

# Pengenalan Pola Tulisan Tangan Suku Kata Aksara Sasak Menggunakan Metode Moment Invariant dan Support Vector Machine

(*Handwritten Sasak Ancient Script Recognition using Moment Invariant and Support Vector Machine*)

Riska Yulianti\*, I Gede Pasek Suta Wijaya, dan Fitri Bimantoro

Prodi. Teknik Informatika, Fakultas Teknik, Universitas Mataram

Jl. Majapahit 62, Mataram, Lombok NTB, INDONESIA

Email: riskakayulianti@gmail.com, gpsutawijaya@unram.ac.id, bimo@unram.ac.id

\*Penulis korespondensi

**Abstract** – *The research of Javanese and Balinese ancient script has been done by some researches. However, researches still have problems, such as image scaling, noise reduction, and image transformation. This research implemented a moment invariant and support vector machine to solve these problems especially on Sasak ancient script. The input data used in this research was 2700 handwritten Sasak ancient script. The testing was done to know the effect of thinning and the number of features by using zoning on the classification performance. Accuracy is used as a performance indicator. The highest average accuracy is 89.76%, in the second scenario, the average accuracy obtained is 92.52%.*

**Key Words:** *image thinning, zoning, sasak ancient script, moment invariant, SVM*

## I. PENDAHULUAN

Aksara merupakan suatu simbol visual yang tertera pada suatu media (kertas, kain) untuk mengungkapkan unsur-unsur yang ekspresif dalam suatu bahasa. Aksara digunakan untuk secara khusus menuliskan bahasa daerah tertentu. Salah satu bahasa daerah nusantara yang digunakan di Lombok adalah bahasa Sasak dan ditulis dengan menggunakan aksara Sasak[1].

Aksara Sasak adalah salah satu aksara tradisional nusantara yang berkembang di Lombok, Nusa Tenggara Barat dan terdiri dari 18 suku kata yang diadaptasi dari aksara Jawa dan Bali. Beberapa penelitian dilakukan untuk membahas mengenai aksara nusantara termasuk di dalamnya aksara Jawa dan aksara Bali yang berasal dari satu rumpun yang sama dengan aksara Sasak. Penelitian sebelumnya membahas mengenai aksara Jawa dan Bali yang menghasilkan akurasi klasifikasi sebesar 38% hingga 80%. Penelitian tersebut dihadapkan oleh beberapa permasalahan yaitu transformasi citra [2], perbedaan skala citra[3], dan *noise reduction*[4]. Pada penelitian ini menggabungkan 2 metode yang belum pernah dilakukan pada penelitian sebelumnya yaitu *Moment Invariant* dan SVM. Metode *Moment Invariant* dikenal menghasilkan ciri geometri objek dari pengenalan karakter yang independen

terhadap translasi, rotasi dan penskalaan citra sedangkan metode SVM merupakan metode *learning machine* yang bekerja atas prinsip *Structural Risk Minimization* dengan tujuan menemukan *hyperplane* terbaik yang memisahkan 2 buah kelas pada *input space*. Berdasarkan teori dari kedua metode tersebut diharapkan penggabungan keduanya dapat mengenali pola tulisan tangan suku kata aksara Sasak dengan tepat serta mengatasi permasalahan yang dihadapi pada penelitian sebelumnya.

## II. TINJAUAN PUSTAKA

Pada penelitian terdahulu yang terkait dengan penelitian ini yaitu penelitian mengenai pengenalan tulisan yang dibangun dengan menggunakan metode *Logika Fuzzy* dan jaringan saraf tiruan memiliki tingkat pengenalan sebesar 84,1% [5]. Penelitian serupa telah dilakukan membahas tentang pengenalan wajah dengan penggabungan *Moment Invariant* dan *Backpropagation Neural Network*. Hasil akurasi dari penelitian tersebut adalah 98,22%[6]. Penelitian ini menunjukkan bahwa *Moment Invariant* sangat baik digunakan dalam ekstraksi fitur.

Penelitian selanjutnya yang mendukung penelitian sebelumnya yaitu penelitian mengenai pengenalan huruf alphabet yang normal. Hasil akurasi data latih mencapai 90% dan angka rata-rata pengenalan sebesar 79,5% dengan huruf kecil dengan *font Times New Roman* yang diputar 90 derajat dan 270 derajat memiliki presentase terkecil dengan 57,7%. Pengenalan mencapai angka 90% ketika diputar 180 derajat baik huruf kapital maupun huruf kecil untuk semua jenis huruf [7].

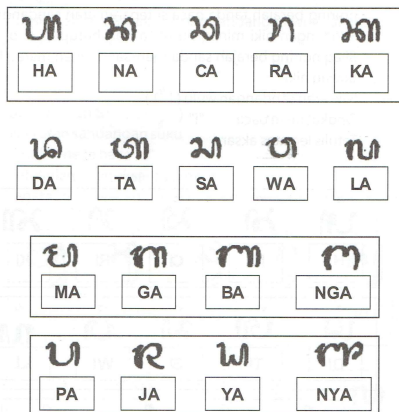
Penelitian yang terkait dengan penelitian ini juga telah dilakukan dan menunjukkan bahwa metode *Moment Invariant* dapat digunakan untuk memperoleh matrik berdimensi rendah sehingga mempercepat waktu komputasi [8]. Penelitian berikutnya mengenai metode SVM yang digunakan dalam proses pengklasifikasian citra, terdapat penelitian terdahulu yang mendukung metode

tersebut yaitu penelitian untuk membangun suatu aplikasi pengenalan wajah. Hasil penelitiannya didapat bahwa keberhasilan metode klasifikasi SVM untuk basis data Yale mencapai 100% dan untuk basis data ORL mencapai 95% [9]. Penelitian lain yang mendukung proses pengklasifikasian menggunakan metode SVM yaitu penelitian pengklasifikasian warna kulit berdasarkan ras dengan menggunakan metode SVM memberikan akurasi sebesar 92,5% [10]. Hasil akurasi dari kedua penelitian ini menunjukkan bahwa aplikasi pengenalan wajah dengan menggunakan metode klasifikasi SVM merupakan sebuah aplikasi yang cukup handal.

Berdasarkan tinjauan pustaka tersebut, dapat diketahui bahwa pengenalan pola menggunakan metode *Moment Invariant* dan SVM memiliki tingkat keberhasilan mencapai lebih dari 70%. Untuk itu, dibuat suatu hipotesa bahwa penelitian ini akan memberikan hasil yang maksimal dalam pengenalan pola aksara Sasak.

#### A. Aksara Sasak

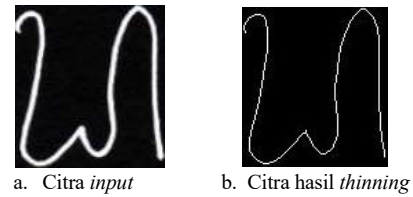
Aksara Sasak atau *hanacaraka* adalah salah satu aksara tradisional nusantara yang berkembang di Lombok, Indonesia dan terdiri dari 18 suku kata. Aksara ini digunakan untuk menulis bahasa daerah yaitu Bahasa Sasak [1]. Keseluruhan suku kata aksara Sasak tertera pada Gambar 1.



Gambar 1. Suku kata aksara Sasak

#### B. Image Thinning

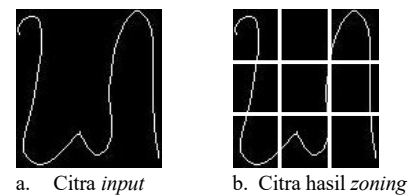
*Image thinning* adalah proses morfologi citra yang merubah bentuk asli citra biner menjadi citra yang menampilkan batas-batas obyek atau *foreground* hanya setebal satu piksel. Algoritma *thinning* secara iterative menghapus piksel-piksel pada citra biner, dimana transisi dari 0 ke 1 (atau dari 1 ke 0) terjadi sampai dengan terpenuhi suatu keadaan dimana satu himpunan dari lebar per-unit (satu pixel) terhubung menjadi suatu garis. Tujuan *image thinning* adalah untuk menghilangkan piksel-piksel yang berada didalam obyek depan pada citra biner [11]. Ilustrasi dari proses *thinning* yang dilakukan pada penelitian ini dapat dilihat pada Gambar 2.



Gambar 2. Image thinning

#### C. Zoning

*Zoning* citra karakter dapat dibagi ke dalam zona  $n \times m$ . Setiap zona memiliki fitur yang diekstraksi ke dalam vektor fitur. Tujuan dilakukan proses *zoning* adalah untuk mendapatkan nilai karakteristik per bagian yang merupakan karakteristik citra karakter secara global [12]. Proses *zoning* pada paper ini disajikan pada Gambar 3.



Gambar 3. Image zoning

#### D. Moment Invariant

*Moment Invariant* merupakan fungsi *non-linear* yang *invariant* terhadap rotasi, translasi dan skala dalam *moment* geometri citra. Jika ada sebuah citra dengan nilai intensitas adalah  $f(x,y)$  nilai  $x$  sebagai baris dan  $y$  sebagai kolom maka *Moment Invariant* yang mentransformasikan fungsi citra  $f(x,y)$  pada sistem diskrit [9].

$$m_{pq} = \sum_{x=0}^{W-1} x \sum_{y=0}^{H-1} y x^p y^q f(x,y) \quad (1)$$

dimana :

- $m$  = Moment Citra
- $p, q$  = Orde Moment
- $f$  = Nilai intensitas warna citra
- $x, y$  = Koordinat piksel
- $H, W$  = Tinggi dan lebar citra.

*Moment* citra yang *invariant* terhadap translasi citra adalah dengan orde  $m00, m01, m02, m03, m10, m11, m12, m20, m21, m31$ . Selanjutnya, ditentukan koordinat pusat citra berdasarkan *moment* citra yang dihitung dengan menggunakan Persamaan (2) dan Persamaan (3).

$$\bar{x} = \frac{m_{10}}{m_{00}} \quad (2)$$

$$\bar{y} = \frac{m_{01}}{m_{00}} \quad (3)$$

$m00$  adalah jumlah total piksel yang membentuk obyek, sedangkan  $m10$  dan  $m11$  adalah pusat massa obyek. Selanjutnya untuk memperoleh *moment invariant* terhadap rotasi maka *moment* pusat diperoleh dengan menggunakan Persamaan (4).

$$\mu_{pq} = \sum_x x \sum_y y (x - \bar{x})^p (y - \bar{y})^q \quad (4)$$

dimana :

- $\mu$  = Moment Pusat
- $\bar{x}, \bar{y}$  = Pusat Citra

*Moment* pusat yang terbentuk sensitif terhadap transformasi rotasi dan penskalaan. Maka dilakukan normalisasi terhadap *moment* pusat (5) melalui Persamaan (5) dan Persamaan (6).

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\frac{p+q}{2}}} \quad (6)$$

$$\gamma = \left(\frac{p+q}{2}\right) + 1$$

Berdasarkan normalisasi *moment* pusat dapat dihitung tujuh *vector moment invariant* dengan Persamaan (7) sampai dengan Persamaan (13).

$$\varphi_1 = \eta_{20} + \eta_{02} \quad (7)$$

$$\varphi_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \quad (8)$$

$$\varphi_3 = (\eta_{03} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \quad (9)$$

$$\varphi_4 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} + 3\eta_{03})^2 \quad (10)$$

$$\varphi_5 = (\eta_{30} - 3\eta_{12})^2(\eta_{30} - 3\eta_{12})^2[(\eta_{30} + \eta_{12})^2 + 3(\eta_{21} + \eta_{03})^2] + (3\eta_{21} - \eta_{03})^2(\eta_{21} + \eta_{03})^2[3(\eta_{30} - \eta_{12})^2 - (\eta_{21} - \eta_{03})^2] \quad (11)$$

$$\varphi_6 = (\eta_{20} - \eta_{02})^2[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}) \quad (12)$$

$$\varphi_7 = (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (3\eta_{12} - \eta_{30})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} - \eta_{03})^2] \quad (13)$$

### E. Support Vector Machine

SVM merupakan metode *learning machine* dengan prinsip kerja *Structural Risk Minimization* (SRM). Tujuan dari SVM ini adalah menemukan *hyperplane* terbaik yang memisahkan dua buah kelas pada ruang *input*. SVM adalah sistem pembelajaran yang menggunakan ruang hipotesa berupa fungsi-fungsi linier dalam sebuah ruang fitur berdimensi tinggi.

Teknik SVM berkaitan dengan *data mining* maupun *machine learning* karena memiliki kemampuan dalam memprediksi kelas suatu data baru. Berikut pembahasan dengan kasus klasifikasi yang secara linier bisa dipisahkan

Fungsi pemisah untuk model SVM pada paper ini adalah fungsi linier. Fungsi ini didefinisikan sebagai:

$$f(x) = w \cdot x + b \quad (14)$$

Setiap data latih dinyatakan oleh  $(x_i, y_i)$ , di mana  $i=1,2,\dots, N$ , dan  $x_i = \{x_{i1}, x_{i2}, \dots, x_{iq}\}^T$  merupakan fitur set untuk data latih ke- $i$ .  $y_i \in \{-1, +1\}$  menyatakan label kelas. *Hyperplane* klasifikasi linier SVM, seperti pada Gambar 2.7, dinotasikan dengan:

$$w \cdot x_i + b = 0 \quad (15)$$

$W$  dan  $b$  adalah parameter model.  $w \cdot x_i$  merupakan *inner-product* dalam antara  $w$  dan  $x_i$ . Data  $x_i$  yang masuk ke dalam kelas -1 adalah data yang memenuhi pertidaksamaan berikut:

$$w \cdot x_i + b \leq -1 \quad (16)$$

Sementara data  $x_i$  yang masuk ke dalam kelas +1 adalah data yang memenuhi pertidaksamaan berikut :

$$w \cdot x_i + b \geq +1 \quad (17)$$

### III. METODE PENELITIAN

#### A. Alat dan Bahan

Alat yang diperlukan pada penelitian ini adalah :

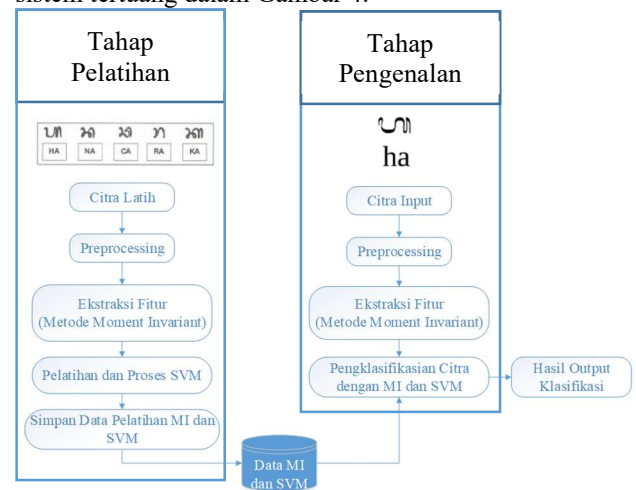
1. *Hardware* Laptop: menggunakan Intel® Core i3® CPU 2020, 2.4 GHz, RAM 2 GB
2. *Operating system*: menggunakan *Operating System Windows 10 Ultimate* 64-bit.
3. *Software java*: menggunakan *software Netbeans IDE 8.0* diperlukan untuk pembuatan program dengan menggunakan bahasa pemrograman Java.
4. *Scanner* Canon LiDE 120. Resolusi 2400 dpi. Maksimum kertas *letter*.

Sedangkan bahan yang diperlukan adalah sebagai berikut:

1. Tulisan tangan suku kata aksara Sasak ditulis oleh 15 orang berbeda sebanyak 10 kali sehingga jumlahnya adalah  $15 \times 10 \times 18 = 2700$  tulisan tangan, ditulis pada kertas putih menggunakan alat tulis yang sama.
2. Citra aksara Sasak dalam format JPEG diperoleh dari hasil *scanning* tulisan tangan aksara Sasak yang berukuran maksimal 300x300 piksel.

#### B. Rancangan Sistem

Perancangan sistem merupakan tahap yang penting dalam mengaplikasikan suatu konsep agar dalam pembuatannya dapat berjalan secara sistematis, terstruktur, dan rapi sehingga hasil program dapat berjalan sesuai dengan apa yang dikehendaki. Proses perancangan sistem tertuang dalam Gambar 4.



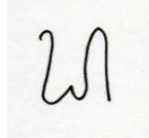
Gambar 4. Blok Diagram Sistem

Secara umum Gambar 4 menggambarkan proses pengenalan sistem, yang terdiri atas 2 tahapan utama yaitu pelatihan dan pengenalan.

#### B.1 Tahap Pelatihan

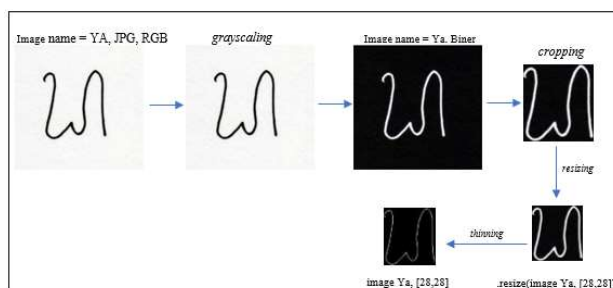
Proses pelatihan bertujuan untuk menemukan model sistem pengenalan citra huruf sasak dengan parameter optimalnya. Proses ini terdiri atas beberapa sub-proses, yaitu:

1. *Input* Citra Latih: Menyiapkan citra digital tulisan tangan 18 suku kata aksara Sasak untuk proses pelatihan. Contoh citra input seperti pada Gambar 5.



Gambar 5. Citra input tahap pelatihan

2. *Preprocessing*: *Preprocessing* merupakan tahap awal pengolahan citra asli sebelum citra tersebut diekstraksi pada proses pelatihan maupun pengklasifikasian yang dimaksudkan untuk memanipulasi citra agar sesuai dengan hasil yang diinginkan. Blok diagram dari *preprocessing* dapat dilihat pada Gambar 6. Pada proses ini digunakan data citra dari 18 suku kata aksara Sasak. Data citra tersebut merupakan citra dengan tipe RGB dan format JPG. Proses *preprocessing* dimulai dengan melakukan proses *resizing* untuk mengubah ukuran citra input menjadi 28x28 piksel. Kemudian citra hasil *resizing* tersebut di manipulasi dengan melakukan proses *grayscale*. *Grayscale* citra dilakukan dengan cara mengambil semua piksel pada gambar kemudian warna tiap piksel akan diambil informasi mengenai 3 warna dasar yaitu merah, biru dan hijau. Ketiga warna dasar ini akan dijumlahkan kemudian dibagi tiga sehingga didapat nilai rata-rata. Nilai rata-rata inilah yang akan dipakai untuk memberikan warna pada piksel gambar sehingga warna menjadi *grayscale*. Hasil citra *grayscale* kemudian di konversi menjadi format biner dengan proses *binarization*. Setelah proses *binarization* kemudian dilakukan proses *cropping* dengan tujuan memotong *noise* pada citra biner sehingga diperoleh citra hasil *cropping* berupa informasi citra tanpa adanya *noise* atau *universe of discourse*. Hasil *cropping* citra kemudian di *resize* kembali menjadi ukuran 28x28 piksel. Citra hasil *resize* tersebut kemudian dilakukan proses *thinning* untuk memperoleh informasi penting dari citra berupa kerangka (*skeleton*). Jadi hasil yang diperoleh dari proses *preprocessing* berupa citra biner *thinning* dengan ukuran 28x28 piksel.

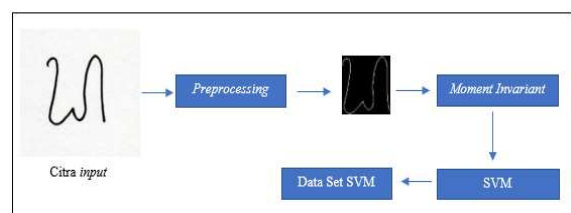


Gambar 6. Proses *preprocessing* pada tahap pelatihan

3. *Ekstraksi Fitur*: Melakukan ekstraksi fitur dengan menggunakan metode *Moment Invariant* pada citra

data latih. Hasil yang diperoleh dari proses ekstraksi fitur ini adalah tujuh nilai *moment invariant*.

4. *Pelatihan SVM*: Melakukan pelatihan SVM untuk mendapatkan bobot data latih yang dapat digunakan untuk mengenali atau mengklasifikasikan aksara Sasak. Secara sederhana, proses pelatihan ditunjukkan pada Gambar 7. Gambar 7 menunjukkan bahwa citra input hasil *preprocessing* berupa citra biner dengan ukuran 28x28 piksel. Citra ini kemudian dilanjutkan ke dalam proses ekstraksi fitur menggunakan perhitungan metode *Moment Invariant* sesuai dengan Persamaan (7) hingga (13) sehingga diperoleh nilai fitur berupa 7 buah nilai *Moment Invariant*. Setelah didapatkan nilai fitur data dimasukkan pada *libSVM*. Hasil proses pelatihan disimpan dalam format xml sebagai *Data Set SVM*. Pembagian kelas dalam SVM menggunakan *tree structure*. Pada proses pertama, 18 buah kelas dikelompokkan sebagai 2 kelas. Kelas pertama merupakan setengah bagian dari jumlah aksara Sasak (9 suku kata) yang memiliki kemiripan fisik satu sama lain. Pada proses kedua, aksara selain dari aksara yang berada di kelas pertama menjadi bagian dari kelas kedua, sehingga diperoleh 2 buah kelas dari kedua proses tersebut. Proses ini terus berlanjut hingga diperoleh kelas untuk 1 suku kata aksara Sasak. *Library* yang digunakan yaitu *openCV*. *Arsitektur SVM* disimpan dalam bentuk xml sehingga didapatkan 17 buah xml dari hasil pelatihan.



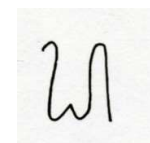
Gambar 7. Proses pelatihan SVM

5. *Simpan data pelatihan MI dan SVM*: hasil dari proses ekstraksi dan proses pelatihan SVM akan disimpan sebagai bobot data latih.

## B.2 Tahap Pengenalan

Merupakan tahapan untuk mengenali setiap citra masukkan kedalam kelas tertentu yang bersesuaian dengan data kelas pelatiha. Sub-proses dari tahapan pengenalan yaitu:

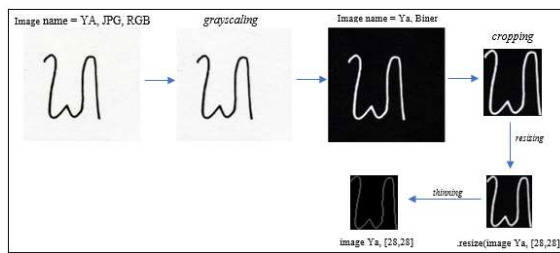
1. *Input Citra*: Memasukkan data citra tulisan tangan suku kata aksara Sasak (Gambar 8) yang akan di klasifikasi ke dalam sistem.



Gambar 8. Citra input tahap pengenalan

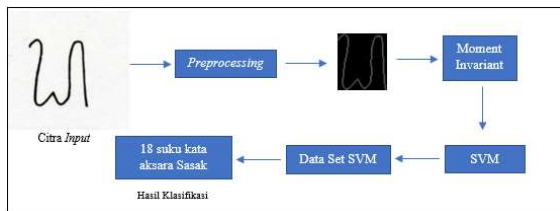
2. *Preprocessing*: Melakukan proses yang bertujuan untuk mengubah besar citra digital sehingga memiliki ukuran yang sama. Hasil yang diperoleh dalam proses

*preprocessing* ini adalah citra biner *thinning* dengan ukuran 28x28 piksel. Algoritma *preprocessing* disajikan pada Gambar 9.



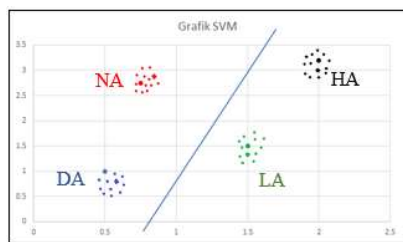
Gambar 9. Proses *preprocessing* pada tahap pengenalan

- Ekstraksi Fitur: Melakukan ekstraksi fitur dengan menggunakan metode *Moment Invariant* sehingga diperoleh tujuh nilai *moment invariant*.
- Pengklasifikasian Citra dengan MI dan SVM: Melakukan klasifikasi berdasarkan hasil ekstraksi fitur citra *input* tulisan tangan aksara sasak dengan metode SVM yang alurnya ditunjukkan pada Gambar 10.



Gambar 10. Alur proses pengenalan

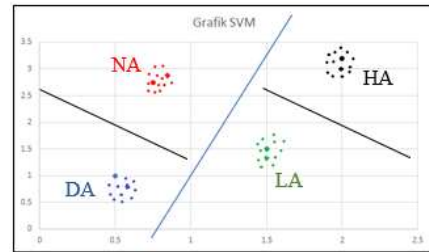
Setelah didapatkan nilai fitur dengan metode *Moment Invariant*, SVM akan membaca kembali data set SVM (xml) yang telah disimpan sebelumnya pada proses pelatihan. Ilustrasi klasifikasi SVM ditunjukkan pada Gambar 11 dan 12.



Gambar 11. Ilustrasi garis klasifikasi SVM pertama

Pada SVM pertama akan dilakukan pengklasifikasian antara kelas 1 (9 suku kata pertama) dengan kelas 2 (9 suku kata kedua). Jika hasil menunjukkan citra merupakan kelas 1 maka proses klasifikasi dilanjutkan ke SVM berikutnya. Perilaku serupa dilakukan jika hasil menunjukkan citra merupakan kelas 2 maka proses klasifikasi dilanjutkan pada SVM berikutnya. Pada SVM kedua akan dilakukan pengklasifikasian antara kelas 1 (NA) dengan kelas (DA). Jika respon bernilai -1 maka citra terklasifikasi sebagai kelas NA dan sebaliknya jika respon bernilai 1 maka citra terklasifikasi sebagai kelas DA. Perilaku serupa dilakukan jika hasil menunjukkan citra merupakan kelas 2(HA) dengan kelas (LA). Jika respon bernilai -1 maka

citra terklasifikasi sebagai kelas HA dan sebaliknya jika respon bernilai 1 maka citra terklasifikasi sebagai kelas LA.



Gambar 12. Ilustrasi garis klasifikasi SVM kedua

- Hasil *Output* Klasifikasi: Diperoleh hasil keluaran klasifikasi aksara Sasak berupa huruf latin citra asli aksara Sasak.

### C. Teknik Pengujian

Teknik pengujian yang digunakan oleh sistem harus diuji dengan parameter-parameter statistik yang dikumpulkan. Parameter-parameter tersebut meliputi tingkat akurasi. Tingkat akurasi merupakan nilai kesesuaian antara *input*-an dengan hasil klasifikasi. Untuk menghitung parameter akurasi digunakan Persamaan (18).

$$\text{Akurasi} = \frac{TP+T}{TP+TN+FP+FN} \times 100\% \quad (18)$$

dimana,

*TP (True Positive)*: banyaknya hasil klasifikasi benar untuk suatu kelas yang bernilai *positive*,

*TN (True Negative)*: banyaknya hasil klasifikasi benar untuk suatu kelas yang bernilai *negative*,

*FP (False Positive)* adalah banyaknya hasil klasifikasi salah untuk suatu kelas yang bernilai *negative*.

*FN (False Negative)* adalah banyaknya hasil klasifikasi salah untuk suatu kelas yang bernilai *positive*

Dalam penentuan presisi data latih dan data uji digunakan pendekatan *K-Fold Cross Validation*. *K-Fold Cross Validation* adalah pendekatan yang digunakan untuk mengestimasi performa atau kualitas suatu model. Cara bekerjanya dengan membagi sebuah data sampel menjadi k bagian secara acak dan sama besarnya. Satu bagian akan menjadi data uji dan model akan melakukan *fitting* terhadap k bagian lainnya. Proses ini dilakukan sebanyak k kali menggunakan k *validation* yang berbeda.

## IV. HASIL DAN PEMBAHASAN

Pengujian terhadap sistem dilakukan secara offline. Tujuan dari pengujian ini adalah untuk mengetahui nilai akurasi yang diberikan oleh metode *Moment Invariant* dalam proses ekstraksi fitur dan SVM dalam pengklasifikasian citra.

### A. Pengujian

Proses pengujian dilakukan dalam 2 skenario berbeda yaitu :

- Skenario pengujian pertama: Pada proses pelatihan, data yang digunakan yaitu  $10 \times 10 \times 18 = 1800$  data latih

maka untuk proses pengujian data yang digunakan yaitu  $10 \times 5 \times 18 = 900$  data uji.

- Skenario pengujian kedua: Proses pengujian menggunakan *K-Fold Cross Validation* dengan menggunakan total data sampel yaitu  $15 \times 10 \times 18 = 2700$  data sampel dimana K yang digunakan sebanyak 10 *K-Fold*. Tabel I menunjukkan tahapan pengujian dengan menggunakan *K-Fold Cross Validation*.

TABEL I. TAHAP PENGUJIAN K-FOLD

Tahap 1	Tahap 2	Tahap 3	Tahap 4	Tahap 5
Fold1 test	Fold2 test	Fold3 test	Fold4 test	Fold5 test
Fold2 train	Fold3 train	Fold4 train	Fold5 train	Fold6 train
Fold3 train	Fold4 train	Fold5 train	Fold6 train	Fold7 train
Fold4 train	Fold5 train	Fold6 train	Fold7 train	Fold8 train
Fold5 train	Fold6 train	Fold7 train	Fold8 train	Fold9 train
Fold6 train	Fold7 train	Fold8 train	Fold9 train	Fold10 train
Fold7 train	Fold8 train	Fold9 train	Fold10 train	Fold1 train
Fold8 train	Fold9 train	Fold10 train	Fold1 train	Fold2 train
Fold9 train	Fold10 train	Fold1 train	Fold2 train	Fold3 train
Fold10 train	Fold1 train	Fold2 train	Fold3 train	Fold4 train
Tahap 6	Tahap 7	Tahap 8	Tahap 9	Tahap 10
Fold6 test	Fold7 test	Fold8 test	Fold9 test	Fold10 test
Fold7 train	Fold8 train	Fold9 train	Fold10 train	Fold1 train
Fold8 train	Fold9 train	Fold10 train	Fold1 train	Fold2 train
Fold9 train	Fold10 train	Fold1 train	Fold2 train	Fold3 train
Fold10 train	Fold1 train	Fold2 train	Fold3 train	Fold4 train
Fold1 train	Fold2 train	Fold3 train	Fold4 train	Fold5 train
Fold2 train	Fold3 train	Fold4 train	Fold5 train	Fold6 train
Fold3 train	Fold4 train	Fold5 train	Fold6 train	Fold7 train
Fold4 train	Fold5 train	Fold6 train	Fold7 train	Fold8 train
Fold5 train	Fold6 train	Fold7 train	Fold8 train	Fold9 train

Beberapa variasi pengujian akan dilakukan untuk mengetahui performa dari teknik klasifikasi. Variasi pengujian tersebut antara lain:

- Pengaruh *thinning* terhadap performa
- Pengaruh jumlah fitur terhadap performa dengan menggunakan proses *zoning*

Parameter yang digunakan untuk menentukan performa adalah akurasi yang dihitung menggunakan persamaan (18).

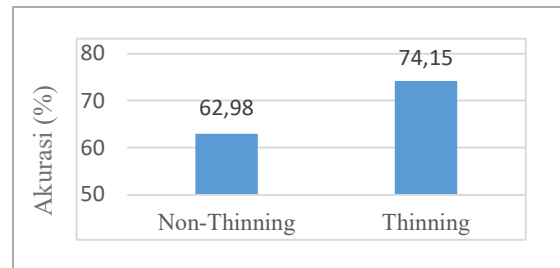
### A.1 Pengujian Skenario Pertama

#### A.1.1 Pengaruh Thinning Terhadap Performa

*Thinning* merupakan proses penghilangan piksel terluar dengan cara *iterative deletion* hingga menghasilkan sebuah piksel tulang (skeleton). Algoritma *thinning* secara iteratif ‘menghapus’ piksel-piksel pada citra biner, dimana transisi dari 0 ke 1 terjadi sampai dengan terpenuhi suatu keadaan dimana satu himpunan dari lebar per unit (satu piksel) terhubung menjadi suatu garis. Hal ini dimaksudkan untuk mengambil informasi yang penting dari citra. Pengaruh *thinning* terhadap performa sistem disajikan dalam bentuk grafik dituang pada Gambar 13.

Gambar 13 menunjukkan bahwa proses *preprocessing* citra sebelum dilakukan proses *thinning* memiliki hasil akurasi 62.98%, Setelah dilakukan proses *thinning* menggunakan metode Zhang-Suen nilai akurasi meningkat sebesar 11.17% dari nilai sebelumnya yaitu 75.25%. Peningkatan hasil akurasi disebabkan oleh pengurangan jumlah *noise* pada citra dengan proses *thinning* sehingga citra yang dihasilkan berupa citra

dengan informasi yang penting saja. Hasil tersebut menunjukkan bahwa proses *preprocessing* dengan menggunakan *thinning* ternyata dapat meningkatkan hasil akurasi klasifikasi citra, akan tetapi hasil yang diperoleh dari proses *preprocessing* tersebut belum mampu memberikan hasil yang maksimal dalam klasifikasi.



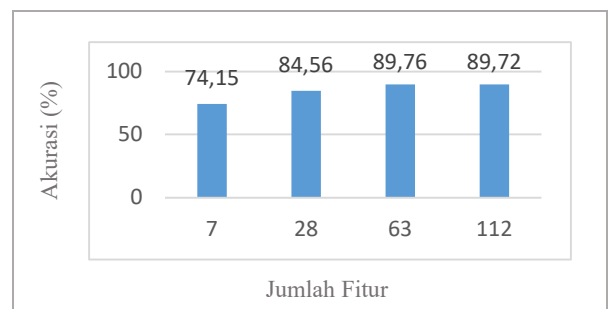
Gambar 13. Grafik performa menggunakan *thinning*

#### A.1.2. Pengaruh Jumlah Fitur Terhadap Performa Dengan Menggunakan Proses Zoning

Tujuan pengujian ini yaitu untuk mengetahui pengaruh penggunaan setiap fitur yang ada dengan performa sistem klasifikasi. Pada pengujian ini, menggunakan statistik fitur berupa 7 nilai *Moment Invariant*. Untuk mengetahui pengaruh fitur terhadap performa dilakukan manipulasi jumlah fitur dimana fitur yang digunakan yaitu 7 fitur, 28 fitur, 63 fitur dan 112 fitur. Jumlah fitur tersebut dilakukan dengan menggunakan proses *zoning*.

*Zoning* merupakan proses pembagian citra menjadi beberapa bagian yang sama. Pada penelitian ini data *input* yang digunakan berukuran  $28 \times 28$  piksel. Kemudian dilakukannya pembagian citra menjadi 4 bagian, 9 bagian dan 16 bagian yang sama. Selanjutnya *input* data yang sudah terblok dimasukkan ke proses ekstraksi fitur. Dari proses tersebut diperoleh fitur sejumlah 4-16 kali dibandingkan dengan sebelum di *zoning*.

Dari proses tersebut didapatkan hasil klasifikasi dari beberapa fitur yang ditunjukkan pada Gambar 14.



Gambar 14. Grafik performa menggunakan berbagai jumlah fitur

Gambar 14 menggambarkan bahwa akurasi meningkat seiring dengan bertambahnya jumlah fitur yang digunakan. Tingkat akurasi maksimum terdapat pada jumlah fitur 63 dengan hasil akurasi klasifikasi 89.76%. Grafik pada Gambar 14 juga membuktikan bahwa proses *zoning* ternyata mampu meningkatkan hasil akurasi klasifikasi meningkat 15.61% dari hasil akurasi

sebelumnya yaitu 74.15%. Peningkatan hasil akurasi dengan *zoning* dikarenakan jika menggunakan *zoning*, citra dibagi menjadi beberapa bagian yang sama besar sebanyak sekian wilayah sehingga jumlah fitur yang dihasilkan dari proses *zoning* menjadi lebih banyak dibandingkan dengan hanya menggunakan 1 buah citra saja. Peningkatan jumlah fitur sama dengan meningkatkan informasi mengenai citra sehingga ciri-ciri citra menjadi lebih jelas.

A.2. Pengujian Skenario Kedua

A.2.1. Pengaruh Thinning Terhadap Performa

Pengujian kedua ini menggunakan proses K-Fold *Cross Validation* dengan K=10. Jika total data *input* yang digunakan yaitu 2700 data maka data *input* untuk masing-masing K-Fold sejumlah 270 data yang terdiri dari masing-masing suku kata berisi 5 data. Pengaruh *thinning* terhadap performa sistem pengenalan disajikan pada pada Tabel II.

TABEL II. TABEL PERFORMA MENGGUNAKAN THINNING DENGAN K-FOLD

Tahap Pengujian K-Fold	Non-Thinning	Thinning
	Akurasi (%)	Akurasi (%)
Tahap 1	67.58	76.43
Tahap 2	66.36	77.85
Tahap 3	71.68	76.40
Tahap 4	61.30	75.70
Tahap 5	62.64	78.04
Tahap 6	64.20	77.48
Tahap 7	63.68	77.16
Tahap 8	69.66	77.24
Tahap 9	63.36	78.18
Tahap 10	62.90	77.07
Rata-rata	65.34	77.16

Tabel II menunjukkan persentase akurasi data *input* dengan dan tanpa menggunakan *thinning*. Hasil persentase dari kedua pengujian tersebut ternyata membuktikan bahwa *thinning* mampu meningkatkan persentase akurasi sebanyak 11.82% dibandingkan tanpa menggunakan proses *thinning*.

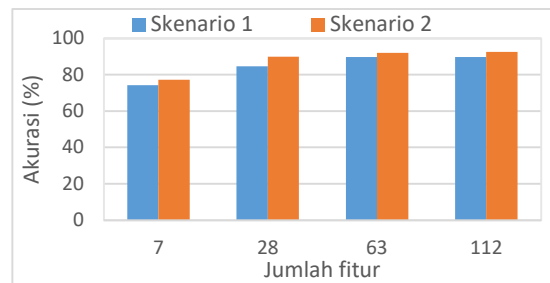
A.2.2. Pengaruh Jumlah Fitur Terhadap Performa Dengan Menggunakan Proses Zoning

Tujuan pengujian ini yaitu untuk mengetahui pengaruh penggunaan setiap fitur yang ada dengan performa sistem klasifikasi. Pada pengujian ini, menggunakan statistik fitur berupa 7 nilai *Moment Invariant*. Untuk mengetahui pengaruh fitur terhadap performa dilakukan manipulasi jumlah fitur dimana fitur yang digunakan yaitu 7 fitur, 28 fitur, 63 fitur dan 112 fitur. Jumlah fitur tersebut dilakukan dengan menggunakan proses *zoning*. Dari proses tersebut didapatkan hasil klasifikasi dari beberapa fitur yang ditunjukkan pada Tabel III. Berdasarkan persentase akurasi yang ditunjukkan pada Tabel III, jumlah fitur terbaik dengan menggunakan pengujian K-Fold adalah fitur 112 dengan persentase yaitu 92.52%. Tabel III juga menunjukkan bahwa dengan menggunakan pengujian K-Fold persentase akurasi meningkat sebanyak 15.37% dibandingkan dengan hanya menggunakan 7 fitur.

Berdasarkan kedua skenario pengujian yang dilakukan, hasil dari kedua skenario tersebut dituang pada grafik dalam Gambar 15.

TABEL III. TABEL PERFORMA MENGGUNAKAN BERBAGAI JUMLAH FITUR DENGAN K-FOLD

Tahap Pengujian K-Fold	7 fitur	28 fitur	63 fitur	112 fitur
	Akurasi (%)	Akurasi (%)	Akurasi (%)	Akurasi (%)
Tahap 1	76.43	88.81	92.40	92.62
Tahap 2	77.85	88.90	91.74	92.73
Tahap 3	76.40	89.78	92.00	92.12
Tahap 4	75.70	90.15	92.53	92.10
Tahap 5	78.04	90.64	91.78	93.21
Tahap 6	77.48	89.30	91.90	91.58
Tahap 7	77.16	88.92	92.82	91.63
Tahap 8	77.24	91.50	91.61	93.73
Tahap 9	78.18	90.23	91.03	93.06
Tahap 10	77.07	89.60	92.28	92.53
Rata-rata	77.16	89.78	92.00	92.53



Gambar 15. Grafik performa 2 skenario pengujian

Sebaran data hasil klasifikasi yang ditunjukkan pada Gambar 15 menunjukkan bahwa hasil akurasi dengan menggunakan K-Fold lebih tinggi dibandingkan jika tanpa menggunakan K-Fold. Hasil akurasi K-Fold terbaik berada pada fitur 112 dengan nilai akurasi 92.52%. Pada pengujian pertama, akurasi dari jumlah fitur 63 yaitu 89.76% menurun pada jumlah fitur 112 yaitu 89.72%. Secara teori, semakin tinggi jumlah fitur yang diperoleh semakin tinggi pula identitas citra yang diketahui sehingga memudahkan dalam proses pengklasifikasian. Penurunan nilai akurasi pada pengujian pertama disebabkan karena SVM merupakan metode dengan sistem pembelajaran yang menggunakan ruang hipotesa berupa fungsi-fungsi linier, ketika SVM memprediksi kelas suatu data baru yang memiliki bobot berbeda dengan kelas yang seharusnya data tersebut beres, maka SVM akan mengklasifikasikan data baru tersebut kedalam suatu kelas yang baru. Berdasarkan hasil yang ditunjukkan tersebut dengan kesesuaian teori mengenai metode SVM yang menjelaskan bahwa konsep klasifikasi dengan SVM adalah sebagai usaha mencari *hyperplane* terbaik yang berfungsi sebagai pemisah dua buah kelas data pada *input space* dimana *hyperplane* merupakan pemisah terbaik antara kedua kelas tersebut dapat ditemukan dengan mengukur *margin hyperplane* dan mencari titik maksimalnya. Teori tersebut membuktikan bahwa SVM dalam penelitian ini ternyata mampu melakukan klasifikasi lebih dari dua kelas dengan baik.

## V. PENUTUP

### A. Kesimpulan

Berdasarkan penelitian pada BAB sebelumnya, maka dapat disimpulkan bahwa:

1. Penelitian dilakukan terhadap beberapa jumlah fitur yaitu 7, 28, 63 dan 112 fitur. Pada skenario pengujian pertama, diperoleh fitur terbaik yaitu 63 fitur dengan nilai akurasi klasifikasi yaitu 89.76% dan pada skenario pengujian kedua diperoleh fitur terbaik yaitu 112 fitur dengan nilai akurasi klasifikasi yaitu 92.52%.
2. Proses ekstraksi fitur *moment invariant* dengan penambahan proses *thinning* mampu meningkatkan akurasi klasifikasi dari 65.34% hingga 77.16% yaitu sebesar 11.82%.
3. Penambahan jumlah fitur dengan proses *zoning* mampu meningkatkan akurasi klasifikasi dari 74.15% hingga 89.76% yaitu sebesar 15.61%.
4. Proses *Preprocessing* dengan menggunakan *thinning* dalam sistem ini membantu mengurangi gangguan pada tulisan tangan suku kata sehingga dengan menggunakan *thinning*, *preprocessing* hanya melakukan pengambilan fitur-fitur yang dikehendaki.
5. Proses *zoning* dalam sistem ini membantu meningkatkan jumlah fitur hingga 4-16 kali dari jumlah fitur citra aslinya. Peningkatan jumlah fitur membantu meningkatkan identitas citra sehingga sistem ini mampu mengenali tulisan tangan suku kata aksara Sasak.
6. Pengklasifikasian data dimensi tinggi dengan pembagian kelas lebih dari dua pada SVM dalam penelitian ini ternyata mampu memberikan hasil akurasi yang baik.

### B. Saran

Jika dilakukan penelitian lebih lanjut pada kasus ini dapat mempertimbangkan saran-saran dan perubahan sebagai berikut:

1. Modifikasi proses dapat dilakukan dengan menambahkan proses pengolahan citra lainnya seperti *filtering* dan *sharpening* atau pengolahan lainnya agar mendapatkan informasi citra yang lebih baik.
2. Metode ekstraksi fitur lainnya yang mampu memberikan jumlah fitur yang lebih banyak dari metode *Moment Invariant*.
3. Modifikasi parameter dapat dilakukan dengan mengubah ukuran alat tulis atau ukuran tulisan tangan aksara Sasak.
4. Diperlukan sistem yang independen terhadap translasi, rotasi dan penskalaan setelah dilakukan proses *zoning*.
5. Diperlukan pembagian kelas yang lebih lengkap lagi untuk kelas yang tidak dikenali sehingga sistem mampu memberikan hasil klasifikasi tidak dikenali untuk setiap citra *input* selain aksara Sasak.

## REFERENCES

- [1] Bahrie., H. Sudirman & Lalu Ratmaja., (2013), *Bahan Ajar Muatan Lokal, Gumi Sasak*, KSU "Prima Guna", Lombok Timur.
- [2] Hidayat, Akik., Rahmi Nur Shofa., (2016), *Self Organizing Maps (SOM) Suatu Metode Untuk Pengenalan Aksara Jawa*. Vol 1, No. 1.
- [3] Wibowo, Madha Christian., I Dewa Gede Rai Mardiana., Sandy Wirakusuma, (2015), *Pengenala Pola Tulisan Tangan Aksara Jawa Menggunakan Multi Layer Perceptron*, STMIK AMIKOM. Yogyakarta.
- [4] Hermanto, I Gede Rudy., Tjokorda Agung Budi Wirayuda., Retno Novi Dayawati ., (2008), *Analisis Dan Implementasi Pengenalan Huruf Bali Menggunakan Modified Direction Feature Dan Jaringan Syaraf Tiruan*, Fakultas Teknik Informatika, Universitas Telkom
- [5] Wibowo, Aditya., Achmad Hidayatno., Ajud Ajulian, (2015), *Pengenalan Huruf Jawa Tulisan Tangan Menggunakan Jaringan Saraf Tiruan Perambatan Balik Dengan Fuzzy Feature Extraction*, Fakultas Teknik, Universitas Diponegoro, Semarang.
- [6] Sholahuddin, A. (2012). *Metode Moment Invariant dan Backpropagation Neural Network pada Pengenalan Wajah*, Prodi Teknik Informatika, Universitas Padjadjaran Jatinangor.
- [7] Ananggadipta, R., Hidayatno, A., & Zahra, A. A., (2014), *Pengenalan Huruf Alfabet Menggunakan 7 Moment Invariant Hu dan Jaringan Saraf Tiruan LVQ (Learning Vector Quantization)*, Jurusan Teknik Elektro, Universitas Diponegoro Semarang.
- [8] Salambue, Roni., (2013), *Pengenalan Pola Tanda Tangan dengan Metode Moment Invariant dan Euclidean Distance* Roni Salambue, Jurusan Matematika, FMIPA, Universitas Riau.
- [9] Andriati, S., 2011, *Aplikasi Pengenalan Wajah Menggunakan Metode Support Vector Machine*, Jurusan Teknik Elektro Politeknik Negeri Bali.
- [10] Ilyarisma, R., (2015), *Pengklasifikasian Warna Kulit Berdasarkan Ras Menggunakan Algoritma Support Vector Machine (SVM)*, Jurusan Teknik Elektro, Universitas Mataram.
- [11] Zurnawita, Zulharbi Suar, (2009), *Algoritma Image Thinning*, Teknik Elektro, Politeknik Negeri Padang.
- [12] David, (2011), *Perancangan Perangkat Lunak Pengenalan Pola Karakter Menggunakan Jaringan Syaraf Tiruan Perceptron*, Program Studi Teknik Informatika, Sekolah Tinggi Manajemen Informatika dan Komputer Pontianak, Kalimantan Barat.