DESAIN DAN IMPLEMENTASI *AUTOMATIC VIDEO CAPTIONING* DENGAN *SPEECH RECOGNITION* MENGGUNAKAN *HIDDEN MARKOV MODEL*

Rama Dimasatria[1], Agus Virgono[2], R. Rumani M.[3]

[1,2,3]Fakultas Teknik Elektro, Universitas Telkom
[1]ramadimasatria@gmail.com, [2]avirgono@telkomuniversity.ac.id, [3]rumani@telkomuniversity.ac.id

Abstrak

Seiring perkembangan teknologi informasi, proses belajar-mengajar semakin banyak menggunakan media alternatif. Salah satu media pembelajaran alternatif yang digunakan adalah video. Untuk mempermudah pemahaman, biasanya video pembelajaran dilengkapi dengan *caption* atau teks keterangan tentang apa yang dibicarakan oleh pembicara. Akan lebih menghemat waktu dan energi apabila *caption* dihasilkan secara otomatis berdasarkan apa yang diucapkan pembicara. Oleh karena itu pada penelitian kali ini akan dibangun sistem *Automatic Video Captioning* menggunakan teknologi *Speech Recognition*. Sinyal suara dari video masukan diekstrak dan diproses dengan sistem *speech recognition* untuk menghasilkan teks yang sesuai. Pada penelitian ini sistem *speech recognition* dibangun dengan *Linear Predictive Coding* untuk ekstraksi ciri dan *Hidden Markov Model* untuk pencocokan ciri. Teks yang dihasilkan dari sistem *speech recognition* tersebut kemudian digunakan sebagai *caption* dari video masukan. Pengujian sistem dilakukan dengan mengubah-ubah jumlah data latih dan parameter HMM yaitu jumlah *state* dan jumlah *cluster* untuk mencari parameter dengan akurasi paling optimal. Dari hasil pengujian, didapatkan akurasi tertinggi sebesar 75,50% pada jumlah *state* 6, jumlah *cluster* 256, dan data latih sebesar 90 untuk setiap suku kata dalam *database*.

Kata Kunci: ***automatic video captioning, speech recognition, Linear Predictive Coding* (LPC), *Hidden Markov Model* (HMM)**

Abstract

As information technology growing further, educational system is more likely using alternative media. One of the alternative educational media that has been widely used is video. In order to get better understanding, usually the educational video is included with caption or text that explaining what the speaker says. It will be much efficient in time and energy if the caption is generated automatically based on what speaker says. Therefore, this research will design and implement Automatic Video Captioning system using speech recognition technology. Speech signal from video is extracted and processed with speech recognition system to generate corresponding text. In this research, the speech recognition system is designed with Linear Predictive Coding as feature extraction method and Hidden Markov Model as feature matching method. The generated text from speech recognition system is then used as the caption for video input. The system is tested by changing the number of data training and the HMM parameters (the number of states and clusters) to find the most optimal parameter with highest accuracy. According the test, the highest accuracy is found at 75.50% when the number of states is 6, number of clusters is 256, and the number of data training is 90 for every syllable in database.

***Keywords*: automatic video captioning, speech recognition, Linear Predictive Coding (LPC), Hidden Markov Model (HMM)**

1. Pendahuluan

Di era teknologi informasi yang semakin maju saat ini, proses pembelajaran menjadi lebih fleksibel dan praktis. Para siswa tidak perlu hadir ke kelas untuk mendapatkan materi yang mereka butuhkan, dan hanya cukup mencari materi tersebut di internet. Melalui internet, mereka dapat menemukan media pembelajaran yang variatif seperti teks, gambar, suara, bahkan video. Pada penelitian ini dibangun sistem pembuat *caption* otomatis (*automatic video captioning*) menggunakan teknologi *speech recognition*.

2. *Speech Recognition*

Speech Recognition atau pengenalan ucapan adalah suatu pengembangan teknik dan sistem yang memungkinkan komputer untuk menerima masukan berupa kata yang diucapkan. Hasil dari identifikasi kata yang diucapkan dapat ditampilkan dalam bentuk tulisan.

2.1. Jenis–jenis *Speech Recognition*

Sistem *Speech Recognition* dapat diklasifikasi berdasarkan kata yang diucapkan dan berdasarkan jumlah pembicara yang dapat dikenali [5].

2.1.1. Berdasarkan Kata yang Diucapkan

Berdasarkan kemampuan dalam mengenal kata yang diucapkan, *speech recognition* dibagi dalam 5 jenis, yaitu [5]:

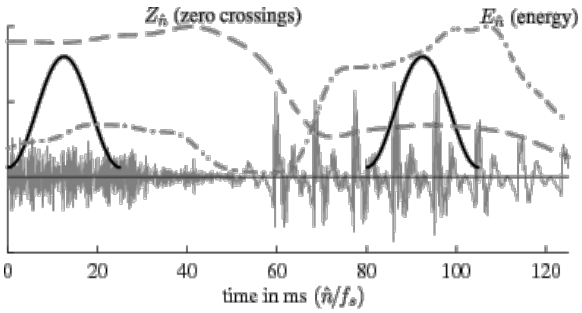
a. Kata-kata yang terisolasi (*Isolated Word*).
b. Kata-kata yang berhubungan dengan proses pengidentifikasian kata yang mirip dengan kata-kata terisolasi.
c. Kata-kata yang berkelanjutan (*Continuous Word*). Pengguna perangkat ini dapat mengucapkan kata-kata secara natural.
d. Kata-kata spontan.
e. Verifikasi atau identifikasi suara.

2.1.2. Berdasarkan Jumlah Pembicara

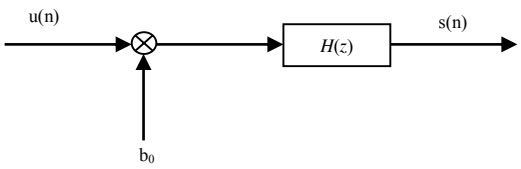
Berdasarkan jumlah pembicara yang dapat dikenali, sistem speech recognition dibagi menjadi dua, yaitu *speaker independent* dan *speaker dependent* [6].

a. *Speaker Independent*
   Sistem *speaker independent* dapat mengenali berbagai macam pembicara.
b. *Speaker Dependent*
   Sistem *speaker dependent* hanya dapat mengenali satu pembicara saja sehingga akurasi dapat lebih tinggi.

Gambar 1. Klasifikasi Sinyal Ucapan [5]



Gambar 2. Contoh Hasil Perhitungan Parameter *Energy* dan *Zero Crossing Rate* [3]



Gambar 3. Model LPC untuk Sinyal Suara

2.2. Sinyal Ucapan

Sinyal Ucapan (*speech signal*) adalah sinyal yang berasal dari ucapan manusia. Sinyal ini bersifat analog sehingga untuk dianalisis lebih lanjut perlu di konversi menjadi sinyal digital menggunakan *analog-to-digital converter* (ADC). Terdapat beberapa tahap dalam proses ADC, yaitu *sampling*, *quantization*, dan *encoding*.

2.2.1. Representasi Sinyal Ucapan

Sinyal ucapan merupakan sinyal yang berubah terhadap waktu dengan kecepatan perubahan yang relatif lambat. Jika diamati pada selang waktu yang pendek (antara 5 sampai 100 milidetik), karakteristiknya praktis bersifat tetap [1].

Ada berbagai cara untuk mengklasifikasikan bagian-bagian atau komponen sinyal ucapan sebagaimana dapat dilihat pada Gambar 1, dengan cara mengklasifikasikannya menjadi tiga keadaan yang berbeda, yaitu:

a. *Silence* (S).
b. *Unvoiced* (U).
c. *Voiced* (V).

2.2.2. Parameter Dasar Sinyal Ucapan

Speech signal memiliki beberapa parameter dasar yang dapat digunakan untuk analisis lebih lanjut. Parameter tersebut antara lain *energy*, *zero crossing rate*, dan *pitch period* [3].

2.2.2.1. *Short Time Energy*

Short Time Energy dapat dihitung dengan membagi sinyal suara kedalam *frame-frame* sepanjang N sampel kemudian dihitung total kuadrat nilai sampel di setiap *frame* (Persamaan 1) [3].

$$Z_n = \sum_{m=-\infty}^{\infty} (x[m]\omega[n-m])^2 = \sum_{m=-\infty}^{\infty} x^2[m]\omega^2[n-m] \quad (1)$$

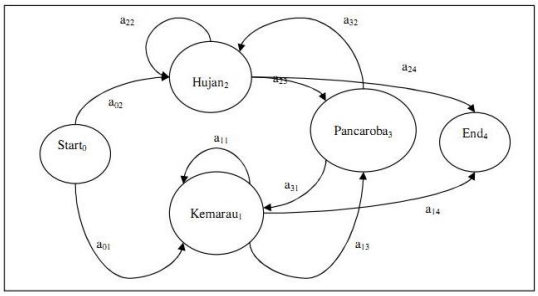
Nilai *short time energy* akan besar apabila dalam *frame* tersebut terdapat sampel-sampel dengan amplitudo yang besar dan sebaliknya (Persamaan 2).

$$Z_n = \sum_{m=-\infty}^{\infty} 0.5\} \,|\, \mathrm{sgn}\{x[m]\} - \mathrm{sgn}\{x[m-1]\} \,|\, \omega[n-m] \quad (2)$$

2.2.2.2. *Short Time Zero Crossing Rate*

Short Time Zero Crossing Rate (Gambar 2) adalah parameter yang menunjukkan banyaknya sampel berganti tanda dalam satu *frame* [3], yaitu:

$$\mathrm{sgn}\{x\} = \begin{cases} 1, x \geq 0 \\ -1, x < 0 \end{cases} \quad (3)$$



Gambar 4. Rantai Markov untuk Cuaca [7]

Parameter ini sering digunakan untuk mendeteksi dan memisahkan bagian *voiced* dan *unvoiced* karena nilai *short time zero crossing rate* akan cenderung besar apabila *frame* tersebut merupakan bagian *unvoiced* dan akan cenderung kecil jika bukan bagian *unvoiced*.

2.3. *Linear Predictive Coding*

Linear Predictive Coding (LPC) adalah salah satu metode dalam ekstraksi ciri yang banyak digunakan dalam sistem pengenalan suara [3].

2.3.1. Model LPC

Tujuan dari LPC adalah untuk mengekstrak paramater–parameter dari sinyal suara. LPC memodelkan sinyal suara berdasarkan fakta bahwa sinyal ucapan bisa direpresentasikan dengan persamaan linear sederhana. LPC memodelkan sinyal suara seperti pada Gambar 3.

Misal diberikan suatu sampel sinyal sepanjang n, $s(n)$ bisa juga dimodelkan sebagai kombinasi linear dari p sampel sebelumnya, sehingga seperti pada persamaan 4 dan 5.

$$s(n) = b_o u(n) + a_1 s(n-1) + a_2 s(n-2) + \ldots + a_p s(n-p) \quad (4)$$

$$s(n) = b0u(n) + \sum_{i}^{i=1p} ais(n-i) \quad (5)$$

Di mana $u(n)$ adalah sinyal yang telah ternormalisasi, b_0 adalah penguat, dan koefisien $a_1, a_2, \ldots a_p$ adalah bobot sampel sinyal sebelumnya. Diasumsikan koefisien–koefisien tersebut selalu konstan. Persamaan tersebut dapat ditulis dalam domain-z menjadi persamaan berikut:

$$S(z) = b_0 U(z) + \sum_{i=1}^{p} a_i S(z) z^{-i} \quad (6)$$

dengan fungsi transfer:

$$H(z) = \frac{S(z)}{U(z)} = \frac{b_0}{1 - \sum_{i=1}^{p} a_i S(z) z^{-i}} \quad (7)$$

Dari fungsi transfer di atas, terlihat bahwa masalah mendasar dari LPC membuat *all-pole model* dari sinyal suara.

2.3.2. Analisis LPC

Masalah mendasar dari analisa peramalan linear adalah menentukan sejumlah koefisien peramalan a_k, langsung dari sinyal suara sehingga sinyal hasil sintesa memiliki spektrum yang sama atau mendekati sama dengan spektrum sinyal aslinya [5].

2.4. *Vector Quantization*

Kuantisasi vektor (*vector quantization*) adalah proses pengelompokan vektor menjadi *cluster-cluster* dimana setiap *cluster* direpresentasikan oleh suatu titik pusat (*centroid*) yang disebut *codeword* [5].

2.5. *Hidden Markov Model*

Sebelum mendefinisikan HMM, perlu dibahas terlebih dulu mengenai *Markov Chain*.

2.5.1. Rantai Markov

Markov Chain merupakan perluasan dari *finite automaton*. *Finite automaton* sendiri adalah kumpulan *state* yang transisi antar *state*-nya dilakukan berdasarkan masukan observasi. Gambar 4 memperlihatkan contoh *Markov Chain* yang menggambarkan kondisi cuaca [3].

2.5.2. Definisi *Hidden Markov Model*

Hidden Markov Model (HMM) adalah sebuah model statistik dari sebuah sistem yang diasumsikan sebuah proses Markov dengan parameter yang tak diketahui (Gambar 4) [7]. HMM didefinisikan sebagai kumpulan lima parameter (N, M, A, B, π). Jika dianggap $\lambda = \{A, B, \pi\}$ maka HMM mempunyai parameter tertentu N dan M.

2.5.3. Parameter Distribusi

HMM mempunyai parameter-parameter distribusi sebagai berikut [3]:

a. Probabilitas Transisi (A)

$$A = \{aij\}, aij = Pr(Xt+1 = qj + Xt = qi), 1 \le j, i \le N \quad (8)$$

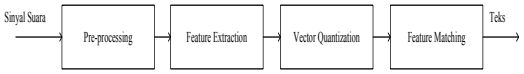
b. Probabilitas observasi (b)

$$B = \{bi\}, bi(k) = Pr(Ot = Vk \mid Xt = qit) \quad (9)$$

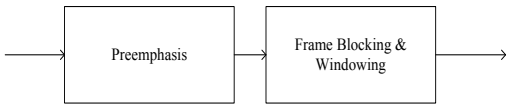
c. Distribusi keadaan awal (π)

$$\pi = \{\pi i\}, \pi i = Pr(Xo = qi) \quad (10)$$

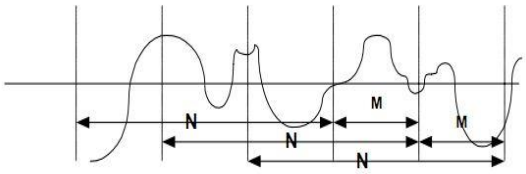
Gambar 5. Skema Umum Sistem



Gambar 6. Diagram Blok *Speech Recognition*



Gambar 7. Diagram blok *Pre-Processing*



Gambar 8. Pembagian Sinyal Menjadi Bingkai-Bingkai [1]

Sedangkan parameter tertentu HMM ada dua yaitu *N* dan *M*.

a. *N* adalah jumlah state atau keadaan model. Dinotasikan himpunan terbatas untuk keadaan yang mungkin adalah $Q = \{q1, \ldots\ldots, qN\}$
b. *M* adalah jumlah dari simbol observasi/keadaan, ukuran huruf diskret. Dinotasikan himpunan terbatas untuk observasi yang mungkin adalah $V = \{V_1, \ldots\ldots, V_M\}$.

3. Perancangan Sistem

Perancangan sistem dilakukan dengan menentukan spesifikasi perancangan sistem dan skema umum sistem.

3.1. Spesifikasi Perancangan Sistem

Dalam perancangan sistem *automatic video captioning*, spesifikasi baik dari segi *hardware*, *software*, maupun *brainware* (pengguna) dibutuhkan.

3.1.1. *Hardware*

Dalam perancangan sistem, *hardware* yang digunakan antara lain adalah:

a. *Personal Computer* (PC)
   Hardware yang digunakan untuk perancangan sistem berupa sebuah *personal computer* dengan spesifikasi sebagai berikut:
   1) Acer Aspireone 532h
   2) Prosessor Intel® Atom™ Processor N450 (1.66Ghz, 512KB Cache)
   3) RAM 1GB
   4) Sistem Operasi Linux Ubuntu 11.04
b. *Microphone*
   Untuk input data latih dan data uji digunakan sebuah *microphone* dari *headset* Philips SHM1900.

3.1.2. *Software*

Dalam perancangan sistem, *Software* yang digunakan antara lain adalah:

a. Code::Block IDE
b. FFMpeg
c. libsndfile C++ library
d. All2Wav Sound Recorder
e. QtOctave
f. Microsoft Visio 2007

3.1.3. *Brainware*

Karena sistem *Automatic Video Captioning* ini bersifat *speaker dependent*, maka pengguna yang dapat menggunakan sistem ini harus diambil data latih terlebih dahulu

3.2. Skema Umum Sistem

Sistem yang diterapkan pada perancangan ini dapat dilihat melalui Gambar 5.

3.2.1. Pembuatan Video

Video disimpan dalam format (*.mpeg) atau (*.avi) dan terdapat kanal suara dengan kualitas *sample rate* 16000 Hz, kanal *Mono*, dan *bitrate* 16 bit.

3.2.2. Pemisahan Video dan Audio

Pada tahap ini kanal suara dari video dipisahkan dan disimpan ke dalam *file* WAV. Proses pemisahan suara dilakukan dengan menggunakan program FFmpeg.

3.2.3. Segmentasi Kata dan Suku Kata

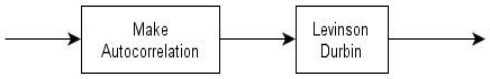
Proses ini terdiri dari dua tahap, yaitu pemisahan kata menggunakan *Word end-point detection algorithm* dan pemisahan suku kata menggunakan *syllable end-point algorithm*.

3.2.4. Proses *Speech Recognition*

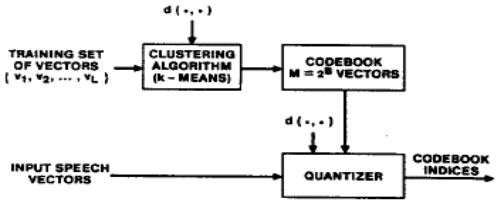
Pada tahap ini dilakukan konversi sinyal audio menjadi teks dalam bahasa Indonesia. Diagram blok proses *speech recognition* dapat dilihat pada Gambar 6. Tahapan dalam proses *speech recognition* yaitu:

3.2.4.1. *Pre-processing*

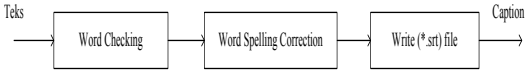
Pre-processing adalah tahap awal sebelum ekstraksi ciri sebagaimana pada Gambar 7.



Gambar 9. Diagram blok *Linear Predictive Coding*



Gambar 10. Diagram Blok Proses Pembentukan Matriks *Sequence* dengan VQ [1]



Gambar 11. Diagram Blok Proses Pembuatan *Caption*

a. *Preemphasis*

Proses ini bertujuan untuk meratakan spektral sinyal dan menghilangkan derau pada sinyal. FIR filter yang digunakan untuk *preemphasis* adalah sebagai berikut [2]:

$$H(z) = 1 - 0.95z^{-1} \qquad (11)$$

b. *Frame Blocking and Windowing*

Pada tahap ini sinyal hasil *Preemphasis* dikelompokkan ke dalam bingkai-bingkai dengan ukuran masing-masing bingkai sebesar N data (Gambar 8). Bingkai ini berurutan dengan pemisahan antara kedua bingkai sebesar M data.

3.2.4.2. *Feature Extraction*

Feature extraction atau ekstraksi ciri bertujuan untuk mendapatkan vektor ciri dari setiap *frame* sinyal. Metode yang digunakan pada tahap *feature extraction* yaitu *Linear Predictive Coding* (LPC) seperti pada Gambar 9.

3.2.4.3. *Vector Quantization*

Untuk mengkuantisasi matriks ciri dibutuhkan suatu *codebook* berukuran 2^B . *Codebook* dibuat dengan algoritma *k-means clustering* seperti pada Gambar 10.

3.2.4.4. *Feature Matching*

Feature Matching adalah tahap membandingkan keluaran hasil ekstraksi ciri dengan data latih yang ada dalam database.

3.2.5. Pembuatan *Caption*

Pada tahap ini, teks hasil keluaran proses *speech recognition* disimpan ke dalam file dengan format (*.srt) disertai dengan timestamp-nya. File (*.srt) tersebut kemudian bisa digunakan dalam aplikasi pemutar video untuk menampilkan *caption*. Ada 2 proses pengolahan yang dilakukan, yaitu *word checking* dan *word spelling correction* (Gambar 11).

4. Pengujian Sistem

Pengujian sistem dalam peneltian ini terbagi menjadi beberapa tahapan seperti yang tersaji pada Gambar 11.

4.1. Pengujian Untuk Mencari Parameter Optimal

Pengujian sistem untuk mencari parameter optimal dalam peneltian ini dilakukan dengan 30, 45, 60, 75, dan 90 data latih.

4.1.1. Pengujian Dengan 30 Data Latih

Pengujian dengan 30 data latih dapat dilihat pada Tabel 1. Dari Tabel 1 terlihat bahwa akurasi tertinggi yang diperoleh sebesar 30% pada pengujian dengan jumlah *state* 4 dan jumlah *cluster* 64.

4.1.1.1. Pengujian Dengan 45 Data Latih

Pengujian dengan 45 data latih dapat dilihat pada Tabel 2. Dari table 2 dapat diketahui akurasi terbesar diperoleh saat pengujian menggunakan sistem dengan jumlah *cluster* 64 dan jumlah *state* 4 yaitu sebesar 52,00%.

4.1.1.2. Pengujian Dengan 60 Data Latih

Pengujian dengan 60 data latih dapat dilihat pada Tabel 3. Dari Tabel 3 dapat diketahui akurasi tertinggi pada pada pengujian ini adalah 65,50% dan terjadi pada dua titik yaitu pada pengujian dengan jumlah *cluster* 64 dan jumlah state 1 dan 6.

4.1.1.3. Pengujian Dengan 75 Data Latih

Pengujian dengan 75 data latih dapat dilihat pada Tabel 4. Dari Tabel 4 dapat diketahui bahwa akurasi sistem masih mencapai titik maksimal pada pengujian dengan menggunakan 64 cluster.

4.1.1.5. Pengujian Dengan 90 Data Latih

Pengujian dengan 90 data latih dapat dilihat pada Tabel 5. Dari Tabel 5 dapat diketahui bahwa sistem mengalami puncak akurasi pada jumlah cluster 256 dan jumlah state 6 yaitu sebesar 75,50%.

Tabel 1. Hasil Pengujian dengan 30 Data Latih

		Cluster			
		32	64	128	256
State	3	30,00%	26,50%	13,00%	7,00%
	4	17,50%	30,00%	14,50%	7,50%
	5	18,00%	28,00%	18,00%	10,00%
	6	27,50%	29,00%	15,00%	8,50%

Tabel 2. Hasil Pengujian dengan 45 Data Latih

		Cluster			
		32	64	128	256
State	3	38,50%	47,50%	37,00%	34,00%
	4	32,00%	52,00%	36,00%	31,00%
	5	37,00%	50,00%	37,50%	32,50%
	6	33,00%	51,00%	35,50%	31,50%

Tabel 3. Hasil Pengujian dengan 60 Data Latih

		Cluster			
		32	64	128	256
State	3	49,00%	65,50%	59,50%	54,50%
	4	47,50%	64,00%	62,50%	53,00%
	5	53,50%	64,50%	61,50%	54,00%
	6	45,50%	65,50%	61,00%	52,00%

Tabel 4. Hasil Pengujian dengan 75 Data Latih

		Cluster			
		32	64	128	256
state	3	54,00%	68,50%	66,00%	65,50%
	4	52,50%	69,50%	67,50%	67,00%
	5	56,00%	68,00%	67,00%	63,50%
	6	49,50%	70,50%	68,00%	68,00%

Tabel 5. Hasil Pengujian dengan 90 Data Latih

		Cluster			
		32	64	128	256
State	3	61,50%	71,50%	72,50%	74,00%
	4	59,50%	72,50%	74,00%	73,00%
	5	62,50%	70,50%	74,00%	72,00%
	6	57,50%	73,00%	73,00%	75,50%

Tabel 6. Perbandingan Jumlah Suku Kata yang Salah dengan Suku Kata yang Tidak Dikenali

Cluster	Salah (S)	Tidak Dikenali (TD)	Rasio (S:TD)
32	304	99	75 : 25
64	113	163	40 : 60
128	46	262	15 : 85
256	8	365	2 : 98

4.1.2. Pengujian dengan Sumber Suara yang Berbeda

Pengujian melibatkan 5 sumber suara. Suara pertama adalah suara *default* yang telah dilakukan pengujian pada pengujian 1. Suara kedua dan ketiga adalah suara pria sedangkan suara ketiga dan keempat adalah suara wanita (Gambar 12).

4.1.3. Pengujian dengan Ukuran *Database* yang Bervariasi

Pengujian kali ini bertujuan untuk mencari hubungan antara ukuran *database* dengan akurasi sistem. Pengujian dilakukan dengan ukuran *database* yang bervariasi dari 6, 10, dan 13 (Gambar 13).

4.2. Analisis

4.2.1. Pengaruh Jumlah Data Latih Terhadap Keakurasian Sistem

Dari pengujian 1, terlihat bahwa akurasi sistem perlahan-lahan membaik setelah jumlah data latih per suku kata ditingkatkan. Hal ini dikarenakan semakin banyak data latih untuk suatu suku kata, parameter HMM yang dihasilkan akan semakin baik.

4.2.2. Pengaruh Jumlah *Cluster* Terhadap Keakurasian Sistem

Berdasarkan pengujian 1, baik dengan 30, 45 maupun 60 data latih, jumlah *cluster* berpengaruh terhadap akurasi sistem (Gambar 14). Tabel 6 memberikan perbandingan jumlah suku kata yang salah dengan jumlah suku kata yang dikenali untuk setiap *cluster* pada pengujian dengan 60 data latih.

4.2.3. Pengaruh Sumber Suara Terhadap Keakurasian Sistem

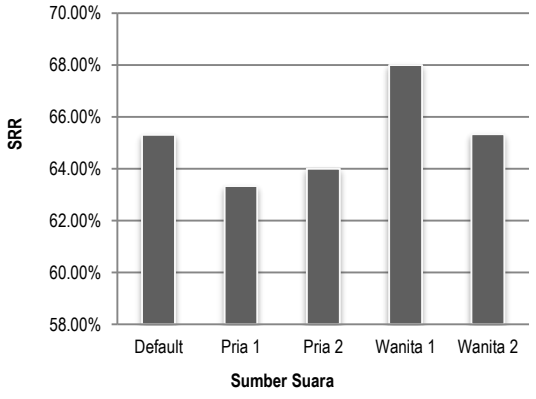
Grafik hasil pengujian 3 (Gambar 12) menunjukkan bahwa terdapat sedikit perbedaan akurasi apabila sistem diuji dengan sumber suara / pembicara yang berbeda.

4.2.4. Pengaruh Ukuran *Database* Terhadap Keakurasian Sistem

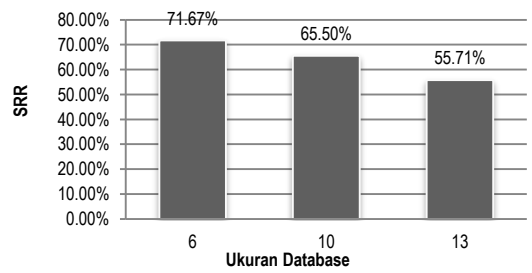
Berdasarkan Gambar 13 didapatkan hubungan antara ukuran *database* dengan akurasi sistem yaitu semakin besar ukuran *database* maka akurasi akan semakin menurun.

4.2.5. Pengaruh Jumlah *State* Terhadap Keakurasian Sistem

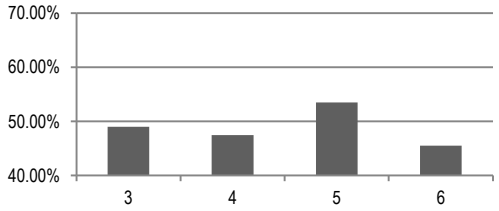
Pengaruh jumlah state terhadap keakurasian sistem dapat dilihat pada Gambar 14 s.d. 16.



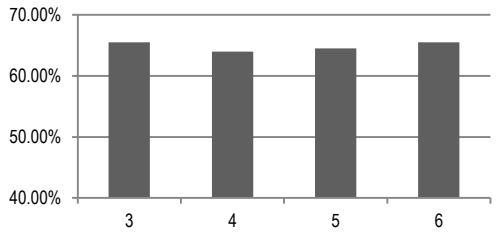
Gambar 12. Grafik Perbandingan Perbedaan Sumber Suara Terhadap Akurasi



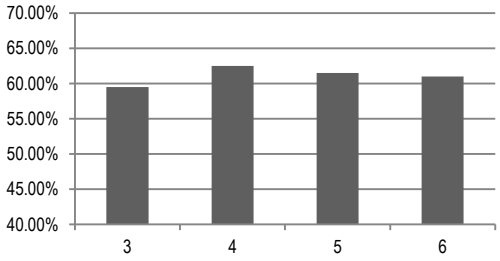
Gambar 13. Grafik Perbandingan Ukuran *Database* Terhadap Akurasi



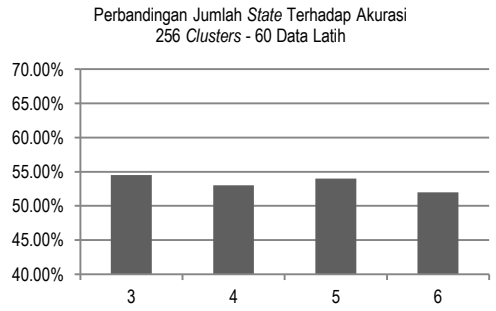
Gambar 14. Grafik Perbandingan Jumlah *State* Terhadap Akurasi 32 *Cluster*



Gambar 15. Grafik Perbandingan Jumlah *State* Terhadap Akurasi 64 *Cluster*



Gambar 16. Grafik Perbandingan Jumlah *State* Terhadap Akurasi 128 *Cluster*



Gambar 17. Grafik Perbandingan Jumlah *State* Terhadap Akurasi 256 *Cluster*

4.3. Kesimpulan

Berdasarkan analisis dan pengujian yang dilakukan, didapatkan beberapa kesimpulan, yaitu:

a. Akurasi maksimum yang didapatkan untuk sistem *Automatic Video Captioning* dengan *speech recognition* menggunakan LPC untuk *feature extraction* dan *hidden markov model* untuk *feature matching* adalah sebesar 75,50% yang didapat pada sistem dengan jumlah *cluster* 256, jumlah *state* 6, dan jumlah data latih 90 untuk setiap suku kata.
b. Faktor–faktor yang berpengaruh dalam kinerja sistem antara lain :
   1) Jumlah Data Latih
   2) Jumlah *Cluster*
   3) Pembicara/Sumber Suara
   4) Ukuran *Database*
c. Untuk variasi jumlah *state* yang kecil, dalam hal ini 4 variasi, tidak terlihat pengaruh yang signifikan antara jumlah *state* dengan akurasi sistem.

Daftar Pustaka

[1] Arman, Arry Akhmad, "*Proses Pembentukan dan Karakteristik Sinyal Ucapan*", Thesis Pascasarjana, Bandung, 2008.

[2] Hasegawa, M. dan Johnson, "*Lecture Notes in Speech Production, Speech Coding, and Speech Recognition*", University of Illinois, Urbana-Champaign, 2000.

[3] Nilsson, M. dan M. Ejnarsson, "*Speech Recognition Using Hidden Markov Model: Performance Evaluation In Noisy Environtment*", Blekinge Institute of Technology, Ronneby, 2002.

[4] Oyelade, O. J. dan O. O. Oladipupo, "*Application of K-Means Clustering Algorithm for Prediction of Students' Academic Performance*", International Journal of Computer Science and Information Security, Vol. 7, No. 1, 2010.

[5] Rabiner, Lawrence dan Biing-Hwang Juang, "*Fundamentals of Speech Recognition*", Prentice, Prentice Hall, New New Jersey, 1993.

[6] Raharjo, Budi, "*Pemrograman C++: Mudah dan Cepat Menjadi Master C++*", Penerbit Informatika, Bandung, 2011.

[7] Wibisono, Yudi, "*Penggunaan Hidden Markov Model Untuk Kompresi Kalimat*", Institut Teknologi Bandung, 2008.