

## SIMULASI DAN ANALISIS PENERJEMAH BAHASA ISYARAT KE TEKS MENGUNAKAN METODE KLASIFIKASI *SUPPORT VECTOR MACHINE*

### *SIMULATION AND ANALYSIS OF SIGN LANGUAGE TRANSLATOR TO TEXT USING SUPPORT VECTOR MACHINE CLASSIFICATION METHOD*

Hendra Priyana Mirantika<sup>1</sup>, Ratri Dwi Atmaja<sup>2</sup>, I Nyoman Apraz Ramatryana<sup>3</sup>

<sup>1</sup>Program Studi Teknik Telekomunikasi, Fakultas Teknik Elektro, Universitas Telkom

<sup>2</sup>Program Studi Teknik Telekomunikasi, Fakultas Teknik Elektro, Universitas Telkom

<sup>3</sup>Program Studi Teknik Telekomunikasi, Fakultas Teknik Elektro, Universitas Telkom

<sup>1</sup>hendrapriyanamirantika99@gmail.com, <sup>2</sup>ratriatmaja@telkomuniversity.ac.id, <sup>3</sup>ramatryana@gmail.com

#### Abstrak

Penelitian ini membahas tentang penerjemah bahasa isyarat ke teks yang mengacu ke SIBI (Sistem Isyarat Bahasa Indonesia) dengan input video. Pada penelitian ini digunakan metode *invariant moment* untuk ekstraksi ciri dan *Support Vector Machine* sebagai *classifier*. Proses yg dilakukan pada penelitian ini berupa *tracking* objek tangan dari peraga isyarat berdasarkan warna kulit yang disegmentasi menggunakan *skin detection* dalam ruang warna YCbCr. Setelah itu dilakukan *cropping* objek tangan kemudian dikonversi ke *grayscale* dan dicari nilai vektor *moment* menggunakan *invariant moment*. Selanjutnya 7 vektor *moment* yang didapat, dilatih untuk mendapatkan data latih lalu diklasifikasikan dengan *Support Vector Machine* (SVM).

Pengujian dari 17 kata mendapatkan akurasi sebesar 80,63%. Pengujian dilakukan dengan mengubah parameter *kernel* pada SVM dan hasil terbaik didapatkan dengan menggunakan *kernel RBF* dan input *classifier* menggunakan ciri dari 7 vektor *moment*.

Kata kunci : Bahasa isyarat, *skin detection*, *Invariant Moments*, *Support Vector Machine*

#### Abstract

This study discusses the sign language translator to text that refers to the SIBI (Indonesian Sign Language System) with video input. In this experiment, *invariant moment* method used for feature extraction and *Support Vector Machine* as classifier. The process in this study are hands tracking based on skin color segmented using *skin detection* in the YCbCr color space. Then *cropping* the hand and converted to grayscale and compute moments vector value using *invariant moment*. Next, train the seven moments to obtain training data and then classified with *Support Vector Machine* (SVM).

Tests from 17 word got 80,63% accuracy rate. Testing is done by changing the kernel parameters on SVM and the best results obtained by using RBF kernel and input for classifier using the seven moments value.

Keyword : Sign Language, *skin detection*, *Invariant Moments*, *Support Vector Machine*

## 1. PENDAHULUAN

Penerjemah bahasa isyarat dibutuhkan agar para tunarungu bisa menyampaikan maksud dan tujuan mereka kepada orang-orang yang biasa berkomunikasi verbal, dimana banyak orang-orang normal yang belum paham setiap pola dari bahasa isyarat.

Dalam jurnal ini, dilakukan proses simulasi dan analisis penerjemah dari Sistem Isyarat Bahasa Indonesia (SIBI) ke dalam teks Bahasa Indonesia. Penerjemah yang dibuat, *input* didapat dari rekaman *video*, kemudian diolah dalam *software* agar mendapatkan *output* berupa teks.

Penelitian sebelumnya sudah pernah menggunakan metode *Hidden Markov Model* (HMM) dalam pembuatan sistem penerjemah bahasa isyarat dengan *input* video dan menghasilkan *output* suara, serta akurasi yang diperoleh sebesar 90% [1]. Selanjutnya, dilakukan penelitian dengan metode yang berbeda, yaitu metode *Linear Discriminant Analysis* (LDA) dengan tingkat akurasi 83,3% [2]. Pengembangan terakhir dari penelitian ini, digunakan metode ekstraksi ciri *Local Binary Pattern* dan HMM untuk klasifikasinya. Penelitian ini sudah mencapai akurasi yang baik, yaitu untuk 1 gerakan mencapai akurasi 76,7%, 2 gerakan mencapai akurasi 93,3% dan untuk lebih dari 2 gerakan mencapai akurasi 96,7% [3]. Perbedaan penelitian ini dengan penelitian sebelumnya adalah penelitian ini dibuat penerjemah yang bisa menerjemahkan per-huruf, dari file video yang setiap video terdiri dari satu kata. Metode Ekstraksi ciri yang digunakan dalam penelitian ini adalah *Moment Invariant*. karena

berdasarkan penelitian yang dilakukan [4], memberikan rata-rata akurasi diatas 90% sehingga memberikan performa yang andal dalam merepresentasikan ciri suatu objek khususnya dalam kasus *pattern recognition*. Sementara untuk metode klasifikasi menggunakan *Support Vector Machine*, karena berdasarkan penelitian yang dilakukan oleh Chamasemani, Fareshteh F dan Singh, Yashwant P bahwa *classifier SVM* khususnya dengan *Multiclass* memberikan performa dan akurasi yang lebih baik dibandingkan dengan *classifier* lain dalam penelitian tersebut [5].

## 2. DASAR TEORI dan METODOLOGI PERANCANGAN

### 2.1 Dasar teori

#### 2.1.1 Bahasa Isyarat

Bahasa isyarat pada dasarnya sama dengan bahasa verbal/lisan, namun bahasa isyarat disampaikan melalui gerakan-gerakan tubuh/tangan, sedangkan bahasa verbal/lisan disampaikan secara lisan menggunakan mulut. Dalam hal penggunaannya, bahasa isyarat tergolong lebih tidak praktis dan susah dimengerti dibandingkan dengan bahasa lisan.

#### 2.1.2 Citra Digital

Citra Digital merupakan suatu teknik pengolahan/manipulasi gambar atau citra yang direpresentasikan dalam bidang dua dimensi. Citra digital merupakan suatu matriks dimana posisi baris dan kolom merupakan penunjuk untuk posisi pada citra dan elemen matriksnya (biasa disebut piksels) merupakan tingkat keabuan pada titik tersebut.

#### 2.1.3 Invariant Moments

*Invariant moments* merupakan suatu metode ekstraksi ciri yang pertamakali diperkenalkan oleh Hu [4]. Metode ini memiliki kelebihan *Rotation Scale Translation (RST)-invariant* atau vektor ciri yang dihasilkan oleh metode ini memiliki kesamaan sebelum dan sesudah mengalami RST.

Jika ada sebuah citra dua dimensi dengan fungsi  $f(x,y)$ , maka secara kontinu persamaan momen dapat dirumuskan sebagai berikut :

$$M_{pq} = \int_{x_0}^{x_1} \int_{y_0}^{y_1} x^p y^q f(x,y) dx dy \quad (1)$$

Berdasarkan persamaan *moment* diatas tidak bersifat *invariant* terhadap RST. Sehingga jika menginginkan *invariant* terhadap RST, diperlukan persamaan baru yang dinamakan momen pusat (*Central Moment*).

$$\mu_{pq} = \int_{x_0}^{x_1} \int_{y_0}^{y_1} (x - \bar{x})^p (y - \bar{y})^q f(x,y) dx dy \quad (2)$$

dimana

$$\bar{x} = \frac{\sum_{x_0}^{x_1} x f(x,y) dx dy}{\sum_{x_0}^{x_1} \sum_{y_0}^{y_1} f(x,y) dx dy} \quad (3)$$

$\bar{x}, \bar{y}$  merupakan *centroid* dari suatu citra dengan fungsi  $f(x,y)$ . Sementara itu, agar fungsi atau persamaan diatas bisa tahan terhadap penskalaan, maka diperlukan normalisasi dengan persamaan sebagai berikut :

$$\bar{\mu}_{pq} = \frac{\mu_{pq}}{M_{00}} + 1 \quad (4)$$

Sehingga dari persamaan diatas, dibentuk 7 *moment* yang *invariant* terhadap RST, dengan persamaan sebagai berikut :

$$\begin{aligned} \bar{\mu}_1 &= \bar{\mu}_{10} + \bar{\mu}_{01} \\ \bar{\mu}_2 &= (\bar{\mu}_{20} - \bar{\mu}_{10}^2) + 4\bar{\mu}_{10}\bar{\mu}_{01} \\ \bar{\mu}_3 &= (\bar{\mu}_{30} - 3\bar{\mu}_{20}\bar{\mu}_{01}) + (3\bar{\mu}_{10}^2 - \bar{\mu}_{01}^2) \\ \bar{\mu}_4 &= (\bar{\mu}_{40} + \bar{\mu}_{04}) + (\bar{\mu}_{10} + \bar{\mu}_{01})^2 \\ \bar{\mu}_5 &= (\bar{\mu}_{40} - 3\bar{\mu}_{30}\bar{\mu}_{01})(\bar{\mu}_{10} + \bar{\mu}_{01}) + [(\bar{\mu}_{30} + \bar{\mu}_{03}) - 3(\bar{\mu}_{20} + \bar{\mu}_{02})] \\ &\quad + (3\bar{\mu}_{10} - \bar{\mu}_{01})(\bar{\mu}_{10} + \bar{\mu}_{01})[3(\bar{\mu}_{10} + \bar{\mu}_{01})^2 - (\bar{\mu}_{10} + \bar{\mu}_{01})] \\ \bar{\mu}_6 &= (\bar{\mu}_{40} - \bar{\mu}_{10})[(\bar{\mu}_{30} + \bar{\mu}_{03}) - (\bar{\mu}_{20} + \bar{\mu}_{02})] + 4\bar{\mu}_{10}(\bar{\mu}_{10} + \bar{\mu}_{01})(\bar{\mu}_{10} + \bar{\mu}_{01}) \\ \bar{\mu}_7 &= (3\bar{\mu}_{30} - \bar{\mu}_{03})(\bar{\mu}_{10} + \bar{\mu}_{01})[(\bar{\mu}_{30} + \bar{\mu}_{03}) - 3(\bar{\mu}_{20} + \bar{\mu}_{02})] \\ &\quad - (\bar{\mu}_{30} - 3\bar{\mu}_{20})(\bar{\mu}_{10} + \bar{\mu}_{01})[3(\bar{\mu}_{10} + \bar{\mu}_{01})^2 - (\bar{\mu}_{10} + \bar{\mu}_{01})] \end{aligned}$$

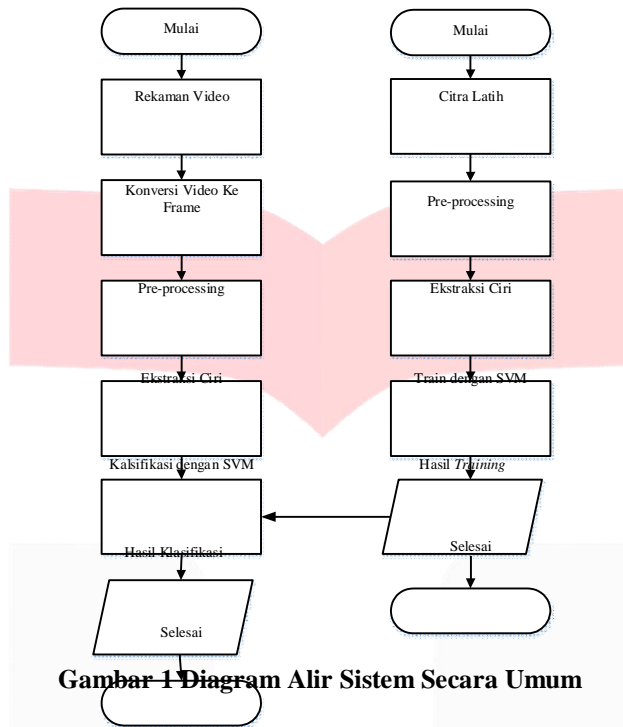
#### 2.1.4 Support Vector Machine (SVM) [6]

Konsep dasar SVM sebenarnya merupakan kombinasi harmonis dari teori-teori komputasi yang telah ada puluhan tahun sebelumnya, seperti *margin hyperplane* (Duda dan Hart tahun 1973, Cover tahun 1965, Vapnik tahun 1964, dan sebagainya.), kernel diperkenalkan oleh Aronszajn tahun 1950, dan demikian juga dengan konsep-konsep pendukung yang lain.

Konsep SVM dapat dijelaskan secara sederhana sebagai usaha mencari *hyperplane* terbaik yang berfungsi sebagai pemisah dua buah class pada *input space*. *Hyperplane* dalam ruang vektor berdimensi  $d$  adalah *affine subspace* berdimensi  $d-1$  yang membagi ruang vektor tersebut ke dalam dua bagian, yang masing-masing berkorespondensi pada kelas yang berbeda. Problem klasifikasi dapat diterjemahkan dengan usaha menemukan garis (*hyperplane*) yang memisahkan antara kedua kelompok tersebut.

**2.2 Metodologi Perancangan**

Gambaran sistem yang akan dibuat secara garis besar dibedakan menjadi dua bagian, yaitu bagian pelatihan (*training*) dan pengujian (*testing*). Diagram alir untuk proses pengujian dan pelatihan/training dapat dilihat pada gambar berikut :



**Gambar 1 Diagram Alir Sistem Secara Umum**

**2.2.1. Rekaman Video**

Rekaman video diperlukan sebagai input dalam penerjemah bahasa isyarat. Perekaman video menggunakan kamera webcam dengan framerate 15 fps, dan ukuran videonya adalah 320x240 pixels, berformat .mp4. Penelitian ini menggunakan 17 kata dimana setiap kata terdapat satu kata isyarat. Kata tersebut diantaranya adalah bandara, bandung, cari, cemas, dimana, fasih, ke, langsung, obat, pergi, pohon, qadar, surabaya, wayang, xilem, vital, dan saya.

Berikut ini contoh frame-frame dari file video isyarat:



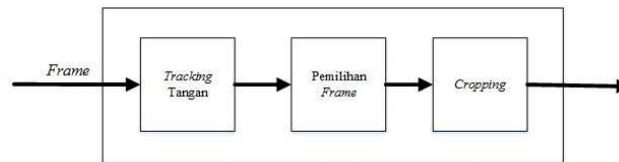
**Gambar 2 Contoh Bahasa Isyarat**

**2.2.2. Konvensi Video Ke Frame**

Pada tahapan ini, file video diekstrak menjadi frame-frame sesuai dengan jumlah frame dalam setiap video, karena ditahap berikutnya yang akan diproses/diolah adalah setiap frame dari masing-masing video.

**2.2.3. Pre-processing**

Pre-processing merupakan proses persiapan citra untuk diolah ketahap berikutnya yakni ekstraksi ciri. Berikut ini merupakan blok diagram pre-processing:



Gambar 3 Blok Diagram Pre-processing

### 1. Tracking Tangan

Tracking tangan diperlukan untuk penentuan *frame* isyarat agar memudahkan saat diolah. *Input* pada proses *tracking* tangan pada diagram alir diatas adalah *frame* yang sebelumnya sudah diekstrak. Hasil dari proses tersebut adalah informasi koordinat kolom dan baris dari objek tangan yang terdeteksi.

### 2. Pemilihan Frame

Pada tahap ini koordinat yang didapatkan dari proses *tracking* akan diolah untuk mendapatkan *range frame* isyarat serta menentukan satu *frame* untuk setiap *rangeny*.

### 3. Cropping

Setelah objek tangan berhasil dideteksi, proses selanjutnya adalah memotong bagian objek tersebut untuk diproses ditahap berikutnya. Objek dicrop berdasarkan luas bounding box pada objek tangan, sehingga ukuran untuk setiap pola isyarat akan berbeda-beda.

#### 2.2.4. Ekstraksi Ciri / Feature Extraction

Pada tahap ini, citra yang sudah diolah di *pre-processing* selanjutnya akan di cari *feature/ciri* menggunakan metode *Invariant Moments*. Metode momen invarian akan menghitung momen suatu citra, kemudian dihasilkan tujuh vektor momen yang akan digunakan sebagai input pada *classifier*.

#### 2.2.5. Classification / Klasifikasi

Tujuan dari klasifikasi adalah untuk menentukan kelas dari suatu objek yang diuji. Dalam penelitian ini, data latih terdiri dari 480 citra dengan 20 citra untuk masing-masing isyarat.

## 3. PEMBAHASAN

### 3.1 Data Uji dan Akurasi Sistem

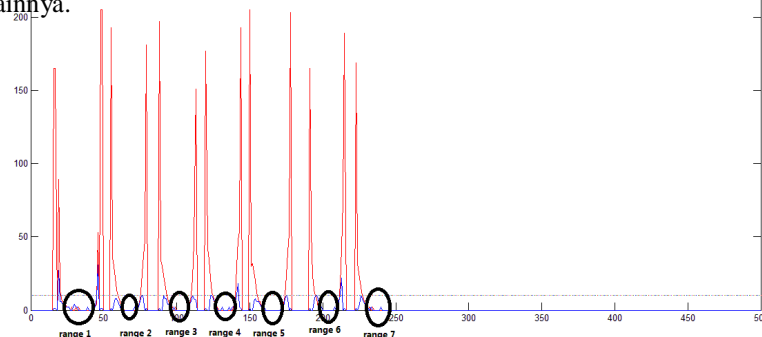
Data yang akan diuji ada sebanyak 17 kata. Akurasi sistem ditentukan dari setiap kata berdasarkan huruf yang benar dari kata tersebut. Kemudian diambil akurasi rata-rata dari setiap akurasi 17 kata.

$$\begin{aligned} \text{Akurasi} &= \frac{\text{Jumlah Huruf Benar}}{\text{Jumlah Huruf}} \\ &= \frac{\sum_{i=1}^{17} \text{Jumlah Huruf Benar}_i}{\text{Jumlah Huruf}_i} \end{aligned}$$

### 3.2 Analisis Pemilihan Frame dan Ekstraksi Ciri

#### 1. Pengaruh threshold selisih baris dan kolom serta counter pada pemilihan frame isyarat

Dalam video isyarat penelitian ini, tidak semua frame dibutuhkan untuk menunjukan suatu isyarat, melainkan hanya diambil frame-frame tertentu sesuai dengan banyaknya jumlah huruf yang ada dalam satu kata. Penentuan ini dapat dilakukan dengan melihat selisih baris dan kolom serta *counter(range)* untuk nilai yang konstan sampai ke gerakan isyarat lainnya.



Gambar 4 Grafik Selisih Baris dan Kolom

Gambar 4 merupakan selisih baris sedangkan biru merupakan selisih kolom. Frame isyarat yang dibutuhkan ada diantara setiap puncak berwarna merah dan biru(bagian lembah). Sehingga langkah awal adalah menentukan *range* isyarat, kemudian mencari selisih terkecil dalam *range* tersebut. Dari grafik kita bisa langsung menentukan kisaran *threshold* yang digunakan.

## 2. Pengaruh Nilai Vektor Momen Terhadap Akurasi Sistem

Pada pengujian penelitian ini, hanya untuk mengetahui pengaruh masing-masing nilai momen serta kombinasi momen terhadap akurasi sistem.

**Tabel 1 Akurasi setiap vektor ciri dari *moment invariant***

KATA	momen ke-1	momen ke-2	momen ke-3	momen ke-4	momen ke-5	momen ke-6	momen ke-7	7 momen
	Akurasi	Akurasi	Akurasi	Akurasi	Akurasi	Akurasi	Akurasi	Akurasi
bandung	28,57	28,57	14,29	14,29	42,86	28,57	28,57	<b>57,14</b>
bandara	57,14	57,14	42,86	42,86	57,14	57,14	57,14	<b>71,43</b>
cari	25,00	25,00	50,00	25,00	50,00	25,00	25,00	<b>50,00</b>
cemas	20,00	40,00	40,00	20,00	20,00	20,00	20,00	<b>80,00</b>
dimana	33,33	33,33	33,33	50,00	50,00	50,00	50,00	<b>83,33</b>
fasih	20,00	20,00	20,00	40,00	20,00	40,00	20,00	<b>60,00</b>
ke	0,00	0,00	0,00	0,00	0,00	0,00	0,00	<b>50,00</b>
langsung	12,50	12,50	12,50	12,50	25,00	12,50	12,50	<b>12,50</b>
obat	50,00	50,00	25,00	25,00	25,00	25,00	25,00	<b>100,00</b>
pergi	0,00	0,00	0,00	0,00	20,00	0,00	20,00	<b>60,00</b>
pohon	0,00	0,00	0,00	0,00	0,00	0,00	0,00	<b>80,00</b>
qadar	40,00	40,00	40,00	60,00	60,00	60,00	60,00	<b>80,00</b>
surabaya	50,00	50,00	37,50	37,50	37,50	37,50	37,50	<b>50,00</b>
wayang	33,33	33,33	33,33	33,33	50,00	33,33	33,33	<b>33,33</b>
xilem	0,00	0,00	0,00	0,00	0,00	0,00	0,00	<b>40,00</b>
vital	20,00	20,00	20,00	20,00	20,00	20,00	20,00	<b>80,00</b>
saya	50,00	50,00	50,00	50,00	50,00	50,00	50,00	<b>50,00</b>
	25,88	27,05	24,64	25,32	31,03	27,00	27,00	<b>61,04</b>

Pada Tabel 1 dapat diketahui bahwa akurasi terbaik adalah dengan menggabungkan ketujuh vektor ciri, karena jika menggunakan satu vektor ciri, untuk sejumlah 24 kelas, terdapat banyak kemiripan antar kelas. Hal tersebut membuat pengenalan menjadi tidak baik dan akurasi menjadi rendah. Untuk itu digunakan kombinasi 7 vektor momen agar setiap kelas memiliki nilai ciri yang berbeda dengan kelas lain, sehingga memungkinkan pengenalan bahasa isyarat menjadi lebih baik dan akurasi menjadi lebih tinggi dibanding menggunakan satu vektor ciri.

### 3.3 Analisis Support Vector Machine

1. Pengujian menggunakan metode SVM One Againsts All
  - a. Pengujian menggunakan Kernel Linear

Pada pengujian ini digunakan kernel *linier* dengan input 7 vektor moment invariant dan parameter C dari 1-40 dengan interval 10.

**Tabel 2 Akurasi menggunakan kernel *Linier***

Kata\C	SVM KERNEL LINEAR				
	c=1	c=10	c=20	c=30	c=40
<b>bandung</b>	57,142857	57,142857	57,142857	57,142857	57,142857
<b>bandara</b>	71,428571	64,285714	64,285714	64,285714	64,285714
<b>cari</b>	62,5	62,5	62,5	62,5	62,5
<b>cemas</b>	60	50	50	50	50
<b>dimana</b>	75	66,666667	66,666667	66,666667	66,666667
<b>fasih</b>	60	60	60	60	60
<b>ke</b>	25	25	25	25	25
<b>langsung</b>	43,75	43,75	37,5	37,5	37,5
<b>obat</b>	50	50	50	50	50

Tabel 2 Akurasi menggunakan kernel *Linier* (lanjutan)

Kata\C	SVM KERNEL LINEAR				
	c=1	c=10	c=20	c=30	c=40
pergi	50	50	50	50	50
pohon	60	60	60	60	60
qadar	80	80	80	80	70
surabaya	43,75	43,75	43,75	43,75	43,75
wayang	50	50	50	50	50
xilem	60	60	60	60	50
vital	40	40	40	40	40
saya	50	50	50	50	50
<b>Rata_rata Akurasi</b>	<b>55,21008</b>	53,711485	53,343838	53,343838	52,167367

Pada Tabel 2 dapat dilihat bahwa parameter C mempengaruhi besar akurasi. Untuk penggunaan C=1 memiliki akurasi yang lebih baik dari yang lain. Semakin besar nilai C membuat antisipasi terhadap kesalahan klasifikasi menjadi kurang baik, karena parameter C berfungsi untuk meminimalkan kesalahan klasifikasi yang disebabkan adanya *datatest* yang masuk dalam area *margin* dari *hyperplane*. Sehingga nilai akurasi terbaik untuk *kernel linear* ini adalah 55,21% dengan parameter C=1.

b. Pengujian menggunakan kernel Polynomial

Pada pengujian ini, sejumlah data uji diklasifikasikan menggunakan *Kernel Polynomial* dan parameter yang diubah adalah parameter p dan parameter C. Parameter p yang digunakan yaitu hasil akurasi dari nilai p=3, p=4, dan p=5

Tabel 3 Akurasi menggunakan kernel *Polynomial*

Kata\C	SVM KERNEL POLYNOMIAL				
	c=1	c=10	c=20	c=30	c=40
<b>Rata-rata Akurasi p=3</b>	<b>62,70308</b>	62,60504	62,60504	62,60504	62,60504
<b>Rata-rata Akurasi p=4</b>	60,763305	60,763305	60,763305	60,763305	60,763305
<b>Rata-rata Akurasi p=5</b>	<b>64,537815</b>	64,537815	64,537815	64,537815	64,537815

Berdasarkan Tabel 3 bahwa paramter p mempengaruhi hasil persamaan *hyperplane* untuk proses pemisahan antar kelas. Rentang nilai C dari 1-40 dengan interval 10 masih belum cukup memberikan dampak antisipasi kesalahan *error* pada klasifikasi baik itu dampak yang meminimalkan kesalahan atau memperbesar kesalahan klasifikasi. nilai parameter C mempengaruhi besar akurasi sistem, dengan semakin tinggi nilai C, maka pengontrolan *error* terhadap klasifikasi menjadi kurang baik. Berdasarkan ketiga tabel diatas, dapat dilihat bahwa akurasi terbaik untuk klasifikasi menggunakan *Kernel Polynomial* adalah dengan menggunakan parameter p=5 dan parameter C untuk segala nilai dari C=1 sampai C=40 dengan interval 10.

c. Pengujian menggunakan kernel RBF

Pada pengujian ini menggunakan kernel RBF (Radial Basis Function). Parameter yang diubah adalah parameter *rbf\_sigma* dengan nilai dari 1 sampai 3. Kemudian ada parameter C dengan rentang 1-40 dan interval 10.

Tabel 4 Akurasi menggunakan kernel *Polynomial*

Kata\C	SVM KERNEL RBF				
	c=1	c=10	c=20	c=30	c=40
<b>Rata-rata Akurasi sigma=1</b>	<b>80,63375</b>	73,228291	71,652661	72,440476	72,860644
<b>Rata-rata Akurasi sigma=2</b>	<b>80,23459</b>	74,887955	73,662465	71,918768	71,428571
<b>Rata-rata Akurasi sigma=3</b>	<b>80,23459</b>	74,887955	73,662465	71,918768	71,428571

Parameter *rbf\_sigma* mempengaruhi besar akurasi, yakni didapatkan akurasi optimal dari perubahan nilai *rbf\_sigma* adalah *rbf\_sigma*=1. Sementara semakin besar nilai *rbf\_sigma*, maka terlihat akurasi semakin menurun.

Paramater C yang digunakan untuk memperkecil klasifikasi error terlihat semakin tidak efektif/akurasi semakin kecil ketika nilai parameter C ditambah. Hal ini berlaku untuk hasil dari ketiga tabel diatas. Sehingga berdasarkan pengujian, untuk mendapatkan akurasi yang optimal dengan menggunakan kernel RBF, parameter *rbf\_sigma* diatur menjadi 1 dan parameter C juga diatur menjadi 1.



2. Pengujian menggunakan metode SVM One Againts All
  - a. Pengujian menggunakan kernel linear

**Tabel 5** Akurasi menggunakan kernel linear

Kata\C	SVM KERNEL LINEAR				
	c=1	c=10	c=20	c=30	c=40
bandung	64,285714	64,285714	64,285714	64,285714	64,285714
bandara	71,428571	71,428571	71,428571	71,428571	71,428571
cari	62,5	62,5	62,5	62,5	62,5
cemas	70	70	70	70	70
dimana	66,666667	66,666667	66,666667	66,666667	66,666667
fasih	80	80	80	80	80
ke	50	50	50	50	50
langsung	81,25	81,25	81,25	81,25	81,25
obat	62,5	62,5	62,5	62,5	62,5
pergi	80	80	80	80	80
pohon	90	90	90	90	90
qadar	80	80	80	80	80
surabaya	68,75	68,75	68,75	68,75	68,75
wayang	66,666667	66,666667	66,666667	66,666667	66,666667
xilem	80	80	80	80	80
vital	100	100	100	100	100
saya	62,5	62,5	62,5	62,5	62,5
<b>Rata-rata Akurasi</b>	<b>72,7381</b>	<b>72,738095</b>	<b>72,738095</b>	<b>72,738095</b>	<b>72,738095</b>

Parameter C tidak memberikan pengaruh terhadap akurasi dikarenakan range pemilihan terlalu kecil sehingga untuk pengurangan error tidak terlihat. Sehingga akurasi yang didapat dari penggunaan kernel linear adalah 72,74%.

- b. Pengujian menggunakan kernel polinomial

Pada pengujian ini, sejumlah data uji diklasifikasikan menggunakan *Kernel Polynomial* dan parameter yang diubah adalah parameter p dan parameter C. Parameter p yang digunakan yaitu hasil akurasi dari nilai p=3, p=4, dan p=5

**Tabel 6** Akurasi menggunakan kernel *Polynomial*

Kata\C	SVM KERNEL POLYNOMIAL				
	c=1	c=10	c=20	c=30	c=40
<b>Rata-rata Akurasi p=3</b>	75,62675	75,626751	75,626751	75,626751	75,626751
<b>Rata-rata Akurasi p=4</b>	77,510504	77,510504	77,510504	77,510504	77,510504
<b>Rata-rata Akurasi p=5</b>	68,785014	68,785014	68,785014	68,785014	68,785014

Parameter C tidak berpengaruh terhadap akurasi. Tidak ada data pencilan yang masuk kedalam *margin hyperplane* antar kelas. Sementara untuk parameter p memberikan pengaruh terhadap perubahan akurasi. Parameter p yang optimal pada penelitian ini adalah p=4, karena jika nilai p bertambah, maka akurasi juga akan semakin kecil. Sehingga akurasi yang didapatkan untuk kernel polinomial adalah sebesar 77,51% dengan parameter p=2 dan parameter C bisa disemua nilai dari rentang 1-40 dengan interval 10.

- c. Pengujian menggunakan kernel RBF

Pada pengujian ini menggunakan kernel RBF (Radial Basis Function). Parameter yang diubah adalah parameter rbf\_sigma dengan nilai dari 1 sampai 3. Kemudian ada parameter C dengan rentang 1-40 dan interval 10.

Tabel 7 Akurasi menggunakan kernel *Polynomial*

Kata\C	SVM KERNEL RBF				
	c=1	c=10	c=20	c=30	c=40
Rata-rata Akurasi sigma=1	79,15966	79,159664	79,159664	79,159664	79,159664
Rata-rata Akurasi sigma=2	77,12535	77,12535	77,12535	77,12535	77,12535
Rata-rata Akurasi sigma=3	71,841737	71,841737	71,841737	71,841737	71,841737

Klasifikasi menggunakan kernel RBF didapatkan hasil yang optimal adalah dengan menggunakan parameter  $rbf\_sigma=1$ . Sedangkan parameter C tidak memberikan pengaruh apapun disaat rangnya hanya dari 1-40. Dapat dilihat bahwa penambahan nilai  $rbf\_sigma$  berdampak pada menurunnya akurasi. Sehingga akurasi yang diperoleh adalah sebesar 79,16%.

#### 4. KESIMPULAN

##### a. Kesimpulan

Berdasarkan hasil analisis dari pengujian penerjemah bahasa isyarat ke teks, dapat ditarik beberapa kesimpulan antara lain adalah :

1. Pada penelitian ini, penggunaan vektor *moment invariant* yang paling efektif untuk akurasi yang tinggi adalah dengan memasukkan ketujuh vektor ciri kedalam pelatihan classifier dan akurasinya sebesar 61,04%.
2. Akurasi tertinggi didapatkan dengan menggunakan metode OAA dan menggunakan *kernel RBF* dengan parameter  $sigma=1$  dan  $C=1$ , dengan akurasi sebesar 80,63%.

##### a. Saran

Berikut merupakan saran dari penulis untuk memperbaiki kekurangan yang ada dalam penelitian ini dan mengembangkan penelitian ini agar berguna sesuai dengan tujuan dalam penelitian ini. Adapun saran yang diberikan adalah:

1. Untuk penelitian selanjutnya sebaiknya dilakukan pre-processing yang lebih baik untuk menambah akurasi sistem dan mempercepat waktu komputasi.
2. Untuk penelitian selanjutnya sebaiknya dibuat sistem yang realtime
3. Untuk penelitian selanjutnya sebaiknya diaplikasikan ke android
4. Untuk penelitian selanjutnya sebaiknya menggunakan ekstraksi ciri dan *classifier* yang lain untuk membandingkan ekstraksi ciri dan *classifier* pada penelitian sebelumnya

#### DAFTAR PUSTAKA

- [1]. Adisti, I. (2010). *Perancangan dan Implementasi Penerjemah Bahasa Isyarat Dari Video Menjadi Suara Menggunakan Ekstraksi Ciri dan Hidden Markov Model*. Bandung: Teknik Telekomunikasi, Institut Teknologi Telkom.
- [2]. Nasution, N. M. (2011). *Desain dan Implementasi Sistem Penerjemah Bahasa Isyarat Berbasis Webcam dengan Metode Linear Discriminant Analysis*. Bandung: Teknik Telekomunikasi, Institut Teknologi Telkom
- [3]. Najiburahman, M. (2015). *Simulasi dan Analisis Sistem Penerjemah Bahasa Sibi Menjadi Bahasa Indonesia Menggunakan Metode Klasifikasi Hidden Markov Model*. Bandung: Manajemen Bisnis Telekomunikasi & Informatika, Universitas Telkom
- [4]. Paschalakis. (1999). Pattern Recognition in Grey Level Images Using Moment Based Invariant Features. *Image Processing and its Applications*, 245-249.
- [5]. Chamasemani, F., & Singh, Y. (2011). Multi-class Support Vector Machine (SVM) Classifiers - An Application in Hypothyroid Detection and Classification. *Theories and Application*, 351-356.
- [6]. Sembiring, K. (2007). *Penerapan Teknik Support Vector Machine untuk Pendeteksian Intrusi pada Jaringan*. Bandung: ITB.