Implementasi *Hidden Markov Model Toolkit* (HTK) pada aplikasi *Speaker Recognition*

Wildan Zulfikar Djunaedi[1], Hendri Maja Saputra[2], Midriem Mirdanies[3]
[1]Universitas Komputer Indonesia
Jl. Dipati Ukur No. 112-116, Bandung 40132 Jawa Barat,
Telp. 022 2504119
[2,3]Pusat Penelitian Tenaga Listrik dan Mekatronik – LIPI
Komp. LIPI Gd.20, Jl. Cisitu No.21/154D, Bandung 40135 Jawa Barat,
Telp. 022 2503055, Fax 022 2504773
wildanzulfikar@email.unikom.ac.id[1], hend018@lipi.go.id[2], midr001@lipi.go.id[3]

Abstrak – *Implementasi dari Hidden Markov Model Toolkit (HTK) pada aplikasi speaker recognition telah dibahas pada tulisan ini. HTK merupakan sebuah toolkit/library untuk membangun dan memanipulasi algoritma Hidden Markov Model (HMM), dimana bahasa pemrograman yang digunakan adalah bahasa c. HTK dapat digunakan untuk membangun aplikasi pada bidang pengenalan suara baik speech recognition atau speaker recognition berdasarkan algoritma HMM. Dalam tulisan ini, telah dijelaskan cara menggunakan HTK khusus untuk aplikasi speaker recognition. Implementasi dari HTK telah dilakukan untuk mendeteksi 3 jenis suara tembakan. Data uji yang digunakan adalah 21 sampel suara (.wav), dimana 20 sampel suara digunakan sebagai data latih (training), dan 1 suara digunakan sebagai data pengujian. Hasil eksperimen menunjukkan bahwa semua suara yang diujikan dapat dideteksi atau dikenali oleh sistem yang telah dibuat.*

Kata kunci*: Hidden Markov Model Toolkit, Hidden Markov Model, speaker recognition, bahasa c*

1. Pendahuluan

Sound recognition merupakan sebuah teknologi yang memungkinkan sebuah mesin untuk dapat mendeteksi jenis suara yang didengar. Sound recognition terdiri dari beberapa jenis antara lain *speech recognition* untuk mendeteksi kata apa yang didengar dan *speaker recognition* untuk mendeteksi siapa atau apa yang bicara. *Speaker recognition* merupakan suatu proses yang kompleks, sehingga diperlukan *toolkit/library* yang mengimplementasikan metode *sound recognition* tertentu untuk dapat diimplementasikan pada aplikasi yang dibuat.

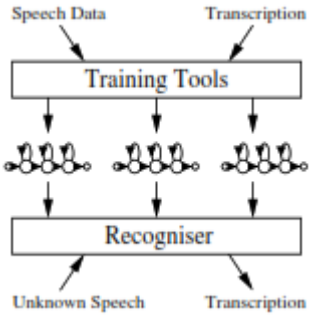
Penelitian mengenai *speaker recognition* ini telah dilakukan antara lain oleh Raja yang menjelaskan mengenai *speaker recognition* pada kondisi suara yang tertekan [1], Lan mengenai metode learning untuk aplikasi *speaker recognition* yang disebut *Extreme learning machine* (ELM) [2]. Selain itu, Jourani juga telah melakukan penelitian mengenai *speaker recognition* menggunakan *Gaussian mixture models* (GMM) [3].

Penelitian yang khusus menggunakan *Hidden Markov Model Toolkit* (HTK) antara lain Adetunmbi yang menggunakan HTK untuk aplikasi speech-to-text pada Standard Yorùbá [4], Khelifa yang menggunakan HTK untuk pengembangan dan eksperimen mengenai *speech recognition* pada bahasa arab [5]. Pada kedua penelitian tersebut, HTK digunakan bukan pada aplikasi *speaker recognition* namun pada *speech recognition*.

Pada tulisan ini, dijelaskan mengenai cara penggunaan HTK yang mengimplementasikan metode *Hidden Markov Model* (HMM) pada aplikasi *speaker recognition*. Ujicoba juga telah dilakukan untuk melihat apakah aplikasi yang dibuat menggunakan toolkit ini dapat berjalan dengan baik, dimana data uji yang digunakan adalah 21 sampel suara (.wav), dimana 20 sampel suara digunakan sebagai data latih (*training*), dan 1 suara digunakan sebagai data pengujian.

2. Metoda Penelitian

HTK merupakan sebuah toolkit yang digunakan untuk membangun model markov tersembunyi (HMM) [6]. HTK terutama dirancang untuk membangun aplikasi *sound recognition* berbasis algoritma HMM. Pada gambar 1 dapat dilihat sebuah blok diagram proses pengenalan suara menggunakan HTK [7]



Gambar 1. Proses pengenalan suara menggunakan HTK

Secara umum, terdapat 2 tahapan pada proses pengenalan suara baik jenis *sound recognition* maupun *speaker recognition* yaitu tahapan training (pada HTK menggunakan *Training Tools*) dan tahapan pengujian (pada HTK menggunakan *Recogniser).* Tahapan training digunakan untuk mengajarkan sistem/mesin agar dapat mengenali suara yang digunakan, misalkan dalam kasus ini, digunakan 3 jenis suara tembakan yaitu AK47, MP5, dan P90, dimana setiap jenis suara terdiri dari 20 sampel suara. Tahapan berikutnya adalah pengujian yang digunakan untuk menguji sistem yang telah ditraining untuk melihat apakah sistem dapat mengenali suara yang telah ditraining tersebut.

HTK *training tools* digunakan untuk mengestimasi parameter yang telah diset oleh HMM menggunakan pelatihan pengucapan dan transkripsi yang terkait. Kemudian pengucapan yang tidak dikenal (*unkwonn Speech)* ditranskripsikan *(transcription)* menggunakan alat pengenalan HTK (*recognizer)*. *Toolkit* ini juga berisi beberapa perintah yang diperlukan untuk membangun sebuah sistem menggunakan algoritma HMM, seperti konversi audio *waveform (.wav)* menjadi sebuah koefisien *Mel-frequency cepstral coefficients* (MFCC), formula *Baum-Welsh*, dan algoritma perhitungan Viterbi.

Sistem *speaker recognition* yang telah dibuat pada penelitian ini dapat dilihat pada Gambar 2.

Gambar 2. Sistem *speaker recognition* pada tulisan ini

ISBN: 978-623-7036-34-0

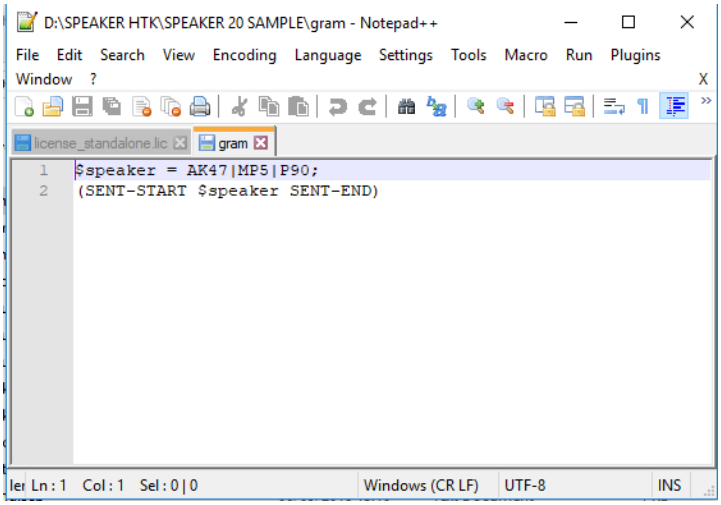
Suara yang digunakan dalam penelitian ini berasal dari file .wav yang diambil dari website sounddogs.com [8]. Toolkit/library HTK ini diinstal pada Laptop dengan spesifikasi processor intel core i5 7200U, memori 8 gb, windows 10 x64-bit, dengan sound card onboard. Speaker digunakan untuk mendengarkan suara yang digunakan dalam penelitian ini.

3. Hasil dan Analisis

Proses pengembangan aplikasi speaker recognition menggunakan HTK diperlukan beberapa tahap proses.

3.1. The Task Grammar

Langkah pertama dalam membangun sistem pengenalan (pengenalan ucapan atau pembicara) adalah untuk membangun file tata Bahasa untuk membiasakan sistem dengan sifat data yang akan dikenali. Tata Bahasa terdiri dari satu set variabel definisi yang diikuti oleh ekspresi regular yang menggambarkan kata-kata atua pembicara untuk mengenali. Untuk sistem pertama ini, file tata Bahasa dapat dilihat pada gambar 3.1 Dibawah ini:



Gambar 3.1 File tatabahasa

Dimana didalam file tersebut didefinisikan 3 orang pembicara dan itu termasuk dengan "SENT-START" dan "SENT-END" didefinisikan sebagai model *silence,* yang digunakan untuk menandai awal dan akhir dari ucapan. Setelah file tata Bahasa telah didefinisikan, diperlukan sebuah jaringan kata (*word network)*. HTK menentukan suatu jaringan kata dengan menggunakan notasi tingkat rendah yang disebut HTK *Standard Lattice Format* (SLF) dimana setiap contoh kata dan transisi kata demi kata terdefatar secara eksplisit. Sebuah jaringan kata dapat dibuat secara otomatis dengan menggunakan tool **HParse**

HParse gram wdnet **(1.1)**

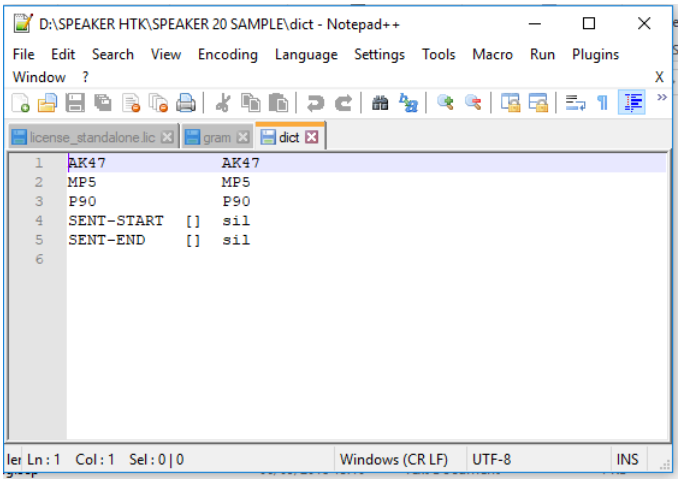
HParse akan membuat sebuah file jaringan kata bernama"wdnet" berdasarkan dari isi file "gram"

3.2. Kamus (*the Dictionary)*

Kamus (*the Dictionary)* adalah sebuah file yang menghubungkan kata atau variabel yang didefinisikan oleh tata Bahasa untuk menyesuaikan dengan model HMM, digunakan untuk menhubungkan model HMM yang akan diekstrak ke definisi variabel dalam tata Bahasa. Sebuah kamus yang dapt dibangun menggunakan database untuk pengucapan kata, list kata, tes kosakata

ISBN: 978-623-7036-34-0

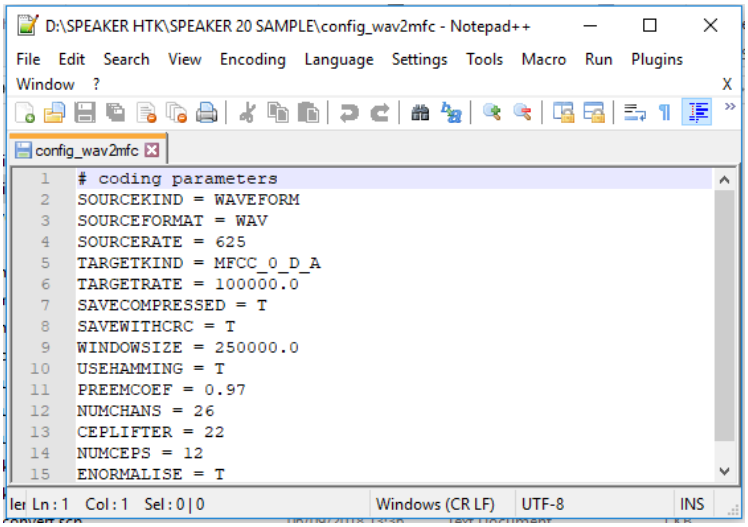
dan yang lainnya. Biasanya dilakukan untuk sebuah data yang besar yang perlu dikenali ataupun untuk proses pengenalan kata. Kamus (*dictionary)* akan seperti Gambar 3.2



Gambar 3.2. File *dictionary*

3.3. Konfigurasi (*the Configuration files)*

File konfigurasi merupakan sebuah skrip yang digunakan HTK untuk parameter data dan mengekstrak informasi yang relevan dari spektrum bicara. Kami menggunakan koefisien MFCC dan pada gambar 3.3 adalah file konfigurasi untuk mengekstrak data MFCC dari spektrum bicara



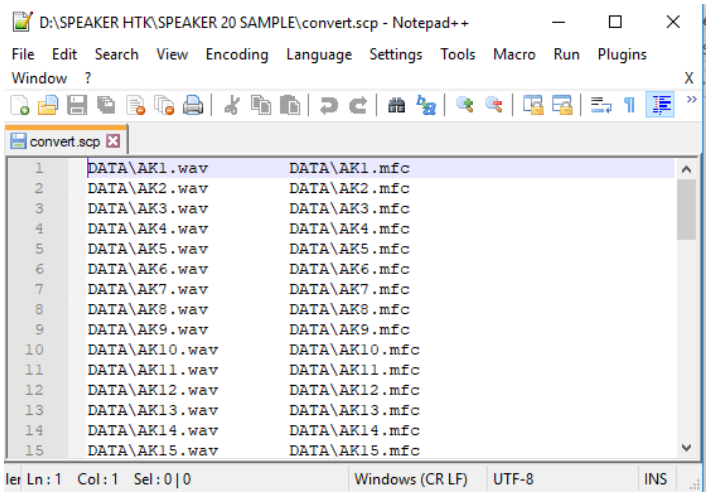
Gambar 3.3 file konfigurasi

Pada gambar xx menunjukan bahwa *source* adalah sebuah file *waveform* dan jenis targetnya (*targetkind)* adalah koefisien MFCC ditambah delta dan akselerasi, *windowsize* digunakan untuk durasi dari frame dan *targetrate* digunakan untuk durasi frame yang dilewati. Semua nilainya dalam bentuk detik hingga 100nS

File MFCC akan di list dalam sebuah file *convert.scp*. Bersama dengan bagian setiap gelombang dan dapat di ekstrak dengan mengeksekusi perintah berikut didalam *command window*:

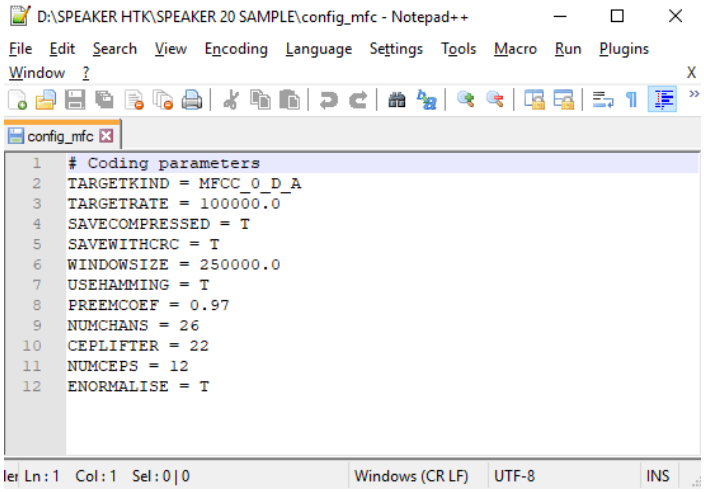
HCopy -T 1 -C config_wav2mfc -S convert.scp **(1.2)**

ISBN: 978-623-7036-34-0



Gambar 3.4 File *convert* .wav menjadi .mfc

Setelah koefisien MFCC dibuat, konfigurasi file yang baru harus dibuat. Pada gambar 3.5 digunakan untuk memberitahui HTK tentang MFCC dan akan digunakan kapanpun ketika MFCC dipanggil

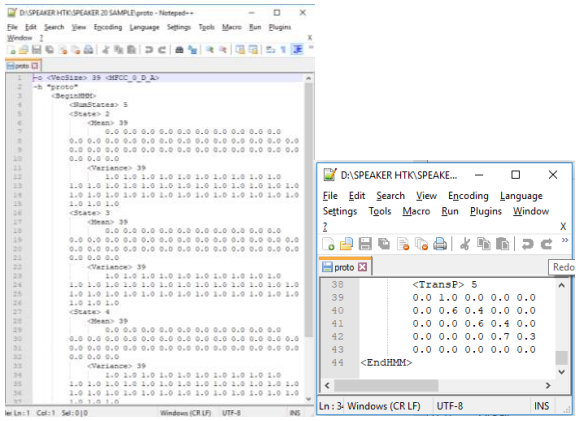


Gambar 3.5. file konfigurasi MFCC

3.4 Topologi model dan definisi HMM

Sebuah model topologi harus didefinisikan ketika membuat sistem pengenalan dan spesifik menggunakan HTK. Topologi ini memberikan bentuk kedalam sistem. Jumlah state (keadaan) dalam HMM, sifat dari HMM(kiri ke kanan atau *ergodic*) dan banyak parameter lainnya. Model didefinisikan pada Gambar 3.6

ISBN: 978-623-7036-34-0



Gambar 3.6 file prototipe atau topologi

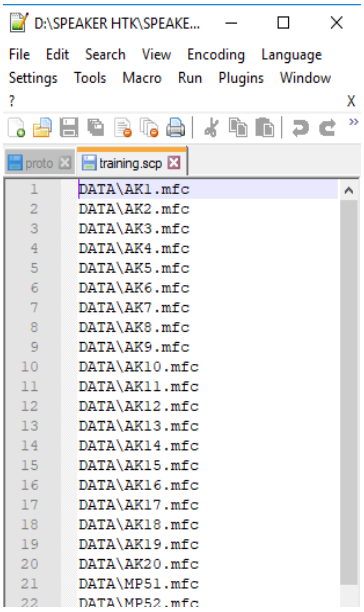
Definisi dimulai dengan meng-spesifikasikan dimensi dari ruang yang akan kita gunakan (nomor dari vector) dan jenis data / fitur MFCC_0_D_A. fitur ini adalah MFCC ditambahkan kedalam koefisien *Delta* (1st derivative) dan *acceleration* (2nd derivative). Definisi HMM berada diantara <beginhmm> dan <endhmm>, didalam perintah ini kita dapat menemukan statistic yang lebih spesifik (*mean* dan *variance*) mengenai masing-masing bagian, dan yang terakhir merupakan sebuah matriks probabilitas transisi

TransP menjelaskan tentang sifat dari sistem, contoh jika itu adalah sebuah matriks dengan bentuk segitiga atas maka itu adalah sistem dari kiri ke kanan, namun jika itu adalah sebuah matriks pebuh maka itu adalah sistem yang saling terhubung *ergodic*. Kebanyakan definisi HMM dengan sistem HTK menggunakan model kiri ke kanan

Setelah mendefinisikan prototipe dari model HMM, kita akan menggunakan perintah berikut untuk mengisialisasi model dengan data digunakan untuk sistem latih

HCompV -C config_mfc -f 0.01 -m -S training.scp -M hmm0 proto **(1.3)**

Gambar 3.7 merupakan File training.scp. file tersebut adalah sebuah skrip tempat data yang akan digunakan untuk pelatihan terdaftar

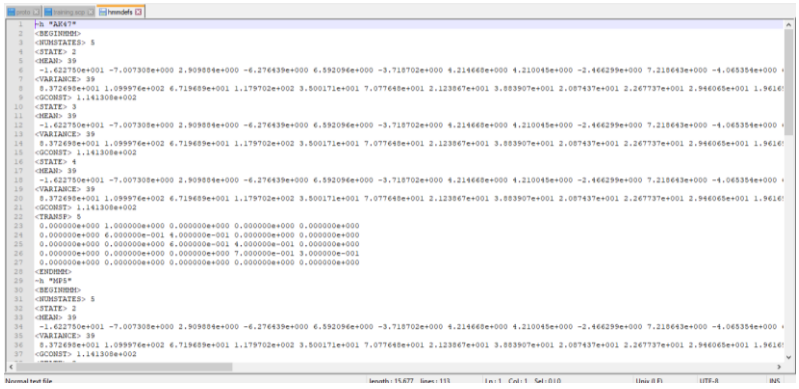


Gambar 3.7. file training.scp

ISBN: 978-623-7036-34-0

Setelah itu, dua buah file akan dibuat didalam sebuah folder hmm0 (*vfloors* dan *proto*). Kedua file ini digunakan untuk mengestimasi ulang model berdasarkan data pelatihan yang nantinya akan digunakan sebagai pengenalan

Hmmdefs dibuat dengan memotong (*cut)* sisa file dari proto dan menempelkan *(paste)* di file teks baru untuk setiap model yang ingin kita buat. Dalam kasus ini, kita memiliki 3 model pembicara kemudian kita salin seperti gambar 3.8



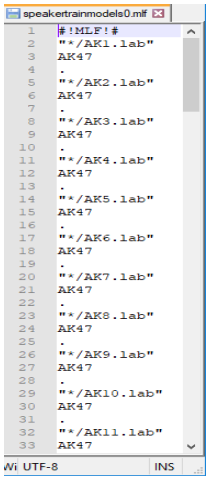
Gambar 3.8 file *hmmdefs*

Kedua file yang barudibuat ini dengan *means* dan *variances* global akan digunakan untuk mengestimasi ulang model dengan perintah berikut

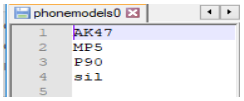
HERest -C config_mfc -I speakertrainmodels0.mlf -t 250.0 150.0 1000.0 -S training.scp -H hmm0/macros -H hmm0/hmmdefs -M hmm1 phonemodels0 (1.4)

sebuah model akan di-estimasi ulang berdasarkan dari formula *Baum-Welsh* dan keluaran filenya (*hmmdefs* dan *macro)* akan berada didalam sebuah folder (*hmm1)*

untuk dapat memperkirakan ulang model parameter menggunakan kamus yang telah dibuat, dua buah file dibutuhkan untuk dibuat seperti pada gambar 3.9



Gambar 3.9. file *speakertest*



Gambar 3.10 file *phonemodels*

ISBN: 978-623-7036-34-0

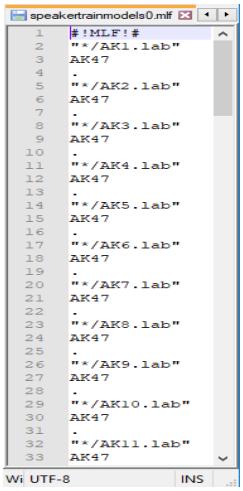
Gambar 3.9 adalah sebuah master label file, fimana file data pelatihan disebutkan disamping model yang data tersebut perlu diestimasi ulang dan ditetapkan

Gambar 3.10 adalah sebuah file teks dimana model dari pembicara HMM terdaftar

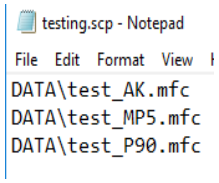
Kita estimasi ulang sebanyak dua kali dan tempatkan file output didalam dua buah folder baru (*hmm2* dan *hmm3)*

3.5 Pengenalan (*the recognition)*

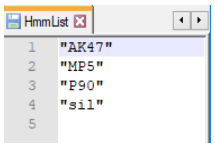
Setelah melakukan estimasi ulang, ada sebuah bagian dimana parameter yang telah diekstrak selama proses pelatihan akan diuji. untuk dapat membuat file itu, kita memerlukan 3 buah file, seperti pada gambar 3.11, gambar 3.12 dan gambar 3.13



Gambar 3.11 file *speakertrainmodels*



Gambar 3.12 file *testing.scp*



Gambar 3.13 file Hmmlist

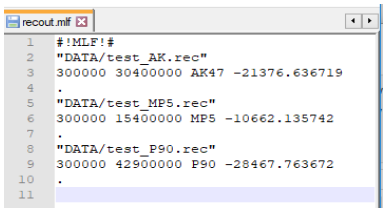
File-file tersebut akan digunakan ketika melakukan proses pengenalan. Masukan perintah berikut didalam *command window*:

HVite -H hmm3/macros -H hmm3/hmmdefs -S testing.scp -I recout.mlf -w wdnet -p 0.0 -s 5.0 dict HmmList (1.5)

HVite berbasis dari algoritma Viterbi, melakukan perhitungan pendekatan dan menyimpan hasilnya didalam file recout.mlf

ISBN: 978-623-7036-34-0

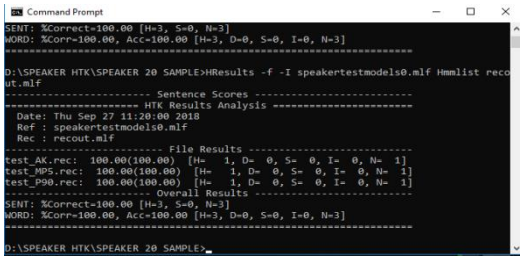
Hasil dari HVite adalah sebuah pembuatan file mlf baru, diabgian ini nama dari recout.mlf berisi informasi pengenalan yang akan digunakan oleh HTK nanti. File recout.mlf digambarkan seperti gambar 3.14



Gambar 3.14 file *recout*

Kemudian berdasarkan hasil dari HVite, kita dapat menampilkan hasil dengan menggunakan perintah berikut

HResults -I speakertestmodels0.mlf HmmList recout.mlf **(1.6)**



Gambar 3.15. Hasil pengenalan

Gambar 3.15 merupakan tampilan hasil dari eksekusi (1.6), Dimana didalamnya terdapat persentase hasil pengujian untuk *correction* dan *accuration*

Tabel 1. Hasil Pengujian

Speakers	**Waktu data *training* (s)**	**Waktu data *testing*(s)**	**Jumlah Data Training**	**Jumlah Data Test**	**Hasil Pengujian**
P90	1-2	3	20	1	100% (dikenali)
AK47	1-2	1	20	1	100 % (dikenali)
MP5	1-2	1	20	1	100% (dikenali)

Pada tabel 1. Terdapat keterangan waktu data training setiap sampel sebanyak 20 sampel berdurasi 1-2 detik, waktu data yang digunakan sebagai testing berdurasi 3 detik untuk speakers P90, 1 detik untuk speaker AK47 dan MP5, hasil pengujian menyatakan bahwa data yang

ISBN: 978-623-7036-34-0

digunakan untuk testing berhasil dikenali semuanya dengan hasil akurasi sebesar 100% dan kebenaran sebesar 100%.

4. Kesimpulan

Penggunaan HTK telah berhasil dilakukan dalam mengimplementasikan algoritma HMM pada aplikasi *speaker recognition*. HTK telah berisi fasilitas yang dapat digunakan dalam membangun aplikasi *speaker recognition* dari tahap training hingga pengujian. Hasil akhir yang diperoleh berupa hasil dari algoritma Viterbi yang selanjutnya dapat digunakan untuk dibandingkan dengan label yang sudah ada. Ujicoba telah dilakukan dalam mentraining dan menguji sistem dalam mengenali 3 jenis suara tembakan. Hasil eksperimen menunjukkan bahwa semua suara yang diujikan dapat dideteksi atau dikenali oleh sistem yang telah dibuat. Namun berdasarkan experimen, hasil akhir dari HTK tidak menampilkan parameter untuk kecocokan berdasarkan *accuration* dan *correction* yang muncul ketika HResults dijalankan, sehingga hasilnya hanyalah berupa Boolean (100% jika dikenali oleh sistem, 0% jika tidak dikenali oleh sistem). Pada penelitian selanjutnya, diharapkan dapat dibuat sebuah GUI atau program lainnya sehingga dapat memaksimalkan penggunaan HTK Toolkit ini.

Ucapan Terimakasih

Penulis megucapkan rasa syukur kepada Alloh SWT, kedua orang tua, serta Universitas Komputer Indonesia khususnya jurusan teknik elektro dan Pusat Penelitian Tenaga Listrik dan Mekatronik - LIPI karena dukungannya selama penelitian ini.

Daftar Pustaka

[1] G. Senthil Raja and S. Dandapat, “Speaker recognition under stressed condition,” *Int. J. Speech Technol.*, vol. 13, no. 3, pp. 141–161, Sep. 2010.

[2] Y. Lan, Z. Hu, Y. C. Soh, and G.-B. Huang, “An extreme learning machine approach for speaker recognition,” *Neural Comput. Appl.*, vol. 22, no. 3–4, pp. 417–425, Mar. 2013.

[3] R. Jourani, K. Daoudi, R. André-Obrecht, and D. Aboutajdine, “Discriminative speaker recognition using large margin GMM,” *Neural Comput. Appl.*, vol. 22, no. 7–8, pp. 1329–1336, Jun. 2013.

[4] O. A. Adetunmbi, O. O. Obe, and J. N. Iyanda, “Development of Standard Yorùbá speech-to-text system using HTK,” *Int. J. Speech Technol.*, vol. 19, no. 4, pp. 929–944, Dec. 2016.

[5] M. O. M. Khelifa, Y. M. Elhadj, Y. Abdellah, and M. Belkasmi, “Constructing accurate and robust HMM/GMM models for an Arabic speech recognition system,” Int. J. Speech Technol., vol. 20, no. 4, pp. 937–949, Dec. 2017

[6] “The Hidden Markov Model Toolkit (HTK)”. Online: http://htk.eng.cam.ac.uk/

[7] Steve Y, Gunnar E. HTK Book, Cambridge University Engineering Department, 2002

[8] “Download suara”. Online: http://sounddogs.com