IMPLEMENTASI ALGORITMA MOTION CHAIN CODE DAN PRINCIPAL COMPONENT ANALYSIS UNTUK PENGENALAN BAHASA ISYARAT INDONESIA

Widyan Fakhruzzaman[1], Nelly Indriani Widiastuti[2]

Teknik Informatika – Universitas Komputer Indonesia
Jl. Dipatikukur 112-114 Bandung
Email: fakhruzzaman.w@gmail.com[1], nelly.indriani@email.unikom.ac.id[2]

ABSTRAK

Bahasa isyarat merupakan bahasa yang tidak menggunakan suara tetapi menggunakan gerakan tangan ataupun isyarat tangan dan mimik wajah yang dibutuhkan oleh penyandang tunarungu untuk memenuhi kebutuhan komunikasi dan kebutuhan informasi. Dalam sistem pengenalan bahasa isyarat terdapat proses yang harus dilakukan yaitu proses ekstraksi ciri. Dalam penelitian ini akan digunakan algoritma *Motion Chain Code* (MCC) untuk bahasa isyarat dinamis dan *Principal Component Analysis* (PCA) untuk bahasa isyarat statis. MCC merupakan metode yang dapat mendeskripsikan pola gerakan tangan sedangkan PCA merupakan metode yang mengambil fitur penting dari data berdimensi tinggi dengan cara mereduksi data tersebut menjadi data berdimensi rendah. Penelitian ini akan mengenali 26 huruf gerakan bahasa isyarat dengan data masukan berupa video dan menggunakan metode klasifikasi *Multiclass Support Vector Machine* (SVM). Dari hasil perancangan dan pengujian yang telah dilakukan, total akurasi tertinggi yang didapatkan untuk mengenali bahasa isyarat statis dan dinamis adalah sebesar 69,32% menggunakan data latih sebanyak 1040 vide dan diuji menggunakan 520 video yang setiap hurufnya memiliki 40 video data latih dan 20 video data uji. Pada pengujian bahasa isyarat statis didapatkan akurasi sebesar 67,25% dan untuk pengujian bahasa isyarat dinamis didapatkan akurasi sebesar 90%.

Kata kunci: Bahasa Isyarat, *Motion Chain Code*, *Principal Component Anlaysis*, *Multiclass*, *Support Vector Machine*

1. PENDAHULUAN

Bahasa isyarat merupakan bahasa yang tidak menggunakan suara tetapi menggunakan gerakan tangan ataupun isyarat tubuh dan mimik wajah yang dibutuhkan oleh penyandang tunarungu untuk memenuhi kebutuhan berkomunikasi dan memenuhi kebutuhan informasi. Keterbatasan komunikasi dengan orang normal menjadi masalah sosial para penyandang tunarungu dan tunawicara karena perbedaan metode komunikasi. Hal ini dapat mengganggu hubungan sosial antara orang normal dengan penyandang tunarungu, maka dibutuhkan perantara yang dapat menterjemahkan bahasa isyarat menjadi tulisan atau suara agar tercipta komunikasi yang baik antara keduanya.

Dalam ranah pengenalan gerakan bahasa isyarat Indonesia dengan masukan berupa citra atau video telah dicoba dalam beberapa penelitian. Salah satunya penelitian yang dikerjakan oleh Wildan [1], yang menggunakan metode ekstraksi ciri *Viola and Jones* untuk mengenali huruf bahasa isyarat indonesia. Dari hasil perancangan dan pengujian didapatkan akurasi sebesar 55%. Penelitian lain yang dilakukan oleh Novrizal [2], yang menggunakan metode *Principal Component Analysis* (PCA) dan menggunakan metode klasifikasi *Naïve Bayesian* mencapatkan akurasi sebesar 69%. Berdasarkan penelitian sebelumnya dibutuhkan algoritma yang dapat meningkatkan akurasi pengenalan gerakan bahasa isyarat dengan baik.

Sistem pengenalan bahasa isyarat memiliki proses penting yang harus dilakukan. Salah satu proses yang harus dilakukan adalah proses ekstraksi ciri. Ekstraksi ciri bertujuan untuk memperoleh informasi yang terkandung dalam suatu citra untuk kemudian digunakan sebagai acuan untuk membedakan antara satu citra dengan citra yang lain. Pada penelitian yang dilakukan oleh Atik [3], penggunaan metode ekstraksi ciri *Principal Component Analysis* (PCA) sebagai ciri pada pengenalan huruf yang mengacu pada Sistem Isyarat Bahasa Indonesia (SIBI) mendapatkan akurasi sebesar 80,42%. Kemudian pada penelitian lain yang dilakukan oleh Arshad Malik dan kawan-kawan [4], yang menggunakan metode ekstraksi ciri PCA digunakan untuk pengenalan *Pakistan Sign Language* (PSL) dengan tingkat akurasi mencapai 85%. Pada penelitian lain yang dilakukan oleh Mahmoud Zaki [5], yang menggunakan fitur dari proses ekstraksi ciri metode *Motion Chain Code* (MCC) untuk

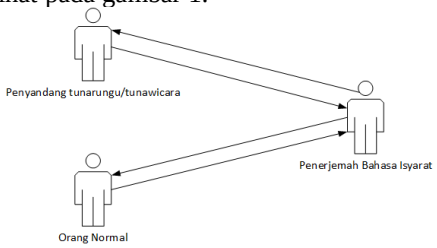
mendeskripsikan pola gerakan tangan bahasa isyarat dinamis dalam Bahasa Arab mendapatkan akurasi sebesar 88%. Dari beberapa penelitian tersebut metode ekstraksi ciri PCA dan MCC terlihat mampu memberikan hasil yang baik untuk pengenalan gerakan bahasa isyarat.

Berdasarkan permasalahan dan solusi yang telah dijelaskan, maka penelitian ini akan mengkombinasikan metode ekstraksi ciri PCA sebagai fitur untuk bahasa isyarat statis dan metode MCC yang sebagai fitur bahasa isyarat dinamis untuk pengenalan gerakan bahasa isyarat Indonesia sesuai dengan kamus Sistem Isyarat Bahasa Indonesia (SIBI).

2. ISI PENELITIAN

2.1. Analisis Masalah

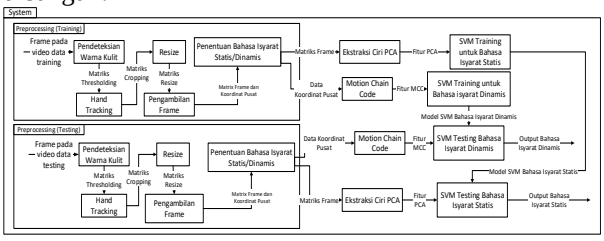
Adapun alur komunikasi yang sedang berjalan antara penyandang tunarungu dengan orang normal dapat dilihat pada gambar 1.



Gambar 1. Alur Komunikasi

2.2. Analisis Sistem

Sistem yang akan dibangun terdapat proses utama yang akan dilakukan, yaitu *preprocessing*, MCC, PCA, SVM *training* dan SVM *testing*. Dapat dilihat pada gambar 2 gambaran umum dari sistem yang akan dibangun.



Gambar 2. Alur Sistem

Berdasarkan gambar 2 dapat dijelaskan, data masukan yang digunakan untuk *training* dan *testing* adalah video dengan format .mp4 yang memiliki resolusi 320x240 piksel dan berdurasi 2 sampai 4 detik. Pada proses *training* yang pertama dilakukan adalah *preprocessing*. Dalam *preprocessing* akan dilakukan beberapa proses, yaitu pendeteksian warna kulit, *hand tracking*, *resize*, pengambilan *frame*, dan penentuan bahasa isyarat statis atau dinamis. Jika hasil dari penentuan bahasa isyarat adalah bahasa isyarat statis, maka data yang digunakan adalah matriks *frame* dilanjutkan kedalam proses ekstraksi ciri PCA dan hasil fitur dari ekstraksi ciri PCA akan digunakan untuk SVM training untuk bahasa isyarat statis. Kemudian jika hasil penentuan bahasa isyarat adalah bahasa isyarat dinamis, maka akan diambil koordinat pusat dari setiap *frame* yang berjumlah 8. Koordinat pusat tersebut akan digunakan untuk proses ekstraksi pola gerakan tangan menggunakan algoritma MCC. Hasil dari MCC akan digunakan SVM training untuk bahasa isyarat dinamis.

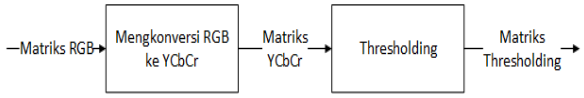
Pada proses *testing* akan dilakukan proses yang sama seperting *training*, yaitu *preprocessing*, ekstraksi ciri PCA jika hasil penentuan bahasa isyarat adalah statis dan dilakukan pengujian menggunakan SVM *Multiclass* menggunakan model prediksi bahasa isyarat statis. Kemudian jika hasil dari penentuan bahasa isyarat adalah dinamis, maka akan dilakukan proses MCC dan dilakukan pengujian menggunakan SVM *Multiclass* menggunakan model prediksi bahasa isyarat dinamis.

2.3. Preprocessing

Preprocessing bertujuan untuk mengolah data masukan sebelum diolah ke tahap selanjutnya.

a. Pendeteksian Warna Kulit

Proses pendeteksian warna kulit merupakan suatu proses yang memisahkan region objek dalam citra dengan berdasarkan pada perbedaan warna kulit. Pada gambar 3 dapat dilihat alur proses pendeteksian warna kulit.



Gambar 3. Alur Proses *Preprocessing*

Proses konversi dari RGB ke YCbCr menggunakan persamaan 1 [5].

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} + \begin{bmatrix} 0{,}299 & 0{,}587 & 0{,}144 \\ -0{,}169 & -0{,}331 & 0{,}500 \\ 0{,}500 & -0{,}419 & -0{,}081 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (1)$$

Dari hasil konversi RGB ke YCbCr, kemudian dilakukan proses thresholding menggunakan persamaan 2.

$$\begin{cases} Y > 80 \\ 85 < Cb < 135 \\ 135 < Cr < 180 \end{cases} \quad (2)$$

Proses thresholding dilakukan agar matriks hanya memiliki dua nilai, yaitu 0 dan 255. Dari persamaan thresholding tersebut, untuk mendapatkan nilai 255 semua kondisi harus terpenuhi [6].

b. Hand Tracking

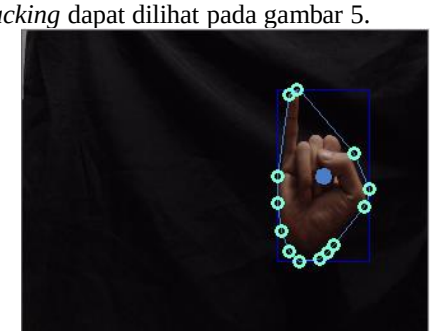
Hand Tracking ini bertujuan untuk melakukan pencarian area tangan dengan luas terbesar dan area tersebut dijadika sebagai *Region of Interest* (ROI).

Kemudian lebar dan panjang dari ROI akan membentuk persegi (*bounding area*) mengikuti kontur terbesar. Alur proses *hand tracking* dapat dilihat pada gambar 4.



Gambar 4. Alur Proses *Hand Tracking*

Data masukan untuk *hand tracking* merupakan matriks *thresholding* dari YCbCr. Dalam penerapannya *hand tracking* dilakukan menggunakan *library* Emgu CV [7]. Bounding area hasil dari proses *hand tracking* dapat dilihat pada gambar 5.



Gambar 5. Bounding Araea

Setelah bounding area didapatkan, selanjutnya adalah melakukan proses *cropping* menggunakan fungsi ROI pada *library* Emgu CV.

c. Resize

Resize merupakan proses mengubah resolusi atau ukuran horizontal dan vertikal suatu citra, hal ini bertujuan untuk menyamakan ukuran citra bahasa isyarat yang menjadi masukan dengan ukuran yang sama pada saat pelatihan maupun pengujian. Metode yang digunakan untuk proses risize adalah metode interpolasi liear. Interpolasi linear merupakan metode yang digunakan untuk mengetahui nilai dari sesuatu yang berada didalam sebuah interval atau didalam dua buah titik yang segaris. Cara kerja dari metode interpolasi linear adalah dengan mengambil empat sisi nilai-nilai piksel dari citra yang akan dilakukan *resize*. Kemudian dicari nilai piksel tengah diantara dua sisi dari citra. Untuk mencari nilai piksel digunakan persamaan 3.

$$i = A + \frac{w}{W}(B - A)$$
$$j = C + \frac{w}{W}(D - C) \quad (3)$$
$$Y = i + \frac{h}{H}(j - i)$$

Dimana:

i, j, y= nilai warna yang akan dicari

A, B, C, D= nilai intensity yang didapatkan dari citra sebelumnya

w= lebar dari ukuran citra yang diinginkan

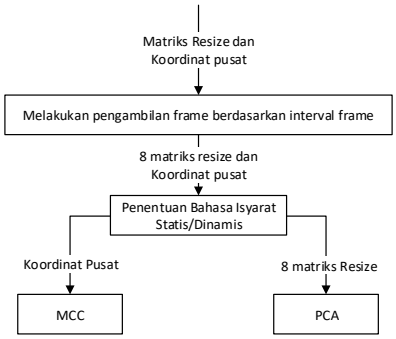
h= tinggi dari ukuran citra yang diinginkan

W= lebar jarak antara nilai warna yang akan dicari dengan warna sebelumnya (misalkan jarak A dengan i)

H= tinggi jarak antara nilai warna yang akan dicari dengan warna sebelumnya (misalkan jarak y dengan i)

d. Pengambilan *Frame*

Pengambilan *frame* bertujuan untuk mengambil matriks dari *frame* yang hanya mengandung gerakan bahasa isyarat saja. Alur proses pengambilan frame dapat dilihat pada gambar 6.



Gambar 6. Alur proses pengambilan *frame*

Tahap pertama adalah melakukan pengambilan *frame* berdasarkan interval menggunakan persamaan 4.

$$Iv = \frac{Jum_frame\ -5}{n} \quad (4)$$

Dimana:

Iv= interval *frame*

Jum_frame = jumlah frame keseluruhan

n = jumlah frame yang akan diambil

Contoh dalam satu video terdapat 65 *frame*, dan akan diambil 8 *frame*. Dalam penerapannya frame yang akan diambil berjumlah 20 *frame*, karena penelitian sebelumnya telah membedakan pengambilan 10, 20 dan 30 *frame* dan hasil yang didapatkan ketika diambil frame sebanyak 20 dan 30 akurasi cenderung meningkat [7].

$$Iv = \frac{Jum_frame\ -5}{n}$$
$$= \frac{65\ -5}{8}$$
$$= 7{,}5 \text{ (dibulatkan kebawah = 7)}$$

Berdasarkan hasil perhitungan yang didapatkan, setiap interval 7 *frame* akan dilakukan pengambilan *frame*. Frame yang didapatkan adalah frame ke- 7, 14, 21, 28, 35, 42, 49, dan 56 akan diambil dan diambil koordinat pusatnya. Proses selanjutnya adalah penentuan gerakan bahasa isyarat termasuk gerakan bahasa isyarat statis atau dinamis. Penentuan dilakukan berdasarkan jarak dari persamaan 5.

$$s = x_{max} - x_{min} \quad (5)$$

Dimana:

s= Jarak

x_{max} = nilai maksimum koordinat x

x_{min} = nilai minimum koordinat x

Hasil dari perhitungan jarak tersebut dilakukan pengecekan kondisi menggunakan persamaan berikut.

$$jenisisyarat = \begin{Bmatrix} Dinamis & s > 15 \\ Statis & s \leq 15 \end{Bmatrix} \quad (6)$$

Contoh koordinat pada huruf A pada tabel 1.

Tabel 1. Koordinat Huruf A

. Frame Ke-	Koordinat Pusat	
	X	Y
1	181	133
2	183	119
3	183	119
4	184	119
5	184	119
6	185	119
7	185	117
8	184	116

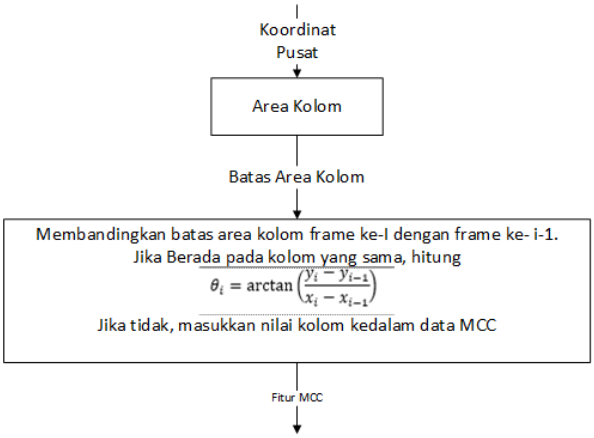
Berdasarkan tabel 1 didapatkan nilai x_{max}= 185 dan nilai x_{min}= 181, maka dapat dihitung jaraknya seperti berikut.

$$s = x_{max} - x_{min}$$
$$s = 185 - 181$$
$$= 4$$

Hasil perhitungan jarak koordinat pusat untuk huruf A adalah 4. Kemudian lakukan pengecekan kondisi dan didapatkan hasil bahwa 4 < 15, maka huruf A termasuk bahasa isyarat statis.

2.4. Motion Chain Code

Pada penelitian ini *Motion Chain Code* (MCC) digunakan untuk mengekstraksi gerakan relatif tangan dengan mengurangkan sebuah centroid tangan di dalam dua buah *frame* [5]. Nilai keluaran dari MCC akan digunakan sebagai fitur bahasa isyarat dinamis. Pada gambar 7 dapat dilihat alur proses dari algoritma *motion chain code*.



Gambar 7. Alur proses Motion Chain Code

Langkah pertama yang dilakukan adalah menentukan area kolom menggunakan persamaan 7.

$$batas1 = \frac{x_{min} + x_{max}}{2}$$
$$batas2 = \frac{batas1 + x_{max}}{2} \quad (7)$$
$$batas3 = \frac{batas2 + x_{max}}{2}$$

Dimana:

batas1, batas2, batas3= Nilai batas area setiap kolom

x_{min} = nilai minimum dari koordinat x gerakan bahasa isyarat

x_{max} = nilai maksimum dari koordinat x gerakan bahasa isyarat

Contoh nilai koordinat pusat pada huruf J pada tabel 2.

Tabel 2. Koordinat Pusat Huruf J

. Frame Ke-	Koordinat Pusat	
	X	Y
1	103	126
2	96	112
3	113	113
4	111	111
5	161	161
6	157	157
7	86	149
8	87	152

Berikut perhitungan untuk menentukan area kolom.

$$batas_1 = \frac{x_{min} + x_{max}}{2} = \frac{79 + 103}{2} = \frac{182}{2} = 91$$
$$batas_2 = \frac{batas_1 + x_{max}}{2} = \frac{91 + 103}{2} = \frac{194}{2} = 97$$
$$batas_3 = \frac{batas_2 + x_{max}}{2} = \frac{97 + 103}{2} = \frac{200}{2} = 100$$

Berdasarkan hasil perhitungan, didapatkan persamaan area setiap kolom menggunakan batas-batas area kolom yang sudah didapatkan adalah sebagai berikut.

$$Kolom = \begin{Bmatrix} 9, & 0 < x < 91 \\ 10, & 91 < x < 97 \\ 11, & 97 < x < 100 \\ 12, & 100 < x < x_n \end{Bmatrix} \quad (8)$$

Area kolom memiliki nilai 9, 10, 11 dan 12 dikarenakan untuk membedakan nilainya dengan nilai arah sudut gerakan (O) yang memiliki nilai dari 1 sampai 8 [5] yang dapat dilihat pada persamaan 9.

$$O = \begin{Bmatrix} 1, & \theta < 22{,}5 \text{ dan } \theta > 337{,}5^o \\ 2, & 22{,}5^o < \theta < 67{,}5^o \\ 3, & 67{,}5^o < \theta < 110{,}5^o \\ 4, & 110{,}5^o < \theta < 157{,}5^o \\ 5, & 157{,}5^o < \theta < 202{,}5^o \\ 6, & 202{,}5^o < \theta < 242{,}5^o \\ 7, & 242{,}5^o < \theta < 292{,}5^o \\ 8, & 292{,}5^o < \theta < 337{,}5^o \end{Bmatrix} \quad (9)$$

Setelah didapatkan area kolom maka langkah selanjutnya adalah melakukan pengecekan kondisi setiap koordinat. Jika koordinat (x_i) dan koordinat sebelumnya (x_{i-1}) tidak berada pada kolom yang

sama, maka tentukan nomor kolom koordinat (x_i) berada. Jika koordinat (x_i) dan koordinat sebelumnya (x_{i-1}) berada pada kolom yang sama, maka lakukan perhitungan nilai sudut menggunakan persamaan berikut [5].

$$\theta_i = \arctan\left(\frac{y_i - y_{i-1}}{x_i - x_{i-1}}\right) \quad (10)$$

Dimana:

θ_i = Nilai sudut gerakan pada frame-i
x_i= Nilai koordinat x
y_i= nilai koordinat y
x_{i-1}= nilai koordinat x sebelumnya
y_{i-1}= nilai koordinat y sebelumnya

Contoh pengecekan kondisi koordinat pusat huruf J frame-2 dengan frame-1 pada tabel 3 jika kedua frame berada pada kolom yang sama.

Tabel 3. Koordinat Pusat Frame-1 dan frame-2

Frame Ke-	Koordinat Pusat
1	X = 103 Y = 126
2	X = 96 Y = 112

Berdasarkan persamaan area kolom pada persamaan 8, nilai koordinat pada frame-2 dan frame-1 pada tabel 3 tidak berada pada kolom yang sama. Kemudian masukan nilai area kolom yaitu 12 untuk frame-2 kedalam data MCC: (12).

Contoh lain pengecekan kondisi koordinat pusat huruf J pada *frame*-7 dengan *frame*-8 pada tabel 4.

Tabel 4. Koordinat Pusat Frame-7 dan Frame-8

Frame Ke-	Koordinat Pusat
7	X = 86 Y = 149
6	X = 87 Y = 152

Berdasarkan persamaan area kolom, nilai koordinat X pada tabel 3 berada pada kolom yang sama. Lakukan perhitungan nilai sudut seperti berikut.

$$\theta_i = \arctan\left(\frac{y_i - y_{i-1}}{x_i - x_{i-1}}\right)$$

$$\theta_5 = \arctan\left(\frac{y_8 - y_7}{x_8 - x_7}\right)$$

$$\theta_5 = \arctan\left(\frac{152 - 149}{87 - 86}\right)$$

$$= \arctan\left(\frac{3}{1}\right)$$

$$= \arctan(3)$$

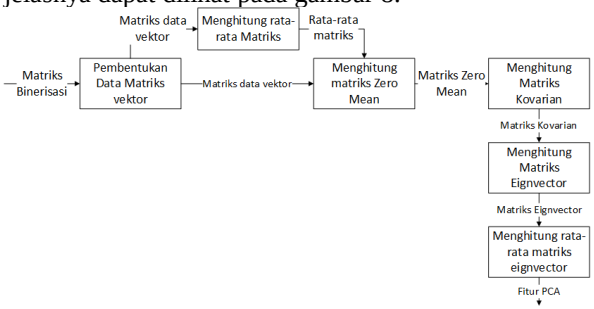
$=71^o$

Berdasarkan persamaan nilai sudut gerakan, maka sudut 71^o memiliki nilai arah sudut gerakan = 3. Masukkan nilai 3 kedalam data MCC: (12, 8, 2, 9, 8, 7, 3).

Perulangan sudah selesai dilakukan, maka data fitur dari proses MCC untuk bahasa isyarat huruf J yang didapatkan adalah (12, 8, 2, 9, 8, 7, 3). Data inilah yang akan digunakan sebagai fitur untuk gerakan bahasa isyarat dinamis pada saat proses pelatihan dan pengujian.

2.5. Principal Component Analysis

Principal Component Analysis (PCA) digunakan untuk mengurangi besarnya dimensi dari data yang diobservasi menjadi dimensi yang kecil tanpa kehilangan informasi yang signifikan dalam menggambarkan keseluruhan data. Untuk lebih jelasnya dapat dilihat pada gambar 8.



Gambar 8. Alur Proses Ekstraksi Ciri PCA

Berdasarkan gambar 8 dapat dijelaskan, langkah pertama yang dilakukan adalah membentuk matriks data menggunakan matriks dari hasil pengambilan frame yang berjumlah 8 matriks *frame* dan diubah menjadi matriks satu dimensi beukuran 6400x1 dan digabungkan menjadi matriks berukuran MxN berukuran 6400x8, dimana M merupakan dimensi matriks dan N merupakan jumlah matriks yang digabungkan.

Tabel 5 Matriks Data

N \ M	1	2	3	4	5	6	7	8
1	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0
14	0	0	0	0	0	1	1	1
15	0	0	0	0	0	1	1	1
16	0	1	0	1	0	1	0	0
17	1	1	0	1	1	1	0	0
18	1	1	0	1	1	1	0	0
19	1	1	1	1	1	1	0	0
20	1	1	1	1	1	0	0	0
	…	…	…	…	…	…	…	…
6400	0	0	0	0	0	0	0	0

Kemudian dilakukan perhitungan rata matriks sebagai berikut.

$$\bar{x}_N = \frac{\sum_{i=1}^{M} x_i}{M} = \frac{x_1 + x_2 + \cdots + x_M}{M} \quad (11)$$

Dimana:

$\bar{x}_N$ = rata-rata matriks kolom ke-n

M = dimensi matriks

Contoh perhitungan rata-rata matriks $\bar{x}_1$:

$$\bar{x}_N = \frac{x_1 + x_2 + \cdots + x_M}{M}$$

$$\bar{x}_1 = \frac{0+0+0+0+0+0+0+0+0+\cdots+0}{6400}$$

$$\bar{x}_1 = \frac{3719}{6400} = 0{,}5810938$$

Perhitungan ini dilakukan dari $\bar{x}_1$sampai $\bar{x}_N$. Hasil rata-rata dapat dilihat pada tabel 6.

Tabel 6. Nilai Rata-rata

$\bar{x}_N$	$\bar{x}_1$	$\bar{x}_2$	$\bar{x}_3$	$\bar{x}_4$	$\bar{x}_5$	$\bar{x}_6$	$\bar{x}_7$	$\bar{x}_8$
x	0.581 09	0.581 09	0.570 31	0.571 25	0.570 15	0.572 03	0.566 56	0.562 65

Setelah didapatkan rata-rata matriks, langkah selanjutnya menghitung matriks *zero mean* menggunakan persamaan berikut.

$$Y = D - \bar{x} \quad (12)$$

Dimana:

D = Matriks Data

$\bar{x}$ = rata-rata matriks

Contoh perhitungan matriks *zero mean* adalah sebagai berikut.

$$Y = D - \bar{x}$$

$$Y = \begin{bmatrix} D_{11} - \bar{x}_1 & D_{12} - \bar{x}_2 & \cdots & D_{1,n} - \bar{x}_n \\ D_{21} - \bar{x}_n & D_{22} - \bar{x}_2 & \cdots & D_{1,n} - \bar{x}_n \\ \cdots & \cdots & \cdots & \cdots \\ D_{m,1} - \bar{x}_1 & D_{m,1} - \bar{x}_2 & \cdots & D_{m,n} - \bar{x}_n \end{bmatrix}$$

$$= \begin{bmatrix} 0 - 0.58109 & 0 - 0.58109 & \cdots & 0 - 0.56265 \\ 0 - 0.58109 & 0 - 0.58109 & \cdots & 0 - 0.56265 \\ \cdots & \cdots & \cdots & \cdots \\ 0 - 0.58109 & 0 - 0.58109 & \cdots & 0 - 0.56265 \end{bmatrix}$$

$$Y = \begin{bmatrix} -0.58109 & -0.58109 & \cdots & -0.56265 \\ -0.58109 & -0.58109 & \cdots & -0.56265 \\ \cdots & \cdots & \cdots & \cdots \\ -0.58109 & -0.58109 & \cdots & -0.56265 \end{bmatrix}$$

Setelah didapatkan matriks *zero mean*, langkah selanjutnya adalah menghitung matriks kovarian menggunakan persamaan berikut [6].

$$C = \frac{1}{M-1} Y^T Y \quad (13)$$

Dimana:

C=Matriks Kovarian

M=Dimensi Matriks

Y=Matriks *Zero Mean*

Contoh perhitungan matriks zero mean adalah sebagai berikut.

$$C = \frac{1}{M-1} Y^T Y$$

$$C = \frac{1}{6400-1} \begin{bmatrix} D_{1,1} & D_{2,1} & \cdots & D_{6400,1} \\ D_{1,2} & D_{2,2} & \cdots & D_{6400,2} \\ \cdots & \cdots & \cdots & \cdots \\ D_{1,8} & D_{2,8} & \cdots & D_{6400,8} \end{bmatrix} \text{x} \begin{bmatrix} D_{1,1} & D_{1,2} & \cdots & D_{1,8} \\ D_{2,1} & D_{2,2} & \cdots & D_{2,8} \\ \cdots & \cdots & \cdots & \cdots \\ D_{6400,1} & D_{6400,2} & \cdots & D_{6400,8} \end{bmatrix}$$

$$C = \frac{1}{6399} \begin{bmatrix} -0.54313 & -0.54313 & \cdots & -0.54313 \\ -0.52392 & -0.52392 & \cdots & -0.52392 \\ \cdots & \cdots & \cdots & \cdots \\ -0.63397 & -0.63397 & \cdots & -0.63397 \end{bmatrix} *$$

$$\begin{bmatrix} -0.54313 & -0.52392 & \cdots & -0.63397 \\ -0.54313 & -0.52392 & \cdots & -0.63397 \\ \cdots & \cdots & \cdots & \cdots \\ -0.54313 & -0.52392 & \cdots & -0.63397 \end{bmatrix}$$

$$C = \frac{1}{6399} \begin{bmatrix} 1557.9 & 1450.9 & \cdots & 1384.48 \\ 1450.91 & 1557.9 & \cdots & 1391.48 \\ \cdots & \cdots & \cdots & \cdots \\ 1384.48 & 1391.4814 & \cdots & 1574.87 \end{bmatrix}$$

$$C = \begin{bmatrix} \frac{1557.9}{6399} & \frac{1450.9}{6399} & \cdots & \frac{1384.48}{6399} \\ \frac{1450.91}{6399} & \frac{1557.9}{6399} & \cdots & \frac{1391.48}{6399} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{1384.48}{6399} & \frac{1391.48}{6399} & \cdots & \frac{1574.87}{6399} \end{bmatrix}$$

$$C = \begin{bmatrix} 0.24346 & 0.22674 & \cdots & 0.21636 \\ 0.22674 & 0.24346 & \cdots & 0.21745 \\ \cdots & \cdots & \cdots & \cdots \\ 0.21636 & 0.21745 & \cdots & 0.24611 \end{bmatrix}$$

Setelah didapatkan matriks kovarian, langkah selanjutnya adalah penentuan matriks *eignvector* dengan mencari lebih dahulu *eignvalue* menggunakan persamaan berikut [9].

$$C - Z = |C - \lambda I|$$
$$|C - \lambda I| = 0 \quad (14)$$

Dimana:

C = matriks kovarian

Z = matriks *eigenvector*

λ = *scalar* pembentuk

I = matriks identitas

Contoh perhitungan *eignvalue* adalah sebagai berikut.

$$C - Z = |C - \lambda I|$$
$$|C - \lambda I| = 0$$

$$\begin{bmatrix} C_{11} & C_{12} & \cdots C_{1,n} \\ C_{21} & C_{22} & \cdots C_{2,n} \\ \cdots & \cdots & \cdots \cdots \\ C_{m,1} & C_{m,2} \cdots & C_{m,n} \end{bmatrix} - \lambda \begin{bmatrix} 1 & 0 \ldots & 0 \\ 0 & 1 \ldots & 0 \\ \cdots & \ldots\ldots & \cdots \\ 0 & 0 \ldots & 1 \end{bmatrix} = 0$$

$$\begin{bmatrix} 0.24346 & 0.22674 & \ldots 0.21636 \\ 0.22674 & 0.24346 & \ldots 0.21745 \\ \cdots & \cdots & \cdots \cdots \\ 0.21636 & 0.21745 & \ldots 0.24611 \end{bmatrix} - \begin{bmatrix} \lambda & 0 \ldots & 0 \\ 0 & \lambda \ldots & 0 \\ \cdots & \ldots\ldots & \cdots \\ 0 & 0 \ldots & \lambda \end{bmatrix} = 0$$

$$\begin{bmatrix} 0.24346 - \lambda & 0.22674 & \cdots & 0.21636 \\ 0.22674 & 0.24346 - \lambda & \cdots & 0.21745 \\ \cdots & \cdots & \cdots & \cdots \\ 0.21636 & 0.21745 & \cdots & 0.24611 - \lambda \end{bmatrix} = 0$$

Maka didapatkan persamaan:

$$-\lambda^8 + 1.95866\lambda^7 - 0.29188\lambda^6 + 0.01857\lambda^5 - 0.0006435\lambda^4 + 0.00001\lambda^3 - 1.574568e7\lambda^2 + 1.033097e - 9\lambda - 2.861807e - 12 = 0$$

Dari persamaan matriks diatas, didapatlah nilai eignvalue berikut.

$\lambda_1 = 1.801801$ $\lambda_2 = 0.47871$ $\lambda_3 = 0.20153$ $\lambda_4 = 0.10211$
$\lambda_5 = 0.04943$ $\lambda_6 = 0{,}03455$ $\lambda_7 = 0.03275$ $\lambda_8 = 0.01818$

Kemudian didapatkan matriks eignvector pada tabel 7.

Tabel 7. Matriks Eignvector

i / j	1	2	3	4	5	6	7	8
1	0.3553 9	0.3549 1	0.3565 2	0.3533 3	0.3545 8	0.3561 9	0.3498 1	0.3475 8
2	-0.2158 9	-0.1424 1	-0.2486 4	-0.4013 3	-0.2489 0	0.1761 4	0.4699 6	0.6296 0
3	-0.2767 2	-0.1542 6	-0.3047 2	0.2113 4	0.4265 5	-0.1567 1	0.6406 0	-0.3811 0
4	-0.2448 1	-0.4781 7	-0.1453 5	0.2911 9	0.4823 3	0.1140 4	-0.4313 6	0.4168 7
5	-0.4178 7	0.4478 5	-0.0133 0	0.4425 5	-0.2111 8	-0.5311 6	-0.0248 4	0.3185 0
6	-0.2310	0.3309 9	-0.4797	0.2570 3	-0.2100	0.6629 7	-0.1444	-0.1906

	9		0		5		7	5
7	-0.00697	-0.53314	0.30863	0.51336	-0.54193	0.13804	0.19685	-0.07366
8	0.68009	-0.08339	-0.60955	0.24328	-0.12381	-0.25262	0.00943	0.14339

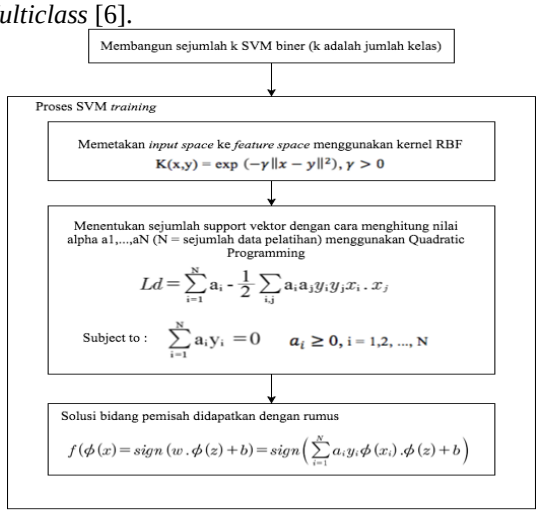
Kemudian hasil dari matriks *eigenvector* dihitung nilai rata-ratanya dari *eigenvector* setiap *eigenvalue* (rata-rata setiap kolom). Pada tabel 8 data dilihat hasil rata-rata eignvector untuk huruf I.

Tabel 8. Hasil Ekstraksi Ciri PCA

Huruf	Nilai Fitur PCA							
I	-0.0447	-0.0322	-0.1420	0.2388	-0.009	0.0633	0.1332	0.1513

2.6. Klasifikasi dengan SVM *Multiclass*

Hasil dari perhitungan ekstraksi ciri PCA akan digunakan untuk pelatihan bahasa isyarat statis dan hasil dari MCC akan digunakan untuk pelatihan bahasa isyarat dinamis. Pada gambar 9 dapat dijelaskan alur proses pelatihan menggunakan SVM *Multiclass* [6].

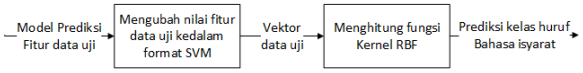


Gambar 9. Alur Proses Pelatihan SVM

Pada proses pengujian setelah didpatkan bidang pemisah, selanjutnya dapat menentukan data uji bahasa isyarat termasuk kedalam kelas positif atau negatif menggunakan fungsi keputusan berikut [8].

$$f(x_{testing}) = \sum_{i=0}^{n} a_i^T \, y_i \, K(x_{training}, x_{testing}) + b$$

Data masukan yang digunakan adalah data uji hasil ekstraksi ciri PCA untuk bahasa isyarat statis dan hasil MCC untuk bahasa isyarat dinamis. Hasil perhitungan dari data pemisah merupakan hasil prediksi kelas. Pada gambar dapat dijelaskan alur proses pengujian SVM *Multiclass One vs All.*



Gambar 10. Alur proses pengujian SVM

2.7. Hasil Pengujian

Pada penelitian ini pengujian dilakukan berdasarkan jumlah data latih dan data uji yang digunakan. Pengujian terdapat dua skenarion pengujian, yaitu :

a. Skenario Pengujian 1

Pada skenario pengujian 1 akan dilakukan pengujian akurasi terhadap metode Motion Chain Code (MCC) sebagai fitur untuk gerakan bahasa isyarat dinamis dan metode Principal Component Analysis (PCA) sebagai fitur untuk gerakan bahasa isyarat statis. Pengujian dilakukan untuk menggunakan 520 video data latih 520 video data uji. Masing-masing huruf gerakan bahasa isyarat memiliki 20 data latih dan 20 data uji. Adapun confusion matrix untuk skenario pengujian 1 dapat dilihat pada tabel 9.

Tabel 9. Confusion Matrix Skenario Pengujian 1

Kelas		Hasil Klasifikasi	
		P	N
Target	T	327	104
	F	89	0

Kemudian hitung akurasi dari *confusion matrix* yang didapatkan untuk bahasa isyarat statis menggunakan rumus (2.35) berikut.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} * 100\%$$

$$= \frac{327}{(327 + 104 + 89 + 0)} * 100\%$$

$$= 62{,}8\%$$

Pengujian akurasi untuk gerakan bahasa isyarat statis dan dinamis menggunakan 520 data latih dan 520 data uji yang masing-masing memiliki 20 data latih dan 20 data ujiuntuk setiap gerakan didapatkan akurasi sebesar 62,8%. Untuk bahasa isyarat dinamis mendapatkan akurasi sebesar 82,5% sedangkan untuk bahasa isyarat statis mendapatkan akurasi sebesar 56,5%.

b. Skenario Pengujian 2

Pada skenario pengujian 2 akan dilakukan pengujian akurasi terhadap metode *Motion Chain Code* (MCC) sebagai fitur untuk gerakan bahasa isyarat dinamis dan metode Principal Component Analysis (PCA) sebagai fitur untuk gerakan bahasa isyarat statis. Pengujian dilakukan untuk menggunakan 1040 video data latih dan 520 video data uji. Masing-masing huruf memiliki 40 data latih dan 20 data uji. Adapun tabel confusion matrix untuk skenario pengujian 2 dapat dilihat pada tabel

Tabel 10. *Confusion Matrix* Skenario Pengujian 2

Kelas		Hasil Klasifikasi	
		P	N
Target	T	360	92
	F	68	0

Adapun hasil perhitungan akurasi pengujian skenarion 2 seperti berikut.

$$Accuracy = \frac{TP + FN}{TP + TN + FP + FN} * 100\%$$

$$= \frac{360 + 0}{(324 + 68 + 92 + 0)} * 100\%$$
$$= 69{,}23\%$$

Pada pengujian bahasa isyarat statis dan dinamis yang telah dilakukan menggunakan menggunakan 1040 video data latih dan 520 video data uji yang masing-masing huruf memiliki 40 video data latih dan 20 video data uji. Pada pengujian untuk bahasa isyarat statis, data uji yang terklasifikasi dengan tepat sebanyak 360 dari 480 data uji. Akurasi yang didapatkan untuk bahasa isyarat statis adalah 67,5%. Kemudian pada pengujian untuk bahasa isyarat dinamis, data uji yang terklasifikasi dengan tepat sebanyak 36 data uji dari 40 data uji. Akurasi yang didapatkan untuk bahasa isyarat dinamis adalah 90%. Berdasarkan hasil pengujian bahasa isyarat statis dan bahasa isyarat dinamis yang telah dilakukan, hasil akurasi yang didapatkan untuk bahasa isyarat dari huruf A-Z adalah 69,23 % dengan *precision* sebesar 79,65% dan *recall* sebesar 100 %.

3. PENUTUP

3.1. Kesimpulan

Berdasarkan hasil percobaan yang dilakukan, jumlah data latih sangat berpengaruh terhadap akurasi yang didapatkan. Semakin banyak data latih semakin besar akurasi yang didaptkan. Berdasarkan hasil pengujian dapat diambil kesimpulan bahwa dengan mengimplementasikan algoritma *Motion Chain Code* dan *Principal Component Analysis* untuk pengenalan bahasa isyarat didapatkan akurasi tertinggi sebesar 69,23% dengan menggunakan 1040 video data latih dan 520 data uji yang masing-masing huruf memiliki 40 data latih dan 20 data uji. Pada pengujian bahasa isyarat statis didapatkan akurasi sebesar 67,25% dan untuk pengujian bahasa isyarat dinamis didapatkan akurasi sebesar 90%. Penggunaan fitur PCA untuk pengenalan bahasa isyarat statis dapat digunakan tetapi belum dapat melakukan pengenalan dengan baik dikarenakan faktor data latih dan ada beberapa huruf dengan bentuk yang hampir sama sehingga sulit untuk dikenali, tetapi dapat mengenali dengan baik huruf-huruf bahasa isyarat dinamis dengan baik menggunakan fitur dari MCC.

3.2. Saran

Berdasarkan hasil penelitian dan pengujian dapat diberikan saran untuk penelitian berikutnya untuk mencoba beberapa metode ekstraksi ciri yang lain utuk pengenalan bahasa isyarat statis. Mencoba algoritma MCC dengan menambahkan bahasa isyarat dinamis yang lain.

DAFTAR PUSTAKA

[1] W. Fauzan, "Finger Spelling untuk Kata Khusus dengan Bahasa Isyarat Tangan Menggunakan Metode Viola and Jones," *Skripsi*, UNIKOM, 2017.

[2] N. E. Sihbudi , "Impelemntasi Metode Naive Bayesian untuk Menerjemahkan Bahasa Indonesia," *Skripsi*, UNIKOM, 2017.

[3] A. Mardiyani, "Pengenalan Bahasa Isyarat Menggunakan Metode PCA dan Haar Like Feature," *ITS*, 2012.

[4] A. M. Malik, N. Kousar, T. Abdullah , M. Ahmed, F. Rasheed, "Pakistan Sign Language Using PCA and KNN," *International Journal of Advanced Computer Sience and Application (IJACSA)*, vol. 9, p. 4, 2018.

[5] M. Z. Abdo, S. A. El-Rahman, A. M. Hamdy and E. M. Saad, "Enhancement of Motion Chain Code for Arabic Sign Language Recognition," *IJACSA*, vol. 6, p. 12, 2015.

[6] G. Kukharev and A. Navosielski, "Visitor Identification Real Time Face Recognition System," *Winter School on Computer Graphics (WSCG)*, pp. 157-164, 2004.

[7] B. Putra AR, Computer Vision & Aplikasinya menggunakan C# & EmguCV, Yogyakarta: Penerbit Andi, 2017.

[8] R. Febriana, "Analisis dan Implementasi Aplikasi Pengenalan Gerak Bibir Menjadi Teks Menggunakan Metode Support Vector Machine," *Skripsi, Universitas Telkom*, 2015.

[9] Linear Algebra, "Eignvalue and Eignveytor," [Online]. Available: http://algebra.math.ust.hk/eigen/01_definition/lecture2.shtml. [Accessed 19 July 2018].

[10] R. Rifkin, A. Klautau, "In Defense of One Vs All Classification," *J. Mach. Learn*, vol. 5, pp. 101-141, 2004.