

ANALISIS DAN IMPLEMENTASI APLIKASI PENGENALAN GERAK BIBIR MENJADI TEKS MENGGUNAKAN METODE SUPPORT VECTOR MACHINE

ANALYSIS AND IMPLEMENTATION OF LIP MOVEMENT TRANSLATOR APPLICATION TO TEXT USING SUPPORT VECTOR MACHINE

Revi Febriana S.¹, Budhi Irawan, S.Si., MT.², Inung Wijayanto, ST., MT.³

^{1,2,3}Fakultas Teknik Elektro, Universitas Telkom

¹revifebrianas@gmail.com, ²bir@ittelkom.ac.id, ³iwijayanto@telkomuniversity.ac.id

Abstrak

Membaca ujaran merupakan suatu metode komunikasi secara verbal dalam memahami pembicaraan orang lain dengan melihat gerakan bibirnya. Dalam proses membaca ujaran diperlukan kunci untuk dapat mengenali kata yaitu huruf vokal. Biasanya membaca ujaran digunakan untuk berkomunikasi oleh penyandang tuna rungu. Hal ini dikarenakan kaum tuna rungu mempunyai keterbatasan dalam mendengar suara, sehingga memanfaatkan penglihatan untuk membaca gerakan bibir lawan bicara.

Pada penelitian ini dirancang sebuah aplikasi pengenalan gerak bibir menjadi teks. Secara umum, sistem terdiri dari tiga tahapan utama yaitu *preprocessing*, ekstraksi ciri, dan pengklasifikasian. Metode ekstraksi ciri yang digunakan adalah *Principal Component Analysis* (PCA). Kemudian, metode klasifikasi yang digunakan adalah metode *Support Vector Machine* (SVM).

Berdasarkan hasil pengujian yang telah dilakukan, sistem pengenalan gerak bibir dapat mengidentifikasi 5 kelas yang merepresentasikan suku kata. Hasil klasifikasi dengan akurasi tertinggi diperoleh pada kernel Gaussian dengan sigma 4000 dan dimensi 80x80 yaitu sebesar 69% untuk data yang diujikan *non realtime* dengan menggunakan 200 data latih dan 100 data uji.

Kata Kunci : ujaran, tuna rungu, *principal component analysis*, *support vector machine*

Abstract

Speechreading is a method of verbal communication to understand the speech of others by see the lip movement. In the process of speechreading, the key to understand lip movement are vowel. Usually speechreading used to communicate by the deaf. This is because the deaf have limitations in hearing the sound, so they take advantage of their sight to read the lip of the speaker.

In this research designed a lip translator applications into text. In general, the system consists of three main stages, namely preprocessing, feature extraction, and classification. Feature extraction method in this research using Principal Component Analysis (PCA). Then, the classification method using Support Vector Machine (SVM).

Based on the results of tests, lip translator application system can identifies 5 classes that represent syllables. The results of the classification with the highest accuracy is obtained on a Gaussian kernel sigma 4000 with dimensions 80x80 and this research resulted in 69% accuracy for data tested in non-realtime with 200 training data and 100 testing data.

Keywords: speech, hearing, *principal component analysis*, *support vector machine*

1. Pendahuluan

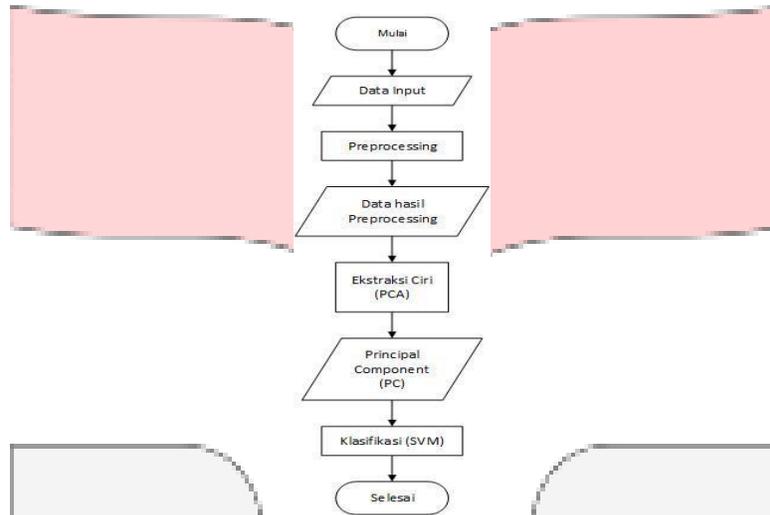
Membaca ujaran merupakan suatu metode komunikasi secara verbal dalam memahami pembicaraan orang lain dengan melihat gerakan bibirnya. Akan tetapi, hanya sekitar 50% bunyi ujaran yang dapat terlihat pada bibir. Sedangkan 50% lainnya diproses pada bagian mulut lainnya atau ada beberapa bunyi ujaran yang secara visual tampak mirip sehingga pembaca gerakan bibir tidak dapat memastikan bunyi apa yang dilihatnya. Dalam proses membaca ujaran diperlukan kunci untuk dapat mengenali kata yaitu huruf vokal. Biasanya membaca ujaran digunakan untuk berkomunikasi oleh penyandang tuna rungu. Hal ini dikarenakan kaum tuna rungu mempunyai keterbatasan dalam mendengar suara, sehingga kemampuan membaca ujaran sangat bermanfaat dalam kehidupan kaum tuna rungu.

Dari perihal di atas, maka pada penelitian ini dibahas tentang pembuatan aplikasi pengenalan gerak bibir menjadi teks. Secara umum, proses terdiri dari tiga tahapan utama yaitu preproses citra masukan, ekstraksi ciri, dan pengklasifikasian. Metode ekstraksi ciri yang digunakan adalah *Principal Component Analysis* (PCA). Kemudian,

metode klasifikasi yang digunakan adalah metode Support Vector Machine (SVM). Selain itu akan dilakukan pengujian beberapa jenis parameter SVM yang digunakan dan berpengaruh terhadap tingkat akurasi dan waktu komputasi dari masukan yang diberikan untuk kemudian dianalisa.

2. Perancangan Sistem

Pengolahan citra adalah istilah umum untuk berbagai teknik yang keberadaannya untuk memanipulasi dan memodifikasi citra dengan berbagai cara^[2]. Pengolahan citra dan pengenalan pola menjadi bagian dari proses pengenalan citra. Kedua aplikasi ini akan saling melengkapi untuk mendapatkan ciri khas dari suatu citra yang hendak dikenali^[6]. Secara umum rancangan sistem dalam penelitian ini digambarkan dalam *flowchart* berikut:

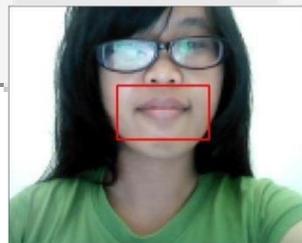


Gambar 2.1 *Flowchart* Sistem

Motion Identification terdiri dari tiga tahapan utama yaitu preproses citra masukan, ekstraksi ciri, dan pengklasifikasian^[6]. Ketika memulai sistem pengenalan gerak bibir, hal yang dilakukan pertama kali adalah sistem menerima input berupa video dari webcam.. Image hasil input akan melalui tahap *preprocessing*. Pada tahap ini terjadi beberapa proses, yaitu deteksi objek, *grayscale*, dan *thresholding*. Data hasil *preprocessing* selanjutnya akan dilakukan proses ekstraksi ciri menggunakan *Principal Component Analysis* (PCA) sehingga akan menghasilkan kumpulan vektor citra dalam bentuk *Principal Component* (PC) untuk kemudian diklasifikasi menggunakan *Support Vector Machine*.

2.1 Preprocessing

Pada penelitian ini, penulis menggunakan *library Haarcascade* yang disediakan EMGU CV. Di dalam *Haarcascade*, sudah terdapat data untuk mendeteksi bibir dan kemudian membandingkan dengan database library yang ada untuk selanjutnya mampu mendeteksi ada atau tidaknya objek dalam *frame*. Jika ada bibir terdeteksi, maka akan terbentuk *bounding rectangle* pada bibir yang terdeteksi untuk selanjutnya akan di segmentasi dan dilanjutkan ke proses berikutnya.



Gambar 2.2 Bibir terdeteksi dalam *bounding rectangle*

Pada proses *grayscale*, citra dari webcam akan dikonversi menjadi citra berskala keabuan (*grayscale*). Tujuan konversi citra RGB ke *grayscale* adalah untuk mempermudah komputasi citra ke proses berikutnya. Gambar dibawah ini merupakan implementasi konversi citra RGB ke citra *grayscale*.

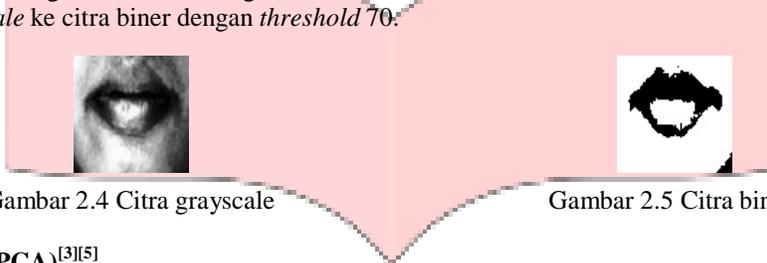


Gambar 2.3 Citra RGB



Gambar 2.4 Citra grayscale

Pada proses *thresholding*, citra grayscale dikonversi menjadi citra biner. Tujuan konversi citra *grayscale* menjadi citra biner adalah untuk memudahkan komputasi karena pada citra biner hanya mengenal nilai bit ‘1’ dan bit ‘0’. Untuk merubah citra *grayscale* menjadi citra biner, maka diperlukan suatu *threshold*. *Threshold* atau nilai ambang digunakan untuk menentukan suatu piksel pada citra *grayscale* termasuk piksel hitam atau piksel putih. Pada sistem pengenalan gerak bibir ini digunakan *threshold* 70. Gambar dibawah ini merupakan implementasi konversi citra *grayscale* ke citra biner dengan *threshold* 70.

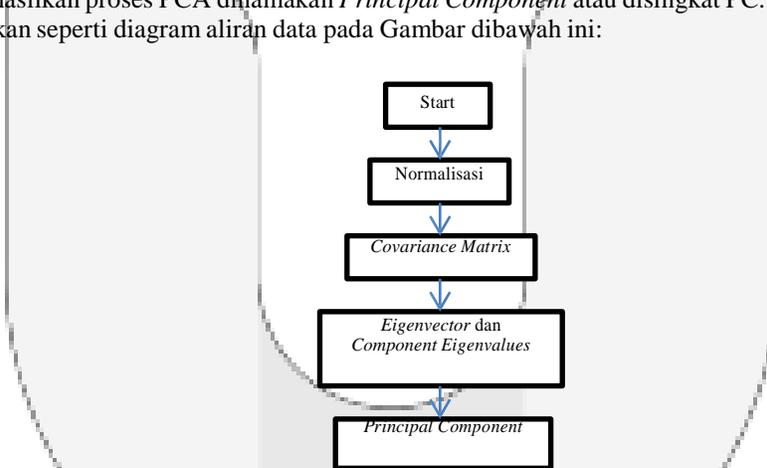


Gambar 2.4 Citra grayscale

Gambar 2.5 Citra biner

2.2 Ekstraksi Ciri (PCA)^{[3][5]}

PCA merupakan salah satu teknik reduksi untuk tipe data numerik. Fokus kerja PCA adalah ‘meringkas’ data, bukan mengelompokkan data seperti *clustering*. PCA adalah salah satu teknik statistika yang umum digunakan pada bidang pengenalan pola, pengenalan wajah, prediksi, kompresi data, dll. PCA mencari pola dan mengambil ciri-ciri penting dari data berdimensi tinggi dengan cara mereduksi data tersebut menjadi data berdimensi rendah. Sekumpulan dimensi baru yang dihasilkan proses PCA dinamakan *Principal Component* atau disingkat PC. Adapun langkah proses PCA dapat digambarkan seperti diagram aliran data pada Gambar dibawah ini:

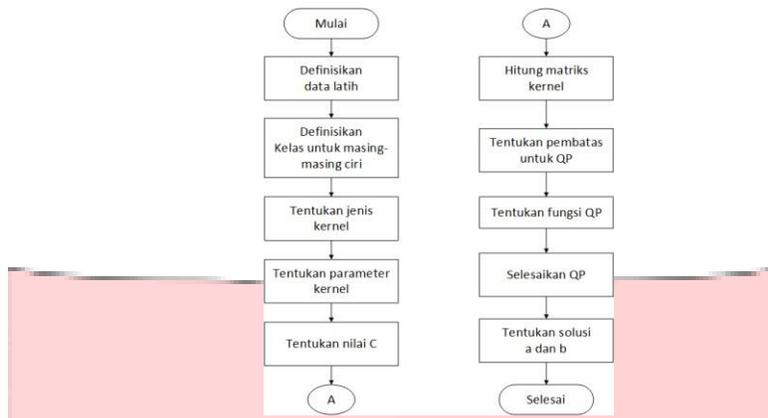


Gambar 2.6 Langkah-langkah pembentukan PCA

Pada tahap ini, metode yang digunakan untuk proses ekstraksi ciri adalah *Principal Component Analysis (PCA)*. Proses ekstraksi ciri bertujuan untuk mendapatkan informasi-informasi penting pada sebuah citra yang membedakan antar citra. Proses ekstraksi ciri dengan PCA menggunakan kumpulan citra yang memiliki dimensi yang sama, maka dilakukan penyeragaman ukuran pada citra menjadi 40 x 40, 60 x 60, dan 80 x 80 yang nantinya akan dianalisa pengaruh dimensi terhadap performansi sistem.

2.3 Klasifikasi dengan SVM

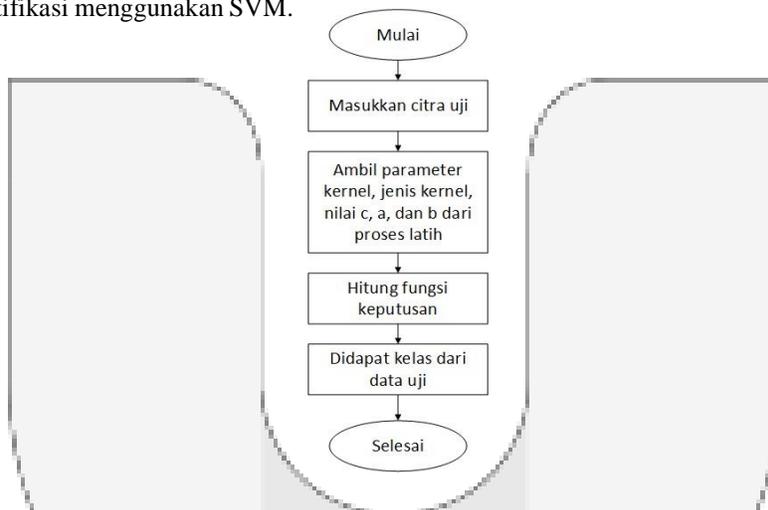
Proses latih pada klasifikasi SVM, digunakan untuk mencari nilai *w (weight)*, *b (bias)*, dan *support machine* dari masing-masing kelas. Gambar 2.7 akan menjelaskan bagaimana proses latih untuk klasifikasi 2 kelas pada SVM.



Gambar 2.7 Diagram Alir Proses Latih Klasifikasi SVM^{[1][4]}

Pada proses uji, parameter-parameter w (*weight*), b (*bias*), dan *support vector* yang didapatkan pada proses latih dan juga data uji diperlakukan sebagai masukan pada fungsi keputusan $f(x) = y_i a_i x_i + b^*$. Nilai *sign*

dari fungsi keputusan yang di dapat merupakan hasil kelas dari data yang diujikan. Gambar 2.8 akan menjelaskan proses uji untuk identifikasi menggunakan SVM.



Gambar 2.8 Diagram Alir Proses Uji Klasifikasi SVM^{[1][4]}

3. Pengujian Sistem dan Analisis

3.1 Pengujian Alpha

Tabel 3. 1 Pengujian Blackbox

Yang Diuji	Data Masukan	Hasil yang diharapkan	Hasil Keluaran	Kesimpulan
Deteksi Bibir	User berada didepan webcam	Menampilkan bibir dalam <i>bounding rectangle</i>	Bibir dalam <i>Bounding rectangle</i> ditampilkan	Diterima
Segmentasi Bibir	User berada didepan webcam	Menampilkan hasil segmentasi bibir	Bibir ditampilkan	Diterima
Mulai Pelatihan	User mulai mengucapkan suku kata	Dapat memulai perekaman <i>frame</i>	Memulai perekaman <i>frame</i>	Diterima
Berhenti Pelatihan	User berhenti mengucapkan suku kata	Dapat menyimpan perekaman <i>frame</i>	Menyimpan perekaman <i>frame</i>	Diterima
Ekstraksi Ciri	Tombol Train ditekan	Dapat menyimpan Data latih dalam bentuk .xls	Menyimpan Data latih dalam bentuk .xls	Diterima
Mulai Pengujian secara <i>real time</i>	User mulai mengucapkan suku kata	Dapat menyimpan perekaman <i>frame</i>	Menyimpan perekaman <i>frame</i>	Diterima
Berhenti Pengujian secara <i>real time</i>	User berhenti mengucapkan suku kata	Dapat mengenali suku kata yang diucapkan	Mengenali suku kata yang diucapkan	Diterima

Pengujian <i>non realtime</i>	Tombol Browse ditekan	Dapat menampilkan <i>frame</i> yang akan diuji	Menampilkan <i>frame</i> yang akan diuji	Diterima
Pengujian <i>non realtime</i>	Tombol Test ditekan	Dapat mengenali <i>frame</i> yang diuji	Mengenali <i>frame</i> yang diuji	Diterima
Pengujian Waktu Komputasi	Tombol Stop (Test), Tombol Test ditekan	Dapat menampilkan waktu komputasi untuk pengenalan	Menampilkan waktu komputasi untuk pengenalan	Diterima

Hasil pengujian *Alpha* yang telah dilakukan menunjukkan bahwa Aplikasi yang dibangun sudah memenuhi persyaratan fungsional. Secara fungsional, aplikasi sudah dapat menghasilkan keluaran seperti yang diharapkan.

3.2 Pengujian Beta

Berdasarkan hasil penilaian MOS, maka dengan menggunakan skala *likert* didapatkan hasil sebagai berikut:

1. Tingkat Kemudahan Aplikasi)))))/)
2. Tingkat Interface Aplikasi)))))/)
3. Tingkat Respon Sistem)))))/)

Dari nilai MOS yang diperoleh, dilihat bahwa tingkat kemudahan Aplikasi yaitu 81%, berarti responden merasa mudah dalam menggunakan aplikasi pengenalan gerak bibir. Kemudian untuk interface sudah baik, dapat dilihat pada nilai pertanyaan kedua yaitu 78%. Dan menurut responden respon aplikasi sudah baik, dapat dilihat pada nilai pertanyaan ketiga yaitu 76%.

3.3 Pengujian Jenis dan Sigma Kernel

Pada pengujian pengaruh jenis dan sigma kernel terhadap waktu komputasi menggunakan kernel *Gaussian*, *Laplacian*, dan *Invers Multiquadric* serta menggunakan sigma kernel 100, 1000, 4000, 7000, dan 10000 dengan data latih sebanyak 200 data dan data uji sebanyak 100 data.

Tabel 3.2 Tabel Pengaruh Jenis dan Sigma Kernel terhadap Akurasi dan Waktu Komputasi

Sigma	Laplacian		Gaussian		Invers Multiquadrik	
	Akurasi	Waktu	Akurasi	Waktu	Akurasi	Waktu
100	32%	7475,05 ms	32%	7553,42 ms	49%	7664,83 ms
1000	39%	7359,73 ms	32%	7522,31 ms	50%	7483,36 ms
4000	68%	7413,48 ms	69%	7606,13 ms	60%	7644,44 ms
7000	68%	7505,98 ms	68%	7571,63 ms	63%	7642,89 ms
10000	66%	7606 ms	67%	7506,78 ms	63%	7533,18 ms

Pada Tabel 3.2 terlihat akurasi tertinggi sebesar 69% diperoleh pada kernel *Gaussian* dengan besar sigma 4000. Parameter sigma tidak berbanding lurus terhadap tingkat akurasi yang dihasilkan dikarenakan ada suatu kondisi optimum yang akan menghasilkan akurasi maksimum dan menurunkan tingkat akurasi setelah parameter optimum tercapai. Berdasarkan hasil pengujian yang dilakukan terlihat bahwa jenis dan sigma kernel tidak berpengaruh terhadap waktu komputasi yang dihasilkan. Tabel menunjukkan bahwa rata-rata sistem pengenalan gerak bibir memerlukan waktu 7 detik untuk memproses pengklasifikasian.

3.4 Pengujian Dimensi

Pada pengujian pengaruh Dimensi terhadap tingkat akurasi dan waktu komputasi, menggunakan Dimensi 40x40 piksel, 60x60 piksel, dan 80x80 piksel dengan data latih sebanyak 200 data dan data uji sebanyak 100 data.

Tabel 3.3 Tabel Pengaruh Dimensi terhadap Akurasi dan Waktu Komputasi

Kelas Data Uji	40 x 40		60 x 60		80 x 80	
	Akurasi	Waktu	Akurasi	Waktu	Akurasi	Waktu
Ba	60%	1814,25 ms	65%	4028,75 ms	65%	7586,85 ms
Be	75%	1847,55 ms	75%	4034,35 ms	75%	7622,1 ms
Bi	60%	1837,45 ms	60%	4025,05 ms	65%	7587,55 ms
Bo	90%	1853,85 ms	90%	4018,85 ms	90%	7575,05 ms
Bu	50%	1835,6 ms	50%	4026,65 ms	50%	7486,6 ms
Rata-rata	67%	1837,74 ms	68%	4026,73 ms	69%	7571,63 ms

Pada Tabel 3.3 terlihat akurasi tertinggi sebesar 69% diperoleh pada dimensi 80x80. Hal ini dikarenakan pada dimensi 80x80 piksel, matriks penyusun citra tersebut lebih banyak sehingga ciri yang dihasilkan akan lebih detail daripada citra dengan dimensi 40x40 piksel dan 60x60 piksel. Sedangkan waktu komputasi tercepat diperoleh pada citra dengan dimensi 40x40. Dengan demikian, dimensi citra yang digunakan berbanding lurus terhadap waktu komputasi yang dihasilkan. Hal ini dikarenakan proses waktu yang komputasi akan menyesuaikan dengan banyaknya matriks penyusun suatu citra.

3.5 Pengujian Jumlah Frame

Pada pengujian pengaruh dimensi terhadap tingkat akurasi dan waktu komputasi, menggunakan Jumlah Frame 10, 20, dan 30 dengan data latih sebanyak 200 data dan data uji sebanyak 100 data.

Tabel 3.4 Tabel Pengaruh Dimensi terhadap Akurasi dan Waktu Komputasi

Jumlah Frame	Akurasi (%)	Waktu (ms)
10	44	8895.88
20	56	10068.08
30	56	11205.52

Pada Tabel 3.4 menunjukkan akurasi tertinggi sebesar 56% diperoleh dengan jumlah frame 20 dan 30. Pada saat jumlah frame yang digunakan adalah 20 frame, tingkat akurasi akan meningkat dari ketika menggunakan 10 frame. Hal ini dikarenakan frame yang terekam lebih banyak sehingga memberikan ciri lebih detail. Akan tetapi ketika jumlah frame yang digunakan adalah 30 frame maka tingkat akurasi akan cenderung konstan. Hal ini dikarenakan jumlah frame sudah mencapai kondisi maksimum. Sedangkan waktu komputasi tercepat diperoleh dengan jumlah frame 10. Jumlah frame yang digunakan berbanding lurus terhadap waktu komputasi yang dihasilkan. Semakin banyak jumlah frame yang digunakan maka semakin lama waktu komputasi yang diperlukan untuk pemrosesan pengklasifikasian, demikian juga sebaliknya.

4. Kesimpulan

Dari hasil pengujian dan analisa yang telah dilakukan pada sistem, dapat diambil beberapa kesimpulan sebagai berikut :

1. Aplikasi sudah dapat mengidentifikasi 5 kelas dengan menggunakan metode Support Vector Machine. Tingkat akurasi sistem maksimal yang didapatkan yaitu 69 % dengan waktu komputasi adalah 9 detik.
2. Pengujian terbaik secara non *realtime* adalah dengan dimensi 80x80, menggunakan kernel *Gaussian* dengan sigma 4000.
3. Pengujian terbaik secara *realtime* adalah dengan jumlah frame 20 dan 30.

Daftar Pustaka

[1] Campbell, Colin. *Support Vector Machine and Kernel Methods*. Note Lecture of Bristol University.

[2] Kadir, Abdul dan Adhi Susanto. 2013. *Teori dan Aplikasi Pengolahan Citra*. Yogyakarta: Andi Publisher.

[3] Rachmat, Adam. 2010. *Sistem Identifikasi Biometrik Ruas Jari Tangan Manusia Menggunakan Principal Component Analysis (PCA) dan Learning Vector Quantization (LVQ)*. Bandung: Institut Teknologi Telkom.

[4] Santosa, Budi. *Tutorial Support Vector Machine*. Surabaya: Institut Teknologi Sepuluh Maret.

[5] Souza, Cesar R. 2012. *A Tutorial on Principal Component Analysis with the Accord.NET Framework*. Universaidade Federal De Sao Carlos.

[6] Wijaya, Marvin Ch, dan Agus Priyono. 2007. *Pengolahan Citra Digital Menggunakan Matlab Image Processing Toolbox*. Bandung: Informatika.