PERANGKAT LUNAK UNTUK MEMBUKA APLIKASI PADA KOMPUTER DENGAN PERINTAH SUARA MENGGUNAKAN METODE *MEL FREQUENCY CEPSTRUM COEFFICIENTS*

Anna Dara Andriana

Program Studi Teknik Informatika

Fakultas Teknik dan Ilmu Komputer Universitas Komputer Indonesia

Jl. Dipati Ukur 114 Bandung

Email : annaDaraandriana@yahoo.co.id

ABSTRAK

Perangkat lunak yang dibangun pada tugas akhir ini adalah sebuah program yang dapat membuka aplikasi pada komputer menggunakan perintah suara (*voice command*). Secara umum suara yang masuk akan dicocokan dengan data suara yang telah ada. Jika hasil dari pencocokan sama, maka sistem akan mengeksekusi aplikasi yang telah ditentukan sebelumnya. Permasalahannya adalah bagaimana komuter dapat mengerti perintah yang diucapkan oleh manusia.

Metode yang digunakan dalam feature extraction adalah *Mel frequency Cepstrum Coefficients* (MFCC). Dalam MFCC sendiri terdapan tujuh tahapan yaitu, *Pre-emphasis, Frame Blocking, Windowing, Fast Fourier Transform, Mel Frequency Waping, Discrete Cosine Transform dan Cepstral Liftering*. Untuk mengurangi waktu pemrosesan saat pencocokan suara maka digunakan K-Means Clustering untuk membuat vektor data sebagai representasi dari keseluruhan sampel data yang ada. Aplikasi *voice command* ini dibangun menggunakan Microsoft visual basic 6.

Pada pengujian didapatkan persentase keberhasilan sebesar 70,5% dalam hal keakuratan perintah suara yang diujikan terhadap seribu sampel data yang ada. Data tersebut didapatkan dari sepuluh orang yang terdiri dari lima orang wanita dan lima orang pria. Untuk memaksimalkan performa aplikasi, diperlukansuasana lingkungan yang tenang, agar aplikasi dapat berjalan dengan benar.

Kata Kunci : *Voice Command, Mel Frequency Cepstrum Coefficients (MFCC)*

1. PENDAHULUAN

Manusia merupakan mahluk sosial yang memerlukan komunikasi dengan sesamanya dalam kehidupan sehari-hari. Suara merupakan salah satu media komunikasi yang paling sering dan umum digunakan oleh manusia. Manusia dapat memproduksi suaranya dengan mudah tanpa memerlukan energi yang besar. Suara merupakan salah satu cara alami manusia untuk berkomunikasi. Dengan suara manusia dapat memberikan informasi maupun perintah. Oleh karena itu dibutuhkan suatu teknologi yang memungkinkan manusia dapat berkomunikasi melalui suara untuk berinteraksi dengan komputer (*voice command*).Perkembangan teknologi, yang semakin pesat telah menciptakan sebuah dunia informasi.

Hal ini semakin memicu kebutuhan akan adanya kemudahan dalam berinteraksi dengan komputer. Suara manusia merupakan salah satu bentuk *biometric* yang dapat digunakan untuk *person identification.* Selain itu dibandingkan *biometric person authentication* yang lain, pengenalan suara pembicara (*speaker recognition*) tidak membutuhkan biaya yang besar. Perangkat lunak pengenalan suara ini merupakan cikal bakal munculnya perangkat lunak pengenalan suara (*voice recognition*). Dengan adanya perangkat lunak pengenalan suara, manusia cukup memberika perintah-perintah secara lisan kepada komputer selayaknya memberikan perintah kepada orang lain

Perangkat lunak yang dibuat dalam tugas akhir ini merupakan salah satu bagian dari *artificial intelligent* yang mereplikasikan organ pendengaran manusia untuk dapat mengenali perintah pembicara berdasarkan suara yang dimasukan. Perangkat lunak ini dapat meminimalisir penggunaan *mouse*.

Perangkat lunak ini dibuat dengan menggunakan metode MFCC (*Mel Frequency Cepstrum Coefficients*) *feature extraction* dan didukung dengan *K-Means clustering.* MFCC *feature extraction* mengkonversikan sinyal suara kedalam beberapa vektor data berguna bagi proses pengenalan pembicara. Terdapat 7 tahapan dalam MFCC yaitu *Pre Emphasize, Frame Blocking, Windowing, Fast Fourier Transform, Mel Frequency*

Warping, Discrete Cosine Transform, dan *Cepstral Liftering.* Hasil dari MFCC *feature extraction* berukuran besar, sehingga membutuhkan waktu proses yang lama bila langsung digunakan untuk proses pengenalan suara. Oleh karena itu, dibutuhkan peranan dari metode *K-Means clustering* untuk membuat beberapa vektor pusat sebagai wakil dari kesuluruhan vektor data yang ada.

Metode *Mel Frequency Cepstrum Coefficients* ini memiliki beberapa kelebihan diantaranya adalah mampu menangkap informasi penting dalam sinyal suara, menghasilkan data seminimal mungkin tanpa menghilangkan informasi-informasi yang ada, dan mereplikasikan organ pendengaran manusia dalam melakukan persepsi terhadap sinyal suara.

Berdasarkan uraian di atas, pembangunan perangkat lunak ini dapat memudahkan user dalam berinteraksi dengan komputer menggunakan perintah suara. Oleh karena itu, tugas akhir ini diberi judul Perangkat Lunak Untuk Membuka Aplikasi Pada Komputer Dengan Perintah Suara Menggunakan Metode *Mel Frequency Cepstrum Coefficients (MFCC)* .

1.1 Maksud Dan Tujuan

1.1.1 Maksud

Maksud dari penulisan tugas akhir ini adalah untuk membangun perangkat lunak yang dapat membuka aplikasi di komputer dengan perintah suara menggunakan metode *Mel Frequency Cepstrum Coefficients* (MFCC) untuk mempermudah user berinteraksi dengan komputer.

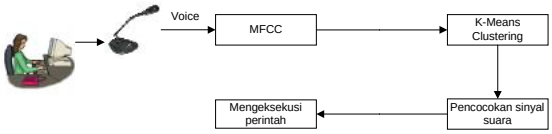
1.1.2 Tujuan

Adapun tujuan dari tugas akhir ini adalah sebagai alat yang mempermudah manusia untuk berinteraksi dengan komputer.

2. MODEL, ANALISA, DESAIN DAN IMPLEMENTASI

Proses *voice command* pada sistem dapat dijelaskan sebagai berikut. Pertama-tama, pengguna menentukan kata yang akan dijadikan sebagai perintah untuk membuka aplikasi tertentu kemudian mengucapkannya pada *microfone* ,dan memilih aplikasi apa yang akan dibuka oleh perintah tersebut. Sistem akan menyimpan data tersebut dalam sebuah database. Setelah itu barulah pengguna mengucapkan kata yang menjadi perintah tadi. Sistem akan mencocokan sinyal suara yang masuk dengan data yang terdapat dalam template. Jika data sinyal sama, maka komputer akan mengeksekusi aplikasi yang telah ditentukan sebelumnya.

Feature extraction mengkonversikan *signal* suara kedalam beberapa vektor data berguna bagi proses pengenalan suara. Dalam MFCC (*Mel Frequency*

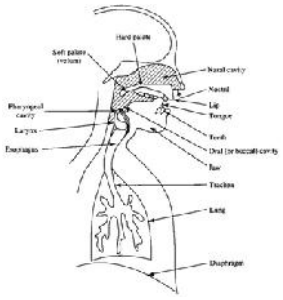


Gambar 1. Gambaran Umum Sistem Aplikasi Voice Command

Cepstrum Coefficients) sendiri terdapat tujuh tahapan yaitu *Pre Emphasize, Frame Blocking, Windowing, Fast Fourier Transform, Mel Frequency Warping, Discrete Cosine Transform,* dan *Cepstral liftering* ,yang telah dijelaskan pada bab sebelumnya. Data hasil MFCC (*Mel Frequency Cepstrum Coefficients*)*Feature extraction* kemudian akan memasuki tahap *K-Means Clustering* tahap ini membuat beberapa vektor pusat sebagai wakil dari kesuluruhan vektor data yang ada. Tahap mencocokan sinyal adalah tahap perhitungan *probabilitas* kemiripan pola dari tiap-tiap model sinyal yang mempunyai *probabilitas* kemiripan yang tertinggi dan berdasarkan pada *template.*

2.1 Suara

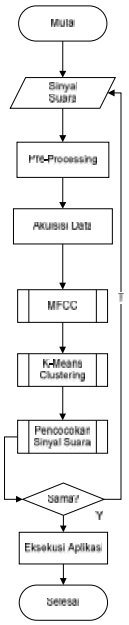
Pada sistem pengenalan suara oleh manusia terdapat tiga organ penting yang saling berhubungan yaitu : telinga yang berperan sebagai *transduser* dengan menerima sinyal masukan suara dan mengubahnya menjadi sinyal syaraf, jaringan syaraf yang berfungsi mentransmisikan sinyal ke otak, dan otak yang akan mengklasifikasi dan mengidentifikasi informasi yang terkandung dalam sinyal masukan.



Gambar 2. Organ Tubuh Manusia

2.2 Analisis Pengenalan Suara

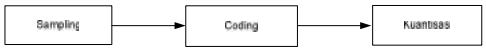
Dalam aplikasi ini terdapat 4 tahapan yaitu, MFCC, K-*Means Clustering* , pencocokan sinyal, dan eksekusi program. Secara umum program mengikuti alur berikut ini :



Gambar 3. Gambaran kerja sistem Aplikasi Voice Command

2.2.1 Pre Processing

Sinyal suara yang akan diproses bersifat analog sehingga jika akan dilakukan pengolahan secara digital, sinyal suara tersebut harus dikonversi menjadi sinyal digital, berupa urutan angka dengan tingkat presisi tertentu yang dinamakan *analog to digital conversion* dengan menggunakan *analog-to-digital converter* (ADC). Konsep Kerja ADC terdiri dari tiga proses :



Gambar 4. Konsep Kerja ADC (Analog To Digital Converter)

Keterangan konsep kerja ADC :

1. *Sampling* adalah konversi sinyal kontinu dalam domain waktu menjadi sinyal diskrit, melalui proses *sampling* sinyal pada selang waktu tertentu. Sehingga jika $x_0(t)$ adalah sinyal input, maka outputnya adalah $x_0(nT)$, dengan T adalah interval *sampling*.
2. Kuantisasi adalah proses untuk membulatkan nilai data kedalam bilangan bilangan tertentu yang telah ditentukan terlebih dahulu.
3. *Coding*, pada proses ini, tiap nilai diskrit yang telah didapat, direpresentasikan dengan angka binary n-bit.

2.2.2 Akusisi Data

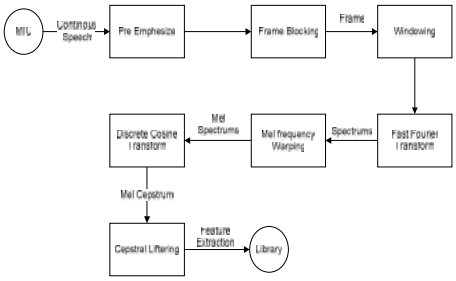
Data berupa sinyal suara diperoleh dengan cara merekam suara melalui mikrofon yang dihubungkan dengan komputer. Perekaman suara di dalam aplikasi menggunakan frekuensi sampling standar 8000Hz. Suara dengan format *.wav* ini bisa menggunakan 16 *bits/sample* dan 1 untuk *channel mono.* Durasi suara yang direkam apabila lebih pendek lebih mudah untuk diambil perbedaan fiturnya. Dalam analisis ini digunakan contoh durasi rekaman yang diambil adalah 2 detik.

$$X = FS\ x\ dt(detik)\ x\left(\frac{bit}{8}\right) x\ j \qquad (1)$$

dimana :
X = data sampling sinyal
Fs = frekuensi sampling
dt = durasi rekaman (detik)
bit = jumlah bit resolusi
j = 1 untuk *mono* atau 2 untuk *stereo*

2.2.3 MFCC

Metode *Mel Frequency Cepstrum Coefficients* (MFCC) ini menggunakan beberapa parameter yang akan berperan penting dalam menentukan tingkat keberhasilan pengenalan *signal* suara. Berikut ini adalah keseluruhan proses MFCC *Feature extraction* :



Gambar 5. Proses metode *Mel Frequency Cepstrum Coefficients* (MFCC)

2.2.3.1 Pre-Emphasis

Pre-emphasis dilakukan untuk memperbaiki *signal* dari gangguan *noise*, sehingga dapat meningkatkan tingkat akurasi dari proses *feature extraction. Default* dari nilai alpha yang digunakan dalam proses *pre-emphasis filtering* adalah 0,97.

$$y\ [n] = s\ [n] - \alpha\ s[n-1] \qquad (2)$$

$y[n] = signal\ hasil\ pre-emphasize$
$s[n] = signal\ sebelum\ pre-emphasize\ filter$

2.2.3.2 Frame Blocking

Hasil perekaman suara merupakan sinyal analog yang berada dalam domain waktu yang bersifat variant time, yaitu suatu fungsi yang bergantung waktu. Oleh karena itu sinyal tersebut harus dipotong-potong dalam slot-slot waktu tertentu agar dapat dianggap invariant. Sinyal suara dipotong sepanjang 20 milidetik .Setiap potongan tersebut disebut *frame*.

Untuk menghitung jumlah Frame digunakan rumus :

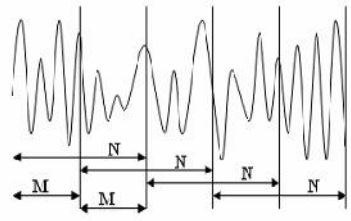
Jumlah frame = ((I-N)/M)+1) (3)

I = Sample rate

N= Sample point (Sample rate * waktu framing (s))

M = N/2

Potongan *frame* digambarkan seperti gambar dibawah ini :



Gambar 6. Framing

2.2.3.3 Windowing

Fungsi window yang paling sering digunakan dalam aplikasi speaker recognation adalah Hamming Window. Fungsi ini menghasilkan sidelobe level yang tidak terlalu tinggi (kurang lebih -43dB) selain itu noise yang dihasilkan pun tidak terlalu besar (kurang lebih 1.36 BINS).

Window hamming :

$$W_{ham}(n) = 0.52 - 0.46\cos\left[\frac{2\pi n}{(N-1)}\right] \quad 0 \le n \le N-1$$

2.2.3.4 FFT

FFT (Fast Fourier Transform) adalah teknik perhitungan cepat dari DFT. FFT adalah DFT dengan teknik perhitungan yang cepat dengan memanfaatkan sifat periodikal dari transformasi fourier. Perhatikan definisi dari FFT :

$$F(k) = \sum_{n=1}^{N} f(n).e^{-j2\pi knT/N}$$

Atau dapat dituliskan dengan :

$$F(k) = \sum_{n=1}^{N} f(n)\cos(2\pi nkT/N) - j\sum_{n=1}^{N} f(n)\sin(2\pi nkT/N)$$

Untuk melihat nilai hasil FFT digunakan rumus

$$|f(u)| = [R^2 + I^2]^{1/2}$$

2.2.3.5 Filterbank

Filterbank menggunakan representasi konvolusi dalam melakukan filter terhadap sinyal. Konvolusi dapat dilakukan dengan melakukan multiplikasi antara spektrum sinyal dengan koeffisien *filterbank*. Berikut ini adalah rumus yang digunaka dalam perhitungan *filterbanks*.

$$Y[i] = \sum_{j=1}^{N} S[j]H_i[j]$$

N = jumlah magnitude spectrum

S[j] = magnitude spectrum pada frekuensi j

$H_i[j]$= koefisien *filterbank* pada frekuensi j $(1 \le i \le M)$

M= jumlah Channel dalam *filterbank*

Dimana $H_i = \frac{mel\ f}{X_{i/2}}$

2.2.3.6 DCT

Proses ini merupakan langkah akhir dari *feature extraction*. Hasil dari DCT ini adalah fitur-fitur yang dibutuhkan oleh penulis untuk melakukan proses analisa terhadap pengenalan suara tersebut. Menggunakan rumus :

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

S_k= keluaran dari proses *filterbank* pada indeks k

K = jumlah koefisien yang diharapkan

2.2.3.7 Cepstral Liftering

Hasil dari fungsi DCT adalah cepstrum yang sebenarnya sudah merupakan hasil akhir dari proses *feature extraction*. Tetapi, untuk meningkatkan kualitas pengenalan, maka cepstrum hasil dari DCT harus mengalami *cepstral liftering*

$$w[n] = \left\{1 + \frac{L}{2}\sin\left(\frac{n\pi}{L}\right)\right\}$$

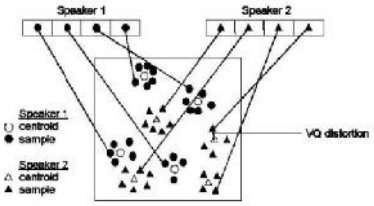
L = jumlah cepstral coefficients

N= index dari cepstral coefficients

2.2.4 K-Means Clustering

Clustering merupakan faktor yanng paling fundamental dalam *pattern recognation*. Masalah utama dari *clustering* adalah mendapatkan beberapa nilai vektor pusat yang dapat mewakili keseluruhan vektor dari hasil *feature extraction*. K-*means Clustering* adalah salah satu metode yang digunakan untuk mempartisi vektor hasil *feature extraction* ke dalam k vektor pusat[3].

K-Means Clustering adalah prroses memetakan vektor-vektor yang berada pada lingkup wilayah yang luas besar menjadi sejumlah tertentu (k) vektor. Wilayah yang terwakili oleh vektor pusat hasil dari proses kuantisasi disebut sebagai *cluster*. Sebuah vektor pusat hasil dari proses kuantisasi dikenal sebagai *codewords*. Sedangkan kumpulan dari vektor pusat dikenal sebagai *codebooks*. Gambar berikut menunjukan hasil ilustrasi *K-Means Clustering.*



Gambar 2.1 K-Means Clustering

Keuntungan dari diimplementasikannya *K-Means Clustering* dalam merepresentasikan *speech spectral vectors* adalah :

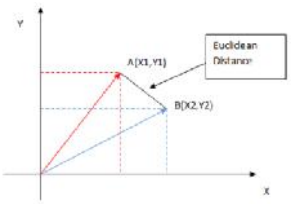
1. Mengurangi *storage memory* yang digunakan untuk analisis informasi spektral.
2. Mengurangi perhitungan yang digunakan untuk menentukan kemiripan dari vektor spektral.

Kelemahan dari penggunaan *K-Means Clustering codebooks* dalam merepresentasikan *speech spectral vectors* adalah :

1. Timbulnya spektral distorsi. Hal ini terjadi karena vektor yang dianalisa bukanlah vektor asli, tetapi sudah mengalami proses kuantisasi
2. *Storage* yang digunakan untuk menyimpan *codebooks* vektor sering kali menjadi masalah. Untuk jumlah *codebooks* yang besar, membutuhkan storage yang cukup besar juga.

2.2.5 Pencocokan suara

Untuk dapat membuka aplikasi yang diinginkan, maka data *signal* baru yang masuk akan dicocokan dengan data yang telah ada dalam database sebelumnya. Setiap vektor dari model yang diujicobakan, dibandingkan dan dihitung *euclidean distance*-nya dengan semua vektor yang ada pada salah satu model database secara bergantian. Kemudian diambil *distance* yang paling minimum antara sebuah vektor pada model yang diuji cobakan dengan semua vektor yang ada pada salah satu model database. Sehingga didapatkan N minimum *distance*. Berikut ini adalah gambar perhitungan distance :



Gambar 7 Perhitungan Distance

$$d(A,B) = \sqrt{(X1 - X2)^2 + (Y1 - Y2)^2}$$

Hitung minimal distance dari setiap codebook yang ada. Disitulah dianggap sebagai kemiripan sinyal.

3. HASIL DAN DISKUSI

Dalam pengujian pengenalan suara ini, terdapat 10 orang yang menjadi bahan uji untuk aplikasi *voice command* . Diantaranya terdapat lima orang wanita dan lima orang pria dengan usia antara 20 tahun sampai 24 tahun. Setiap orang mendapatkan kesempatan sebanyak 10 kali untuk setiap aplikasi. Sehingga dalam satu kata terdapat 100 kali percobaan. Perhitungan akurasi, digunakan untuk mengetahui seberapa besar kesuksesan aplikasi voice command ini. Keterangan dikenali merupakan kecocokan perintah dengan aplikasi yang dieksekusi

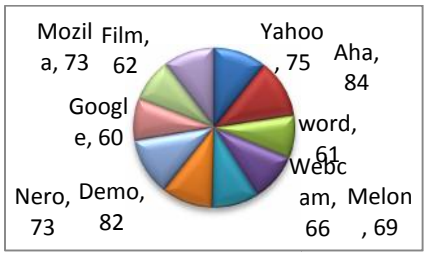
3.1 Hasil Pengujian Perintah

Dari hasil persentase masing-masing kata maka didapatkan nilai persentase keseluruhan yaitu :

$$\frac{75 + 84 + 61 + 69 + 66 + 73 + 82 + 60 + 62 + 73}{10} * 100\ \% = \frac{705}{10} * 100\ \% = 70{,}5\ \%$$

3.1.1 Grafik Hasil Pengujian

Berikut ini adalah grafik keseluruhan hasil pengujian dari sepuluh aplikasi yang diujikan :



Gambar 8 Grafik Hasil Pengujian

4. KESIMPULAN DAN SARAN

Kesimpulan merupakan ringkasan yang diambil dari hasil analisis yang telah dilakukan. Berisi tentang hasil kinerja metode *Mel Frequency Cepstrum Coefficients* (MFCC) pada aplikasi *voice command* yang telah dibuat. Sedangkan saran ditujukan kepada pengguna agar dapat memaksimalkan kinerja perangkat lunak *voice command* ini.

4.1 Kesimpulan

Berdasarkan analisis yang telah dilakukan terhadap pembangunan aplikasi *voice command* menggunakan metode mel frequency cepstrum coefficients (MFCC) maka dapat disimpulkan bahwa:

1. Aplikasi ini memiliki tingkat keberhasilan sekitar 70,5 %.
2. Keberhasilan aplikasi ini masih tergantung pada seberapa besar noise yang datang.
3. Metode Mel Frequency Cepstrum coeffisien (MFCC) ini dapat dipakai sebagai metode yang baik dalam melakukan feature extraction.
4. Tingkat sensitifitas microphone yang digunakan juga dapat mempengaruhi hasil dari aplikasi ini.
5. Pengguna dapat berinteraksi dengan komputer menggunakan perintah suara.

4.2 Saran

Agar perangkat lunak ini dapat berkembang ke depannya, penulis menyarankan beberapa hal sebagai berikut yaitu :

1. Meningkatkan kemampuan sistem untuk membedakan suara noise. Menambahkan modul-modul perintah seperti shortcut pada aplikasi microsoft word dan lain-lain.
2. Agar aplikasi *voice command* ini dapat bermanfaat bagi pengguna komputer yang mengalami cacat fisik, maka diperlukan perluasan perintah pada aplikasi.
3. Untuk kedepannya diharapkan aplikasi ini dapat mengcover semua perintah dalam komputer, sehingga dapat membantu interkasi manusia dengan komputer menjadi lebih mudah.
4. Diharapkan ada metode yang benar-benar dapat menghilangkan noise, sehingga keberhasilan aplikasi akan semakin besar

DAFTAR PUSTAKA

[1]. Kusumadewi, S., *Artificial Intelligent*, ed. G. Ilmu. 2003, Yogyakarta.
[2]. people.revoledu.com/cardi.teknomo. [cited 2011 14 juni].
[3]. prawira, i., *software pembuka aplikasi komputer*. 2009.
[4]. Bala,A., *Voice command recognation system based mfcc* .2003.
[5]. Bimanto, Iwan., *Multimedia Digital* , ed.andi.2003, Yogyakarta
[6]. hadi,p., *pengelompokan usia berdasarkan suara menggunakan metode K-means Clustering.*2009.
[7]. Somantri,y ., *pengenalan pembicara dengan ekstraksi ciri MFCC.* 2006.
[8]. tanudjaja, h., *pengolahan sinyal digital &sistem pemrosesan sinyal*, ed. andi. 2009, yogyakarta
[9]. Resmawan,a., All About Technology. [cited 2011 20 maret]