan semakin besar.

## V PENUTUP

### 5.1 Kesimpulan

Dari pengujian yang telah dilakukan dapat diambil beberapa kesimpulan sebagai berikut:

1. Rata-rata persentase pengenalan tertinggi terhadap pengujian data rekaman mencapai 68,5 % dan terjadi pada saat kombinasi nilai parameter orde LPC 12, panjang *frame* 20 mS, ukuran *codebook* 16 dan orde LPC 12, panjang *frame* 20 ms, ukuran *codebook* 32.
2. Rata-rata persentase pengenalan tertinggi terhadap pengujian *realtime* mencapai 58 % dan terjadi pada saat kombinasi parameter orde LPC 8, panjang *frame* 10 mS, ukuran *codebook* 64.
3. Keadaan lingkungan dan kondisi dari pengucap pada saat perekaman suara untuk basisdata dan perekaman suara untuk pengujian akan memberikan pengaruh terhadap hasil pengenalan.

### 5.2 Saran

1. Penelitian ini bisa dikembangkan dengan menggunakan metode ekstraksi ciri yang lain seperti *Mel-Frequency Cepstrum Coefficients* (MFCC),.
2. Penelitian ini bisa dilakukan dengan menggunakan metode pencocokan pola yang lain seperti HMM(*Hidden Markov Model*) dan Jaringan Saraf Tiruan.
3. Penambahan data latihan (untuk *database*) dari tiap responden sehingga melengkapi ciri suara dari tiap-tiap responden.

## DAFTAR PUSTAKA

1. Cemal, Yakup, and Onder Filiz, *Speaker Verification*, Bogazici University, Iatanbul, 2001.
2. Do, Minh N, *An Automatic Speaker Recognition System*, Swiss Federal Institute of Technology, Switzerland.
3. Gold, Ben, and Nelson Morgan, *Speech and Audio Signal Processing : Processing and Perception of Speech and Music*, John Willey & Sons, Inc., New York, 1999.
4. Hanselman, Duane, and Bruce Littlefield, *MATLAB : Bahasa Komputasi Teknis*, Penerbit ANDI, Yogyakarta, 2000.
5. Ivana, *Pengenalan Ucapan Vokal Bahasa Indonesia Dengan Jaringan Saraf Tiruan Menggunakan Linear Predictive Coding*, Jurusan Teknik Elektro Universitas Diponegoro, Tugas Akhir, 2002.
6. Kinnunen, Tomi, Teemu Kilpelainen and Pasi Franti, *Comparison of Clustering Algorithms in Speaker Identification*, University of Joensuu, Finlandia.
7. Nurdian, Hendra, *Pengenalan Pengucap Berbasis Ekstraksi Komponen Cepstral Menggunakan Linear Predictive Coding*, Jurusan Teknik Elektro Universitas Diponegoro, Tugas Akhir, 2004
8. May, Ignatius Leo, *Pengenalan Vokal Bahasa Indonesia dengan Jaringan Syaraf Tiruan melalui Transformasi Wavelet Diskret*, Jurusan Teknik Elektro Universitas Diponegoro, Tugas Akhir, 2002.
9. Phan, Till T, and Thomas Soong, *Text Independent Speaker Identification*, 1999.
10. Proakis, John G., and Monolakis Dimitris G., *Pemrosesan Sinyal Digital ; Prinsip, Algoritma dan Aplikasi*, PT Prenhallindo, Jakarta. 1997.
11. Rabiner, R. Lawrence, and Biing-Hwang Juang, *Fundamentals of Speech Recognition*, PTR Prentice-Hall Inc., New Jersey, 1993.
12. Rastogi, Swati, and David Mayor, *An Automatic Speaker Recognition System*, [swati.rastogi@epfl.ch](mailto:swati.rastogi@epfl.ch).
13. Rabiner, R. Lawrence, and Ronald W. Schafer, *Digital Processing of Speech Signals*, Prentice Hall, New Jersey, 1978.
14. Syarip, Iip Achmad, *Penerapan Model Markov Tersembunyi dan Penyandian Ramalan Linier Untuk Pengenalan Kata Tersisolasi*, Jurusan Teknik Elektro, Universitas Diponegoro, Tugas Akhir, 2004.
15. Sitanggang, Doni, *Pengenalan Vokal Bahasa Indonesia dengan Jaringan Saraf Tiruan melalui Transformasi Fourier*, Jurusan Teknik Elektro Universitas Diponegoro, Tugas Akhir, 2002.
16. Xafopoulos, Alexandros, *Speaker Verification ( An Overview )*, TICSP (Tampere International Center for Signal Processing), Tampere Finland, 2001.

## FACHRUDIN AN – L2F 099 604

Lahir di Semarang-Jawa Tengah. Saat ini sedang menjalani proses untuk menyelesaikan studi Strata-1 pada Jurusan Teknik Elektro, Fakultas Teknik, Universitas Diponegoro Semarang.

Mengetahui / Mengesahkan  
Pembimbing I Pembimbing II

Wahyudi, ST,MT.  
NIP. 132 086 662

Achmad Hidayatno, ST, MT.  
NIP. 132 137 933