P-ISSN : 2622-1594
E-ISSN : 2685-449X

Pengenalan Lirik Lagu Otomatis Pada Video Lagu Indonesia Menggunakan Hidden Markov Model Yang Dilengkapi Music Removal

Luhfita Tirta Swarga[a], Joan Santoso[b], Endang Setyati[c*]
[a]*Departemen Teknologi Informasi, Institut Sains dan Teknologi Terpadu Surabaya*
[b]*Departemen Teknologi Informasi, Institut Sains dan Teknologi Terpadu Surabaya*
[c]*Departemen Teknologi Informasi, Institut Sains dan Teknologi Terpadu Surabaya*
E-mail: luhfitatirta@gmail.com, joan@stts.edu, endang@stts.edu

***Abstrak*—Video sangat penting untuk membuat informasi berupa suara dalam video agar dapat dipahami oleh semua kalangan masyarakat, dan orang-orang yang memiliki masalah pendengaran yaitu dengan cara paling alami terletak pada penggunaan subtitle. Oleh karena itu, peneliti membuat pengenalan lirik lagu otomatis pada video lagu Indonesia menggunakan Hidden Markov Model yang dilengkapi music removal. Dalam pengenalan suara lebih akurat dilakukan dengan menggunakan model HMM yang dilengkapi oleh MFCC (kata yang cocok 81% dan WER 19%) dibandingkan dengan model LDA + MFCC (kata yang cocok 71% dan WER 29%) dan DWT + MFCC (kata yang cocok 61% dan WER 39%). Jumlah kata dan sample suara pada library Bahasa Indonesia yang digunakan cukup sangat mempengaruhi MFCC dan CMU Sphinx-4, Nada pada inputan lagu yang akan diproses CMU Sphinx-4 juga sangat berpengaruh pada tingkat keberhasilan, dikarenakan CMU Sphinx-4 sangat sensitif dengan nada yang terlalu tinggi dan noise yang ada pada inputan lagu tersebut sehingga peneliti menambahkan fitur ekstraksi pada suara yaitu menggunakan MFCC. Dalam hal ini menggunakan dataset kecil terlebih dahulu untuk memastikan metode Hidden Markov Model yang dilengkapi MFCC dan CMU Sphinx-4 dapat berjalan dengan baik, Dari penelitian beberapa peneliti sebelumnya, maka hasil akhir yang diperoleh dengan menggunakan metode HMM yang dilengkapi oleh MFCC dan CMU Sphinx-4 dalam penelitian ini mendapatkan hasil akurasi training 78% dan testing 81% kecocokan kata pada video lagu.**

Kata Kunci*— Video, Subtitle, *Hidden Markov Model

I. PENDAHULUAN

Sistem teknologi sangat penting untuk membuat informasi yang dicakup dalam bentuk suara beserta video untuk dapat dipahami bagi masyarakat umum. Cara yang paling alami terletak pada sistem penggunaan subtitle [1]. *Subtitle* merupakan terjemahan teks dari dialog dalam video yang ditampilkan secara realtime selama pemutaran video dibagian bawah layar, sehingga penggunaan subtitle menjadi sangat penting dalam memahami video.

Naskah Masuk : 31 Mei 2022
Naskah Direvisi : 17 Oktober 2022
Naskah Diterima : 26 Oktober 2022
*Corresponding Author : endang@stts.edu

Subtitle merupakan hal penting bagi semua kalangan untuk memahami informasi didalam video, informasi tersebut yang diterima berupa teks. Subtitle juga bermanfaat bagi orang-orang yang memiliki kondisi tubuh yang kurang sempurna dalam hal kurang pendengaran, kurang mampu dalam hal membaca, kurang mampu dalam masalah keaksaraan dan serta bagi mereka yang sedang belajar membaca. Oleh karena itu *subtitle* merupakan penerjemahan antara teks dalam dialog serta video yang ditampilkan secara realtime bagi video dibagian bawah layar [2]. Yang selanjutnya antara dialog dalam video akan dapat memberikan kemudahan dan sinkronisasi dalam proses sinkronisasi antara suara terhadap teks. Sehingga tujuan dari penelitian ini adalah memanfaatkan metode *Hidden Markov Model*, sejauh mana kemampuan mengidentifikasi sinkronisasi antara suara dan teks pada video lagu. Sehingga penelitian ini dapat mampu untuk memberikan manfaat dalam membantu masyarakat yang tidak (kurang mampu) didalam memahami dialog yang diucapkan.

II. TINJAUAN PUSTAKA

Representasi pada makna kecocokan suara dan teks pada video lagu dapat dilakukan dengan pendekatan tradisional yaitu dengan cara pembuatan subtitle atau teks terjemahan yang dapat mempermudah masyarakat umum tidak (kurang) mampu didalam memahami dialog yang diucapkan pada video. Oleh karena itu peneliti akan membahasanya pada tinjauan pustaka ini sebagai berikut.

A. Video

Video merupakan salah satu multimedia paling populer yang digunakan masyarakat umum [3]. Jadi sangat penting untuk membuat informasi berupa suara dalam video agar dapat dipahami oleh orang-orang yang memiliki masalah pendengaran yaitu dengan cara paling alami terletak pada penggunaan *subtitle*. Namun, pembuatan subtitle manual adalah aktivitas yang membosankan dan membutuhkan partisipasi aktif dari pengguna. Oleh karena itu, studi pembuatan subtitle otomatis hadir sebagai subjek penelitian yang valid.

B. Subtitle

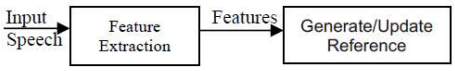
Subtitle adalah terjemahan teks dari dialog dalam video yang ditampilkan secara realtime sealama pemutaran video

DOI : 10.37823/insight.v4i2.225

berlangsung dibagian bawah layar. Terjemahan mungkin dalam bahasa yang sama dengan video atau bahasa lain. Teks ini dimaksudkan untuk membantu orang yang menderita masalah pendengaran, orang tidak terbiasa dengan Bahasa tersebut, atau bahkan orang yang sedang belajar membaca. File subtitle adalah tulang punggung dari subtitle ini. Beberapa format file subtitle dapat berupa *.srt. *.ssa, *.sub dan lain-lain [4].

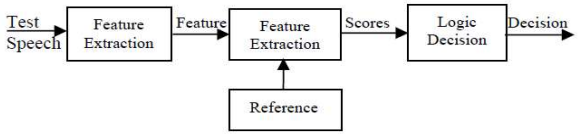
C. *Automatic Speech Recognition*

Speech (Suara) merupakan bentuk percakapan manusia yang paling umum dan sangat penting untuk tujuan komunikasi. *Speech Recognition* yaitu terjemahan kata-kata yang diucapkan kedalam teks dan juga dikenal sebagai *Automatic Speech Recognition* (*ASR*). Pengenalan Suara Komputer, atau *Speech To Text* (*STT*) [5]. *Automatic Speech Recognition* (Pengenalan Suara Otomatis) adalah salah satu teknologi komunikasi yang sangat diperlukan antara manusia serta antara manusia dan mesin yang telah membuat kemajuan luar biaasa dalam 20 tahun terakhir, mulai dari laboratorium hingga pasar. Baru-baru ini, kemajuan dalam Teknik *Automatic Speech Recognition* telah memungkinkan mesin untuk berkomunikasi secara efektif dengan manusia [6]. Dan dengan perkembangan pengenalan suara dapat mendorong munculnya dan kemajuan pengenalan musik. Sama seperti system pengenalan pola lainnya, proses melakukan pengenalan suara terdiri atas dua fase yaitu : fase pelatihan dan fase pengujian. Fase pelatian merupakan proses membiasakan system dengan karakteristik suara dari speaker yang merupakan input dengan mengekstrak fitur dari masing-masing speaker. Blok diagram fase pelatihan Vektor fitur yang mewakili karakteristik suara pembicara diekstraksi dari ucapan pelatihan dan digunakan untuk membuat model referensi sebagaimana diperlihatkan pada gambar 1.



Gambar. 1. Blok diagram Fase Pelatihan

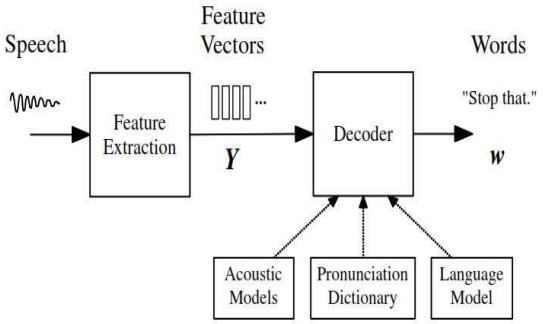
Pada blok diagram fase pengujian, vektor fitur serupa dienktraksi dari ujicoba tes, dan tingkat kecocokannya dengan referensi diperoleh dengan menggunakan beberapa algoritme pencocokan. Dan dimana proses pencocokan fitur dilakukan untuk memutuskan apakah fitur-fitur ini memiliki pola pengeras suara yang diketahui sebelumnya atau tidak yang sebagaimana dapat diperlihatkan pada gambar 2.



Gambar. 2. Blok diagram Fase Pengujian

D. *Hidden Markov Model*

Hidden Markov Model diperkenalkan pada awal 1970-an, menjadi solusi sempurna untuk masalah pengenalan suara otomatis. Sinyal akustik wicara dimodelkan oleh seperangkat unit akustik kecil, yang dapat dianggap sebagai bunyi dasar Bahasa. Secara tradisional, unit yang dipilih adalah fonem, dengan demikian kata tersebut dibentuk dengan cara menggabungkannya. Komponen utama dari pengenalan ucapan kontinu kosakata besar diilustrasikan pada gambar 3.



Gambar. 3. Arsitektur Recogniser Berbasis HMM

Unit dasar bunyi diwakili oleh model akustik adalah telepon. Misalnya, kata "bat" terdiri dari tiga hurf /b/ /ae/ /t/. Sekitar 40 telepon semacam itu diperlukan untuk bahasa Inggris. Untuk apapun yang diberikan w, model akustik yang sesuai disintesis dengan menggabungkan model huruf untuk membuat kata-kata seperti yang didefinisikan oleh kamus pengucapan. Parameter model telepon ini diperkirakan dari data pelatihan yang terdiri dari bentuk gelombang ucapan dan transkripsi ortografisnya. Model bahasa biasanya model N-gram di mana probabilitas setiap kata hanya dikondisikan pada N 1 pendahulu. parameter N -gram diperkirakan dengan menghitung N -tuples dalam korpora teks yang sesuai.

Dekoder beroperasi dengan mencari melalui semua urutan kata yang mungkin menggunakan pemangkasan untuk menghilangkan hipotesis yang tidak mungkin sehingga menjaga pencarian tetap dapat dilakukan. Ketika akhir ujaran tercapai, urutan kata yang paling mungkin adalah keluaran. Seperti disebutkan diatas, setiap kata yang diucapkan w didekomposisi menjadi barisan K_w suara dasar yang disebut kata dasar. Untuk memungkinkan kemungkinan beberapa pengucapan, kemungkinan $p(\boldsymbol{Y}|\boldsymbol{w})$ dapat dihitung melalui beberapa pengucapan.

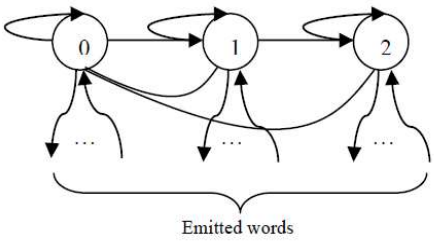
$$p(Y|w) = \frac{\Sigma}{Q}\, p(Y|Q)P(Q|w) \quad (1)$$

Dimana penjumlahan atas semua urutan pengucapan yang valid untuk *w, Q* adalah urutan pengucapan tertentu.

$$P(Q|w) = \frac{L}{I=1}\, P(q^{(w1)}|w1) \quad (2)$$

Alternatifnya, dekoder modern dapat menghasilkan kisi yang berisi representasi ringkas dari hipotesis yang paling mungkin[5]. Sinyal suara dapat disamakan dengan serangkaian unit. Dalam konteks Markov ASR. Unit akustik dimodelkan oleh HMM Gambar. 4 yang biasanya berupa tristat kiri-kanan.

 DOI : 10.37823/insight.v4i2.225



Gambar. 4. Topologi *Hidden Markov Model*

Hidden Markov Model telah menjadi paradigma yang paling banyak digunakan untuk sistem pengenalan suara kontemporer. Didalamnya terdapat transisi dan observasi probabilitas distribusi dari state tersebut merupakan *Speech Durational* dan karakteristik spektral masing-masing. Untuk mendefinisikan *Hidden Markov Model* sepenuhnya [7], berikut elemen yang diperlukan.

- Jumlah status model N
- Jumlah simbol observasi dalam alfabet M
  Jika pengamatannya terus menerus maka M tidak terbatas
- Satu set probabilitas transisi keadaan A = $\{a_{ij}\}$

$$a_{ij} = p\{q_{t+1} = j \backslash q_t = i\}, 1 \leq i, j \leq N \quad (3)$$

  Dimana q_t menunjukkan keadaan saat ini.
- Distribusi probabilitas disetiap negara B = $\{b_j$ (k)$\}$

$$b_j(k) = p\{o_t = v_k \backslash q_t = j\}, 1 \leq j \leq N, 1 \leq k \leq M \quad (4)$$

  Dimana v_k menunjukkan symbol observasi k^{th} dalam alfabet dan o_t vektor parameter saat ini.
- Distribusi keadaan awal π = $\{\pi_i\}$

$$\pi_i = p\{q_1 = i\}, 1 \leq j \leq N \quad (5)$$

- Oleh karena itu peneliti dapat menggunakan notasi kompak λ = (A, B, π)

Peneliti mengacu pada penelitian-penelitian yang sudah ada. Beberapa penelitian sebelumnya yang menggunakan metode *Hidden Markov Model* adalah sebagai berikut.

1. *Automatic Word Recognition Based on Second-Order Hidden Markov Model* yaitu penelitian *word recognition* yang dilakukan oleh Jean Francois Mari et al[8]. Penelitian ini mengusulkan perpanjangan dari algoritma viterbi yang membuat *Second-Order Hidden Markov Model* menjadi efesien secara komputasi. Dataset penelitian berupa basis data *TI-NIST* yang berhubungan dengan *word recognition*. Data set yang digunakan terdiri dari 23 model – satu per digit dan jenis kelamin dan satu untuk latar belakang yang memiliki *noise*. Pada penelitian ini peneliti menggunakan metode HMM yang telah dimodifikasi menjadi HMM1 dan HMM2 yang memiliki hasil akurasi penelitian *WER* sebesar 2.4%.
2. Pada penelitian *Audio-Visual Continuous Speech Recognition Using A Coupled Hidden Markov Model* yang dilakukan oleh Xiaoxing Liu et al[9]. Tujuan dari penelitian ini adalah untuk mencari integrase fitur *audio-visual* yang akan digunakan dalam sistem antara *audio-visual* tersebut yang dapat memungkinkan sinkron secara alami sekaligus memaksakan status sinkronisasi pada batas model, dan metode yang digunakan oleh peneliti yaitu *CHMM* yang dapat digunakan untuk memodelkan setiap kata. Dataset penelitian berupa satu set 1.450 digit urutan pencacahan yang diambil dari 200 pembicara untuk pelatihan dan satu set 700 urutan dari 95 pembicara lainnya yang digunakan untuk *decoding* pada database *XM2VTS*. Pada hasil dari penelitian yang telah dilakukan pada database XM2VTS menunjukkan bahwa sistem peneliti meningkatkan tingkat sistem pengenalan suara audio secara konsisten disemua tingkat SNR yang mencapai hasil akurasi WER lebih dari 55%.
3. Pada penelitian *Automatic Urdu Speech Recognition Using Hidden Markov Model* yang dilakukan oleh Asadullah et al[10] yaitu penelitian yang melakukan pengembangan pendekatan untuk pengenalan suara otomatis dari kata-kata yang terisolasi dalam Bahasa urdu dengan menggunakan metode HMM. Dataset penelitian yang diambil oleh peneliti yaitu dengan melakukan dua eksperimen berdasarkan pengaturan dataset yang berbeda yaitu eksperimen pertama menggunakan dataset sebanyak 100 data, sedangkan eksperimen kedua yaitu menggunakan sebanyak 250 dataset kata. Demikian hasil yang telah dihasilkan dari penelitian ini yaitu dengan akurasi keberhasilan 78.2%.
4. Penelitian yang telah dilakukan oleh Bezoui Mouaz et al *Speech Recognition of Moroccan Dialect Using Hidden Markov Models* bertujuan untuk memverifikasi kemampuan sistem pengenalan suara HMM. Peneliti untuk membedakan vocal pembicara dan mengidentifikasi mereka dengan memberikan kelas tertentu kepada masing-masing. Hal ini dilakukan melalui pembuatan sistem pengenalan suara dan dapat menerapkannya pada pidato dialek maroko. Faktanya, bahwa dialek maroko sangat berbeda dengan Bahasa arab standar modern yang sangat dipengaruhi oleh Bahasa perancis. Oleh sebab itu peneliti mengusulkan untuk menggunakan metode HMM yang dilengkapi fitur *Mel Frequency Cepstral Coefficients* (*MFCC*) untuk menentukan sistem indetifikasi speaker terbaik. Dataset yang digunakan dalam pelatihan dan pengujian penelitian ini untuk setiap dialek adalah kombinasi dari 20 pembicara, 11 pria dan 9 wanita. Peneliti memilih 4 orang yaitu 3 wanita dan 2 pria untuk mengucapkan 4 kata dialek maroko yaitu (م) = (Hai), (ديكاير) = (Apa Kabar), (سابلا) = (Tidak ada yang salah), (ريِ) = (Baik) dan peneliti merekam suara speaker dalam format file (.wav). Dan hasil akurasi keberhasilan yang telah di peroleh dari penelitian ini yaitu sekitar 90%.

Hasil dari penelitian dari peneliti-peneliti diatas maka, peneliti menggunakan metode *Hidden Markov Model* (HMM) sebagai acuan untuk pengenalan lirik lagu otomatis pada video lagu Indonesia menggunakan *Hidden Markov Model* yang dilengkapi *Music Removal*. Sehingga, informasi

 DOI : 10.37823/insight.v4i2.225

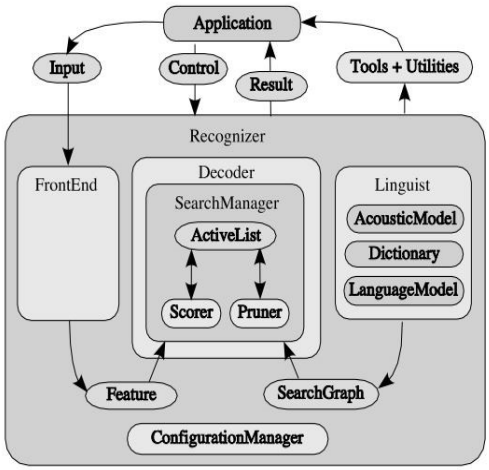
berupa suara beserta teks dalam video dapat dinikmati oleh banyak orang.

E. CMU Sphinx-4

CMU Sphinx-4 merupakan kerangka kerja modular, fleksibel, dan dapat dipasang dengan menggabungkan pola desain dari sistem yang adapun cukup untuk membantu mendorong inovasi baru dalam penelitian *Speech Recognition* sistem dengan *Hidden Markov Model* [11].

1) Kerangka *CMU Sphinx-4*

Kerangka *CMU Sphinx-4* telah dirancang dengan tingkat modularitas dan disibilitas yang tinggi. Arsitektur sistem secara keseluruhan dari setiap elemen berlabel mewakili modul, tanpa memungkinkan peneliti untuk bereksperimen dengan implementasi modul yang berbeda tanpa perlu memodifikasi bagian lain dari sistem. Ada tiga modul kerangka kerja *CMU Sphinx-4* yaitu terdiri dari *FrontEnd*, *Linguist* dan *Decoder*, sedangkan blok pendukung terdiri dari blok *ConfigurationManager* dan *Tools* yang sebagaimana dapat diperlihatkan pada gambar 4.

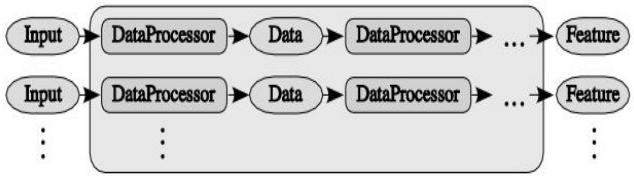


Gambar. 5. Kerangka Kerja *CMU Sphinx-4*

CMU Sphinx-4 ConfigurationManager digunakan untuk mengkonfigurasi dan memuat modul secara dinamis pada waktu berjalan, dan menghasilkan sistem yang fleksibel dan dapat dipasang. Misalnya, *CMU Sphinx-4* biasanya dikonfigurasi dengan FrontEnd yang menghasilkan *Mel-Frequency Cepstral Coefficients* (*MFCCs*). *CMU Sphinx-4* juga menyediakan dukungan untuk utilitas yang mendukung pemrosesan tingkat aplikasi hasil pengenalan.

2) *FrontEnd*

FrontEnd bertujuan untuk membuat parameter sinyal input (contohnya audio) menjadi urutan fitur output seperti sebagaimana diperlihatkan pada gambar 6.



Gambar. 6. *Sphinx-4 FrontEnd*

FrontEnd terdiri dari datu atau lebih rantai parallel dari modul pemrosesan sinyal komunikasi yang dapat diganti disebut *DataProcessors*. Pembuatan sistem secara bersamaan dapat mendekode menggunakan jenis parameter yang berbeda, seperti *MFCC* dan *Perceptual Linear Prediction (PLP)*, dan bahkan bisa jenis parameter yang diturunkan dari sinyal non-suara seperti video [12].

3) *Linguist*

Linguist adalah modul yang dapat dicocokkan dan dapat memungkinkan orang untuk mengetes secara dinamis mengkonfigurasi sistem dengan implementasi *Linguist* yang berbeda. Implementasi *Linguist* yang khas dapat membangun *SearchGraph* dengan menggunakan struktur bahasa seperti yang diwakili oleh *LanguageModel* dan struktur topologi *AcousticModel*. Linguist juga dapat menggunakan *Dictionary* untuk memetakan kata-kata dari model bahasa ke dalam urutan elemen *AcousticModel*. Dengan mengizinkan implementasi yang berbda dari *Linguist* untuk dicocokkan pada waktu berjalan, *CMU Sphinx-4* mengizinkan individu guna memberikan konfigurasi berbeda dan persyaratan pengenalan untuk sistem yang berbeda [11].

4) *Decoder*

CMU Sphinx-4 Decoder merupakan fitur yang digunakan dari *FrontEnd* yang berhubungan dengan *SearchGraph* dari *Linguist* untuk menghasilkan hipotesis hasil dari penelitian. *Decoder* adalah *SearchManager* yang dapat dicocokkan dan kode pendukungnya yang lain dapat menyederhanakan proses decoding untuk aplikasi. *Decoder* hanya memberi *SearchManager* untuk dapat mengenali sekumpulan fitur yang pada setiap langkah proses *SearchManager* membuat hasil objek yang berisi semua jalur yang telah mencapai status non-emisi akhir.

F. FFMPEG

Ffmpeg merupakan aplikasi perangkat lunak *open source* yang terdiri dari sejumlah besar pustaka. Ffmpeg adalah program computer yang dapat merekam dan mengkonversi audio, video digital serta file multimedia lainnya dalam berbagai format. Program ffmpeg telah dirancang untuk pemrosesan file video dan audio berbasis baris perintah yang banyak digunakan untuk transcoding format, pengeditan dasar (pemangkasan dan penggabungan), video scalling, efek pasca produksi video, dan standart compliance (*SMPTE, ITU*). Ffmpeg terdiri dari *libavcodec* yaitu *library* untuk audio/video codec yang digunakan oleh beberapa proyek perangkat lunak komersial dan gratis. Dan *libavformat* (*Lavf*) yaitu library untuk audio/video *container mux* dan *demux library* serta program command line ffmpeg untuk transcoding ke file multimedia lainnya [13]. Ffmpeg merupakan proyek aplikasi yang berasal dari grup video standart *MPEG* yang lalu ditambahkan "*FF*" yang artinya "*fast forward*" oleh *Fabrice Bellard* [14].

III. HMM PADA PENGENALAN TEKS LAGU

Pada dasarnya HMM digunakan untuk kosakata kecil dan kasus yang mempunyai kerumitan terbatas seperti kamus antar bahasa. HMM model kurang cocok apabila digunakan untuk tingkat kerumitan yang tinggi, serta kosakata besar

 DOI : 10.37823/insight.v4i2.225

seperti acara breaking news. Pada bab ini beberapa penambahan yang digunakan untuk meningkatkan kinerja sistem ASR dan memungkinkan digunakan untuk pengenalan teks lagu. Peneliti menambahkan dynamic bayesian networks digunakan untuk mengembangkan HMM dan sistem ASR. Selain itu juga peneliti menggunakan toolkit CMU-Sphinx 4 sehingga bisa digunakan sebagai pengenalan teks lagu.

A. Model Dasar

HMM adalah model generatif dan meskipun model akustik berbasis HMM dikembangkan terutama untuk pengenalan suara, hal ini sangat dipertimbangkan untuk menghasilkan ucapan yang sesuai. Tidak hanya digunakan untuk aplikasi sintesis dimana flexibility dan compact membuat HMM menawarkan banyak manfaat. Tetapi hal ini juga memberikan kesempatan lebih lanjut tentang penggunaannya dalam recognition. Kunci dari HMM ini adalah membuat parameter statis menjadi lebih akurat.

B. Library HMM

HMM menggunakan beberapa library, antara lain library acoustic, dictionary dan language model path. Pada library acoustic menggunakan beberapa sample kata yang berbentuk suara. Dimana satu kata diwakili oleh suara pria dan suara wanita. Rata-rata panjang file audio sekitar 1 detik dan rata-rata besar size filenya 26kb. Sebelum file suara bisa digunakan sebagai library, File suara harus dilakukan convert, yang awalnya mempunyai rate 44100Hz dan ber-ekstensi *.mp4 menjadi *.wav yang mempunyai rate 16000Hz dan menjadi mono (single channel). Beberapa sample atribut pada library bisa dilihat pada Tabel I.

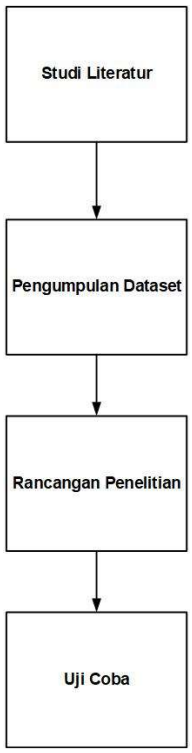
TABEL I
SAMPLE ATRIBUT PADA LIBRARY

Kata	Jenis Kelamin	Kosa Kata
Ada	Wanita	A DA
Ada	Pria	A DA
Adalah	Wanita	A DA LA H
Adalah	Pria	A DA LA H
Jika	Wanita	JI KA
Jika	Pria	JI KA

Pada tabel I bisa dilihat bahwa dalam satu kata diwakili dua gender, yaitu pria dan wanita. Selain itu di dalam library juga ada penjabaran dari kata menjadi kosakata.

IV. METODOLOGI PENELITIAN

Pada metodologi penelitian ini menunjukkan urutan proses terbentuknya penelitian ini hingga aplikasinya, sehingga informasi yang disampaikan pada jurnal ini tetap relevan dengan bidang penelitian yang spesifik. Tahapan alur metodologi penelitian yang akan dilakukan dapat dilihat pada gambar 7 sebagai berikut.



Gambar. 7. *Alur Penelitian*

A. Studi Literatur

Pada tahapan studi literatur yaitu proses yang mencakup pemilihan teori dan kajian berdasarkan penelitian sejenis yang terpublikasi di jurnal bereputasi, dan yang berfokus dalam penelitian bidang *subtitle generation* dan *speech recognition* dengan menggunakan metode *Hidden Markov Model*.

B. Pengumpulan Dataset

Pada tahapan pengumpulan, memiliki beberapa tahap rancangan yang merupakan studi pustaka yang akan dilakukan pada sistem. Diantaranya adalah menentukan kebutuhan data yang akan digunakan, dan data yang digunakan pada penelitian ini adalah data video lagu indonesia.

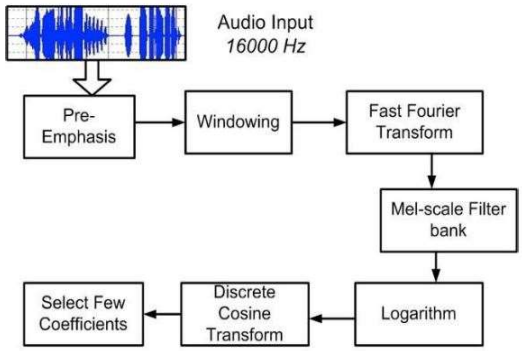
C. Rancangan Penelitian

Pada tahapan ini menjelaskan proses menentukan alur sistem dan pendekatan terbaik yang digunakan untuk mencapai tujuan penelitian sebagai berikut.

a. Konsep Model Dasar

Model ini dibangun berdasarkan model pada penelitian sebelumnya oleh Asadullah et al, "*Automatic Urdu Speech Recognition* berbasis *Hidden Markov Model* [10]. Penelitian tersebut menggunakan model pengembangan pendekatan sphinx. Peneliti menggunakan *Mel Frequency Cepstral Coefficients* (*MFCC*) dalam ekstraksi fitur sehingga dapat membantu membangun *HMM* untuk melakukan pengenalan suara. Hal ini tampak jelas terlihat pada gambar 7. yang menjelaskan proses penelitian dari mulai *input* audio, *pre-emphasis*, *windowing*, *fast fourier transform*, *mel-scale filter bank*, *logarithm*, *discrete cosine transform*, *select few coefficients* sampai *output* dengan menghitung *MFCC* menggunakan *Sphinxtrain* dari *Sphinx*.

 DOI : 10.37823/insight.v4i2.225



Gambar. 8. *Model Dasar dari Penelitian Asadullah*

Tahap pertama yaitu pengambilan *Audio*. Setiap audio yang didapat dari *Urdu Speech Recognition Dataset* dicari fiturnya untuk mewakili masing-masing perkata/kata dari audio tersebut. Fitur-fitur ini nantinya akan melalui beberapa proses sebagai berikut. Fitur pertama yaitu proses *Pre-Emphasis* merupakan proses yang digunakan untuk mengurangi noise pada suara masukan, sehingga dapat meningkatkan kualitas signal dan dapat menyeimbangkan spektrum dari signal voice sound yang terekam oleh microphone. Fitur selanjutnya yaitu proses *Windowing* setiap *frame* untuk meminimalisir diskontinuitas sinyal pada permulaan dan akhir setiap *frame*. Apabila *windowing* didefinisikan $w(n), 0 \leq n \leq N-1$, dengan N merupakan jumlah sampel dalam setiap *frame*, maka hasil dari proses ini dapat ditunjukkan pada rumus sebagai berikut.

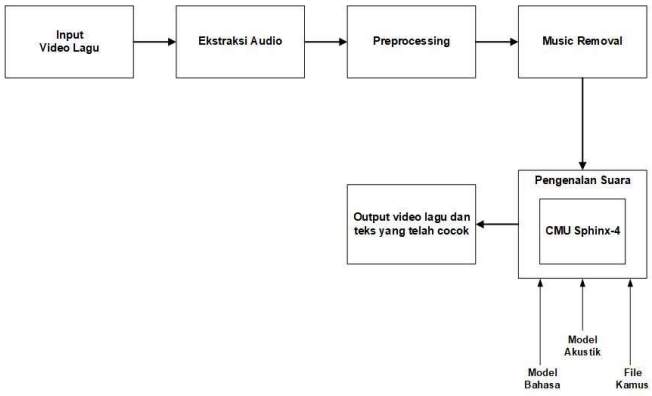
$$y(n) = x(n)w(n), 0 \leq n \leq N-1 \quad (6)$$

Setelah dilakukan pengambilan fitur *windowing*, dataset akan melalui tahap ketiga dari *pre-processing*. Tahap selanjutnya yaitu *Fast Fourier Transform* yang mengkonversi setiap *frame* yang berisi N sample dari waktu ke frekuensi. Setelah dilakukan pengambilan fitur *Fast Fourier Transform*, selanjutnya *dataset* akan melalui tahap ketiga dari *pre-processing*.

Pada tahap *Mel-scale Filter Bank* salah satu bentuk filter linear frekuensi scale pada frekuensi dibawah 1000 Hz, dan merupakan *logarithmic scale* pada frekuensi diatas 1000 Hz yang dilakukan untuk mengetahui ukuran energi dari frequency band tertentu dalam signal suara yang dapat diterapkan dalam domain frekuensi, selanjutnya *dataset* akan melalui ke tahap selanjutnya yaitu *Discrete Cosine Transform* sangatlah berguna dalam menentukan end point dari suatu suku kata maupun kata, hal ini disebabkan karena pada end point dari suku kata maupun kata mempunyai energi yang lebih rendah daripada point-point lainnya. Sedangkan untuk tahapan yang terakhir yaitu tahap *Select Few Coefficients* pada sistem mfcc berfungsi untuk menyaring koefisien pada sinyal input. MFCC menggunakan antara 8-13 tingkat koefisien.

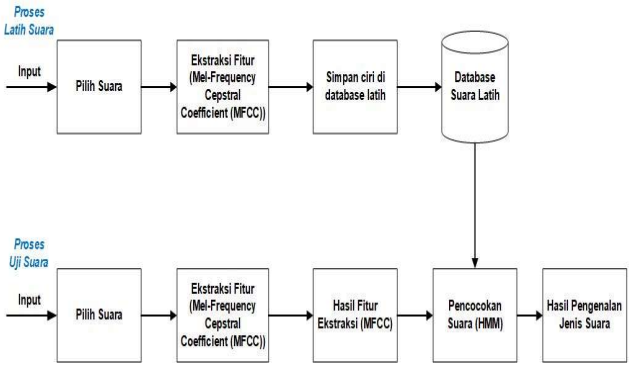
b. Pengembangan Model

Pada penelitian ini berfokus pada pengembangan pemilihan pengenalan lirik lagu menggunakan *HMM* yang dilengkapi *Music Removal*. Aspek penggunaan *HMM* dalam penelitian dianggap kurang memadai untuk permasalahan pengenalan lirik lagu. Oleh karena itu, peneliti menggunakan HMM dengan dibantu oleh *MFCC* dan *CMU Sphinx-4* dalam proses pengenalan suara dan pembuatan kata. Terdapat uji coba model dengan pelatihan sendiri. Untuk aspek fitur preprocessing, peneliti mencoba menambahkan metode *MFCC*. Keputusan ini berdasarkan dari hasil penelitian-penelitian sebelumnya yang menunjukkan keunggulan akurasi yang baik dalam menggunakan *MFCC* dalam proses pengelolaan pengenalan suara.



Gambar. 9. *Arsitektur Model Penelitian*

Yang terlihat pada gambar 8. Bahwa peneliti menggunakan *Ekstraksi audio* dalam tahapan penelitian ini untuk proses pemisahan antara audio dari video dan lalu disimpan ke dalam file berbentuk *.wav. Selanjutnya yaitu *Preprocessing*, disini menggunakan ekstraksi fitur *Mel-Frequency Cepstral Coefficients* (*MFCC*) untuk membantu membangun HMM untuk melakukan pengenalan suara. Setelah dilakukan tahap *Preprocessing*, selanjutnya yaitu tahap *Music Removal* yaitu akan terjadi pemisahan antara suara penyanyi dan instrument musik sehingga hanya tinggal suara penyanyinya saja. Tahap selanjutnya yaitu Pengenalan Suara yang merupakan proses konversi dari hasil sinyal suara menjadi file teks yang berbentuk *.srt dalam Bahasa Indonesia. Pada tahapan ini peneliti memanfaatkan *CMU Sphinx-4* pada *HMM* sebagai library untuk membangun model penelitian ini. Tahapan terakhir setelah tahapan pengenalan suara yaitu *Output* video lagu dan teks yang telah cocok yang dapat langsung diputar di aplikasi *media player*.



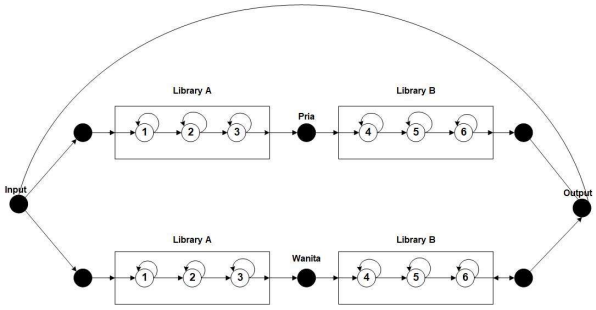
Gambar. 10. *Arsitektur Preprocessing*

Terlihat dalam ilustrasi arsitektur preprocessing pada gambar 9 bahwa terdapat dua proses yang akan digunakan yaitu proses pertama adalah proses latih suara. Langkah pertama memiliki *input* suara latih yang didapatkan dari

 DOI : 10.37823/insight.v4i2.225

proses ekstraksi audio video lagu, suara tersebut memiliki bentuk file *.wav dengan format mono dan frekuensi 16000Hz, lalu dari suara tersebut didapatkan nilai berupa *array* latih suara yang akan dilakukan pada tahap ekstraksi fitur dengan menggunakan *MFCC*. Langkah kedua proses *ekstraksi fitur* yang dilakukan dengan menggunakan *MFCC* untuk mendapatkan beberapa parameter yang dapat dianalisis berupa vektor fitur yang dapat dilihat pada gambar 4.4. Selanjutnya yaitu hasil dari proses ekstraksi didapatkan suatu vektor fitur yang akan disimpan dalam *database* yang telah ditentukan. Setelah dataset latih suara disimpan ke dalam database, selanjutnya *database* dari latih suara akan digunakan sebagai bahan untuk pencocokan data latih dan data uji dalam proses pengenalan suara.

Proses kedua adalah proses uji suara, langkah pertama memiliki *input* suara latih yang didapatkan dari proses ekstraksi audio video lagu, suara tersebut memiliki bentuk file *.wav dengan format mono dan frekuensi 16000Hz. Langkah kedua proses *ekstraksi fitur* data uji yang digunakan *MFCC* sama seperti pada proses ekstraksi fitur latih suara. Selanjutnya vektor fitur dari input suara dicocokan dengan *database* sebelumnya yang telah tersimpan pada proses latih suara. Pencocokan ini digunakan untuk menguji kecocokan antara proses latih suara dan uji suara sehingga didapatkan proses pengenalan suara nilai minimum dari masukan pengguna terhadap aplikasi yang akan dibangun.



Gambar. 11. *Arsitektur Model SearchGraph HMM*

Dapat terlihat pada gambar 10. menunjukkan pada model SearchGraph saat terdapat inputan, maka inputan tersebut akan dilakukan proses pengecekan dari *library* A dan *library* B. dimana setiap library terdapat pengecekan yang diulang sebanyak tiga kali. Serta dalam proses pengecekan sisipkan pengecekan suara pria atau wanita. Karena di dalam *library HMM* terdapat suara pria dan suara wanita dalam satu kata. Apabila inputan sudah cocok pada suatu kata, maka hasilnya akan keluar.

c. Environtment dan Library

Model dari penelitian ini dibuat di *Visual Basic* dengan Bahasa pemrograman Java dan *vb.net. *Visual basic* merupakan sebuah platform aplikasi Open Source yang dapat diunduh secara gratis melalui website *https://www.microsoft.com*. *Visual basic* dapat memberikan kemudahan kepada penggunanya dengan mudah mengakses menggunakan Laptop HP dengan AMD Ryzen 5 3500U with Radeon Vega Mobile Gfx 2.10 Ghz, LCD 14 Inch, RAM 16GB, SSD 256GB, Harddisk 1TB. Library yang digunakan dalam program penelitian ini adalah *Library Indonesian Language* yang didapat dari proses generate menggunakan Sphinxtrain**.** Library ini berfungsi untuk menerjemahkan dari suara lagu menjadi teks subtitle. Pada *Library Indonesian Language* ini berisi kurang lebih 127.000 kosakata.

d. Persiapan Data

Pada persiapan data ini membahas tentang detail program saat *preprocessing* dan pengambilan fitur suara dari data *video lagu* yang telah terkumpul. Proses import *library* juga akan dijabarkan dalam subbab ini. Kemudian, dalam subbab ini juga akan membahas tentang fitur yang akan dicari, yaitu: kata yang tidak cocok, kata yang cocok, dan jumlah kata, sehingga agar mendapatkan hasil kecocokan antara suara dengan teks pada video lagu.

e. Preprocessing

Pada *Preprocessing* ini dilakukan untuk merapikan data pengenalan suara dari sisi pemilihan suara yang akan dilatih dan diuji, persiapan fitur dan pencocokan suara yang akan digunakan, sehingga mendapatkan hasil pengenalan suara yang diinginkan yang terdiri dari :.

1) Cut Video
2) Convert to wav
3) Remove Music

V. HASIL UJI COBA

Pada hasil uji coba yang akan dilakukan, yaitu : 1) jenis kata, 2) jenis kalimat, 3) pembuatan subtitle pada video lagu. Inti dari tiga uji coba tersebut adalah sebagai perbandingan berapakah jenis kecocokan kata, pelatihan kecocokan kata dan kombinasi dari penggunaan fitur yang akan digunakan.

A. Uji Coba Jenis Kata

Pada uji coba jenis kata yaitu peneliti membandingkan performa antara *HMM* yang menggunakan ekstraksi fitur dari *MFCC*. Penentuan *HMM* yang dilengkapi dengan *MFCC* sebagai ekstraksi fitur pada pengenalan suara menjadi perbandingan yang mengacu pada penelitian sebelumnya yang menggunakan *HMM* tersebut untuk mengetahui *Word Error Rate* (*WER*) dengan menggunakan rumus sebagai berikut.

$$WER = \frac{Kata\ yang\ tidak\ cocok\ per\ video}{Total\ kata\ per\ video} X\ 100 \qquad (7)$$

Pada kosa kata yang diucapkan pembicara melalui microphone internal yang ada pada laptop. Dari 100 kata yang diuji coba, terdapat beberapa hasil yang berbeda terdapat pada metode masing-masing penelitian yang terlihat pada tabel II sebagai berikut.

 DOI : 10.37823/insight.v4i2.225

TABEL II
HASIL UJI COBA PERBANDINGAN MENGGUNAKAN 100 KATA

Pendekatan	F1 Kata Cocok	Word Error Rate (WER)
HMM + MFCC	81%	19%
HMM + MFCC	79%	21%
LDA + MFCC	71%	29%
DWT + MFCC	61%	39%
Rata - rata	73%	27%

Dari uji coba menggunakan 100 kata, bahwa ada perbedaan akurasi sebesar 10% antara rata-rata *F1 score* model dengan *LDA* (71%) terhadap model dengan *DWT* (61%). Sedangkan penggunaan fitur *MFCC* paling akurat dengan menggunakan *MFCC* pada *HMM* pengenalan suara otomatis Bahasa Urdu dengan mencapai akurasi 79% dan penggunaan *HMM* yang dilengkapi fitur *MFCC* dan *CMU Sphinx-4* pada pengenalan lirik lagu otomatis pada video lagu Bahasa Indonesia dengan hasil akurasi mencapai 81% yang dapat dilihat dari *F1 score HMM* yang dilengkapi *MFCC* mempunyai hasil yang paling baik. Hal ini dikarenakan cara prosesnya yang lebih baik daripada menggunakan *LDA* dan *DWT* dan *library* yang digunakan *HMM* lebih mudah teridintifikasi. Kelebihan dari *F1 score HMM* yang dilengkapi *MFCC* dapat disebabkan oleh kemampuan metode tersebut dalam memproses *Vocabulary words* melalui aspek yang didapat dari proses *MFCC* dan *CMU Sphinx-4* yang telah dilakukan pada *HMM*.

B. Uji Coba Kalimat

Pada uji coba kalimat peneliti membandingkan hasil nilai *WER* dan nilai akurasi dari uji coba suara pengucapan sebuah kalimat pada suara berita. Sama halnya dengan pengujian kata, nilai *WER* dari metode *HMM* yang menggunakan *MFCC* mendapatkan nilai yang lebih baik daripada yang lain yang dapat dilihat pada tabel III sebagai berikut.

TABEL III
HASIL UJI COBA PERBANDINGAN KALIMAT

Pendekatan	F1 Kata Cocok	Word Error Rate (WER)
Pelatihan Sendiri		
HMM + MFCC	79%	21%
HMM + MFCC	76%	24%
LDA + MFCC	70%	30%
DWT + MFCC	60%	40%
Rata - rata	71.25%	28.75%

Walaupun hasil metode *HMM* tidak mempunyai nilai yang sempurna, yaitu mendapatkan nilai akurasi sebanyak 79%. Hal ini dikarenakan pengaruh dari jumlah kata dan ucapan pada suatu kata pada library yang telah dibuat oleh peneliti.

C. Uji Coba Video Lagu Menggunakan HMM

Pada uji coba ini peneliti menggunakan *HMM* dengan *MFCC* untuk membuat subtitle secara otomatis pada video lagu. Adapun *preprocessing* yang harus dilakukan untuk bisa mendapatkan hasil yang maksimal yaitu meliputi pemotongan video lagu menjadi durasi 30 detik, yang bertujuan untuk mempercepat proses pembuatan subtitle, *convert* dari video ke audio dengan format *.wav. Serta menghilangkan background music, sehingga dapat terdengar jelas suara dari penyanyi.

TABEL IV
HASIL UJI COBA LAGU MENGGUNAKAN HMM

Lagu	Waktu ke-	F1 Kata Cocok	Word Error Rate (WER)
Pelatihan Sendiri			
Afgan – Bawalah Cintaku	30 detik	77.8%	22.2%
Afgan – Bawalah Cintaku	1 menit	77.8%	22.2%
Afgan – Bawalah Cintaku	1 menit 30 detik	75%	25%
Sammy Simorangkir – Dia	1 menit	71%	29%
Sammy Simorangkir – Dia	1 menit 30 detik	85%	15%
Sammy Simorangkir – Dia	2 menit	78%	22%
Syahrini – Kau Yang Memilih Aku	2 menit	74%	26%
Syahrini – Kau Yang Memilih Aku	2 menit 30 detik	84%	16%
The Overtunes – Sayap Pelindungmu	1 menit	84%	16%
The Overtunes – Sayap Pelindungmu	2 menit	87.5%	22.5%
Rata - rata		79%	21%

Sebagai salah satu contoh yang dapat dilihat pada tabel IV pengujian teks kata pada video lagu Sammy Simorangkir yang berjudul Dia pada waktu ke 1 menit, "*Sungguh semua tanya yang terindah*", memiliki kesalah pendeteksian teks pada kata "*semua*" yang seharusnya kata tersebut yang keluar adalah "*sebuah*". Sehingga hasil dari kata yang cocok 71% dan hasil *WER* sebanyak 29%.

Dari hasil uji coba perbandingan dengan penelitian-penelitian yang sebelumnya diatas, dalam memperoleh hasil akhir uji coba dengan menggunakan metode *HMM* yang dilengkapi oleh *MFCC* dan *CMU Sphinx-4* dalam penelitian ini dibagi menjadi 2 hasil uji coba yaitu uji coba data *training* dan uji coba data *testing*. Yang dimana untuk masing-masing hasil uji coba tersebut rata-rata mendapatkan 78% dan 81% kecocokan kata.

VI. KESIMPULAN

Dari penelitian yang telah dilakukan oleh peneliti pada proses pengerjaan penelitian ini, terdapat beberapa kesimpulan yang telah diambil oleh peneliti sebagai berikut :

1. Dalam menghitung hasil dari *F1 score* kata yang cocok dan *Word Error Rate* (*WER*) pada pengenalan suara lebih akurat dilakukan dengan menggunakan model *HMM* yang dilengkapi oleh *MFCC* (kata yang cocok 81% dan *WER* 19%) dibandingkan dengan model *LDA* + *MFCC* (kata yang cocok 71% dan *WER* 29%) dan *DWT* + *MFCC* (kata yang cocok 61% dan *WER* 39%). Karena dapat memberikan hasil yang lebih maksimal dan lebih baik lagi.
2. Jumlah kata dan sample suara pada library Bahasa Indonesia yang digunakan cukup sangat

 DOI : 10.37823/insight.v4i2.225

mempengaruhi *MFCC* dan *CMU Sphinx-4* pada metode *HMM* dalam meningkatkan perfoma dari inputan suara video lagu menjadi sebuah teks. Karena suara pada video lagu mempunyai nada yang berbeda-beda, sehingga satu kata mewakili beberapa sample suara.

3. Nada pada inputan lagu yang akan diproses *CMU Sphinx-4* juga sangat berpengaruh pada tingkat keberhasilan, dikarenakan *CMU Sphinx-4* sangat sensitif dengan nada yang terlalu tinggi dan noise yang ada pada inputan lagu tersebut sehingga peneliti menambahkan fitur ekstraksi pada suara yaitu menggunakan *MFCC*.
4. Sehingga dalam memperoleh hasil akhir dengan menggunakan metode *HMM* yang dilengkapi oleh *MFCC* dan *CMU Sphinx-4* dalam penelitian ini dibagi menjadi 2 hasil uji coba yaitu uji coba data *training* dan uji coba data *testing*. Yang dimana untuk masing-masing hasil uji coba tersebut rata-rata mendapatkan 78% dan 81% kecocokan kata.

Setelah mendapatkan beberapa kesimpulan dari penelitian yang telah dilakukan oleh peneliti terdapat beberapa saran yang mungkin penelitian ini dapat digunakan di kemudian hari. Beberapa saran yang diberikan yaitu sebagai berikut :

1. Memperbanyak kata atau sampel suara sebagai library Bahasa Indonesia. Terutama dibagian sampel suara, baik menggunakan nada maupun yang tanpa nada. Sehingga hasil kecocokan suara dan teks dapat lebih baik.
2. Pada uji coba awal diharapkan menggunakan dataset kecil terlebih dahulu untuk memastikan metode *Hidden Markov Model* yang dilengkapi *MFCC* dan *CMU Sphinx-4* dapat berjalan dengan baik, dan jangan terlalu optimis terlebih dahulu dengan menggunakan data yang sangat banyak pada uji coba awal.
3. Apabila dataset yang digunakan pada *training* dan *testing* berupa video, harus dilakukan *cutting* terlebih dahulu pada video tersebut. Dan disarankan video tersebut harus menggunakan video yang berkualitas HD yaitu 720pixels. Dikarenakan hasil yang dihasilkan akan lebih baik daripada menggunakan video yang berkualitas di bawah HD.

DAFTAR PUSTAKA

[1] K. Mishra, P. Bhagat, and A. Kazi, “Automatic Subtitle Generation for Sound in Videos,” in *International Journal of Engineering and Technology (IRJET)*, 2016, vol. 3, no. 2, pp. 915–918.

[2] A. Jakhotiya, K. Kulkarni, C. Inamdar, B. Mahajan, and A. Londhe, “Automatic Subtitle Generation for English Language Videos,” in *International Journal of Computer Science and Engineering*, 2015, vol. 2, no. 10, pp. 5–7, doi: 10.14445/23488387/ijcse-v2i10p102.

[3] B. Mouaz, B. H. Abderrahim, and E. Abdelmajid, “Speech recognition of Moroccan dialect using hidden Markov models,” *Procedia Comput. Sci.*, vol. 151, no. 2018, pp. 985–991, 2019, doi: 10.1016/j.procs.2019.04.138.

[4] A. Nilakhe and S. Shelke, “A design for wireless music control system using speech recognition,” in *Conference on Advances in Signal Processing, CASP 2016*, 2016, pp. 337–339, doi: 10.1109/CASP.2016.7746191.

[5] R. Sridhar, S. Aravind, H. Muneerulhudhakalvathi, and M. Sibi Senthur, “A hybrid approach for Discourse Segment Detection in the automatic subtitle generation of computer science lecture videos,” *Proc. 2014 Int. Conf. Adv. Comput. Commun. Informatics, ICACCI 2014*, pp. 284–287, 2014, doi: 10.1109/ICACCI.2014.6968422.

[6] Y. C. Mu, J. S. Hwa, and S. K. Hyung, “Speech/music discrimination for robust speech recognition in robots,” in *Proceedings - IEEE International Workshop on Robot and Human Interactive Communication*, 2007, vol. 200, pp. 118–121, doi: 10.1109/ROMAN.2007.4415064.

[7] L. M. Lee, “Duration high-order hidden Markov models and training algorithms for speech recognition,” *J. Inf. Sci. Eng.*, vol. 31, no. 3, pp. 799–820, 2015.

[8] J. F. Mari, J. P. Haton, and A. Kriouile, “Automatic word recognition based on second-order hidden markov models,” *IEEE Trans. Speech Audio Process.*, vol. 5, no. 1, pp. 22–25, 1997, doi: 10.1109/89.554265.

[9] X. Liu, Y. Zhao, X. Pi, L. Liang, and A. V. Nefian, “Audio-visual continuous speech recognition using a coupled hidden Markov model,” in *7th International Conference on Spoken Language Processing, ICSLP 2002*, 2002, pp. 213–216.

[10] A. Shaukat, H. Ali, and U. Akram, “Automatic Urdu Speech Recognition Using Hidden Markov Model,” in *2016 XXI Symposium on Signal Processing, Images and Artificial Vision (STSIVA)*, 2016, pp. 135–139.

[11] W. Walker *et al.*, “Sphinx-4: A Flexible Open Source Framework for Speech Recognition,” *Language (Baltim).*, pp. 1–9, 2004.

[12] M. Mohri, “Finite-State Transducers in Language and Speech Processing,” 1997.

[13] FFmpeg, “FFmpeg.” http://www.ffmpeg.org/ (accessed May 05, 2021).

[14] F Bellard, “FFmpeg naming and logo,” 2006. http://www.ffmpeg.org/about.html (accessed May 05, 2021).

 DOI : 10.37823/insight.v4i2.225