

# **FINGERSPELLING SISTEM ISYARAT BAHASA INDONESIA MENGUNAKAN *LEARNING VECTOR QUANTIZATION* 3**

Dinar Arisandra<sup>1</sup>, Galih Hermawan<sup>2</sup>

Program Studi Teknik Informatika

<sup>1,2</sup> Fakultas Teknik dan Ilmu Komputer Universitas Komputer Indonesia

Jl. Dipati Ukur 114 Bandung

E-mail : arisandra54@gmail.com<sup>1</sup>, galih.hermawan@email.unikom.ac.id<sup>2</sup>

## **ABSTRAK**

*Fingerspelling* digunakan untuk memberi informasi nama, orang dan tempat dan objek lain yang belum dikenal atau tidak diketahui oleh bahasa isyarat seperti nama orang, perusahaan, merk dan lain sebagainya. Penerjemah dalam bahasa isyarat dari *fingerspelling* sudah banyak dibuat dengan berbagai metode, namun selalu terdapat kendala atau akurasi yang masih kurang dalam menerjemahkan huruf bergerak yaitu huruf J dan Z. Program pengenalan yang akan dibuat akan menggunakan metode *learning vector quantization* 3 yang di mana metode ini merupakan optimasi dari metode sebelumnya yaitu *learning vector quantization*. Pendeteksian isyarat tangan menggunakan alat media *leap motion controller*, karena alat ini sudah dikhususkan hanya akan mendeteksi pada bagian tangan saja. *Leap motion controller* akan membuat ruang deteksi dengan tinggi 600 mm, panjang 150 derajat, dan lebar 120 derajat. Keluaran yang dihasilkan dari *leap motion controller* sudah berupa vektor terdiri dari palm1 (*palm normal*), palm2 (*palm position*), palm3 (*palm velocity*), hp (sudut x), roll (sudut z), dan yaw (sudut y). Berdasarkan hasil penelitian, analisis dan perancangan, implementasi serta tahap pengujian, *fingerspelling* sistem isyarat bahasa indonesia dengan menggunakan metode *learning vektor quantization* 3 menghasilkan akurasi sebesar 54%.

**Kata Kunci** : *fingerspelling*, *learning vector quantization* 3, *leap motion controller*

## **1. PENDAHULUAN**

Bahasa Isyarat merupakan bahasa utama kaum penyandang tunarungu, tunawicara atau orang tuli. Indonesia memiliki dua bahasa isyarat yang aktif digunakan, yaitu Sistem Isyarat Bahasa Indonesia (SIBI) dan Bahasa Isyarat Indonesia (BISINDO). Meskipun hanya SIBI yang dibuatkan kamus resmi oleh pemerintah demi untuk menyesuaikan bahasa isyarat dengan ejaan yang disempurnakan (EYD), keduanya tetap aktif digunakan meskipun untuk BISINDO sendiri memiliki sedikit perbedaan kosa

kata di setiap daerah tertentu. Kedua bahasa tersebut tidak dipelajari oleh semua orang, hanya beberapa orang yang berkepentingan dan yang memiliki minat.

*Fingerspelling* umumnya digunakan untuk memberi informasi nama orang, tempat dan objek lain yang belum dikenal atau tidak diketahui oleh bahasa isyarat seperti nama orang, perusahaan, merk dan lain sebagainya. Dalam penelitian sebelumnya *fingerspelling* sudah uji menggunakan metode *Viola & Jones* yang menghasilkan akurasi pengujian sebesar 55% [1], dan metode *Naive Bayesian* pengujian menghasilkan akurasi sebesar 69% [2]. Penelitian sebelumnya belum bisa mendeteksi huruf bergerak dari SIBI, yaitu huruf J dan huruf Z. Adapun penelitian memanfaatkan teknologi *Leap Motion Controller* dengan basis metode klasifikasi *Backpropagation* dapat mendeteksi huruf bergerak J dan Z serta menghasilkan akurasi sebesar 90% [3]. Sedangkan pada penelitian yang akan dilakukan ini adalah menggunakan teknologi *Leap Motion Controller* dan basis metode klasifikasi menggunakan algoritma pengembangan LVQ yaitu LVQ3. Dalam penelitian klasifikasi identitas wajah menggunakan LVQ memperoleh akurasi yang cukup tinggi yaitu 92% [4]. Untuk LVQ3 sendiri telah dibandingkan dengan LVQ dalam penelitian klasifikasi status gizi anak, penelitiannya memperoleh hasil akurasi 88% untuk LVQ dan untuk LVQ3 adalah 95,2% [5].

LVQ adalah suatu metode klasifikasi pola yang masing-masing unit output mewakili kategori atau kelompok tertentu. Pemrosesan yang terjadi pada setiap neuron adalah mencari jarak terdekat antara suatu vektor masukan ke bobot yang bersangkutan [6]. Kelebihan metode ini adalah selain mencari jarak terdekat, selama pembelajaran unit output diposisikan dengan mengatur dan memperbaharui bobot melalui pembelajaran yang terawasi untuk memperkirakan keputusan klasifikasi. Sedangkan untuk LVQ3, dua vektor (pemenang dan runner-up) diperbaharui jika beberapa kondisi dipenuhi. Ide pengembangan algoritma LVQ adalah jika input memiliki taksiran jarak yang sama dengan vektor pemenang dan runner-up, maka masing-masing vektor tersebut harus melakukan pembelajaran [7].

Berdasarkan penjelasan di atas, penelitian *fingerspelling* Sistem Isyarat Bahasa Indonesia

Menggunakan Learning Vektor Quantization 3 ini dimaksudkan untuk memanfaatkan teknologi *Leap Motion Controller* dan algoritma pengembangan LVQ yaitu LVQ3 pada penelitian kali ini, memungkinkan agar *fingerspelling* dapat menghasilkan akurasi yang berbeda dari penelitian sebelumnya dalam deteksi huruf yang mempunyai banyak *gesture* yaitu huruf J dan Z.

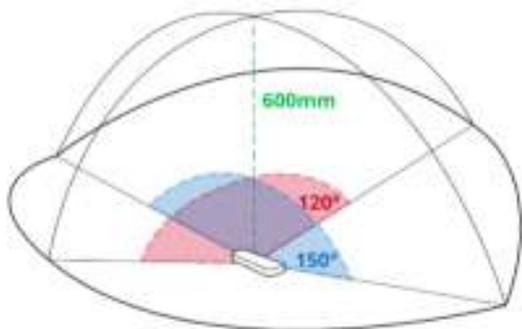
## 2. LANDASAN TEORI

### 2.1 Sistem Isyarat Bahasa Indonesia (SIBI)

Sistem Isyarat Bahasa Indonesia (SIBI) yang dibakukan itu merupakan salah satu media yang membantu komunikasi sesama kaum tunarungu di dalam masyarakat yang lebih luas. Wujudnya adalah tatanan yang sistematis tentang seperangkat isyarat jari, tangan, dan berbagai gerak yang melambangkan kosa kata bahasa Indonesia[8].

### 2.2 Leap Motion Controller

Leap Motion Controller adalah perangkat yang berguna untuk mendeteksi gerak isyarat tangan yang tidak bisa diwakili oleh tetikus dan papan tombol. Leap Motion Controller mampu mendeteksi jari tangan dari jarak hingga 600 milimeter. Alat ini memiliki dua kamera monokromatik dan tiga LED inframerah dengan akurasi hingga 1/100 milimeter. Area jangkauan penjejakan Leap Motion Controller diilustrasikan pada gambar 1[9].



Gambar 1. Area Jangkauan Leap Motion Controller

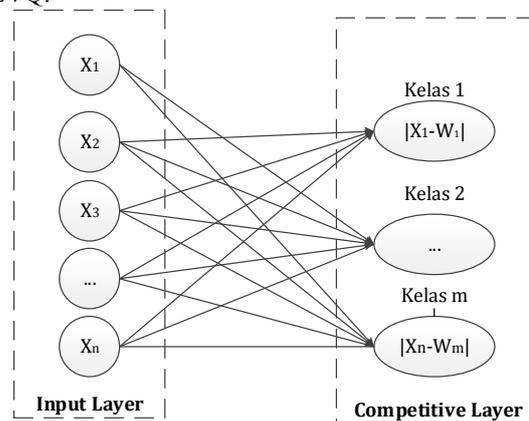
LMC mempunyai *library* yang menyediakan nilai berupa vektor dari setiap *gesture* tangan yang di tangkap menggunakan LMC. Vektor yang di ambil dari LMC terdiri dari 6 vektor, yaitu :

1. *Palm Position*: Pusat telapak tangan yang terbagi menjadi tiga vektor yaitu *palm1*, *palm2*, dan *palm3*
2. *Hand Pitch* : Sudut sumbu x
3. *Yaw* : Sudut sumbu y
4. *Roll* : Sudut sumbu z

### 2.3 Klasifikasi Learning Vector Quantization (LVQ)

LVQ adalah metode klasifikasi pola yang masing-masing unit keluarannya mewakili kategori atau kelas tertentu. Metode LVQ digunakan untuk pengelompokan dengan jumlah target atau kelas sudah ditentukan sebelumnya [10].

Setiap kelas diwakili atas satu set data yang digunakan dalam pelatihan, yang kemudian disebut bobot. Suatu lapisan kompetitif secara otomatis belajar untuk mengklasifikasikan vektor-vektor masukan. Kelas-kelas yang didapatkan sebagai hasil dari lapisan kompetitif ini hanya tergantung pada jarak antara vektor-vektor masukan dengan vektor bobot, yang kemudian bobot dari kelas pemenang direvisi. Jika dua vektor masukan mendekati sama, maka lapisan kompetitif meletakkan kedua vektor masukan tersebut ke dalam kelas yang sama. Prinsip kerja dari algoritma LVQ adalah pengurangan *node-node* yang pada akhirnya hanya ada satu *node* output yang terpilih. Gambar 2 menunjukkan arsitektur LVQ.



Gambar 2. Arsitektur LVQ

keterangan:

$X_1-X_n$  adalah vektor masukan

$W_1-W_m$  adalah vektor bobot pada lapisan kompetitif.

Tahap awal pembelajaran LVQ inialisasi bobot untuk tiap-tiap *node* yang didapatkan dari *node-node* input yang mewakili masing-masing kelas. Setelah menentukan bobot, maka jaringan diberi masukan sejumlah dimensi *neuron* masukan. Setelah masukan diterima jaringan, maka jaringan mulai melakukan perhitungan jarak vektor yang didapatkan dengan jarak antara vektor masukan dengan vektor bobot.

Akhir pelatihan diperoleh bobot akhir ( $W$ ). Bobot-bobot ini nantinya digunakan untuk identifikasi data yang lain. Pada saat pelatihan hanya bobot pemenang dengan nilai minimum yang dilakukan revisi. Tahapan algoritma LVQ, yaitu:

Pembelajaran LVQ3 dalam sistem dikembangkan berdasarkan algoritma LVQ1 dan ketentuan dasar teori LVQ3.

1. Tetapkan:
  - a. Bobot ( $w$ ),

- b. Maksimum iterasi (*MaxEpoch* )
  - c. Laju pembelajaran ( $\alpha$ )
  - d. Laju pembelajaran minimum ( $\epsilon$ )
  - e.  $Epoch = 0$
2. Kerjakan jika iterasi belum mencapai maksimum ( $Epoch < MaxEpoch$ ) atau laju pembelajaran lebih besar dari laju pembelajaran minimum ( $\alpha > \epsilon$ )
- a.  $Epoch = epoch + 1$
  - b. Kerjakan untuk  $i = 1$  sampai maksimum iterasi (*MaxEpoch* )
  - c. Tentukan jarak ( $J$ ) untuk mencari  $w$  minimum dengan Persamaan 2.7.

$$J = \sqrt{(x_{11} - w_{11})^2 + \dots + (x_{1n} - w_{1n})^2} \quad (1)$$

- d. Nilai window ( $\epsilon$ ), digunakan sebagai daerah yang harus dipenuhi untuk memperbaharui vektor referensi pemenang ( $Y1$ ) dan *runner-up* ( $Y2$ ) jika berada di kelas yang berbeda. Persamaan window ( $\epsilon$ ):

$$\text{Min} = \left( \frac{dc1}{dc2}, \frac{dc2}{dc1} \right) > (1 - \epsilon)(1 + \epsilon) \quad (2)$$

Nilai window ( $\epsilon$ ) = 0.3 (Fausett:1994)

- e. Jika memenuhi kondisi window ( $\epsilon$ ), maka pembaharuan menggunakan rumus:

$$w_{1,t+1} = w_{1,t} - \alpha(x - w_{1t}) \quad (3)$$

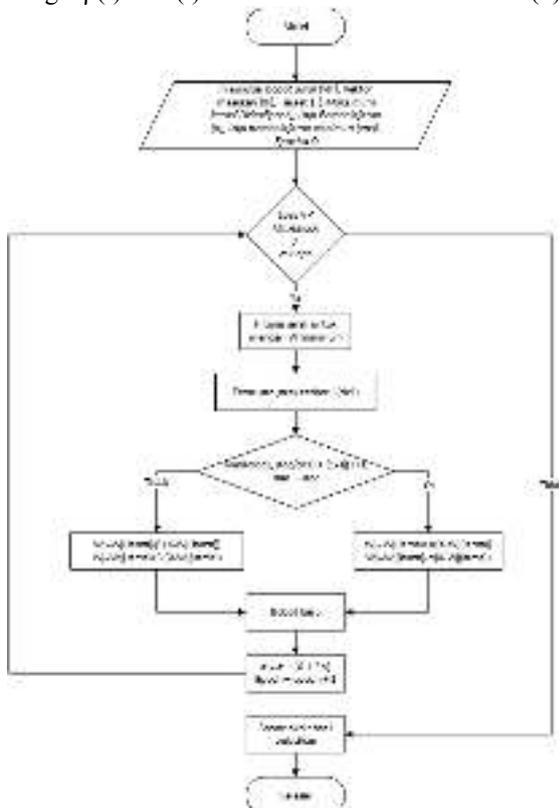
$$w_{1,t+1} = w_{1,t} + \alpha(x - w_{1t})$$

- f. Sedangkan jika tidak memenuhi maka pembaharuan menggunakan rumus:

$$w_{1,(t+1)} = w_{1(t)} - \beta(t) (x - w_{1(t)}) \quad (3)$$

$$w_{1,(t+1)} = w_{1(t)} + \beta(t) (x - w_{1(t)})$$

- g.  $\beta(t) = \epsilon\alpha(t)$  (4)



**Gambar 3.** Bagan Alir Pembelajaran (*Training*) LVQ3

### 3. METODE PENELITIAN

Metodologi penelitian yang digunakan dalam penelitian ini adalah penelitian kuantitatif. Metode yang digunakan dalam penulisan laporan penelitian ini menggunakan dua metode, yaitu metode pengumpulan data dan metode pembangunan perangkat lunak.

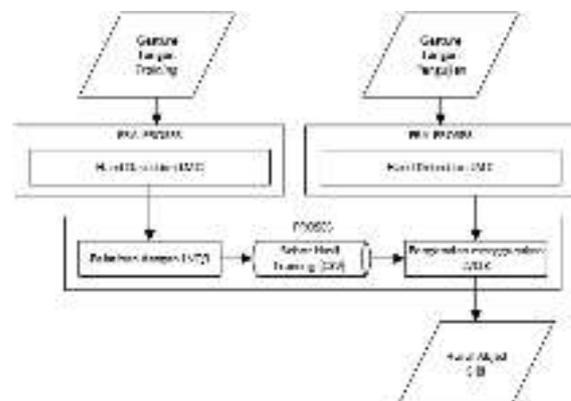


**Gambar 4.** Metode Penelitian

### 4. HASIL DAN PEMBAHASAN

#### 4.1 Gambaran Umum Sistem

Gambaran umum sistem yang akan di bangun dapat dilihat pada gambar 5.



**Gambar 5.** Gambaran Umum Sistem

Pembangunan sistem *fingerspelling* akan melalui beberapa tahapan. Tahapan pembuatan bobot baru

untuk selanjutnya yang akan digunakan klasifikasi terhadap data uji terdapat pada gambar 6

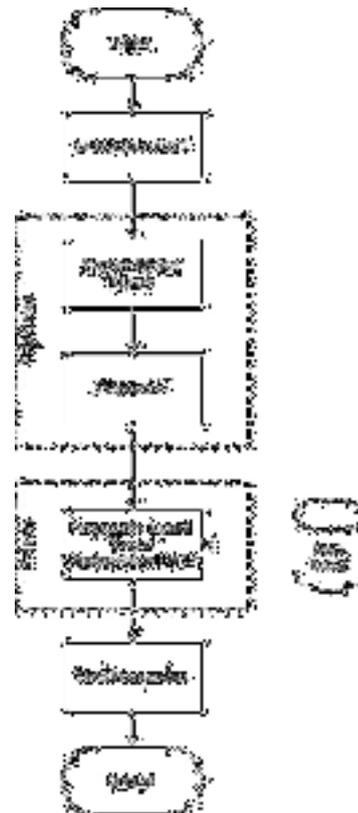


**Gambar 6. Klasifikasi Data Uji**

Berdasarkan gambar, sistem pembuatan bobot baru dari sistem yang akan dibangun, proses utama yaitu pengguna memasukan *gesture* tangan dari LMC dan memberi label sesuai *gesture* yang diambil yaitu huruf abjad alfabet. Proses yang ada pada tahapan pembuatan bobot baru yaitu:

- a. Pengguna mengaktifkan LMC
- b. *Capture gesture* tangan  
Pengambilan *gesture* tangan pada media masukan ke dalam sistem menggunakan LMC.
- c. *Praprocess*  
Proses sebelum proses utama, yaitu dilakukan *cropping*, *grayscale*, *thresholding*, dan konversi matrik pada citra masukan. Semua proses ini di *skip* karena LMC sudah melakukan *praprocess* dengan perhitungan yang tidak ditampilkan.
- d. *Gesture* tangan yang diambil pertama dijadikan bobot. *Gesture* tangan yang kedua dan seterusnya dijadikan data latih.
- e. Proses *training* menggunakan LVQ3  
Proses *training* data latih terhadap data bobot yang akan menghasilkan bobot baru yang akan digunakan untuk klasifikasi terhadap data uji.
- f. Proses menyimpan data *gesture* tangan  
*Gesture* tangan yang ditangkap disimpan menggunakan file .CSV.

Tahapan klasifikasi isyarat tangan dalam sistem terdapat pada gambar 7.



**Gambar 7. Klasifikasi Isyarat Tangan**

Berdasarkan gambar pada sistem klasifikasi isyarat tangan terbagi menjadi tahapan *praprocess* dan *process*. Proses utama yaitu pengguna mengaktifkan LMC kemudian meng-*capture gesture* tangan sesuai dengan isyarat tangan.

Tahap pertama yaitu *praprocess*. Proses yang ada dalam tahap ini yaitu:

- a. Pengguna mengaktifkan LMC
- b. *Capture gesture* tangan.
- c. *Praprocess*.

Proses selanjutnya adalah proses utama. Proses klasifikasi isyarat tangan menggunakan LVQ3, klasifikasi data hasil *praprocess* sebelumnya dengan data bobot hasil *training* LVQ3 yang sudah tersimpan dalam file .CSV.

#### 4.2 Pra-proses

Pra-proses merupakan proses pengambilan data untuk seterusnya di *training* oleh algoritma LVQ3. Data diperoleh dengan melakukan pengambilan data dari LMC. Posisi tangan tepat berada di depan LMC disertai tidak menggunakan aksesoris seperti cincin, gelang atau aksesoris lain yang dikenakan tangan seperti pada gambar 8.



**Gambar 8.** Capture data tangan dari LMC

Dalam pengambilan data dari LMC tidak dicantumkan perhitungan karena data yang diambil sudah berupa vektor. Banyaknya vektor masukan adalah sebanyak 6, yaitu *palm position* yaitu (*palm1*, *palm2*, *palm3*) dan sumbu *z* (*hp*), sumbu *x* (*roll*), dan sumbu *y* (*yaw*). Semua vektor ini didapatkan karena sistem kerja LMC adalah membentuk ruang sensor seperti pada gambar 1.

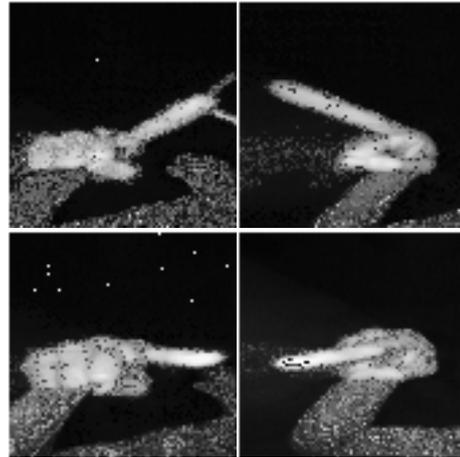
**Tabel 1.** Tabel Vektor LMC

No	Palm1	...	Roll	Kelas
1	-30.3971	...	-26.8880	I
2	-35.3162	...	-6.0653	J1
3	-37.7638	...	-71.7844	J2
4	-14.7153	...	-49.6420	J3
5	-29.6046	...	-16.0272	Z1
6	-10.0783	...	13.4136	Z2
7	-34.1530	...	-35.6873	Z3
8	-21.4457	...	-26.0776	Z4

Dalam pengambilan ciri untuk huruf bergerak, penelitian ini hanya menggunakan 4 *frame* utama dalam penerapannya. Semua *frame* diambil secara manual yang terdiri dari *frame* I, J1, J2, dan J3 untuk huruf J sedangkan untuk huruf Z adalah Z1, Z2, Z3, dan Z4.



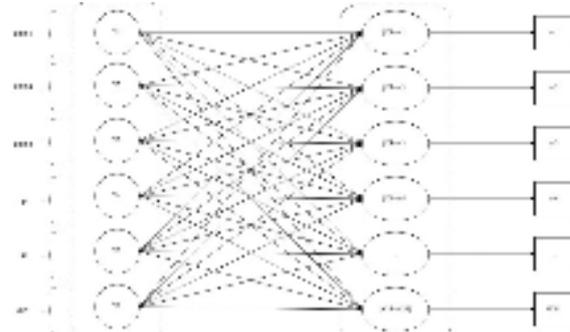
**Gambar 9.** Kombinasi Huruf J



**Gambar 10.** Kombinasi Huruf Z

### 4.3 Training

Pada fase *training* ini menggunakan LVQ3 terhadap identitas huruf. Jumlah neuron input memiliki 6 neuron, yang terdiri dari vektor *palm1*, *palm2*, *palm3*, *hp*, *roll*, *yaw*. Sementara itu, output layer memiliki dua puluh enam neuron merujuk pada arsitektur LVQ pada Gambar 11 Arsitektur LVQ untuk sistem klasifikasi dapat dilihat pada Gambar 3.5.



**Gambar 11.** Arsitektur LVQ

Berdasarkan rujukan penelitian sebelumnya, ditetapkan laju pembelajaran ( $\alpha$ ) sebesar 0.01 dengan pengurangan setiap satu kali iterasi  $0.1 * \alpha$ , laju pembelajaran minimum (*eps*) sebesar 0.001, maksimum *epoch* 1.000[4], nilai *window*  $\epsilon$  0.3 [15]. Diketahui 31 masukan vektor dalam dua puluh enam kelas sesuai.

**Tabel 2.** Vektor Bobot (W)

No	Palm1	...	Roll	Kelas
1	-30.3971	...	-26.8880	I
2	-35.3162	...	-6.0653	J1
3	-37.7638	...	-71.7844	J2
4	-14.7153	...	-49.6420	J3
5	-29.6046	...	-16.0272	Z1
6	-10.0783	...	13.4136	Z2

No	Palm1	...	Roll	Kelas
7	-34.1530	...	-35.6873	Z3
8	-21.4457	...	-26.0776	Z4

**Tabel 3.** Vektor Data Latih

No	Palm1	...	Roll	Kelas
1	-7.0845	...	-31.9043	I
2	-18.0686	...	-31.6101	I
3	-23.6199	...	-29.1312	I
4	-11.2545	...	-30.142	I
5	-4.8634	...	-29.303	I
6	-13.9992	...	-30.5507	I
7	-9.0316	...	-29.4871	I
8	1.6348	...	-26.0745	I
9	11.5798	...	-26.3806	I
10	-17.5413	...	-28.7256	I
11	-37.7202	...	-25.6646	I
12	-43.2258	...	-29.0739	I
13	-25.3187	...	-24.7178	I
14	-1.9447	...	-20.8329	I
15	1.9808	...	-20.1757	I
...	...	...	...	...
170	523.27	...	123.9453	Z4

**Tabel 4.** Vektor Bobot Hasil *Training*

Bobot	Palm1	...	Roll
W1	-30.4234	...	-26.9224
W2	-30.3708	...	-26.8536
W3	-37.7638	...	-71.7842
W4	-14.7153	...	-49.6420
W5	-29.5731	...	-16.0195
W6	-10.0728	...	13.3032
W7	-34.1291	...	-35.6737
W8	-21.4514	...	-25.9679

#### 4.4 Klasifikasi

Pada tahap klasifikasi data masukan berupa vektor hasil pra-proses dilakukan perhitungan dengan menggunakan bobot akhir hasil pembelajaran LVQ3.

Diketahui vektor masukan dari *gesture* data uji J1:

-14.7233 88.6025 ... -52.1370

Setelah menghitung jarak antara vektor masukan dengan bobot, dapat diketahui jarak terkecil pada

bobot ke-4, Sehingga pola tersebut masuk pada kelas ke J1 yaitu kelas dikenali.

#### 4.5 Deteksi Huruf Dnamis

Tahap ini adalah tahap deteksi huruf dinamis, yaitu huruf J dan Z. alur deteksi huruf J dan Z terdapat pada gambar 12.



**Gambar 12.** Proses Deteksi Huruf Dinamis

Tahapan deteksi huruf dinamis dalam gambar 3.6. yaitu:

1. Memasukan nilai input atau data uji
2. Pengujian menggunakan LVQ3
3. Apakah terbaca I atau Z1 bila ya maka masuk ke proses selanjutnya, bila tidak maka menampilkan hasil pengujian
4. Bila ya, maka memasukan kembali nilai input atau data uji
5. Proses pengujian menggunakan LVQ3
6. Bila J atau Z2 dan epoch < 3 bila ya epoch + 1, bila tidak menampilkan hasil pengujian
7. Bila melakukan pengujian sampai terdeteksi huruf J/Z atau epoch > 3

#### 4.6 Pengujian

Pengujian metode merupakan tahap pengujian performansi metode-metode yang diimplementasikan untuk membangun sistem pengenalan isyarat. Metode yang akan diujikan adalah metode Learning Vektor Quantization 3. Metode diujikan dengan menggunakan 8 data latih dan 3 data uji. Berikut rincian dapat dilihat pada tabel 5.

**Tabel 5.** Pengujian

No	Jenis Isyarat	Jumlah Data Latih	Jumlah Data Uji
1	I	3	3
2	J1	3	3

No	Jenis Isyarat	Jumlah Data Latih	Jumlah Data Uji
3	J2	3	3
4	J3	3	3
5	Z1	3	3
6	Z2	3	3
7	Z3	3	3
8	Z4	3	3
9	Bukan Gesture	0	3

### 1. Skenario Pengujian 1

Pada pengujian 1 ini dilakukan secara manual atau tidak secara *real time*, data yang digunakan merupakan data hasil dari *capture gesture* secara manual

**Tabel 6.** Confusion Matrix Pengujian 1

		Predicted Class	
		Positif	Negatif
Actual Class	Positif	7	17
	Negatif	0	2

Setelah melakukan klasifikasi, hitung nilai akurasinya menggunakan persamaan.

$$\begin{aligned}
 Akurasi &= \frac{TP + TN}{TP + FP + TN + FN} \\
 &= \frac{7+2}{7+17+2+0} \\
 &= \frac{9}{26} = 0,34
 \end{aligned}$$

Hitung nilai *precision* dan *recall* dengan menggunakan persamaan 2.13.

$$\begin{aligned}
 Precision &= \frac{TP}{TP + FP} \\
 &= \frac{7}{7+17} \\
 &= \frac{7}{24} = 0,29
 \end{aligned}$$

$$\begin{aligned}
 Recall &= \frac{TP}{TP + FN} \\
 &= \frac{7}{7+0} \\
 &= \frac{7}{7} = 1
 \end{aligned}$$

Pengujian akurasi menggunakan data uji sebanyak 26, yang terdiri dari 8 yang termasuk *gesture* bahasa isyarat dan 2 *gesture* tidak termasuk bahasa isyarat. Klasifikasi yang dihasilkan oleh sistem sebanyak 7 terklasifikasi bahasa isyarat dengan tepat, 17 terklasifikasi tidak tepat pada bahasa isyarat dan 2 terklasifikasi bukan bahasa isyarat dengan tepat, maka jumlah klasifikasi yang tepat adalah 20 bahasa isyarat.

Berdasarkan pengujian akurasi, didapatkan hasil akurasi sebesar 34% dengan *precision* sebesar 29% dan *recall* 100%.

### 2. Skenario Pengujian 2

Pada pengujian 2 dilakukan secara *real time* menggunakan LMC, data yang diujikan yaitu 3 data.

**Tabel 7.** Confusion Matrix Pengujian 2

		Predicted Class	
		Positif	Negatif
Actual Class	Positif	12	12
	Negatif	0	2

Setelah melakukan klasifikasi, hitung nilai akurasinya menggunakan persamaan.

$$\begin{aligned}
 Akurasi &= \frac{TP + TN}{TP + FP + TN + FN} \\
 &= \frac{12+2}{12+12+2+0} \\
 &= \frac{14}{26} = 0,54
 \end{aligned}$$

Hitung nilai *precision* dan *recall* dengan menggunakan persamaan 2.13.

$$\begin{aligned}
 Precision &= \frac{TP}{TP + FP} \\
 &= \frac{12}{12+12} \\
 &= \frac{12}{24} = 0,50
 \end{aligned}$$

$$\begin{aligned}
 Recall &= \frac{TP}{TP + FN} \\
 &= \frac{12}{12+0} \\
 &= \frac{12}{12} = 1
 \end{aligned}$$

Pengujian akurasi menggunakan data uji sebanyak 26, yang terdiri dari 8 yang termasuk *gesture* bahasa isyarat dan 2 *gesture* tidak termasuk bahasa isyarat. Klasifikasi yang dihasilkan oleh sistem sebanyak 7 terklasifikasi bahasa isyarat dengan tepat, 17 terklasifikasi tidak tepat pada bahasa isyarat dan 2 terklasifikasi bukan bahasa isyarat dengan tepat, maka jumlah klasifikasi yang tepat adalah 20 bahasa isyarat.

Berdasarkan pengujian akurasi, didapatkan hasil akurasi sebesar 54% dengan *precision* sebesar 50% dan *recall* 100%.

### 3. Skenario Pengujian 3

Pengujian dilakukan dengan mengujikan beberapa isyarat untuk dapat mengenali huruf bergerak, dalam hal ini huruf J dan Z. Berikut hasil pengujian pada tabel .

**Tabel 8.** Skenario Pengujian 3

No	Para meter Uji	Keluaran	Hasil Pengujian	
			Klasifik asi Faktual	Klasifik asi Sistem
1	J	I,J2,J2,J2	Positif	Negatif

No	Para meter Uji	Keluaran	Hasil Pengujian	
			Klasifik asi Faktual	Klasifik asi Sistem
2	J	I,J1,J2,J3	Positif	Positif
3	J	I,J3,J2,J3	Positif	Negatif
4	J	I,J1,J2,J3	Positif	Positif
5	J	I,J3,J2,J2	Positif	Negatif
6	Z	Z1,Z2,Z3,Z4	Positif	Positif
7	Z	Z1,Z4,Z3,Z4	Positif	Negatif
8	Z	Z1,Z2,Z3,Z4	Positif	Positif
9	Z	Z1,Z2,Z3,Z4	Positif	Positif
10	Z	Z1,Z2,Z3,Z4	Positif	Positif

Berikut adalah tabel dari *confusion matrix*:

**Tabel 7.** *Confusion Matrix* Pengujian 1

		Predicted Class	
		Positif	Negatif
Actual Class	Positif	6	4
	Negatif	0	0

Setelah melakukan klasifikasi, hitung nilai akurasi menggunakan persamaan.

$$\begin{aligned}
 \text{Akurasi} &= \frac{TP + TN}{TP + FP + TN + FN} \\
 &= \frac{6+0}{6+4+0+0} \\
 &= \frac{6}{10} = 0,6
 \end{aligned}$$

Hitung nilai *precision* dan *recall* dengan menggunakan persamaan.

$$\begin{aligned}
 \text{Precision} &= \frac{TP}{TP + FP} \\
 &= \frac{6}{6+4} \\
 &= \frac{6}{10} = 0,6 \\
 \text{Recall} &= \frac{TP}{TP + FN} \\
 &= \frac{6}{6+0} \\
 &= \frac{6}{6} = 1
 \end{aligned}$$

Berdasarkan pengujian akurasi, didapatkan hasil akurasi sebesar 60% dengan *precision* sebesar 60% dan *recall* 100%. Maka diketahui bahwa sistem kurang mengenali huruf J dan Z dengan akurat.

## 5. KESIMPULAN

Berdasarkan hasil penelitian, analisis dan perancangan, implementasi serta sampai pada tahapan pengujian, dapat diambil kesimpulan bahwa

*fingerspelling* sistem isyarat bahasa indonesia menggunakan *Learning Vektor Quantization* 3 menghasilkan akurasi pengujian sebesar 54%. Saran untuk penelitian kedepannya vektor ciri yang diambil dari tiap frame perlu penambahan dengan jumlah yang banyak dan penambahan jumlah frame untuk deteksi huruf bergerak agar akurasi menjadi lebih akurat dan pendeteksian huruf bergerak menjadi lebih efektif dan efisien.

## 6. DAFTAR PUSTAKA

- [1] Wildan Fauzan, Nelly Indriani W, "Finger Spelling untuk kata Khusus dengan Bahasa Isyarat Tangan Menggunakan Metode Viola and Jones," *Teknik Informatika*, p. 8, 2017.
- [2] N. E. Sihbudi, "Implementasi Metode Naive Bayesian untuk Menerjemahkan Bahasa Isyarat Indonesia," *Teknik Informatika*, p. 8, 2017.
- [3] Wijayanti Nurul, Risal Andika, Nanik Suciati, Ridho Rahman Hariadi, "Alphabet Sign Language Recognition Using Leap Motion Technology and Rule Based Backpropagation-Genetic Algorithm Neural Network (RBBPGANN)," *Jurnal Ilmiah Teknologi Informasi*, vol. 15, p. 95, 2017.
- [4] Gilang Ramadhan, Esmeralda C Djamal, Tedjo Darmanto, "Klasifikasi Identitas Wajah Untuk Otoritasi Menggunakan Deteksi Tepi dan LVQ," *Seminar Nasional Aplikasi Teknologi Informasi (SNATi)*, pp. D-37, 2016.
- [5] Elvia Budianita, Widodo Prijodiprodjo, "Penerapan Learning Vector Quantization (LVQ) untuk Klasifikasi Status Gizi Anak," vol. 7, no. 2, p. 155, 2013.
- [6] Kusumadewi S, Hartati S, Neuro Fuzzy : Integrasi Sistem Fuzzy dan Jaringan Syaraf, Yogyakarta: Graha Ilmu, 2006.
- [7] L. Fausett, *Fundamentals of Neural Networks ; Architectures, Algorithms, and Applications*, Pretince Hall: Englewood Cliffs, 1994.
- [8] Kamus, "Sistem Isyarat Bahasa Indonesia," Jakarta: Departemen Pendidikan Nasional, 2010.
- [9] L. Motion, "Leap Motion Documentation," Leap Motion, 11 February 2015. [Online]. Available: [developer-archive.leapmotion.com](http://developer-archive.leapmotion.com). [Diakses 1 January 2019].
- [10] M. D. Wuryandari dan I. Afrianto, "Perbandingan Metode Jaringan Syaraf Tiruan Backpropagation dan Learning Vector Quantization pada Pengenalan Wajah," *KOMPUTA*, vol. 1, pp. 45-51, 2012.