

# A Principle Component Analysis Cascade with Multivariate Regression for Statistical Downscaling Technique : A Case Study in Indramayu District

Agus Buono<sup>1</sup>, Akhmad Faqih<sup>1</sup>, Adi Rakhman<sup>1</sup>, I Putu Santikayasa<sup>2</sup>, Arief Ramadhan<sup>3</sup>, M. Rafi Muttqien<sup>3</sup>, M. Asyhar A<sup>3</sup>, and Rizaldi Boer<sup>1</sup>

<sup>1</sup>Center for Climate Risks and Opportunity Management in Southeast Asia and Pacific (CCROM-SEAP) Bogor Agriculture University, Bogor-West Java, Indonesia

<sup>2</sup>Department of Geophysics and Meteorology, Faculty of Mathematics and Natural Sciences, Bogor Agricultural University, Bogor-West Java, Indonesia

<sup>3</sup>Department of Computer Sciences, Faculty of Mathematics and Natural Sciences, Bogor Agricultural University, Bogor-West Java, Indonesia

Email: pudesha@yahoo.co.id, rizaldiboer@gmail.com

**Abstract**—This research is focused on the development of statistical downscaling model using principle component analysis (PCA) as feature extraction cascade with multivariate regression as model prediction. Sea Surface Temperature (SST) and the General Circular Model (GCM) derived from 18 ensemble members of ECHAM5 model under A1B scenario are used as predictors, and the September, October, November, and december (SOND) rainfall in each station in Indramayu as the response variables. SST domains were reduced using the PCA that explain 80%, 85%, and 95% of the data variability, which were divided into three sets, namely SST lag 1 (August), lag 2 (July), and lag 3 (June). GCM domain-sized 5x5 with 2.8x2.8 km resolution is reduced by PCA. The number of components that are taken are such that the variables must still explain 90% of original data variability. The new variables yielded by PCA are then take as the input of the multivariate regression with 13 observation data (from 13 station) as the output. The regression parameters are estimated by using the least sum of square error criteria. By assigning 0.4 as the correlation boundary to select the domain SST, the SST lag 3 (June) yields the best regression model with a correlation of about 0.7. But if the limit is increased to 0.55, only the SST lag 1 (August) that meet, and produce a model with a correlation above 0.7.

## I. INTRODUCTION

Facts indicated that agriculture system productions are influenced significantly by climate conditions. There are many model have been developed that related climate variables with the agriculture system. There are usually two main obstacles, first is the limitation of historical climate data with a sufficiently long series, and second is the need of future climate projections

(under certain scenarios), especially in the study the impacts of climate change.

General Circulation Model (GCM) provides a solution to this problem and the data has been widely used for climate change studies. However, due to its coarse resolution, that is about 2x2 degrees, or about 200x200 km, the model is unable to capture local variability that is needed in the analysis of a smaller coverage area, such as district level. Therefore there is a gap between the GCM output and the observed data. In this case, the GCM is only able to capture the pattern of average, whereas variability mainly influenced by local factors is not accommodated.

This research is addressed to develop a methodology of statistical downscaling model using a PCA cascading with multivariate regression. This model links the rainfall data from GCMs and Sea Surface Temperature (SST) with the observational data to predict rainfall intensity in Indramayu district. Downscaling techniques will be applied to estimate the total rainfall on SOND (September, October, November, and December) season. With 24 time periods of data, an 8-fold cross validation technique is implemented to evaluate the methodology.

The remainder of this paper is organized as follows: Section 2 presents the principles of statistical downscaling. Section 3 describes the data and experimental setup. Section 4 is addressed to explain in brief the concept of principle component analysis and multivariate regression technique. Result and discussion are presented in Section 5, and finally, Section 6 is dedicated to the conclusions of this research study.

## II. STATISTICAL DOWNSCALING

Downscaling is defined as an effort to connect between global-scale (explanatory variables) and local scale climate variables (response variables), [1]. Figure 1 illustrates the process of downscaling. There are two approaches for downscaling, using regional data (obtained from a regional climate model, RCM), or global data (obtained from the general circulation models, GCM). The first approach is known as statistical dynamical downscaling, while the second is known as statistical downscaling (SD). Statistical downscaling based on the relationship between coarse-scale grid (predictor) with local-scale data (response) is expressed by a statistical model that can be used to translate a global scale anomalies which became an anomaly of some variables of local climate (Zorita and Storch 1999, in [2]). In this case the SD is a transfer function that describes the functional relationship of global atmospheric circulation with elements of the local climate, which is formulated as follows:

$$R_{t,q} = f(G_{t,p,s,g})$$

where :

- $R$  : Local climate variables as the responds
- $G$  : global climate variables (provided by GCM model), as the predictors
- $t$  : time period
- $q$  : dimension of  $R$
- $p$  : dimension of  $G$
- $s$  : layers in the atmosphere
- $g$  : GCM domain

Until now the SD models that have been developed are generally categorized into five, i.e., i) based on regression techniques or classification, ii) based on linear or non linear model, iii) based on parametric and non parametric, iv) based on projection and selection, and v) based on model-driven or data-driven techniques. Nevertheless, an SD model can be included in the combination of the five categories, for example PCR (principle component regression) that were categorized as regression-based methods, linear, parametric, projections and data-driven. In this research, we developed multivariate regression model for statistical downscaling using data from the GCM and sea surface temperature as explanatory variables. The use of SST data is specifically intended to capture the El Nino phenomenon, so the model is expected to produce better prediction results.

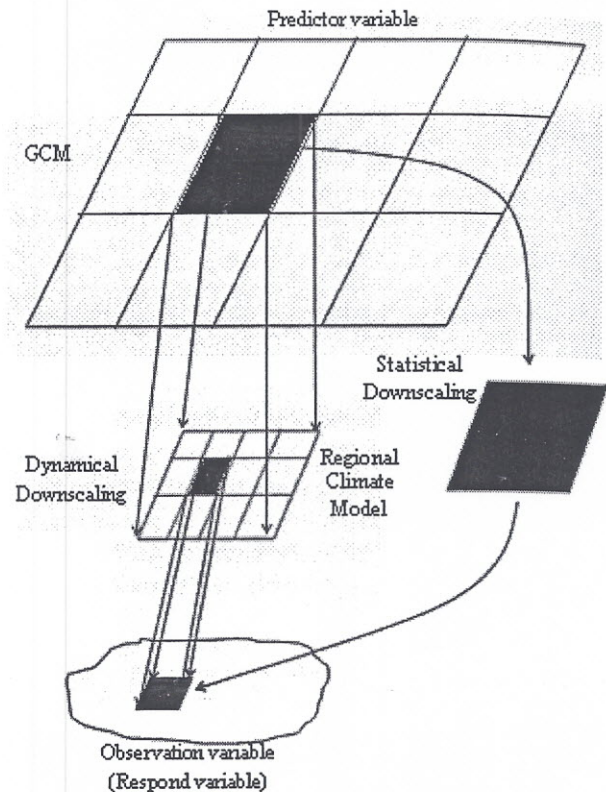


Fig. 1. Illustration of the downscaling (Source : [3])

## III. DATA AND EXPERIMENTAL SETUP

### A. Data

The research involved three types of data, i.e. i) precipitation data from GCM model (with the A1B scenario, ECHAM5 model (with 18 members and  $2.8^\circ \times 2.8^\circ$  resolution), ii) SST data (with  $2^\circ \times 2^\circ$  resolution), and iii) rainfall data from 17 rain gauge stations in Indramayu. All datasets have the time period from 1979 to 2002 (24 periods).

### B. Experimental Setup

Generally, the experiment divides into four steps as presented in Figure 2, i.e.:

- a. Preprocessing : This process consists of (1) checking the rainfall observation data (validity and consistency) using the method described by [4]; (2) calculating the SOND season climate variables (rainfall observation of all stations, GCM for all ensemble members). Especially for the SST, the data is divided into three months, i.e. June (lag 3), July (lag 2) and August (lag 1); and (3) calculating the normalization of SOND data for rainfall observations and GCM data, and the normalization of SST for each lag.
- b. SST domain selection: for every point in the SST domain, calculates the correlation coefficient between

normalize SST in the point with the observation rainfall in the existing station, and then take the average correlation. If the average correlation is greater than 0.4, we put the point to the SST domain that gives a significant influence to the observation rainfall. In order to reduce the number of point, we applied PCA to the domain and take a number of coefficients that explain the original variability of the data around 90 .

c. GCM data reduction: In this research we use 5x5 grid of the GCM data, and the dimension is reduced by using PCA. We take a number of coefficients that contribute around 90 of the original data variability.

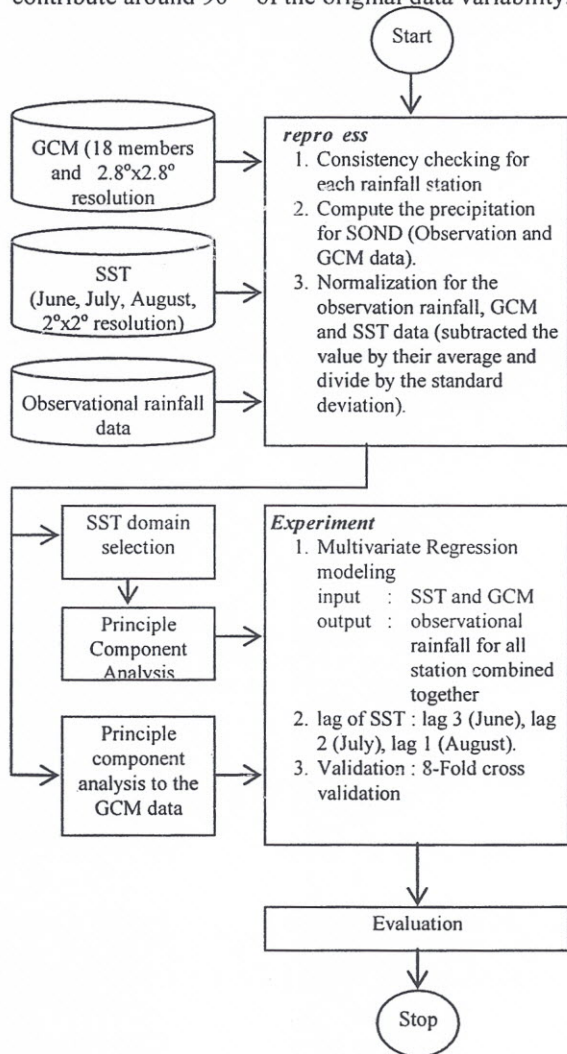


Fig. 2. The flow diagram of the experiments

d. Modeling and testing multivariate regression model: the models has a number of predictors that yielded by step 2 and 3, and the responses variables are normalize observation rainfall from the consistence station yielded by step 1. Multivariate regression parameters are estimate by using the least squares

error criteria as described in [5]. A considered factor in this experiment is the lag time of the SST data, i.e. the SST in June (lag 3), July (lag 2), and August (lag 1). Then we use an 8-fold cross validation to test the model by comparing the predicted data with the observed.

IV. PRINCIPLE COMPONENT ANALYSIS AND MULTIVARIATE REGRESSION

A. Principle Component Analysis

The fundamental idea of the PCA is as follows: Let  $x$  is a vector with  $m$  elements of a random variables:

$$x = (x_1, x_2, \dots, x_m)^T,$$

then we defined a new vector,  $y$ , consist of  $n$  new elements :

$$y = (y_1, y_2, \dots, y_n),$$

where,  $y_j, j=1, 2, 3, \dots, n$ , as a linear combination of the  $m$  elements in the vector  $x$  :

$$\begin{aligned} y_1 &= a_{11}x_1 + a_{12}x_2 + \dots + a_{1m}x_m = a_1^T x \\ y_2 &= a_{21}x_1 + a_{22}x_2 + \dots + a_{2m}x_m = a_2^T x \\ &\dots \\ y_n &= a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nm}x_m = a_n^T x \end{aligned}$$

The problem is how to decide the coefficients vectors,  $a$  such that the new variables could capture the variability of the original variables ( $x$ ). In other word, the vector  $a$  have to maximized the variance of the  $y_j, j=1, 2, 3, \dots, n$ , that formulated as  $Var(y_i) = \sigma_{y_i}^2 = a_j^T \Sigma a_j$ , where  $\Sigma$  is the covariance matrix of vector  $x$ .

In geometrical aspect, the new variables generate a vector space with  $a$  as the axis. Without change the meaning, we can constraint the vectors  $a$  as follows:  $a^T a = 1$  and  $cov(a_i, a_j) = 0$  for  $i \neq j, i, j=1, 2, 3, \dots, n$ . Then, the problem is transformed to the maximization with conraint, and formulated as follows :

$$\text{Maximize : Variance}(y) = a_j^T \Sigma a_j$$

$$\text{Constraint : } a^T a = 1 \text{ and } cov(a_i, a_j) = 0 \text{ for } i \neq j, i, j=1, 2, 3, \dots, n$$

By using the Lagrange multiplier technique,  $a_j$  is the characterisic vector of  $\Sigma$  corresponding to the largest characteristic roots  $j, \lambda_j$ .

B. Multivariate Regression

Multivariate regression analysis is an input-output system that transforms  $p$  input variables into  $q$  responses variables, i.e.:

$$\begin{aligned} q \text{ responses variables} &: z_1, z_2, \dots, z_q \\ p \text{ predictor variables} &: x_1, x_2, \dots, x_p \end{aligned}$$

Each respond variable,  $z_i$ , formulated as follows:

$$z_1 = \beta_{01} + \beta_{11}x_1 + \dots + \beta_{p1} x_p + \varepsilon_1$$

$$z_2 = \beta_{02} + \beta_{12}x_1 + \dots + \beta_{p2} x_p + \varepsilon_2$$

$$\dots$$

$$z_q = \beta_{0q} + \beta_{1q}x_1 + \dots + \beta_{pq} x_p + \varepsilon_q$$

For simplify in the mathematical formulation, suppose we have r observations, then we have some matrix notation:

$$Z_{rxq} = \begin{bmatrix} z_{11} & z_{12} & \dots & z_{1q} \\ z_{21} & z_{22} & \dots & z_{2q} \\ \dots & \dots & \dots & \dots \\ z_{r1} & z_{r2} & \dots & z_{rq} \end{bmatrix}$$

$$X_{rx(p+1)} = \begin{bmatrix} 1 & x_{11} & \dots & x_{1p} \\ 1 & x_{21} & \dots & x_{2p} \\ \dots & \dots & \dots & \dots \\ 1 & x_{r1} & \dots & x_{rp} \end{bmatrix}$$

and

Whereas, the regression parameters matrix and the error matrix are:

$$\beta_{(p+1)xq} = \begin{bmatrix} \beta_{01} & \beta_{02} & \dots & \beta_{0q} \\ \beta_{11} & \beta_{12} & \dots & \beta_{1q} \\ \dots & \dots & \dots & \dots \\ \beta_{p1} & \beta_{p2} & \dots & \beta_{pq} \end{bmatrix}$$

$$\varepsilon_{nxq} = \begin{bmatrix} \varepsilon_{11} & \varepsilon_{12} & \dots & \varepsilon_{1q} \\ \varepsilon_{21} & \varepsilon_{22} & \dots & \varepsilon_{2q} \\ \dots & \dots & \dots & \dots \\ \varepsilon_{n1} & \varepsilon_{n2} & \dots & \varepsilon_{nq} \end{bmatrix}$$

and

Then, the regression model is formulated as follows:

$$Z_{(rxq)} = X_{rx(p+1)}\beta_{(p+1)xq} + \varepsilon_{rxq}$$

With some assumptions:

$$E(\varepsilon_i) = 0, \text{ and } Cov(\varepsilon_i, \varepsilon_k) = \sigma_{ik} I, \text{ for } i,k:1, 2, \dots, p$$

By using the least of square error (LSE), the regression parameters,  $\beta_{(p+1)xq}$ , are formulated as follows, [5]:

$$\hat{\beta}_{(p+1)xq} = (X^T X)^{-1} X^T Z$$

Then, the observation could be predict by the formulae as follows:

$$\hat{Z} = X\hat{\beta} = X(X^T X)^{-1} X^T Z$$

V. RESULT AND DISCUSSION

Consistency analysis on 17 rainfall stations in Indramayu found that four stations are inconsistent. Those stations are Losarang, Indramayu, Bulak Kandanghaur, and Tugu. These are indicated by the values of F statistics that are greater than the F threshold (4.3248) at 95 confidence level as shown in Figure 3.

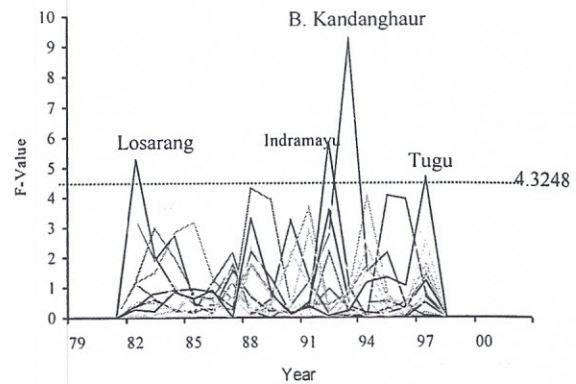


Fig. 3. Pattern F statistic values for 17 stations.

More clear illustration of the inconsistencies is shown in Figure 4. The figure shows that the SOND rainfall inconsistent in 1993-1994. Based on the results above, the four stations are not included for further analysis, which mean that only 13 stations will be analyzed in this study.

Based on the 3393 grid points of the SST domain, the correlation distribution skew to the right (Figure 5). We can see that the number of points in the SST domain that has a high correlation is increase started from June, July and August. We put the point with correlation greater than 0.4 as the SST domain included in downscaling modeling.

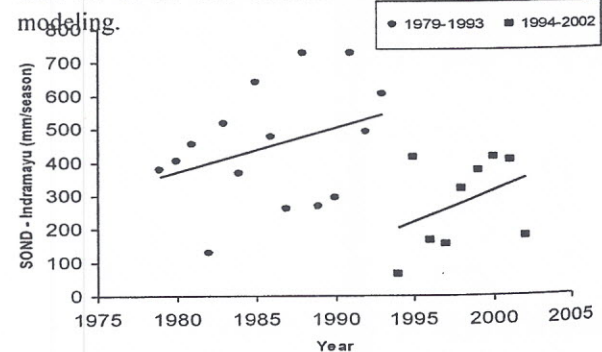


Fig. 4. SOND rainfall pattern for Indramayu station

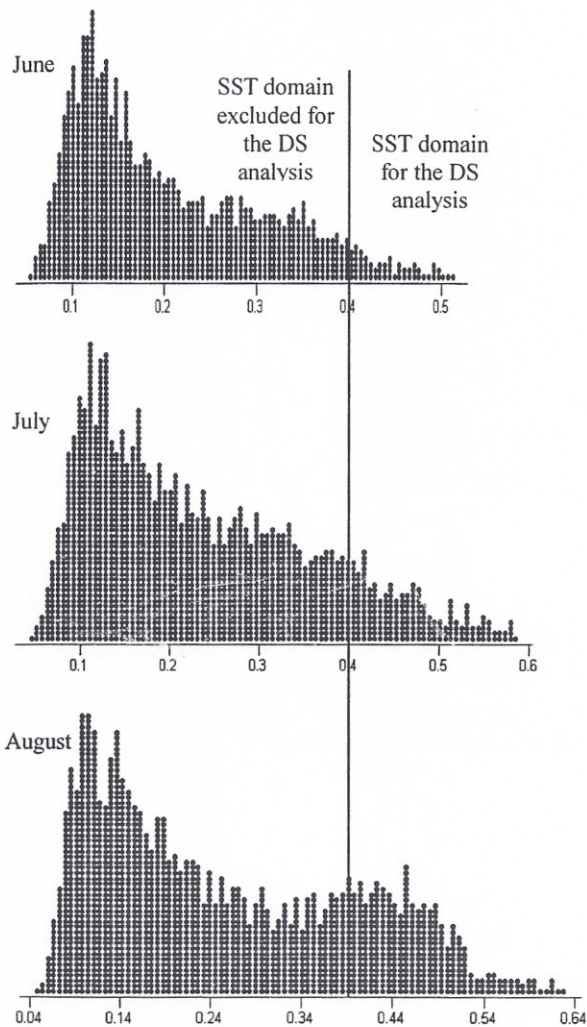


Fig. 5. Dotplot of the domain based on the correlation between SST with the observation rainfall

Based on the SST domain which its correlation greater than 0.4, and 5x5 GCM data, the regression model give the predicted SOND rainfall with similar distribution to the one for the true value (Figure 6 and Figure 7). It mean, the model have a good prediction for the value around the center. But for an extreme condition, the prediction is to far from the true value.

However, if we investigate the correlation between observations and predicted values, showed that SST in June to provide the best prediction model. The correlation value is 0.70, 0.57, and 0.59 each for the SST in June, July and August, respectively. One of the reasons why the SST in July and August produces models that are relatively less good than the model yielded by using SST in June is more grid in the SST domain included in the model building. This causes increased variability prediction, and ultimately correlation between prediction with the observation goes

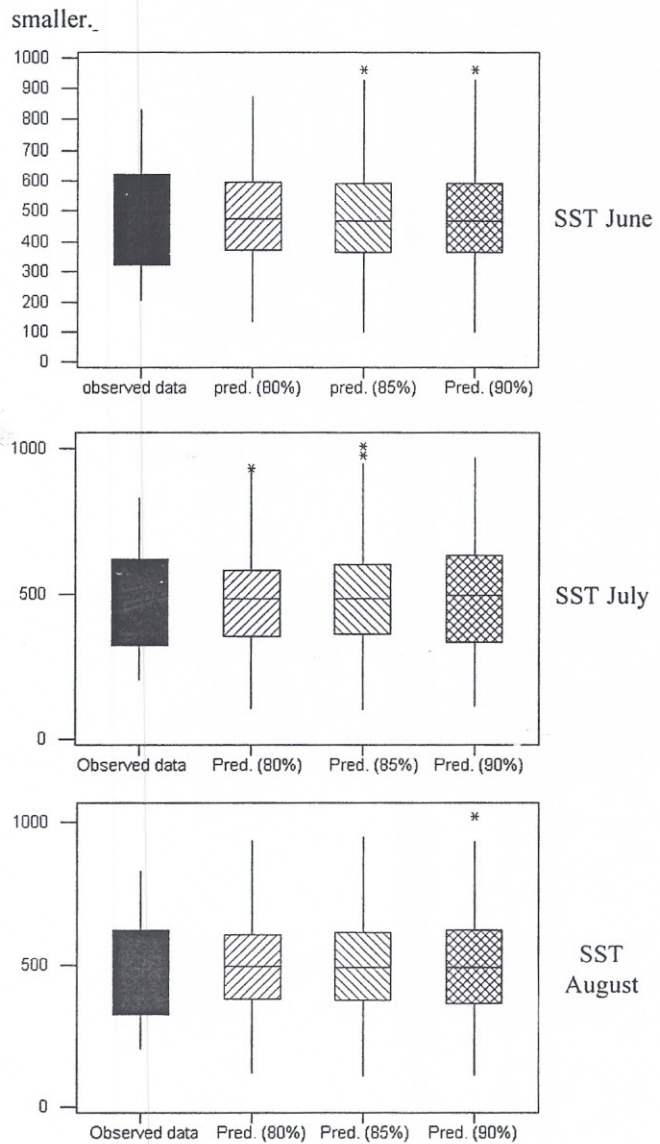


Fig. 6. Boxplot of the observation SOND rainfall and the prediction with PCA 80 , 85 , and 90

By determining the correlation of 0:55 as the threshold in selecting the SST domain, SST lag 1 produces the best model, the correlation between observations and predictions slightly above 0.7. It also demonstrated with the fact that the regression equation between the predictions and the observations by using SST lag 1 looks better (Figure 8) compare with the three model before as presented in Figure 7. The gradient of the equation closer to 1 (0.97) and the intercept closer to 0 (9.15), as presented in figure 8. Meanwhile, the gradient for the relationship between predictions and observations for the three previous models are 0.87, 0.70, and 0.75, each for SST lag 3, lag 2 and lag 1, respectively. And, the intercept are far from 0, i.e.

66.14, 147.08, and 112.83, each for SST lag 3, lag 2 and lag 1, respectively.

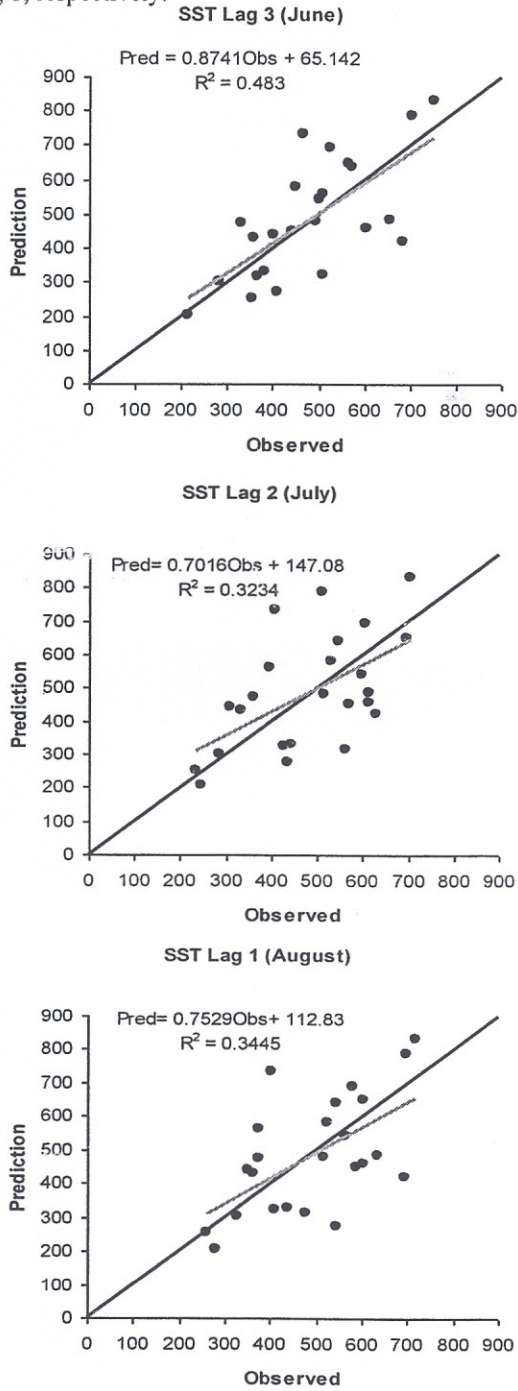


Fig. 7. Scattered plot of the observation vs. prediction (SST lag 3, lag 2 and lag 1)

**SST Lag 1 (August, domain>0.55)**

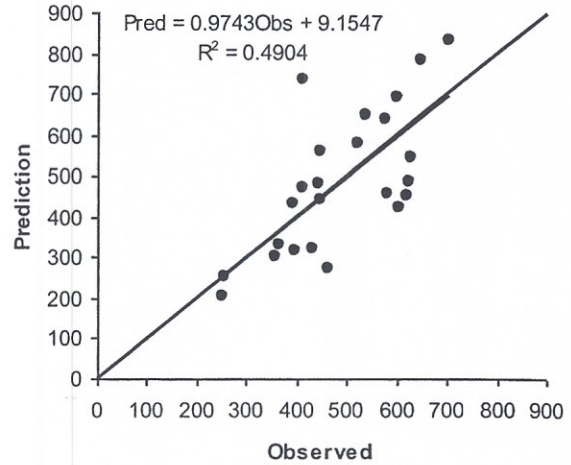


Fig. 8. Scattered plot of the observation vs. rediction (SST lag 1, with 0.55 as the threshold of the correlation in the SST domain selection)

VI. CONCLUSIONS

Based on the experiment, we can conclude several things:

- a. SST domain influencing the observations at a particular station varies temporally, the smaller the lag, the more points in the domain that correlates higher with the observation. The maximum correlation increase from 0.52, 0.58 and 0.63, each for SST lag 3, lag 2, and lag 1, respectively.
- b. If the SST domain selection using 0.4 as cut of point, SST lag 3 (June) give the best model with correlation between the observation and the prediction is 0.70. Whereas, the such value are only 0.57 and 0.59, each for SST lag 2 and lag 1, respectively.
- c. If the cut of point is increased to 0.55, the SST lag 1 is the only one that can generate the SST domain for further analysis. The model yielded by this domain look better than before. Its correlation is slightly above than 0.7, and the prediction more closely to the true value.

In order to increase the system accuracy, for future word, we will focus on the SST selection that appropriate to the individual station.

## ACKNOWLEDGMENT

This research is part of Bogor Agricultural University's IM-HERE B2c Project, which is funded by the National Government under contract number: 7/13.24.4/SPP/IM-HERE/2010. The authors thank to International Research Institute (IRI), Columbia University for their permission to access the GCM datasets and to BMKG for providing the rainfall data.

## REFERENCES

- [1] Wilby, R.L. and T.M.L. Wigly, "Downscaling General Circulation Model Output : a Review of Methods and Limitations", *Progress in Physical Geography* 21, 4, pp. 530-548, 1997.
- [2] Wigena, A.H. *Pemodelan Statistical Downscaling dengan Regresi Projection Pursuit untuk Peramalan Curah Hujan Bulanan : Kasus Curah Hujan Bulanan di Indramayu*. PhD Dissertation, Department of Statistics, Graduate School, Bogor Agriculture University, Bogor 2006.
- [3] Sutikno. *Statistical Downscaling Luaran GCM dan Pemanfaatannya untuk Peramalan Produksi Padi*. PhD Dissertation, Department of Meteorology, Graduate School, Bogor Agriculture University, Bogor. 2008.
- [4] Boer, R. *Metode untuk mengevaluasi Keandalan model Prakiraan Musim*. Climatology laboratory, Departement of Meteorology, Faculty of Mathematics and Natural Sciences, Bogor Agriculture University Bogor. 2006.
- [5] Johnson, R.A., and D.W. Wichern. *Applied Multivariate Statistical Analysis*. Fifth Edition, Prentice-Hall, Inc., New Jersey. 1998.

# Pengenalan Kata Berbahasa Indonesia dengan *Hidden Markov Model (HMM)* menggunakan Algoritme *Baum-Welch*

Agus Buono, Arief Ramadhan, Ruvinna

Departemen Ilmu Komputer, FMIPA - IPB

## Abstract

*Speech recognition is the process of converting an acoustic signal, captured by a microphone or a telephone, to a set of words. Speech can be defined as waves of air pressure created by airflow pressed out of the lungs and going out through the mouth and nasal cavities. The air passes through the vocal folds (chords) via the path from the lungs through the vocal tract, vibrating them at different frequencies. To make a computer system reacts as a human being in recognizing a word is not an easy task. A good model is needed to represent the speech signal as the input of the speech system.*

*This research used Baum-Welch training algorithm to train HMM as the model of a word. The purpose of this research is to implement HMM using Baum-Welch training algorithm to recognize an isolated word. Words of this research are ranged into 2 types of syllable; they are 2 syllables and 3 syllables. Speaker of this research is also ranged into 2 trained woman speaker and 2 trained men speaker, therefore this system is said to be speaker-dependent. In general this research resulted some HMM, that represent speech signal input as an Indonesian word. The best HMM to recognize an isolated word is HMM using 3 hidden states that were trained up to 10 epochs and the best accuracy is 83.125%.*

## PENDAHULUAN

### Latar Belakang

Mengenali sebuah kata atau kalimat bukanlah hal yang sulit dilakukan bagi manusia. Apalagi kata tersebut merupakan 'Bahasa Utama' yang digunakannya sehari-hari. Berbagai logat ataupun cara bicara tidak menjadi halangan untuk mengenali kata tersebut. Namun pekerjaan ini bukanlah hal yang mudah dilakukan oleh sebuah sistem komputer.

Berbagai sistem pengenalan suara atau yang dapat disebut juga *Automatic Speech Recognition (ASR)* telah banyak dikembangkan di berbagai negara dengan berbagai bahasa. Berikut merupakan beberapa sistem pengenalan suara yang telah dikembangkan:

- *Spoken Dialogue System*, sistem yang dapat melakukan dialog singkat guna mendapatkan informasi tertentu. Seperti pada seorang *customer service*, pengguna hanya perlu menjawab 'ya' atau 'tidak' untuk mendapatkan informasi tertentu.
- *Speed Dialing System*, sistem yang dapat mengenali sebuah nama atau ID seseorang dan mencarinya dalam buku telepon untuk segera dihubungi. Pengguna tidak perlu mencari nomor telepon seseorang, biasanya dalam telepon selular, untuk dapat menghubungkannya, namun cukup dengan menyebutkan nama atau ID orang yang akan dihubungi dan sistem secara otomatis menghubungkannya.

- *Speech to Text Translation System*, sistem yang secara otomatis mengetikkan kata-kata yang diucapkan pengguna.

Sistem-sistem tersebut memang telah banyak dikembangkan, namun kata yang dikenali ialah kata berbahasa Inggris. Oleh sebab itu, pengembangan sistem pengenalan kata berbahasa Indonesia perlu dilakukan mengingat bahasa Indonesia memiliki pola dan cara pengucapan yang berbeda dengan bahasa Inggris.

Agar sistem komputer dapat mengenali sebuah kata, maka dibutuhkan representasi yang baik terhadap sinyal-sinyal yang masuk berikut perubahan frekuensinya terhadap rentang waktu tertentu. Hal ini tidak mudah dilakukan mengingat Indonesia merupakan sebuah bangsa yang sangat besar dengan berbagai ragam suku dan logat atau cara bicara. Kesulitan lainnya ialah sistem tidak dapat membedakan sinyal suara yang masuk dengan sinyal noise.

### Tujuan Penelitian

Tujuan penelitian ini ialah menerapkan *Hidden Markov Model (HMM)* menggunakan algoritme Baum-welch untuk mengenali sebuah kata.

### Ruang Lingkup

Adapun ruang lingkup dari penelitian ini antara lain :

1. Kata-kata yang digunakan ialah kata berbahasa Indonesia.

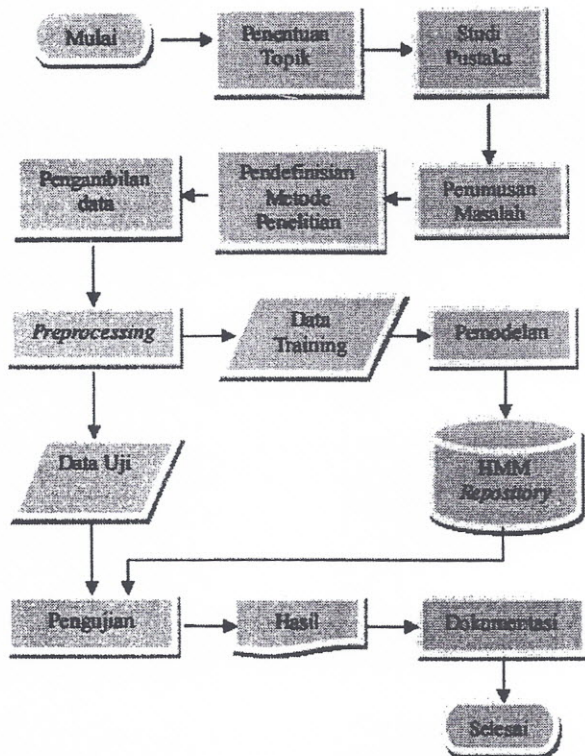


2. Penelitian ini terbatas pada pengenalan kata (*Isolated word*), bukan pengenalan kalimat.
3. Kata yang digunakan sebanyak 40 kata yang dibedakan ke dalam 2 suku kata dan 3 suku kata.
4. Kata yang dikenali harus berasal dari pembicara yang telah terlatih (*Dependent speaker*).

## METODE PENELITIAN

### Kerangka Pemikiran

Langkah-langkah yang dilakukan pada penelitian ini sesuai dengan apa yang disarankan oleh Jurafsky ataupun Rabiner. Namun dilakukan beberapa penyesuaian yang diperlukan. Secara umum, langkah-langkah yang dilakukan dalam penelitian ini digambarkan pada **Gambar 1** berikut:



**Gambar 1** Proses Pengenalan Kata.

### Studi Pustaka

Studi pustaka dilakukan guna memahami langkah-langkah dalam metode yang digunakan dalam penelitian ini. Selain itu, perlu dipelajari perkembangan mengenai *Signal Processing* pada umumnya dan *Speech Recognition* pada khususnya, agar metode yang digunakan tepat sasaran. Referensi-referensi yang digunakan pada penelitian ini dapat dilihat pada daftar pustaka.

### Pengambilan Data Suara

Pengambilan data suara dilakukan dengan *Frekuensi Sampel (Fs)* 11 KHz selama 5 detik untuk setiap kata, karena menurut Do (1994) frekuensi ini dapat meminimalisasi efek *aliasing* saat konversi sinyal analog ke sinyal digital.

Data suara sendiri terbagi dalam 2 macam jumlah suku kata, yaitu: 2 suku kata dan 3 suku kata. Pemilihan ini dilakukan karena sebagian besar kata dalam bahasa Indonesia terdiri oleh 2 atau 3 suku kata. Setiap kelompok kata terdiri dari 20 kata sehingga total seluruh kata yang digunakan ialah 40 kata. Daftar kata yang digunakan dapat dilihat pada **Tabel 1** di bawah ini:

**Tabel 1** Daftar Kata

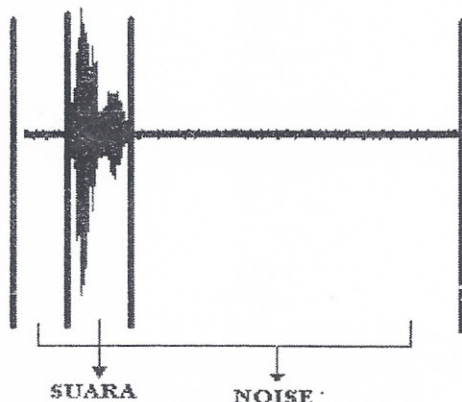
2 Suku kata	3 Suku Kata
Alam	Aljabar
Bogor	Bahasa
Citra	Digital
Data	Empati
Ganda	Fakultas
Hasil	Gelombang
Ilmu	Institut
Jumlah	Jaringan
Keras	Kembali
Lulus	Komputer
Matriks	Laporan
Nama	Metode
Program	Numerik
Robot	Ornamen
Sistem	Perangkat
Temu	Revisi
Umum	Sarjana
Virus	Teori
Warna	Usaha
Yakin	Wisuda

Sistem ini dibatasi dalam hal pembicaranya (*Speaker Dependent*), sehingga pembicara hanya terdiri dari 4 orang, yaitu: 2 orang wanita dan 2 orang laki-laki. Setiap Pembicara mengucapkan kata dengan pengulangan sebanyak 10 kali untuk setiap kata.

Tempat yang digunakan untuk proses pengambilan suara bersifat hening, karena jenis *noise* yang digunakan bersifat *Low* yaitu di bawah 30 db. Bila *Noise* yang terdapat pada ruangan terlalu besar, maka hal tersebut akan menyulitkan saat proses pembersihan/*cleaning* data suara. Selain itu, sangat sulit bagi sistem untuk dapat membedakan gelombang suara dengan *noise* dari lingkungan.

**Preprocessing**

Data suara yang terkumpul merupakan data suara kotor, karena masih terdapat *blank* atau jeda pada awal atau akhir suara, seperti yang terlihat pada **Gambar 2** di bawah ini. Data suara tersebut selanjutnya dibersihkan dari *blank* pada awal atau akhir suara, proses ini disebut sebagai proses pembersihan data.

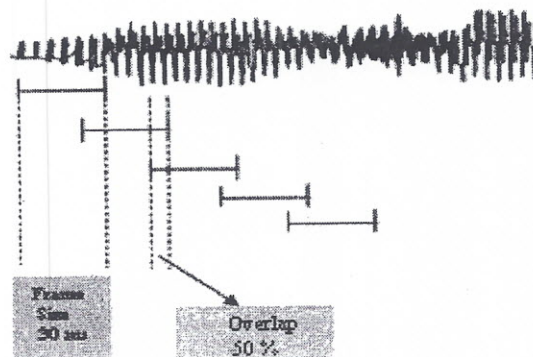


**Gambar 2** Data Suara Kotor.

Apabila *noise* yang terdapat pada suara terlalu besar, maka proses pembersihan ini tidak dapat berjalan optimal. Hal ini dikarenakan sistem tidak mampu membedakan lagi antara gelombang suara dengan *noise*.

Sinyal suara berubah secara perlahan seiring dengan berjalannya waktu dan sepanjang itu, sinyal yang dihasilkan akan berubah karakteristiknya sesuai dengan kata yang disebutkan. Berdasarkan penelitian *Davis* dan *Mermelstein (1980)* dalam *Ganchev (2005)*, *MFCC* dapat merepresentasikan sinyal lebih baik dari *LPC*, *LPCC* dan yang lainnya, dalam pengenalan suara. Oleh sebab itu, penelitian ini menggunakan *MFCC* *FB-40* untuk merepresentasikan karakteristik sinyal suara. Tujuan dari *feature extraction* ini ialah untuk merepresentasikan gelombang sinyal yang masuk ke dalam vektor-vektor ciri akustik, dimana setiap vektornya merepresentasikan informasi dalam beberapa *frame* sinyal.

Pada penelitian ini, *frame* yang digunakan sebesar 30 ms, dimana terjadi *overlap* pada setiap *frame* sebanyak 50 % seperti yang terlihat pada **Gambar 3**. Hal ini mengingat cara bicara orang Indonesia yang cenderung cepat, sehingga *frame* sebesar 30 ms dianggap cukup representatif dalam mencirikan sebuah potongan kata. Masing-masing *frame* sendiri menghasilkan sebuah vektor ciri yang terdiri dari 13 koefisien *cepstral*.



**Gambar 3** Proses *Frame Blocking*.

**Pemodelan Kata**

Proses pemodelan kata dibagi dalam 2 tahapan, yaitu: inialisasi *HMM* dan pelatihan *HMM*.

**A. Clustering data/Inialisasi Model Kata**

Inialisasi *HMM* dimulai dari pengelompokan (*Clustering*) *Cepstral Coeficients* yang telah didapatkan dari proses ekstraksi fitur di atas. Pertama-tama, vektor ciri suara disatukan menurut katanya,

$$\text{Gabung} \begin{cases} \text{kata 1} = O_{t+1}, O_{t+2}, \dots, O_k \\ \text{kata 2} = O_{t+1}, O_{t+2}, \dots, O_k \\ \vdots \\ \text{kata N} = O_{k+1}, O_{k+2}, \dots, O_T \end{cases}$$

Pada penelitian ini digunakan 6 macam jumlah state, mulai dari 3, 4, hingga 8 state *HMM* untuk setiap katanya. Oleh sebab itu, matriks yang didapat selanjutnya dikelompokkan menjadi 3, 4, hingga 8 kelompok. Pengelompokan ini digunakan untuk mendapatkan nilai inialisasi *HMM* yang akan dilatih.

$$\pi = \begin{bmatrix} \frac{\sum \text{kluster 1}}{\sum \text{Observasi}} \\ \frac{\sum \text{kluster 2}}{\sum \text{Observasi}} \\ \vdots \\ \frac{\sum \text{kluster N}}{\sum \text{Observasi}} \end{bmatrix}$$

$$A = \begin{bmatrix} \frac{\sum (1 \rightarrow 1)}{\sum 1 \rightarrow (1-N)} & \dots & \frac{\sum (1 \rightarrow N)}{\sum 1 \rightarrow (1-N)} \\ \vdots & \ddots & \vdots \\ \frac{\sum (N \rightarrow 1)}{\sum N \rightarrow (1-N)} & \dots & \frac{\sum (N \rightarrow N)}{\sum N \rightarrow (1-N)} \end{bmatrix}$$

$$B = \begin{bmatrix} \mu_1 \Sigma_1 \\ \vdots \\ \mu_N \Sigma_N \end{bmatrix}$$

Karena setiap kata dikelompokkan ke dalam 6 macam kelompok, maka setiap kata tersebut akan memiliki 6 buah inialisasi *HMM*. Pada penelitian ini sendiri, digunakan 40 kata, sehingga hasil Inialisasi *HMM* ialah sebanyak 240 macam *HMM*

**B. Pelatihan Model kata**

Pelatihan *HMM* dilakukan dengan menggunakan algoritme *Baum-Welch* dan distribusi *Gaussian*. Pada penelitian ini digunakan *Algoritme Baum-Welch* karena menurut *Shu, et al. (2003)* banyak studi yang telah membuktikan bahwa algoritme *Baum-Welch* mampu melatih *HMM*, untuk sinyal akustik, lebih baik dibanding viterbi. Selain itu, *Baum-Welch* tidak memerlukan nilai inisialisasi yang cukup dekat untuk menghasilkan *HMM* yang baik.

Distribusi *Gaussian* yang digunakan ialah *Distribusi Gaussian Multivariate*, karena setiap pada matriks observasi bukan merupakan nilai skalar melainkan sebuah vektor ciri. Karena terdiri dari 13 koefisien *cepstral*, maka dimensi (d) yang digunakan dalam *Gaussian Multivariate* ialah 13.

Fungsi *Scaling* berikut:  $C_t = \frac{1}{\sum_{i=1}^N \alpha_i(t)}$  juga digunakan dalam pelatihan *HMM*. Fungsi ini berguna untuk menskalakan nilai *Alfa* ( $\alpha$ ) dan *Beta* ( $\beta$ ) yang dihasilkan agar tidak terlalu kecil sehingga mendekati nol. Langkah-langkah yang dilakukan pada pelatihan *HMM* ini ialah :

1. Menghitung nilai  $\gamma$  dan  $\xi$  dengan menyertakan fungsi *Scaling*.

a) *Forward*

Inisialisasi :

$$\alpha_i(1) = \pi_i b_i(O_1)$$

$$\hat{\alpha}_i(1) = C_1 \alpha_i(1)$$

Rekursi :

$$\alpha_i(t+1) = b_j(O_{t+1}) \sum_{i=1}^N \alpha_i(t) \cdot a_{ij}$$

$$\hat{\alpha}_i(t+1) = [\prod_{s=1}^{t+1} C_s] \alpha_i(t+1)$$

Terminasi :

$$\log[P(O|\lambda)] = - \sum_{t=1}^T \log C_t$$

b) *Backward*

Inisialisasi :

$$\beta_i(T) = 1$$

$$\hat{\beta}_i(T) = C_T \beta_i(T)$$

Rekursi :

$$\beta_i(t) = \sum_{j=1}^N \beta_j(t+1) a_{ij} b_j(O_{t+1})$$

$$\hat{\beta}_i(t) = [\prod_{s=t}^T C_s] \beta_i(t)$$

2. Menghitung nilai  $\gamma$  dan  $\xi$ .

$$\gamma_i(t) \equiv P(Q_t = i | O, \lambda) = \frac{\hat{\alpha}_i(t) \hat{\beta}_i(t)}{\sum_{i=1}^N \hat{\alpha}_i(t) \hat{\beta}_i(t)}$$

$$\xi_{ij}(t) \equiv P(Q_t = i, Q_{t+1} = j | O, \lambda)$$

$$= \frac{\hat{\alpha}_i(t) a_{ij} \hat{\beta}_j(t+1) b_j(O_{t+1})}{\sum_{i=1}^N \sum_{j=1}^N \hat{\alpha}_i(t) a_{ij} \hat{\beta}_j(t+1) b_j(O_{t+1})}$$

3. Update nilai ( $\pi, A, B$ )

$$\hat{\pi}_i = \sum_{k=1}^{jml \text{ kata}} \gamma_i^k(1)$$

$$\hat{a}_{ij} = \frac{\sum_{k=1}^{jml \text{ kata}} \sum_{t=1}^{T-1} \xi_{ij}^k(t)}{\sum_{k=1}^{jml \text{ kata}} \sum_{t=1}^{T-1} \gamma_i^k(t)}$$

$$\hat{b}_i(O_t) \left\{ \begin{array}{l} \hat{\mu}_i = \frac{\sum_{k=1}^{jml \text{ kata}} \sum_{t=1}^T \gamma_i^k(t) \cdot O_t^k}{\sum_{t=1}^T \gamma_i(t)} \\ \hat{\sigma}_i = \frac{\sum_{k=1}^{jml \text{ kata}} \sum_{t=1}^T \gamma_i^k(t) \cdot (O_t^k - \mu_i^k)(O_t^k - \mu_i^k)}{\sum_{k=1}^{jml \text{ kata}} \sum_{t=1}^T \gamma_i^k(t)} \end{array} \right.$$

4. Proses di atas dilakukan hingga didapat nilai yang dianggap cukup baik.

**Pengujian**

Pengujian dilakukan dengan membandingkan hasil kata yang diberikan oleh *HMM* dengan kata yang dimasukkan sebenarnya.

Persentase tingkat akurasi dihitung dengan fungsi berikut:  $hasil = \frac{\sum \text{kata yang benar}}{\sum \text{kata yang diuji}} \times 100\%$ .

**Lingkungan Pengembangan**

Pada pengembangan sistem pengenalan suara ini digunakan perangkat keras dan perangkat lunak dengan spesifikasi berikut :

a) *Perangkat Keras*

- Processor Intel(R) Pentium(R) 4 CPU 2,40GHz
- Memori DDR 768 MB
- Harddisk 160 GB
- Microphone
- Monitor
- Keyboard dan Mouse

b) *Perangkat Lunak*

- Sistem Operasi Windows XP Professional
- Matlab 7.0.1

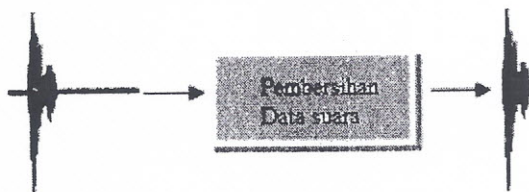
**HASIL DAN PEMBAHASAN**

**Preprocessing Data Suara**

Data suara yang telah berhasil direkam pada *Frekuensi Sampel (Fs)* 11 KHz selama 5 detik untuk setiap kata merupakan data suara kotor. Hal ini dikarenakan data tersebut tidak hanya mengandung sebuah kata, namun terdapat pula jeda waktu pada awal dan akhir pengucapan kata, seperti yang terlihat pada **Gambar 4** sebelumnya.

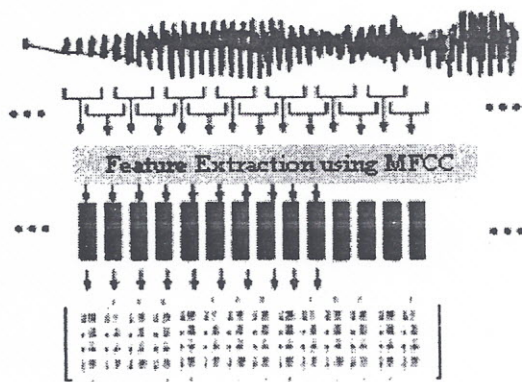
Pada tahap *preprocessing*, data suara dibersihkan dari jeda pada awal dan akhir pengucapan kata tersebut, sehingga dihasilkan data suara yang hanya mengandung sebuah kata dan memiliki dimensi yang jauh lebih kecil (*Gambar 4*). Di sisi lain, apabila data suara mengandung *noise* yang cukup besar, bisa jadi

hasil pembersihan kata masih mengandung jeda di awal dan akhir kata. Oleh sebab itu, pada saat perekaman suara, keadaan sekitar harus hening, agar tahap *preprocessing* dapat berjalan sesuai harapan.



Gambar 4 Pembersihan data suara.

Data suara yang telah dibersihkan, selanjutnya dianalisis atau diekstrak ciri-cirinya. Proses ini dilakukan dengan menggunakan *toolbox* yang telah tersedia, yaitu : *Auditory Toolbox* milik Slaney (1998). Seperti yang telah dijelaskan sebelumnya, pada penelitian ini *frame* yang digunakan sebesar 30 ms, dimana terjadi *overlap* pada setiap *frame* sebanyak 50 %, sedangkan *Cepstral Coeficients*-nya berjumlah 13 untuk setiap *frame*. Proses di atas menghasilkan matriks ( $T \times 13$ ) untuk setiap kata, dimana 13 merupakan jumlah koefisien ciri dan  $T$  merupakan jumlah vektor observasi setiap kata.



Gambar 5 MFCC data suara.

**Baum-Welch Training**

Data yang telah mengalami *preprocessing* selanjutnya dibagi ke dalam 2 bagian, yaitu : data latih (*Training*) dan data uji (*Testing*). Masing – masing memiliki jumlah yang sama, yaitu 20 data untuk setiap katanya. Data latih digunakan untuk melatih seluruh *HMM*, mulai dari *HMM* dengan 3 *Hidden State* hingga *HMM* dengan 8 *Hidden State*.

Pada proses pelatihan ini, Metode *Baum-Welch* yang digunakan merupakan Metode *Baum-Welch* untuk *Multiple Observation Sequences*. Proses perhitungannya sendiri telah dijelaskan sebelumnya dalam metode penelitian. Sebelum data dilatih dengan metode *Baum-Welch*, nilai inisialisasi *HMM* ditentukan dengan pengelompokkan *K-Means*.

Model yang telah dilatih selama 5, 10, dan 50 epoh, selanjutnya diuji menggunakan data uji. Hasil pengujiannya diukur melalui tingkat akurasi, sesuai formulasi yang telah dijelaskan sebelumnya. Berikut ini merupakan penjelasan mengenai hasil pengujian yang telah dilakukan dengan data uji :

**1. Pelatihan menggunakan 3 Hidden State**

*HMM* menggunakan 3 *Hidden State* menunjukkan kinerja yang sangat baik dibandingkan *HMM* lainnya, seperti yang ditunjukkan oleh Tabel 2 di bawah ini.

Tabel 2 Hasil Pengujian HMM dengan 3 Hidden State

Jumlah Epoh	Tingkat Akurasi		Rataan
	2 Suku Kata	3 Suku Kata	
5	81.5%	84.5%	83%
10	82.75%	83.5%	83.125%
50	81.75%	82.25%	82%

Hal ini terlihat dari tingkat akurasi pengujian yang keseluruhannya di atas 80%. Hasil terbaik dicapai oleh pelatihan 3 suku kata selama 5 epoh, yaitu 84.5%. Beberapa kata juga berhasil dikenali hingga 100%, di antaranya kata: Citra, Ilmu, Lulus, Yakin, Institut, dan Komputer.

Kecenderungan pengaruh epoh tidak terlihat pada hasil di atas, khususnya pada pelatihan *HMM* untuk 2 suku kata. Rataan tingkat akurasi terbaik didapat saat pelatihan selama 10 epoh, yaitu : 83.125%. Di lain pihak, pelatihan dengan 5 epoh mencapai rata-rata 83%, dimana waktu yang dihabiskan jauh lebih sedikit dibanding dengan pelatihan selama 10 epoh. Oleh sebab itu, dapat disimpulkan bahwa jumlah epoh tidak mempengaruhi hasil pelatihan *HMM* dengan 3 *hidden state*.

**2. Pelatihan menggunakan 4 Hidden State**

Secara umum persentase hasil pengujian terhadap data uji dengan *HMM* yang memiliki 4 *Hidden state* ditunjukkan pada Tabel 3.

Tabel 3 Hasil Pengujian HMM dengan 4 Hidden State

Jumlah Epoh	Tingkat Akurasi		Rataan
	2 Suku Kata	3 Suku Kata	
5	80%	83.25%	81.625%
10	80%	82.25%	81.125%
50	79.5%	83.25%	81.375%

Berdasarkan tabel di atas, terlihat bahwa pelatihan *HMM* dengan 5 epoh menunjukkan kinerja

yang terbaik dibandingkan yang lainnya. Selain rataan persentase yang lebih tinggi, pelatihan dengan 5 epoh ini tidak menghabiskan waktu yang lama seperti pada pelatihan 10 epoh dan 50 epoh, sehingga kinerja model tersebut dinilai lebih efektif dibanding pelatihan dengan epoh yang lebih tinggi.

Melalui tabel di atas juga terlihat bahwa jumlah epoh tidak terlalu berpengaruh terhadap kinerja akhir *HMM* dengan 4 *hidden state*. Hal ini ditunjukkan oleh kecilnya perbedaan tingkat akurasi yang dihasilkan oleh masing-masing jumlah epoh. Secara umum, perbedaan tingkat akurasi pada 5, 10 dan 50 epoh tidak lebih dari 0.5%. Selain itu, tidak terdapat kecenderungan peningkatan tingkat akurasi pada setiap peningkatan jumlah epoh.

Kata yang memiliki jumlah suku kata 3 juga cenderung lebih mudah dikenal, terlihat dari tingkat akurasi yang lebih tinggi dibanding tingkat akurasi pengujian 2 suku kata. Dengan kata lain pengujian terhadap kata dengan jumlah suku kata 3 selama 5 epoh menunjukkan hasil yang terbaik dengan persentase mencapai 83,25 %.

**3. Pelatihan menggunakan 5 Hidden State**

Pada pelatihan *HMM* dengan 5 *hidden state* terdapat penurunan tingkat akurasi yang cukup signifikan dibandingkan dengan model-*HMM* sebelumnya. Rataan tingkat akurasinya tidak satupun yang berhasil mencapai 80%. Rataan tingkat akurasi tertinggi didapat oleh *HMM* dengan pelatihan selama 5 epoh sebesar 79.875%.

**Tabel 4** Hasil Pengujian *HMM* dengan 5 *Hidden State*

Jumlah Epoh	Tingkat Akurasi		Rataan
	2 Suku Kata	3 Suku Kata	
5	78.25%	81.5%	79.875%
10	77.5%	81.5%	79.5%
50	77.75%	79.75%	78.75%

Bila mengacu pada rataan tingkat akurasi, banyaknya jumlah epoh cukup mempengaruhi hasil pengenalan kata, walaupun nilainya tidak terlalu signifikan. Epoh yang semakin banyak justru memperburuk hasil pengenalan kata. Selain itu, waktu yang dibutuhkan untuk pelatihan juga semakin lama, sehingga pada penelitian ini *HMM* dengan 5 *hidden state* dinilai tidak efektif dalam mengenali kata.

**4. Pelatihan menggunakan 6 Hidden State**

Sama seperti pelatihan *HMM* dengan 5 *hidden state*, pelatihan *HMM* dengan 6 *hidden state* juga menghasilkan tingkat akurasi yang kurang baik.

Bahkan bila kita membandingkan antara **Tabel 4** dan **tabel 5**, terlihat jelas bahwa *HMM* dengan 6 *hidden state* lebih buruk dibandingkan *HMM* dengan 5 *hidden state*. Pada *HMM* dengan 5 *hidden state* rataan tingkat akurasi tertinggi, yaitu: 79.875%, dicapai saat jumlah epoh 5, sedangkan pada *HMM* dengan 6 *hidden state* rataan tingkat akurasi tersebut baru dapat dicapai saat epoh telah mencapai 10.

**Tabel 5** Hasil Pengujian *HMM* dengan 6 *Hidden State*

Jumlah Epoh	Tingkat Akurasi		Rataan
	2 Suku Kata	3 Suku Kata	
5	77.5%	80%	78.75%
10	78.25%	81.5%	79.875%
50	75%	79.5%	77.25%

Berbeda dengan *HMM* dengan 5 *hidden state*, model ini tidak menunjukkan kecenderungan pengaruh jumlah epoh terhadap hasil pengenalan kata. Seperti yang terlihat pada **Tabel 5** di bawah, kata bersuku kata 2 maupun 3 dikenali dengan baik oleh *HMM* ini saat epoh mencapai 10.

**5. Pelatihan menggunakan 7 Hidden State**

Pelatihan *HMM* menggunakan 7 *hidden state* untuk kata bersuku kata 3 menghasilkan tingkat akurasi yang sedikit lebih baik dari *HMM* menggunakan 5 atau 6 *hidden state*. Namun secara umum, model ini pun tidak mampu menghasilkan tingkat akurasi yang optimal dibanding *HMM* lainnya. Tingkat akurasi maksimum didapat saat pelatihan mencapai 5 epoh, yaitu : 78.5%.

**Tabel 6** Hasil Pengujian *HMM* dengan 7 *Hidden State*

Jumlah Epoh	Tingkat Akurasi		Rataan
	2 Suku Kata	3 Suku Kata	
5	75.25%	81.75%	78.5%
10	75.5%	81%	78.25%

Pada **tabel 6** di atas terlihat bahwa pelatihan hanya dilakukan hingga 10 kali epoh, sedangkan pelatihan hingga 50 epoh tidak dilakukan. Hal ini, dikarenakan waktu yang dibutuhkan untuk melatih model ini sangatlah lama, sehingga pelatihan hanya dilakukan hingga 10 kali epoh. Selain itu, pada pelatihan *HMM* yang sebelumnya terlihat bahwa peningkatan jumlah epoh tidak meningkatkan persentase tingkat akurasi.

Pada tabel di atas juga terlihat bahwa

peningkatan jumlah epoch pada pelatihan *HMM* untuk 3 suku kata menurunkan tingkat akurasi pengenalan kata. Di lain pihak, pelatihan *HMM* untuk 2 suku kata mampu meningkatkan tingkat akurasi pengenalan kata, walaupun jaraknya tidak cukup signifikan.

**6. Pelatihan menggunakan 8 Hidden State**

Sama seperti pelatihan *HMM* menggunakan 7 *hidden state*, pelatihan *HMM* dengan 8 *hidden state* juga hanya dilakukan hingga 10 kali epoch. Hal ini, dikarenakan waktu yang dibutuhkan untuk melatih model ini jauh lebih lama dibanding pelatihan *HMM* sebelumnya.

Tabel 7 di bawah ini juga menunjukkan hasil yang tidak jauh berbeda dengan *HMM* menggunakan 7 *hidden state*. Selain tidak adanya kecenderungan pengaruh perubahan jumlah epoch terhadap tingkat akurasi pengenalan kata, model ini juga menghasilkan rata-rata tingkat akurasi terendah dibanding *HMM* sebelumnya. Tingkat akurasi pengenalan kata yang dihasilkan oleh pelatihan selama 5 ataupun 10 kali epoch menghasilkan persentase sebesar 77.125%.

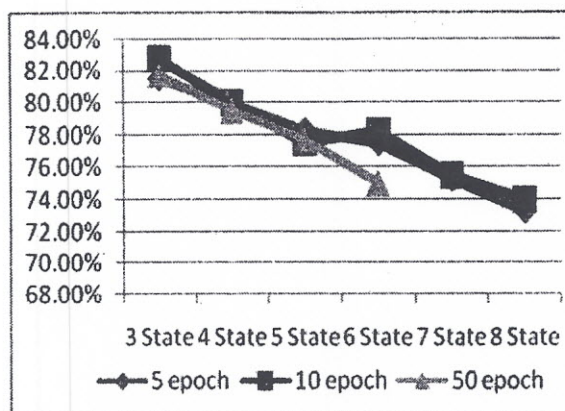
Tabel 7 Hasil Pengujian *HMM* dengan 8 *Hidden State*

Jumlah Epoch	Tingkat Akurasi		Rataan
	2 Suku Kata	3 Suku Kata	
5	73.25%	81%	77.125%
10	74%	80.25%	77.125%

**Hasil Pengenalan Kata**

Dari penjelasan-penjelasan sebelumnya dan dengan melihat hasilnya secara umum pada Gambar 6, hasil pengenalan kata bersuku kata 2 tidak dipengaruhi oleh banyaknya jumlah epoch yang dilakukan. Perbedaan hasil pengenalan kata ini lebih dipengaruhi oleh banyaknya jumlah state. Secara nyata terlihat bahwa penambahan jumlah state mampu menurunkan presentase hasil pengenalan kata. Tingkat akurasi terbaik untuk pengenalan kata bersuku kata 2 ialah 82.75%, dimana model tersebut menggunakan 3 *hidden state* dan dilatih hingga 10 kali epoch.

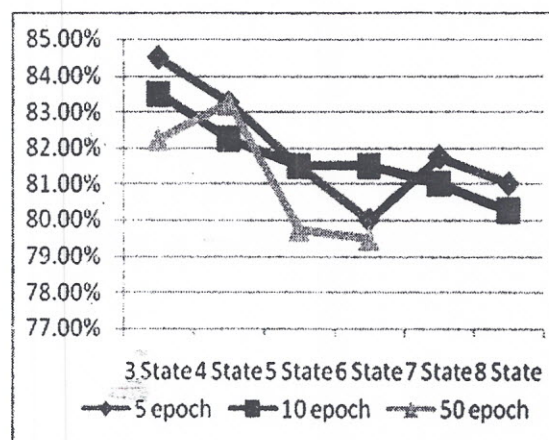
Serupa dengan hasil yang didapat oleh pengenalan kata bersuku kata 2, pengenalan kata bersuku kata 3 juga lebih banyak dipengaruhi oleh banyaknya jumlah state. Hasil pengenalan kata terbaik didapat oleh *HMM* menggunakan 3 *hidden state* dengan pelatihan selama 5 epoch, yaitu: 84.5%. Pada Gambar 7, terlihat kecenderungan penurunan tingkat akurasi terkait dengan penambahan jumlah



Gambar 6 Grafik Hasil Pengenalan Kata Bersuku Kata 2.

state. Pada beberapa titik peningkatan hasil pengenalan kata terjadi dengan cukup signifikan, namun tidak cukup mengubah kecenderungan penurunan hasil pengenalan kata. Grafik di bawah juga semakin menegaskan tidak adanya pengaruh jumlah epoch terhadap hasil pengenalan kata.

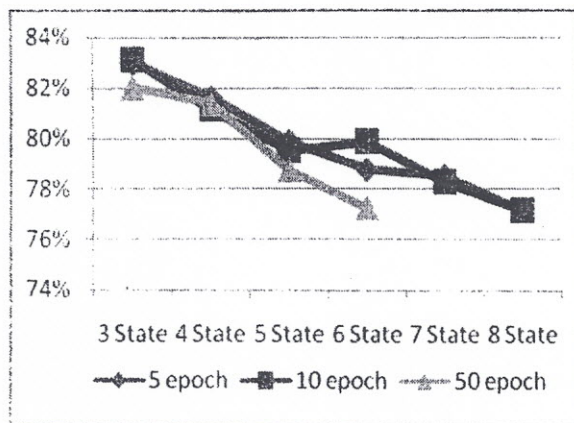
Pada penelitian ini, Hasil terbaik didapat oleh *HMM* menggunakan 3 *hidden state* dengan pelatihan selama 10 kali epoch sebesar 83.125%, Penelitian sebelumnya yang dilakukan oleh Yani (2005) dihasilkan *HMM* terbaik dengan tingkat akurasi 70.56%. Selain itu, penelitian sebelumnya hanya menggunakan 20 kata, sedangkan penelitian ini menggunakan hingga 2 kali lipatnya, yaitu 40 kata. Dengan kata lain, pelatihan *HMM* berbasis suara menggunakan algoritme *Baum-Welch* dinilai lebih baik dibanding pelatihan menggunakan pelatihan *Viterbi*. Selain itu, ia juga dapat meningkatkan tingkat akurasi pengenalan kata.



Gambar 7 Grafik Hasil Pengenalan Kata Bersuku Kata 3.

Grafik pada Gambar 8 memperlihatkan kecenderungan bahwa peningkatan jumlah state

secara umum menurunkan hasil pengenalan kata. Hal ini bisa saja disebabkan oleh terlalu banyaknya jumlah state yang dapat mengurangi perbedaan atau *variance* antar state itu sendiri. Pada saat proses pengenalan kata, sistem tidak mampu membedakan setiap observasi yang masuk dan memberikan bobot yang serupa untuk setiap observasi yang masuk, karena perbedaan antar state itu sendiri tidak terlalu jelas. Hal ini menyebabkan sebuah kata dapat dianggap sebagai kata lainnya walau sebenarnya keduanya sangat berbeda, karena bobot yang diberikan hampir sama.



Gambar 8 Grafik Hasil Pengenalan Seluruh Kata.

## KESIMPULAN DAN SARAN

### Kesimpulan

Dari penelitian ini dihasilkan beberapa *HMM* yang merepresentasikan sinyal suara yang masuk menjadi sebuah kata berbahasa Indonesia. *HMM* terbaik untuk pengenalan kata bersuku kata 2 ialah *HMM* menggunakan 3 *hidden state* dan dilatih hingga 10 kali epoch. Tingkat akurasi tertinggi yang didapat untuk pengenalan kata bersuku kata 2 ialah sebesar 82.75% dan tingkat akurasi terendahnya sebesar 73.25%. Demikian halnya dengan *HMM* untuk kata bersuku kata 3, tingkat akurasi terbaik juga didapat oleh *HMM* menggunakan 3 *hidden state*, namun pelatihan yang dilakukan cukup dengan 5 kali epoch. Pengenalan kata untuk kata bersuku kata 3 secara umum lebih baik dari pada pengenalan kata bersuku kata 2. Hal ini terlihat dari tingkat akurasi yang didapat keduanya, dimana tingkat akurasi terbaik untuk kata bersuku kata 3 mencapai 84.50%, sedangkan tingkat akurasi terendahnya sebesar 79.50%.

Secara umum *HMM* terbaik yang dihasilkan ialah *HMM* menggunakan 3 *hidden state* yang telah dilatih selama 10 epoch, dimana tingkat akurasinya mencapai 83.125%. Di sisi lain, *HMM* yang menggunakan 8

*hidden state* menghasilkan rata-rata tingkat akurasi terburuk, yaitu : 77.125%.

Berdasarkan hasil yang telah dijabarkan sebelumnya, terlihat bahwa jumlah epoch pelatihan tidak mempengaruhi tingkat akurasi pengenalan kata. Begitupun jumlah suku kata tidak mempengaruhi jumlah state yang harus digunakan.

### Saran

Pada dasarnya, penelitian ini masih sangat memungkinkan untuk dikembangkan lebih lanjut. Pembatasan *noise* dan jumlah kata yang digunakan pada penelitian ini membuat sistem yang dihasilkan belum memungkinkan untuk langsung digunakan dalam kondisi nyata. Selain itu, penelitian ini belum membahas lebih lanjut mengenai pengaruh *preprocessing* data suara terhadap hasil pengenalan kata, khususnya pengaruh jumlah *Cepstral Coefficients* sebagai hasil *MFCC*.

## DAFTAR PUSTAKA

- Allen, J. F. 2007. An Overview of Speech Recognition. [www.cs.rochester.edu/u/james/CSC248/Lec12.pdf](http://www.cs.rochester.edu/u/james/CSC248/Lec12.pdf). [2 Oktober 2007]
- Al-Akaidi, M. 2007. Fractal Speech Processing. Cambridge University Press.
- Dugad, R. dan Desai, U. B. 1996. A Tutorial in Hidden Markov Models. Indian Institute of Technology, India.
- Do, MN. 1994. DSP Mini-Project: An Automatic Speaker Recognition System. [http://www.ifp.uiuc.edu/~minhdo/teaching/speaker\\_recognition.doc](http://www.ifp.uiuc.edu/~minhdo/teaching/speaker_recognition.doc). [30 Mei 2007]
- Ganchev, T. D. 2005. Speaker Recognition. University of Patras, Greece.
- Jackson, P. 2004. HMM Tutorial 4. [http://www.ee.surrey.ac.uk/Personal/P.Jackson/tutorial/hmm\\_tut4.pdf](http://www.ee.surrey.ac.uk/Personal/P.Jackson/tutorial/hmm_tut4.pdf). [5 Mei 2007]
- Jelinek, F. 1995. Training and Search Methods for Speech Recognition. Proc. Natl. Acad. Sci. USA, Vol. 92, pp. 9964-9969, October 1995.
- Jurafsky, D. dan Martin, J. H. 2007. An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, Second Edition. [2 Oktober 2007]
- Rabiner, L. R. 1989. A Tutorial in Hidden Markov Models and Selected Applications in Speech

- Recognition. Proc. IEEE, vol. 77, pp. 257-287, February 1989.
- Shu, H., et al., 2003. Baum-Welch Training for Segment-Based Speech Recognition. Massachusetts Institute of Technology, USA.*
- Yani, M. 2005. Pengembangan model markov tersembunyi untuk pengenalan kata berbahasa Indonesia. Skripsi. Departemen Ilmu Komputer, FMIPA, Institut Pertanian Bogor.*
- Young, S., et al., 2001. HTK Book. Cambridge University Engineering Department.*
- Zue, V., et al., 2007. Speech Recognition. <http://cslu.cse.ogi.edu/HLTsurvey/ch1/node4.html>. [30 Mei 2007]*