

## PENERJEMAHAN BAHASA ISYARAT MENGGUNAKAN METODE GENERALIZED LEARNING VECTOR QUANTIZATION (GLVQ)

<sup>[1]</sup>Dwi Gustiar, <sup>[2]</sup>Sampe Hotlan Sitorus, <sup>[3]</sup>Dwi Marisa Midyanti

<sup>[1][2][3]</sup>Jurusan Rekayasa Sistem Komputer, Fakultas MIPA Universitas Tanjungpura

Jalan Prof. Dr. H. Hadari Nawawi Pontianak

Telp./Fax : (0561) 577963

e-mail: <sup>[1]</sup>dwigustiar@student.untan.ac.id, <sup>[2]</sup>sitorus.hotland@gmail.com,

<sup>[3]</sup>dwi.marisa@siskom.untan.ac.id.

### Abstrak

Komunikasi merupakan kegiatan untuk berinteraksi dan berhubungan satu sama lainnya. Aktivitas komunikasi dapat dilakukan ketika pelaku komunikasi saling memahami bahasa yang digunakan. Umumnya bahasa yang digunakan merupakan bahasa verbal. Namun bagi penderita tunarungu, sulit untuk berkomunikasi menggunakan bahasa verbal. Keberadaan bahasa isyarat dapat membantu komunikasi antara penderita tunarungu dengan sesama penderita tunarungu atau penderita tunarungu dengan orang normal atau sebaliknya. Kemajuan teknologi memungkinkan masyarakat yang sebelumnya tidak mengerti bahasa isyarat dapat belajar untuk mengenalinya melalui sebuah sistem penerjemahan. Adapun bahasa isyarat yang digunakan adalah sistem isyarat bahasa Indonesia (SIBI). Agar bahasa isyarat dapat diterjemahkan oleh sistem, maka dilakukan penelitian ini menggunakan metode jaringan saraf tiruan *Generalized Learning Vector Quantization* (GLVQ). Metode ini digunakan untuk mendapatkan bobot terbaik yang akan digunakan untuk mengenali pola huruf bahasa isyarat pada sistem. Penggunaan metode GLVQ dalam sistem penerjemahan bahasa isyarat mendapatkan akurasi tertinggi pegujian data latih sebesar 68,32% dari 546 data dan akurasi pengujian data uji sebesar 71,37% dari 234 dengan menggunakan nilai *alpha* sebesar 0,9.

**Kata kunci:** bahasa isyarat, citra, glvq, jaringan saraf tiruan.

### 1 PENDAHULUAN

Komunikasi merupakan salah satu kegiatan yang dibutuhkan dalam kehidupan sehari-hari pada manusia. Komunikasi memainkan peran penting bagi manusia untuk dapat berinteraksi dan berhubungan satu sama lainnya. Melalui komunikasi seseorang dapat menyampaikan berbagai hal yang ada dipikirkannya kepada orang lain sehingga mencapai suatu pengertian makna pesan yang sama. Makna pesan yang tersampaikan dengan baik dapat membuat tujuan penyampaian pesan seseorang tercapai. Komunikasi dilakukan oleh siapa saja, dalam bentuk seperti melalui komunikasi verbal maupun komunikasi non verbal. Komunikasi verbal merupakan komunikasi dalam bentuk kata-kata, sedangkan komunikasi non verbal merupakan komunikasi yang melalui gesture atau bahasa tubuh seseorang [1].

Untuk dapat melakukan komunikasi, pelaku komunikasi harus saling memahami bahasa yang digunakan. Umumnya bahasa yang digunakan merupakan bahasa verbal. Namun bagi penderita tunarungu, sulit untuk berkomunikasi menggunakan bahasa verbal. Keberadaan bahasa

isyarat dapat membantu komunikasi antara penderita tunarungu dengan sesama penderita tunarungu atau penderita tunarungu dengan orang normal atau sebaliknya. Bahasa isyarat adalah bahasa yang menggunakan bahasa tubuh, gerak bibir dan komunikasi manual. Bahasa isyarat untuk tunarungu terdiri dari kombinasi bentuk dan gerakan tangan, lengan, tubuh dan ekspresi wajah. Sulit bagi orang normal untuk berkomunikasi dengan penderita tunarungu tanpa menggunakan bahasa isyarat.

Permasalahan yang diangkat dalam kasus penelitian ini dilatarbelakangi oleh ketidaktahuan masyarakat umum yang mengerti bahasa isyarat. Kemajuan teknologi memungkinkan masyarakat yang sebelumnya tidak mengerti bahasa isyarat dapat belajar untuk mengenali bahasa isyarat dengan sebuah sistem yang dapat menerjemahkan bahasa isyarat. Adapun bahasa isyarat yang digunakan adalah sistem isyarat bahasa Indonesia (SIBI).

Penelitian ini dapat menangani masalah tersebut dengan menggunakan aplikasi perangkat lunak yang *userfriendly* untuk melakukan

penerjemahan Bahasa Isyarat. Aplikasi Perangkat Lunak tersebut berfungsi untuk mengambil dan mengolah hasil tangkapan gambar agar dapat menerjemahkan kedalam Sistem Isyarat Bahasa Indonesia (SIBI).

Penelitian ini mengangkat kasus mengenai penerjemahan bahasa isyarat SIBI menggunakan metode GLVQ. Metode GLVQ dipilih karena telah dilakukan penelitian oleh [2] dan [3] dengan data berupa citra yang menghasilkan akurasi yang lebih baik dari pada akurasi metode LVQ yang diteliti oleh [4]. Pada penelitian yang dilakukan oleh [4] dengan judul “Pengenalan Bahasa Isyarat Huruf Abjad Menggunakan Metode *Learning Vector Quantization* (LVQ)” menghasilkan sistem pengenalan bahasa isyarat huruf abjad yang dapat mengenali 26 huruf isyarat, dengan tingkat akurasi sebesar 61,54%. Penelitian yang dilakukan oleh Sulistia Rauf Yulian ini merupakan referensi untuk kasus yang diangkat.

Penelitian mengenai metode GLVQ pernah diteliti oleh [2] dengan judul “Pengembangan Sistem Pengenalan Wajah Menggunakan Metode *Generalized Learning Vector Quantization* (GLVQ)” Pada sistem pengenalan pengenalan wajah untuk uji 120 citra wajah, 160 citra wajah dan 200 citra wajah.

Penelitian lain mengenai metode GLVQ juga diteliti oleh [3] dengan judul “Pengenalan Aksara Bali Menggunakan Metode *Modified Direction Feature* dan Algoritma *Generalized Learning Vector Quantization* (GLVQ)”. Penelitian ini menghasilkan sistem pengenalan aksara bali dengan pengujian data karakter aksara Bali yang di tulis menggunakan *mouse*, untuk pengujian data karakter aksara Bali yang ditulis tangan lalu di scan dengan menggunakan nilai *Learning Rate* yang menghasilkan akurasi tertinggi yaitu 0,9, menggunakan *minimum error* 0.0001, dan penurunan rasio *learning rate* sebesar 0.01.

Berdasarkan uraian diatas dapat diketahui bahwa penelitian yang dilakukan oleh [4] yang mempunyai kasus sama dengan penelitian ini, tetapi menggunakan metode yang berbeda yaitu LVQ. Pada penelitian tersebut menghasilkan akurasi sebesar 61,54%. Berdasarkan tingginya tingkat akurasi pada pengenalan yang menggunakan metode GLVQ maka dibuat sebuah aplikasi penerjemahan bahasa isyarat dengan jaringan saraf tiruan menggunakan metode *Generalized Learning Vector Quantization* (GLVQ).

## 2 LANDASAN TEORI

### 2.1 Definisi Citra

Citra adalah kumpulan piksel-piksel yang disusun dalam larik dua dimensi. Indeks baris dan kolom (x,y) dari sebuah piksel dinyatakan dalam bilangan bulat. Piksel (0,0) terletak pada sudut kiri atas pada citra, indeks x bergerak ke kanan dan indeks y bergerak kebawah. Konvensi ini dipakai merujuk pada cara penulisan larik yang digunakan dalam pemrograman komputer [5].

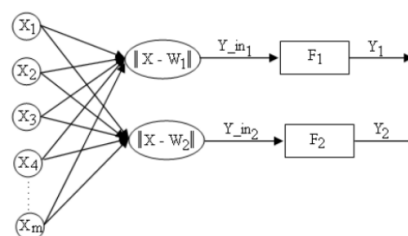
### 2.2 Jaringan Saraf Tiruan

Dalam [6], Putra menyatakan bahwa jaringan saraf tiruan atau *Artificial Neural Network* yang sering disingkat ANN merupakan model jaringan syaraf yang meniru prinsip kerja syaraf otak manusia. ANN pertama kali muncul setelah model sederhana jaringan saraf diperkenalkan oleh McCullosh dan Pitts pada 1943. Model sederhana tersebut dibuat berdasarkan fungsi syaraf biologis.

#### 2.2.1 *Learning Vector Quantization*

Salah satu metode untuk melakukan pembelajaran lapisan kompetitif yang terawasi adalah *Learning Vector Quantization* (LVQ). Lapisan kompetitif ini akan belajar secara otomatis untuk melakukan klasifikasi terhadap vektor input yang diberikan. Kelas yang diberikan berdasarkan pada jarak vektor tersebut. Jika ada dua vektor yang mempunyai jarak yang berdekatan maka akan dikelompokkan menjadi satu kelas yang sama [6].

Sebuah jaringan LVQ memiliki lapisan kompetitif pertama dan lapisan linier kedua. Para lapisan kompetitif belajar untuk mengklasifikasikan vektor masukan. Lapisan linier mengubah kelas lapisan kompetitif ke klasifikasi target yang didefinisikan oleh pengguna. Hal ini mengacu pada kelas dipelajari oleh lapisan kompetitif sebagai subkelas dan kelas-kelas dari linier lapisan sebagai kelas target [7]. Arsitektur LVQ dapat dilihat pada Gambar 1.



Gambar 1 Arsitektur LVQ  
Sumber: [10]

Kedua lapisan kompetitif dan linier memiliki satu neuron per (sub atau target) kelas. Dengan demikian, lapisan kompetitif bisa belajar sampai

dengan subclass S1. Ini, pada gilirannya, digabungkan oleh lapisan linier untuk membentuk kelas menargetkan S2 (S1 adalah selalu lebih besar dari S2) [7].

### 2.2.2 Generalized Vector Quantization

*Generalized Learning Vector Quantization (GLVQ)* dikembangkan oleh Atsushi Sato dan Keiji Yamada pada tahun 1996. Sebuah Metode pembelajaran untuk meminimalkan fungsi biaya (*cost function*). Perumusan perbedaan jarak relatif  $\mu(x)$  pada metode GLVQ dapat dilihat pada persamaan 1 [8].

$$\mu(x) = \frac{d_1 - d_2}{d_1 + d_2} \quad (1)$$

Dimana  $d_1$  adalah jarak  $x$  dengan  $w_1$ , dan  $d_2$  adalah jarak antara  $x$  dengan  $w_2$ .  $\mu(x)$  bernilai antara -1 sampai 1. Jika  $\mu(x)$  bernilai negatif maka  $x$  diklasifikasikan benar, jika  $\mu(x)$  bernilai positif maka  $x$  diklasifikasikan salah. Dengan demikian ukuran pembelajaran diformulasikan dengan meminimalkan *cost function* S dapat dilihat pada persamaan 2 [8].

$$S = \sum_{i=1}^N f(\mu(x_i)) \quad (2)$$

Dimana  $N$  adalah vektor input pada pembelajaran. Untuk meminimalkan  $S$ ,  $w_1$ , dan  $w_2$  yaitu dengan cara diperbarui dengan menggunakan persamaan 3 [8] dan persamaan 4 [8].

$$w_1 = w_1 + \alpha \frac{\delta f}{\delta \mu} \frac{d_2}{(d_1 + d_2)^2} (x - w_1) \quad (3)$$

$$w_2 = w_2 - \alpha \frac{\delta f}{\delta \mu} \frac{d_1}{(d_1 + d_2)^2} (x - w_2) \quad (4)$$

Dimana  $\frac{\delta f}{\delta \mu}$  adalah turunan dari fungsi sigmoid

$$f(\mu, t) = \frac{1}{1 + e^{-\mu t}} \quad [8] \quad (5)$$

$$\text{Sehingga } f'(\mu, t) = \frac{1}{1 + e^{-\mu t}} \left\{ 1 - \frac{1}{1 + e^{-\mu t}} \right\} \quad (6)$$

Fungsi pada persamaan 6 dapat disederhanakan menjadi  $f'(\mu, t) = \frac{e^{\mu t}}{e^{\mu t} + 1}$  (7)

GLVQ merupakan sebuah metode pembelajaran pada lapisan kompetitif yang terawasi. Suatu lapisan kompetitif akan secara otomatis belajar untuk mengklasifikasikan vektor-vektor input. Adapun algoritma dari GLVQ adalah [8]:

1. Tetapkan bobot ( $w$ ), maksimum epoch (MaxEpoch), *error minimum* yang diharapkan (Eps), *Learning rate* ( $\alpha$ ), Pengurang rasio (dec).
  - a. Tentukan
    - i. Input :  $x(m,n)$

- ii. Target :  $T(1,n)$
2. Tetapkan kondisi awal epoch = 0
3. Kerjakan jika : (epoch < MaxEpoch) atau ( $\alpha > \text{eps}$ )
  - a. Epoch = epoch + 1
  - b. Kerjakan untuk  $i = 1$  sampai  $n$ 
    - i. Tentukan  $j$  sedemikian hingga  $\|x - w_j\|$  minimum (sebut sebagai  $C_j$ )
    - ii. Perbaiki  $w_j$  dengan ketentuan
    - iii. Jika  $T = C_j$  maka, hitung  $w_j$  (baru) dengan rumus pembaruan  $w_1$  berikut:
 
$$w_j(\text{baru}) = w_j(\text{lama}) + \alpha \frac{e^{\mu t}}{e^{\mu t} + 1} \frac{d_2}{(d_1 + d_2)^2} (x - w_j(\text{lama})) \quad (8)$$
    - iv. Jika  $T \neq C_j$  maka, hitung  $w_j$  (baru) dengan rumus pembaruan  $w_2$  berikut :
 
$$w_j(\text{baru}) = w_j(\text{lama}) - \alpha \frac{e^{\mu t}}{e^{\mu t} + 1} \frac{d_1}{(d_1 + d_2)^2} (x - w_j(\text{lama})) \quad (9)$$
4. Kurangi *learning rate*

$$\alpha = \alpha - (\alpha * \text{pengurang rasio}).$$

### 2.3 Pendeteksian Tepi (Edge Detection)

Suatu objek yang berada dalam bidang citra dan tidak bersinggungan dengan batas bidang citra, berarti objek tersebut dikelilingi daerah yang bukan objek yaitu latar belakang. Pertemuan antara bagian obyek dan bagian latar belakang disebut tepi obyek. Dari uraian diatas, dapat kita mengerti bahwa tepi merupakan salah satu fitur citra yang penting karena dapat mewakili informasi yang penting dari obyek [5].

### 2.4 Grid Partition

Segmentasi merupakan teknik untuk membagi suatu gambar menjadi beberapa daerah (*region*) dimana setiap daerah memiliki kemiripan atribut. *Grid Partitioning* merupakan metode sederhana untuk melakukan segmentasi gambar dengan menggunakan *grid* (kotak) persegi panjang yang sifatnya *fixed-size* serta *slides over* (dapat tumpang tindih) pada gambar. Untuk setiap *grid* persegi panjang dari gambar, fitur vektornya akan diekstrak. Ukuran dari persegi panjang dapat bervariasi untuk membuat versi multi skala dari *grid partitioning*. Penggabungan antara *overlapping* dan multi skala memungkinkan untuk mengatasi perubahan posisi dan perubahan skala dari suatu gambar. Dengan menggunakan *grid* mendapatkan beberapa keuntungan. Kinerja dari *grid* persegi panjang lebih baik daripada metode *Region Segmentation*. Selain itu, terjadi penurunan yang signifikan dalam waktu komputasi yang diperlukan untuk segmentasi gambar [9].

### 3 METODE PENELITIAN

Tahapan metode penelitian yang digunakan untuk merealisasikan penelitian ini berupa studi literatur, metode pengumpulan data, analisis kebutuhan, perancangan sistem, implementasi, dan pengujian.

#### 3.1 Studi Literatur

Pada tahap ini dilakukan mencari teori- teori pendukung agar sistem dapat terealisasikan. Teori- teori pendukung yang didapat berupa data dari instansi terkait, buku- buku, dan jurnal yang berkaitan dengan penelitian ini.

#### 3.2 Metode Pengumpulan Data

Metode pengumpulan data untuk kebutuhan penelitian ini meliputi data-data yang dapat menunjang penelitian. Data yang didapat berupa pola tangan yang memiliki arti dalam SIBI. Data pola tangan tersebut merupakan data primer. Data diperoleh dengan menggunakan webcam yang terhubung langsung dengan sistem aplikasi yang akan dibangun. Data yang diperoleh dibagi menjadi 2 yakni, data pelatihan dan data pengujian. Data pelatihan sebanyak 546 citra yang didapat dari 21 orang dimana tiap orang membentuk masing-masing 26 pola tangan sesuai dengan huruf yang ada pada SIBI. Data pengujian sebanyak 234 citra yang didapat dari 9 orang dimana tiap orang membentuk masing-masing 26 pola tangan sesuai dengan huruf yang ada pada SIBI.

#### 3.5 Implementasi

Implementasi dari hasil perancangan berupa aplikasi penerjemahan bahasa isyarat dengan menggunakan metode GLVQ. Hasil perancangan diimplementasikan berupa kode program dan tampilan antarmuka.

#### 3.6 Pengujian

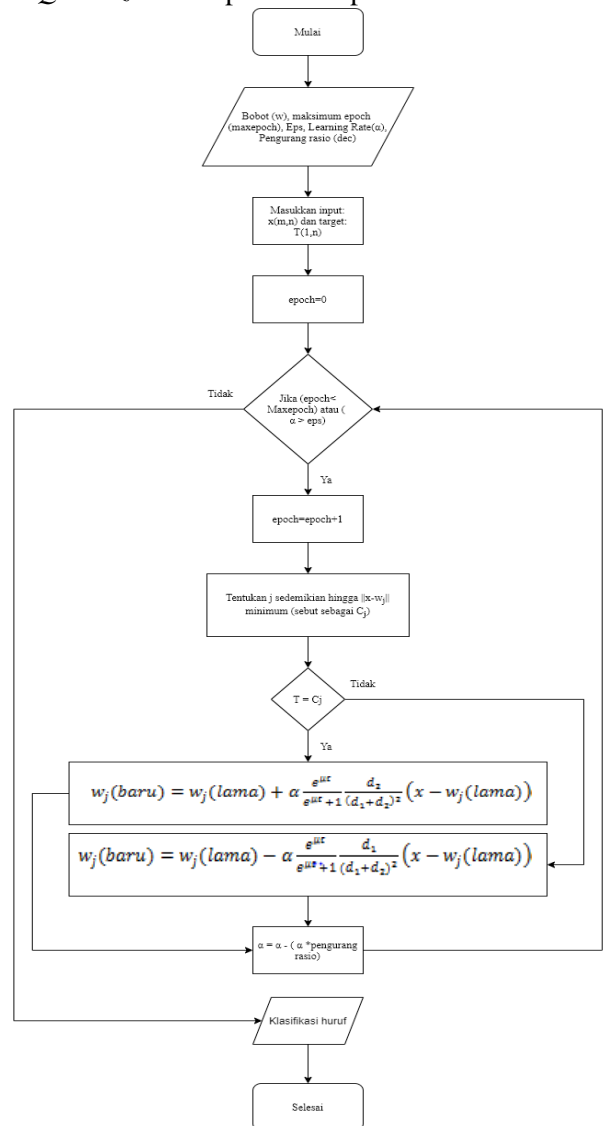
Pengujian sistem dilakukan untuk mengetahui akurasi penerjemahan bahasa isyarat yang dihasilkan oleh sistem. Pengujian sistem dalam penelitian ini menggunakan skenario pengujian klasifikasi huruf dengan variasi nilai grid yang diijinkan dan nilai laju pembelajaran yang telah ditentukan. Pada bagian ini juga ditambahkan hitungan manual dari metode yang digunakan beserta pembahasannya.

## 4 PERANCANGAN SISTEM

### 4.1 Flowchart Metode Generalized Learning Vector Quantization

Flowchart sistem prediksi jumlah kebutuhan

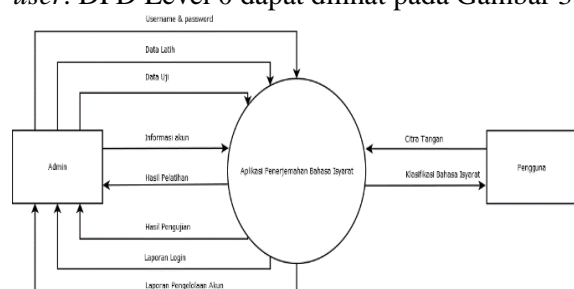
obat menggunakan metode *Generalized Learning Vector Quantization* dapat dilihat pada Gambar 2.



Gambar 2. Flowchart Metode Generalized Learning Vector Quantization

### 4.2 Data Flow Diagram (DFD)

Pada perancangan sistem digunakan *Data Flow Diagram (DFD)* untuk menggambarkan ruang lingkup sistem yang akan dibangun. Pada sistem ini terdapat 2 pengguna, yaitu admin dan user. DFD Level 0 dapat dilihat pada Gambar 3.



Gambar 3. DFD Level 0

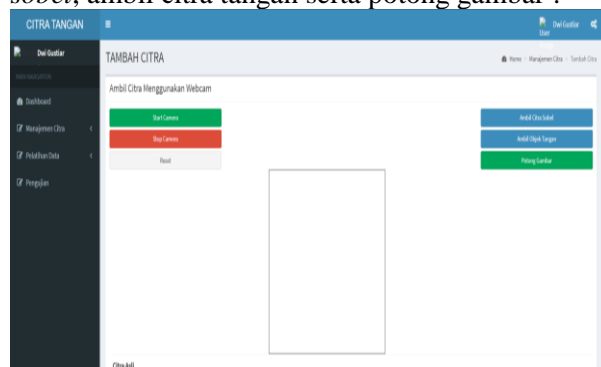
## 5 IMPLEMENTASI, PENGUJIAN DAN PEMBAHASAN

### 5.1 Implementasi Perangkat Lunak

Berikut ini merupakan tampilan antarmuka aplikasi sesuai dengan hak aksesnya masing-masing, yaitu admin dan *user*.

#### 1. Halaman Ambil Citra

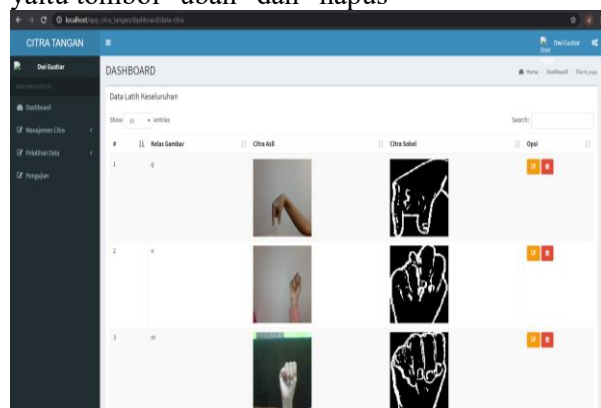
Gambar 4 merupakan implementasi halaman ambil citra yang memiliki tombol *start camera*, *stop camera*, *reset*. Terdapat juga ambil citra *sobel*, ambil citra tangan serta potong gambar .



Gambar 4. Halaman Ambil Citra

#### 2. Halaman Citra Data Seluruhnya

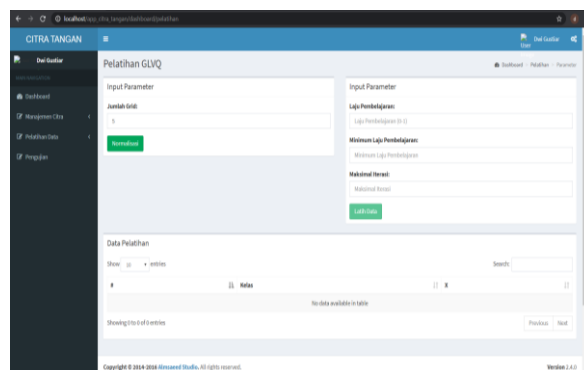
Gambar 5 merupakan implementasi halaman untuk mengelola data citra secara keseluruhan yang menampilkan data-data citra. Terdapat dua tombol yang digunakan untuk mengelola data citra yaitu tombol “ubah” dan “hapus”



Gambar 5. Halaman data Citra Keseluruhan

#### 3. Halaman Pelatihan GLVQ

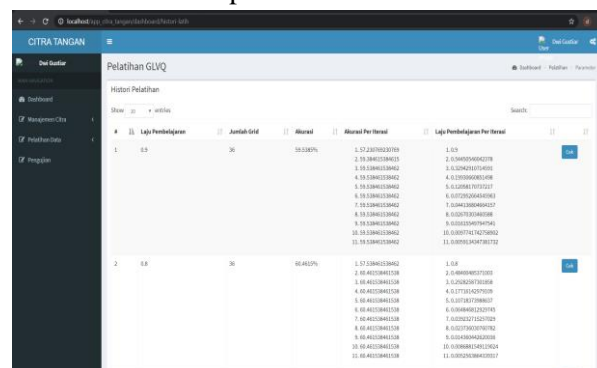
Gambar 6 merupakan implementasi pelatihan GLVQ yang memiliki *form* input parameter yang berisi jumlah *grid* dan tombol normalisasi. Halaman juga memiliki *form* yang berisi laju pembelajaran, minimum laju pembelajaran, dan tombol latih data.



Gambar 6 Halaman Pelatihan GLVQ

#### 4. Halaman Histori Pelatihan

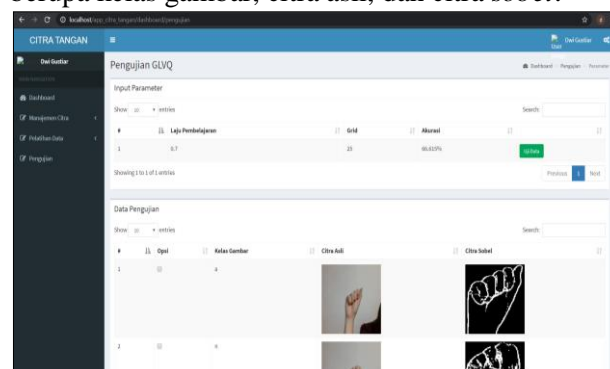
Gambar 7 merupakan halaman histori pelatihan. Halaman ini menampilkan tabel yang berisi laju pembelajaran, jumlah *grid*, akurasi, serta memiliki tombol cek untuk menampilkan hasil klasifikasi tiap data latih.



Gambar 7. Halaman Histori Pelatihan

#### 5. Halaman Pengujian

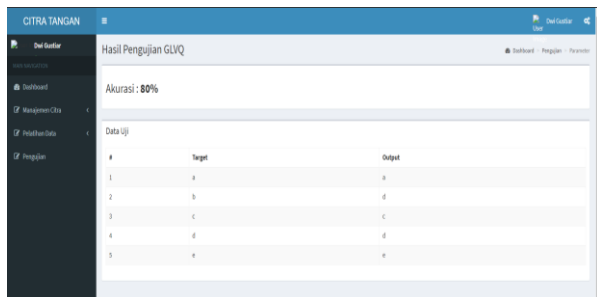
Gambar 8 merupakan halaman pengujian. Halaman ini menampilkan histori pengujian dan data uji. Informasi data uji yang ditampilkan berupa kelas gambar, citra asli, dan citra *sobel*.



Gambar 8. Halaman Pengujian

#### 6. Halaman Hasil Pengujian

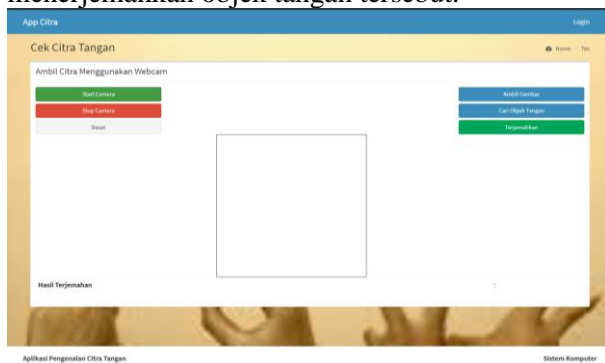
Gambar 9 merupakan halaman hasil pengujian. Halaman ini menampilkan hasil pengujian berupa target dan kelas klasifikasi dari data yang dipilih untuk diuji.



Gambar 9. Halaman Hasil Pengujian

## 7. Halaman Pengguna

Gambar 10 merupakan halaman pengguna. Halaman ini menampilkan gambar secara *realtime* untuk pengguna agar dapat melakukan penerjemahan secara langsung. Terdapat tombol untuk memulai kamera, memberhentikan kamera, dan *me-reset*. Terdapat juga tombol ambil citra tangan untuk menangkap gambar dari kamera, tombol cari objek tangan untuk memulai pencarian objek tangan dari gambar yang telah diambil, dan tombol terjemahan untuk menerjemahkan objek tangan tersebut.



Gambar 10. Halaman Pengguna

## 4.2 PENGUJIAN

Bobot-bobot hasil pelatihan yang tersimpan akan digunakan untuk pengujian menggunakan data latih dan data uji. Untuk data latih pengujian menggunakan semua bobot dari masing- masing pelatihan sedangkan untuk data uji menggunakan bobot terbaik berdasarkan hasil pengujian menggunakan data latih.

### 1. Data Latih

Pada pengujian ini dilakukan sebanyak 500 kali dengan menggunakan 546 data latih. Pengujian ini dilakukan untuk mengetahui akurasi dari tiap- tiap pelatihan yang telah dilakukan. Pada pengujian ini menggunakan parameter sebagai berikut:

- a. *Max Epoch* : 500
- b. *Alpha* ( $\alpha$ ) : 0.1- 0.9
- c. *Min alpha* : 0.00001
- d. *Dec alpha* : 0.01
- e. *Grid* : 9, 25, 36, 100, 225, 625, 900, 2500

Hasil semua pengujian data latih dapat dilihat pada Tabel 1 dan Gambar 14

Tabel 1. Akurasi Pengujian Data Latih

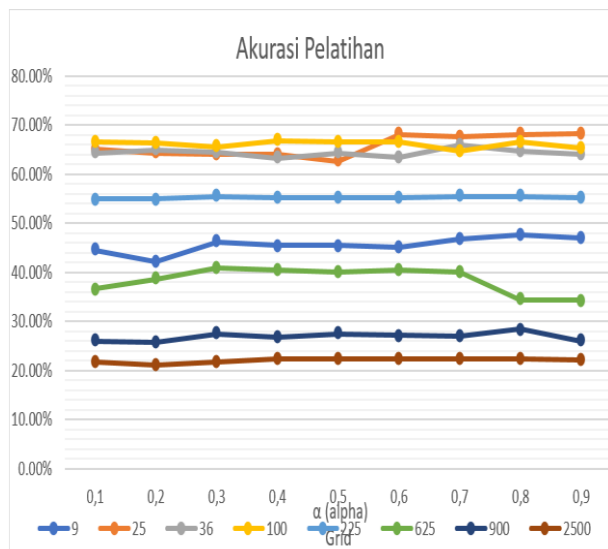
Grid\ $\alpha$	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9
9	44.51%	42.12%	46.15%	45.42%	45.42%	45.05%	46.70%	47.62%	46.89%
25	65.02%	64.29%	64.10%	64.10%	62.64%	68.13%	67.58%	68.13%	68.32%
36	64.29%	64.84%	64.47%	63.19%	64.29%	63.37%	65.93%	64.65%	64.10%
100	66.48%	66.30%	65.57%	66.85%	66.67%	66.48%	64.65%	66.48%	65.20%
225	54.95%	54.95%	55.31%	55.13%	55.13%	55.13%	55.49%	55.31%	55.13%
625	36.63%	38.64%	40.84%	40.48%	40.11%	40.48%	40.11%	34.43%	34.25%
900	25.91%	25.64%	27.47%	26.74%	27.47%	27.11%	26.92%	28.39%	26.01%
2500	21.61%	21.03%	21.79%	22.34%	22.34%	22.34%	22.34%	22.34%	22.34%

Berdasarkan Tabel 1 dan Gambar 14 didapat bahwa pada *grid* 9 akurasi terendah yaitu 42,12% dan yang tertinggi yaitu 47,62% . Pada *grid* 25 akurasi terendah yaitu 62,64% dan tertinggi yaitu 68,32%. Pada *grid* 36 akurasi terendah adalah 63,19% dan tertinggi adalah 65,93%. Pada *grid* 100 akurasi terendah adalah 64,65% dan tertinggi 66,85%. Pada

*grid* 225 akurasi terendah adalah 54,95% dan tertinggi adalah 55,49%. Pada *grid* 625 akurasi terendah adalah 34,25% dan tertinggi adalah 40,84%. Pada *grid* 900 akurasi terendah adalah 25,64% dan tertinggi adalah 28,39%. Pada *grid* 2500 akurasi terendah adalah 21,03% dan tertinggi

adalah 22,34%. Sehingga diketahui akurasi terbaik dari pengujian data latih adalah 68,32%.

Adapun Gambar 14 adalah grafik akurasi pelatihan untuk mempermudah membaca perbedaan akurasi pelatihan data latih tiap *grid* dan *alpha*.



Gambar 14 Grafik Akurasi Pelatihan

## 2. Data Uji

Berdasarkan Tabel 1 didapat parameter-parameter yang digunakan untuk pengujian sebagai berikut :

- a. *Alpha* : 0.9
- b. *Max Epoch* : 500
- c. *Min alpha* : 0.0001
- d. *Dec alpha* : 0.01
- e. *Grid* : 25

Hasil pengujian ini dapat dilihat pada Tabel 2

Tabel 2. Pengujian Data Uji

Data Ke-	Target	Hasil Klasifikasi	Validasi
1	a	a	<b>Benar</b>
2	b	j	Salah
3	c	c	<b>Benar</b>
4	d	d	<b>Benar</b>
5	e	b	Salah
6	f	f	<b>Benar</b>
7	g	c	Salah
8	h	h	<b>Benar</b>
9	i	f	Salah
10	j	j	<b>Benar</b>
11	k	k	<b>Benar</b>

Data Ke-	Target	Hasil Klasifikasi	Validasi
12	l	l	<b>Benar</b>
13	m	m	<b>Benar</b>
14	n	s	Salah
15	o	o	<b>Benar</b>
16	p	p	<b>Benar</b>
17	q	q	<b>Benar</b>
18	r	r	<b>Benar</b>
19	s	a	Salah
20	t	t	<b>Benar</b>
21	u	r	Salah
22	v	k	Salah
23	w	a	Salah
24	x	m	Salah
25	y	y	<b>Benar</b>
26	z	d	Salah
27	a	a	<b>Benar</b>
28	b	f	Salah
29	c	c	<b>Benar</b>
30	d	d	<b>Benar</b>
31	e	b	Salah
32	f	f	<b>Benar</b>
33	g	g	<b>Benar</b>
34	h	h	<b>Benar</b>
35	i	y	Salah
36	j	j	<b>Benar</b>
37	k	v	Salah
38	l	l	<b>Benar</b>
39	m	b	Salah
40	n	x	Salah
41	o	o	<b>Benar</b>
42	p	p	<b>Benar</b>
43	r	r	<b>Benar</b>
44	s	f	Salah
45	t	t	<b>Benar</b>
46	u	r	Salah
47	v	v	<b>Benar</b>
48	q	q	<b>Benar</b>
49	w	w	<b>Benar</b>
50	x	x	<b>Benar</b>



Tabel 2 hanya menampilkan 50 dari 234 data. Adapun tabel pengujian data uji yang lengkap bisa dilihat pada lampiran. Berikut validasi hasil klasifikasi menggunakan data uji yaitu :

Jumlah benar = 167

Jumlah salah = 67

Berdasarkan jumlah benar dan salah, maka didapat akurasi menggunakan data uji yaitu:

$$\text{Akurasi} = \frac{\text{Jumlah benar}}{\text{Jumlah total}} \times 100\% = \frac{167}{234} \times 100\% = 71,37\%$$

## 6. KESIMPULAN DAN SARAN

### 6.1 Kesimpulan

Berdasarkan penelitian yang telah dilakukan, diperoleh kesimpulan sebagai berikut:

1. Nilai *learning rate* / *alpha* mempengaruhi besaran nilai perbaikan bobot. Semakin besar nilai *alpha* maka akan semakin besar nilai perbaikan bobot yang didapatkan dan sebaliknya. Nilai *alpha* sebesar 0,9 menghasilkan akurasi terbaik pada penelitian ini.
2. Penggunaan nilai *alpha* sebesar 0,9 menghasilkan akurasi tertinggi sebesar 68,32% dari 546 data latih pada pelatihan dan 71,37% dari 234 data uji pada pengujian.
3. Sistem Isyarat Bahasa Indonesia (SIBI) memiliki banyak pola yang mirip seperti (a,e,dan s), (m dan n), (k dan v). Kemiripan ini mempengaruhi hasil akurasi.

### 6.2 Saran

Adapun saran untuk penelitian selanjutnya yaitu;

1. Penelitian ini dapat dikembangkan pada *platform* berupa *android/ios* untuk mempermudah penerjemahan Bahasa isyarat secara *real time*.
2. Citra yang akan dideteksi tepi harus diperbaiki terlebih dahulu agar mendapatkan hasil yang optimal. Perbaikan kualitas citra dapat menggunakan teknik seperti perbaikan kontras gelap/terang, perbaikan tepian objek (*edge enhancement*) dan penajaman (*sharpening*).
3. Untuk mendapatkan fitur yang lebih baik, maka hindari *resize* sama rata antar huruf. Penulis menyarankan untuk menambah latar polos agar mempunyai ukuran yang sama.

## DAFTAR PUSTAKA

- [1] Nurudin. (2010). *Sistem Komunikasi Indonesia*. Jakarta: Rajawali Pers.
- [2] Yulian, S. R. (2016). *Pengenalan Bahasa Isyarat Huruf Abjad (LVQ)*
- [3] Hermawan, I. (2015). *Pengembangan Sistem Pengenalan Wajah Menggunakan Metode Generalized Learning Vector Quantization (GLVQ)*.
- [4] Septiari, N. W. (2015). *Pengenalan Bahasa Akasara Bali Menggunakan Metode Modified Direction Feature dan Algoritma Generalized Learning Vector Quantization (GLVQ)*.
- [5] Ahmad, U. (2005). *Pengolahan Citra Digital dan Pemrogramannya*. Yogyakarta: Graha Ilmu
- [6] Khairy, M. K., Sitorus, S. H., & Midyanti, D. M. (2019). *Sistem Pengukuran Anthropometri Badan Untuk Klasifikasi Ukuran Pakaian Menggunakan Metode Learning Vector Quantization 2*. *Jurnal Komputer dan Aplikasi Volume 07*, 1-10.
- [7] Demuth, H., & Beale, M. (2002). *Neural Network Toolbox User's Guide*. Massachusetts: The Math Work, Inc
- [8] Sato, A., & Yamada, K. (1996). *Generalized Learning Vector Quantization*. 424-426
- [9] Pham, T.-t. (2006). *Automatic Image Annotation: Towards a Fusion of Region-based and Saliency-based Models*. *Universite Pierre Et Marie Curie Master IAD*
- [10] Agustina, A. C., Suwarno, S., & Proboyekti, U. (2012). *Pengenalan Aksara Jawa Menggunakan Learning Vector Quantization (LVQ)*. 3