

**Pengenalan Pola Tulisan Tangan Aksara Bima Menggunakan Metode
Ekstraksi Ciri Local Binary Pattern, Metode Reduksi Data Latih K-
Support Vector Nearest Neighbour, Dan Metode Klasifikasi K-Nearest
Neighbour**

Tugas akhir
untuk memenuhi sebagian persyaratan
mencapai derajat Sarjana S-1 Program Studi Teknik Informatika



Oleh:
Muhammad Ilham Fidatama
FID 016 057

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS MATARAM
2020

DAFTAR ISI

DAFTAR ISI.....	i
DAFTAR GAMBAR.....	iii
DAFTAR TABEL.....	iv
BAB I PENDAHULUAN.....	1
1.1 Latar belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah.....	3
1.4 Tujuan.....	3
1.5 Manfaat.....	3
1.6 Sistematika Penulisan.....	4
BAB II TINJAUAN PUSTAKA DAN DASAR TEORI	5
2.1 Tinjauan Pustaka	5
2.2 Dasar Teori.....	7
2.2.1 Local Binary Pattern.....	7
2.2.2 K-Support Vector Nearest Neighbour	7
2.2.3 K-Nearest Neighbour.....	7
2.2.4 Performa	8
BAB III METODOLOGI PENELITIAN	10
3.1 Alat dan Bahan Penelitian	10
3.2 Rancangan Penelitian	11
3.3 Studi Literatur	11
3.4 Pengumpulan Data	11
3.5 Rancangan Algoritma.....	11
3.5.1 Tahap <i>Pre-processing</i>	12
3.5.2 Tahap Ekstraksi Ciri	12
3.5.3 Tahap Reduksi Data Latih	14
3.5.4 Tahap Klasifikasi.....	16
3.6 Pengujian Sistem.....	17
3.7 Skenario Pengujian.....	18
BAB IV HASIL DAN PEMBAHASAN	19
4.1 Akuisisi Data	19
4.2 Data Pre-processing.....	23

4.3 Ekstraksi Ciri	24
4.4 Mekanisme Pengujian	24
4.5 Hasil Pengujian	25
BAB V KESIMPULAN DAN SARAN	34
5.1 Kesimpulan.....	34
5.2 Saran.....	34
DAFTAR PUSTAKA	35

DAFTAR GAMBAR

Gambar 2.1 Ilustrasi penentuan piksel tetangga searah jarum jam.....	7
Gambar 3.1 Citra pengambilan data tulisan tangan aksara Bima	10
Gambar 3.2 Diagram alir penelitian.....	11
Gambar 3.3 Blok diagram rancangan algoritma	12

DAFTAR TABEL

Tabel 2.1 Model Confussion matrix 2 kelas	8
Tabel 3.1 Hubungan antara jumlah piksel tetangga dan radius	14
Tabel 3.2 Tabel contoh data latih.....	15

BAB I

PENDAHULUAN

1.1 Latar belakang

Indonesia merupakan negara kepulauan dengan berbagai macam suku dan budaya. Setiap suku memiliki berbagai jenis bahasa dan aksara yang berbeda-beda. Salah satu suku yang ada di Indonesia adalah suku Mbojo (Bima). Suku Mbojo bertempat di Pulau Sumbawa bagian timur, yang sekarang terbagi menjadi tiga bagian yaitu Kota Bima, Kabupaten Bima, Kabupaten Dompu. Suku Mbojo memiliki sistem penulisan atau aksara yang biasa disebut dengan aksara Mbojo atau aksara Bima[1]. Aksara Bima digunakan pada pembelajaran sekolah dasar yang dicantumkan sebagai mata pelajaran. Setelah menyelesaikan jenjang studi sekolah dasar, tidak ada lagi pelajaran aksara Bima pada jenjang sekolah berikutnya. Dengan berkurangnya penggunaan aksara Mbojo pada saat ini ditakutkan dapat menyebabkan hilangnya kebudayaan yang diwariskan oleh Suku Mbojo.

Berdasarkan hasil survei yang telah dilakukan dengan jumlah responden sebanyak 87 orang dengan rentang usia 17 sampai 38 tahun, yang terdiri dari 81 orang responden yang berasal dari Bima. Berdasarkan survei yang didapatkan dari responden yang berasal dari Bima, sebanyak 45.7% diantaranya masih belum mengetahui tentang keberadaan aksara Bima, dan sebanyak 48.1% diantaranya belum pernah mempelajari aksara Bima. Berdasarkan data tersebut dapat disimpulkan bahwa masih banyak penduduk Bima yang belum familiar dengan penggunaan aksara Bima, bahkan di Pendidikan formal pun tidak seluruhnya mendapatkan pelajaran muatan lokal yang mempelajari aksara Bima. Hal ini dapat mengakibatkan hilangnya warisan salah satu Budaya Bima yaitu aksara Bima. Oleh karena itu penting untuk dilakukan edukasi tentang bentuk dan pola tulisan aksara Mbojo kepada masyarakat agar dapat mempertahankan kebudayaan yang telah diwariskan oleh suku Mbojo.

Dengan berkembangnya teknologi digital, saat ini media pembelajaran dapat diakses melalui aplikasi ataupun website edukasi. Dengan memasukkan aksara Bima ke media pembelajaran berbasis teknologi digital, maka dapat mempertahankan budaya sistem penulisan aksara Bima. Pengenalan pola tulisan tangan aksara Bima merupakan salah satu solusi yang dapat dihadirkan dalam pengembangan aplikasi dan website edukasi. Pengenalan pola adalah pengelompokan data numerik dan simbolik (seperti citra) secara otomatis oleh mesin. Tujuan pengenalan pola adalah untuk mengenali atau mengklasifikasikan suatu citra tertentu dalam suatu kelas.

Tahap awal untuk melakukan pengenalan pola pada citra digital adalah dengan melakukan ekstraksi ciri terhadap citra digital. Salah satu proses ekstraksi ciri pada pengenalan pola adalah LBP (*Local Binary Pattern*). Cara kerja LBP adalah dengan membandingkan piksel pusat dengan 8 piksel tetangganya sehingga diperoleh nilai biner pada matriks, nilai biner tersebut dihitung dan diubah ke dalam bentuk desimal[2]. LBP memiliki beberapa kelebihan yaitu kesederhanaan perhitungannya yang menyebabkan waktu komputasi menjadi singkat dan sifatnya yang invarian terhadap perubahan fotometri dari objek yang sama. LBP telah digunakan untuk melakukan pengenalan aksara Bali dengan menggunakan metode klasifikasi K-NN (*K-Nearest Neighbour*) dan SVM (*Support Vector Machine*) dengan tingkat akurasi sebesar 74,6% mengenali aksara Bali[3]. Terdapat percobaan lainnya untuk melakukan pengenalan aksara Jawa *hanacaraka* dengan menggunakan metode klasifikasi KNN dengan tingkat keberhasilan sebesar 82,5% mengenali aksara Jawa[2].

Untuk melakukan proses pengenalan pola, sistem memerlukan data ciri-ciri pada masing-masing data untuk dapat mengelompokkannya. Semakin banyak data ciri yang digunakan dapat meningkatkan keberhasilan dalam pengelompokannya. Hal ini menyebabkan jumlah data yang besar sehingga waktu yang dibutuhkan untuk pengelompokan menjadi lebih lama. Pada proses pengelompokan, data ciri yang digunakan untuk pelatihan dalam memberikan hasil prediksi perlu menjadi perhatian. K-SVNN (*K-Support Vector Nearest Neighbour*) adalah algoritma yang dikembangkan berdasarkan metode K-NN untuk mereduksi data latih agar mampu mengurangi beban sistem pada tahap pelatihan[4]. K-SVNN telah digunakan untuk pengenalan karakter tulisan tangan, hasil akurasi yang didapatkan mencapai 85,81% dengan menggunakan nilai $k=1$, untuk pengenalan tulisan tangan huruf besar tingkat akurasi mencapai 78,98% dengan menggunakan nilai $k=3$ [5]. Terdapat percobaan lainnya untuk membandingkan metode K-SVNN dengan metode DT (*Decision Tree*) dan NB (*Naive Bayes*) dengan kesimpulan bahwa metode K-SVNN lebih baik daripada metode DT dan NB dengan selisih lebih baik paling kecil adalah 18,20%[6].

K-NN (*K-Nearest Neighbour*) adalah metode untuk melakukan klasifikasi atau pengelompokan berdasarkan data latih yang memiliki jarak berdekatan[7]. Keunggulan metode K-NN adalah mampu mengelompokkan data yang memiliki banyak *noise*[3]. Metode K-NN telah digunakan untuk melakukan pengenalan aksara Jawa dengan akurasi tertinggi sebesar 82,5% dengan nilai $k=3$ [2]. Terdapat percobaan lainnya untuk melakukan pengenalan aksara Bali dengan akurasi 74,6%[3]. Percobaan lainnya dilakukan untuk mengelompokkan tulisan tangan angka dengan data set berasal dari MNIST (*Modified National Institute of Standards and Technology*) database dengan akurasi tertinggi 89,81%[7].

Berdasarkan uraian di atas, penulis mengajukan penelitian untuk mengetahui performa dari metode ekstraksi ciri *Local Binary Pattern*, metode reduksi data latih *K-Support Vector Nearest Neighbor*, dan metode klasifikasi *K-Nearest Neighbor* yang digunakan untuk mengenali pola tulisan tangan aksara Bima.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dipaparkan, rumusan masalah yang ingin dijawab pada penelitian ini adalah mengetahui performa dari pengenalan pola aksara Bima menggunakan metode ekstraksi ciri *Local Binary Pattern*, metode reduksi *K-Support Vector Nearest Neighbour*, dan metode klasifikasi *K-Nearest Neighbour*.

1.3 Batasan Masalah

Penelitian ini memiliki batasan-batasan masalah untuk memberikan lingkup penelitian agar lebih terfokus ketika pengerjaan. Adapun batasan masalah yang diberikan adalah sebagai berikut :

1. Karakter aksara Bima yang digunakan berjumlah 22 karakter.
2. Data citra diperoleh dari tulisan tangan menggunakan spidol dan kertas yang sama.
3. Data citra aksara Bima diperoleh dari dua jenis responden yaitu orang asli Bima dan bukan orang bima.
4. Data citra aksara Bima diperoleh dari dua puluh orang responden yang masing-masing menuliskan tiap huruf sebanyak dua belas kali.

1.4 Tujuan

Tujuan yang diharapkan dari penelitian ini adalah untuk mengetahui performa pengenalan pola tulisan tangan aksara Bima menggunakan metode ekstraksi ciri *Local Binary Pattern*, metode reduksi *K-Support Vector Nearest Neighbour*, dan metode klasifikasi *K-Nearest Neighbour*.

1.5 Manfaat

Manfaat dari penelitian ini dapat diperoleh oleh dua subjek antara lain sebagai berikut :

1. Bagi Penulis
 - a. Dapat menerapkan pengetahuan selama di perkuliahan terutama pengetahuan dalam bidang pengenalan pola.
 - b. Dapat menambah pengetahuan di bidang pengenalan pola.
2. Bagi Pembaca
 - a. Dapat mengetahui performa penggunaan metode ekstraksi ciri *Local Binary Pattern*, metode reduksi *K-Support Vector Nearest Neighbour*, dan metode klasifikasi *K-Nearest Neighbour*.
 - b. Dapat menggunakan hasil tulisan sebagai *benchmark* untuk penelitian lebih lanjut.

1.6 Sistematika Penulisan

Sistematika penulisan dalam penyusunan tugas akhir ini adalah sebagai berikut:

1. Bab I Pendahuluan

Bab ini menjelaskan dasar-dasar dari penulisan laporan tugas akhir, yang terdiri dari latar belakang, rumusan masalah, batasan masalah, tujuan, serta sistematika penulisan laporan tugas akhir.

2. Bab II Tinjauan Pustaka dan Dasar Teori

Bab ini membahas teori-teori yang berhubungan dengan topik penelitian, meliputi Pengenalan Pola, metode ekstraksi ciri *Local Binary Pattern*, metode reduksi *K-Support Vector Nearest Neighbour*, dan metode klasifikasi *K-Nearest Neighbour*.

3. Bab III Metodologi Penelitian

Bab ini membahas tentang metodologi yang digunakan untuk membangun model pengenalan pola tulisan aksara Bima.

4. Bab IV Analisis Dan Perancangan

Pada bab ini merupakan pembahasan tentang analisis perangkat lunak, meliputi analisis, analisis masalah, analisis metode, analisis kebutuhan sistem, serta perancangan. Perancangan sistem yang terdiri dari perancangan diagram alir (*flowchart*).

5. Bab V Implementasi Dan Pengujian Metode

Bab ini membahas implementasi yang dilakukan terhadap pengolahan citra digital pada citra menggunakan Bahasa *python* dengan metode ekstraksi ciri *Local Binary Pattern*, metode reduksi *K-Support Vector Nearest Neighbour*, dan metode klasifikasi *K-Nearest Neighbour*.

BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Penelitian mengenai pengenalan tulisan aksara telah dilakukan pada beberapa jenis aksara Nusantara antara lain Bali[3], Lampung[8], dan Jawa[2], [9]. Penelitian aksara Bali dilakukan dengan menggunakan metode ekstraksi ciri LBP (*Local Binary Pattern*) dan metode klasifikasi K-NN (*K-Nearest Neighbour*) dengan akurasi 74,6% berhasil mengenali aksara Bali[3]. Penelitian aksara Lampung dilakukan dengan menggunakan metode ekstraksi ciri LBP dan metode klasifikasi SVM (*Support Vector Machine*) dengan hasil LBP dengan nilai radius = 2 menghasilkan 88,93% akurat dan dengan nilai radius = 3 menghasilkan 87,02% akurat mengenali aksara Lampung[8]. Penelitian aksara Jawa dilakukan dengan menggunakan metode ekstraksi fitur *zoning* dan metode klasifikasi SOM (*Self Organizing Maps*) dengan menggunakan 140 data dan menghasilkan 73,57%[9], dan menggunakan metode ekstraksi ciri LBP dan metode klasifikasi K-NN menggunakan 160 dataset citra yang dibagi menjadi 40 citra uji dan 120 citra latih. Pada penelitian tersebut menghasilkan akurasi tertinggi sebesar 82,5% dengan menggunakan parameter pada nilai $k = 3$ untuk metode klasifikasi[2].

Penelitian yang menggunakan metode LBP (*Local Binary Pattern*) sebagai metode ekstraksi ciri telah dilakukan sebelumnya, beberapa penelitian di antaranya yaitu, penelitian tentang pengenalan tulisan tangan angka Bangla. Penelitian tersebut dilakukan dengan menggunakan 6000 data set yang terdiri dari 1000 data pada masing-masing angka dari angka nol sampai angka sembilan. Data tersebut dibagi menjadi 1000 data uji yang diambil secara acak dengan masing-masing angka terdapat 100 data dan 5000 data uji dengan masing-masing angka terdapat 500 data. Proses penelitian tersebut dilakukan dengan 3 variasi dari metode LBP yaitu, *Basic* LBP, *Uniform* LBP, *Simplified* LBP. Hasil terbaik yang diperoleh pada penelitian tersebut adalah dengan menggunakan metode *Basic* LBP dengan ukuran zona 8×8 sebesar 96,7%[10]. Penelitian lainnya dilakukan untuk mengidentifikasi penulis huruf Arab dengan 130 responden. Masing-masing responden menulis satu kata sebanyak 30 kali untuk dijadikan data latih dan 20 kali untuk dijadikan data uji. Penelitian tersebut menghasilkan akurasi sebesar 83%[11]. Penelitian lainnya dilakukan untuk mengelompokkan aksara Jawa dengan menggunakan 120 data set yang dibagi menjadi 80 data latih dan 40 data uji. Penelitian tersebut menggunakan 5 parameter di antaranya, [16 16], [32 32], [64 64], [128 128], dan [256 256]. Penelitian tersebut menghasilkan akurasi terbaik dengan menggunakan parameter [64 64] sebesar 82,5%[12].

Penelitian menggunakan metode K-SVNN (*K-Support Vector Nearest Neighbour*) sebagai metode reduksi data latih telah dilakukan sebelumnya, beberapa penelitian di antaranya yaitu, penelitian tentang penggunaan metode K-SVNN untuk meningkatkan hasil klasifikasi pada metode K-NN (*K-Nearest Neighbour*). Pada penelitian tersebut disimpulkan bahwa metode K-SVNN mampu memberikan hasil yang lebih baik terhadap akurasi pada proses klasifikasi[13]. Pada tahun 2015 dilakukan penelitian tentang penggunaan metode K-SVNN untuk mereduksi data latih sebagai proses awal pada metode klasifikasi *Back-Propagation* untuk mengurangi waktu pelatihan. Pada penelitian tersebut disimpulkan bahwa metode K-SVNN mampu mengurangi waktu pelatihan sebesar 15% hingga 80% dan menyebabkan penurunan akurasi prediksi atau klasifikasi sebesar 0% hingga 4,76%[14]. Penelitian lainnya dilakukan untuk membandingkan K-SVNN dengan 5 metode lain terhadap akurasi prediksi. 5 metode diantaranya yaitu, K-NN (*K-Nearest Neighbour*), SVM (*Support Vector Machine*), *Back-Propagation*, DT (*Decision Tree*), NB (*Naive Bayes*). Pada penelitian tersebut disimpulkan bahwa metode K-SVNN menempati peringkat ke-3 terbaik dalam hal akurasi[15]. Penelitian menggunakan metode K-NN (*K-Nearest Neighbour*) sebagai metode klasifikasi telah dilakukan sebelumnya, beberapa penelitian di antaranya yaitu, penelitian tentang pengenalan angka tulisan untuk diterapkan pada formulir C1 KPU (Komisi Pemilihan Umum). Skenario pada penelitian tersebut dilakukan dengan menggunakan data set dari yang berasal dari MNIST (*Modified National Institute of Standards and Technology*). Dengan menggunakan data dari MNIST tersebut menghasilkan akurasi tertinggi sebesar 89,81% dengan menggunakan nilai $k = 10$. Lalu digunakan data set yang berasal dari formulir C1. Dengan menggunakan data dari formulir C1 sebagai data latih tersebut menghasilkan akurasi tertinggi sebesar 70,9091%[7]. Penelitian lainnya dilakukan untuk mengenali pola karakter angka dan huruf pada tulisan tangan. Penelitian tersebut dilakukan dengan 5 skenario yaitu pengenalan angka, huruf kecil, huruf besar, (huruf kecil + huruf besar), dan (angka + huruf kecil + huruf besar). Pada penelitian tersebut menghasilkan akurasi tertinggi pada skenario pengenalan angka sebesar 85,10%[5].

Berdasarkan uraian di atas, penulis mengajukan untuk melakukan penelitian tentang pengenalan pola tulisan tangan aksara Bima dengan menggunakan metode LBP (*Local Binary Pattern*) untuk ekstraksi ciri karena kesederhanaan perhitungannya yang menyebabkan waktu komputasi menjadi singkat, metode K-SVNN (*K-Support Vector Nearest Neighbour*) untuk mereduksi data latih karena mampu mengurangi waktu komputasi ketika melakukan klasifikasi dengan metode K-NN (*K-Nearest Neighbour*), dan untuk klasifikasi menggunakan metode K-

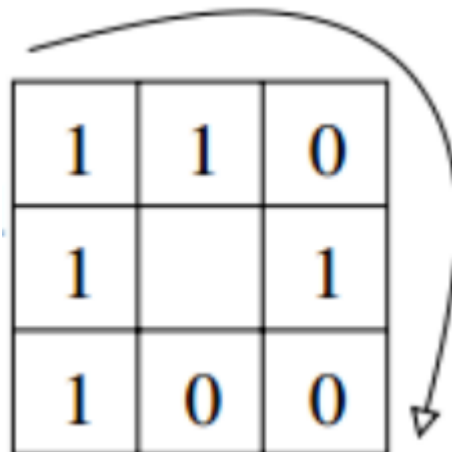
NN karena keunggulan metode ini yang mampu mengelompokkan data yang memiliki banyak *noise*.

2.2 Dasar Teori

2.2.1 Local Binary Pattern

LBP (*Local Binary Pattern*) merupakan metode ekstraksi ciri dengan cara membandingkan nilai piksel pusat dengan 8 piksel di sekelilingnya[16]. LBP pertama kali diperkenalkan oleh Timo Ojala. Terdapat 2 keunggulan dari metode LBP yaitu memiliki waktu komputasi yang lebih cepat dikarenakan kesederhanaan dari proses perhitungannya dan sifatnya yang invarian terhadap perubahan fotometri dari objek yang sama dikarenakan pada prosesnya LBP mengukur intensitas relatif suatu piksel dengan intensitas piksel disekitarnya[8].

Langkah-langkah yang dilakukan pada proses LBP yaitu dengan mengubah citra masukan yang berwarna menjadi citra *grayscale*. Kemudian menentukan piksel pusat pada citra untuk dilakukan proses perbandingan dengan 8 piksel tetangganya. Penentuan piksel tetangga dihitung searah putaran jarum jam seperti pada Gambar 2.1 [2].



Gambar 2.1 Ilustrasi penentuan piksel tetangga searah jarum jam

2.2.2 K-Support Vector Nearest Neighbour

K-SVNN (*K-Support Vector Nearest Neighbour*) adalah algoritma yang dikembangkan berdasarkan metode K-NN untuk mereduksi data latih agar mampu mengurangi beban sistem pada tahap pelatihan. Proses reduksi K-SVNN dihitung berdasarkan ketinggian nilai SD (*Significant Degree*) pada masing-masing data. Entropy digunakan untuk dapat menghitung SD yang memiliki lebih dari 2 kelompok data[17].

2.2.3 K-Nearest Neighbour

K-NN (*K-Nearest Neighbour*) adalah algoritma *supervise learning* yang digunakan dalam proses klasifikasi. Secara umum cara kerja algoritma ini dengan membandingkan data uji

dengan data latih untuk mendapatkan informasi mengenai jarak terdekat dari data tersebut[2]. Secara rinci, algoritma dari klasifikasi K-NN adalah sebagai berikut[18] :

1. Menentukan nilai k yang merupakan nilai ketetanggaan. Pada umumnya nilai k merupakan bilangan ganjil (1,3,5,7,dst).
2. Menghitung nilai jarak pada data latih yang diperoleh berdasarkan rumus *Euclidean* sebagai berikut.

$$d_i = \sqrt{\sum_{i=1}^p (x_2 - x_1)^2} \quad (2-1)$$

3. Mengurutkan hasil perhitungan nilai jarak berdasarkan nilai ketetanggaan.
4. Memeriksa nilai minimum untuk menentukan kelas data uji termasuk dalam kelas.
5. Didapatkan data yang dapat dikenali sesuai dengan jenis kelas.

2.2.4 Performa

Untuk dapat menentukan performa pada metode yang diterapkan, penulis menggunakan *confussion matrix* sebagai alat ukur performa terhadap metode yang diterapkan. *Confussion matrix* atau matriks kebingungan adalah sebuah perhitungan yang membandingkan jumlah data yang berhasil diprediksi dengan jumlah seluruh data yang digunakan. *Confussion matrix* berisi informasi perbandingan label prediksi dengan label yang sebenarnya[19].

Tabel 2.1 Model Confussion matrix 2 kelas

Klasifikasi		Prediksi	
		Kelas: positif	Kelas: negatif
Kenyataan	Kelas: positif	True Positive (TP)	False Negative (FN)
	Kelas: negatif	False Positive (FP)	True Negative (TN)

Keterangan :

TP : hasil prediksi sesuai dengan kenyataan kelas positif

FN : hasil prediksi negatif dengan kelas kenyataan positif

FP : hasil prediksi positif dengan kelas kenyataan negatif

TN : hasil prediksi sesuai dengan kenyataan kelas negatif

Informasi pada tabel *confussion matrix* akan menghasilkan nilai akurasi, presisi, dan *recall*. Akurasi adalah persentase ketepatan data yang diprediksi sesuai dengan data yang sebenarnya setelah dilakukan pengujian pada hasil klasifikasi. Parameter akurasi digunakan sebagai

parameter persentase ketepatan data karena jumlah data pada masing-masing yang dimiliki sama. Perhitungan akurasi dapat dilihat pada persamaan (2-2):

$$akurasi = \frac{TP+TN}{TP+FN+FP+TN} \text{ atau } akurasi = \frac{TB}{TS} \quad (2-2)$$

Dengan:

TB : total seluruh kasus prediksi sesuai dengan kenyataan

TS : total seluruh kasus kejadian

Presisi adalah proporsi kasus prediksi sesuai dengan kenyataan, dengan seluruh kasus prediksi yang tidak sesuai dengan kenyataan. Presisi digunakan karena dapat memberikan nilai keyakinan terhadap data yang dihasilkan dari proses klasifikasi. Perhitungan presisi dapat dilihat pada persamaan berikut (2-3):

$$presisi = \frac{TP}{TP+FP} \text{ atau } presisi = \frac{TP}{TPP} \quad (2-3)$$

Dengan:

TPP : Total seluruh kasus prediksi sesuai dengan kenyataan dan tidak sesuai dengan kenyataan

Recall adalah proporsi kasus data kenyataan tepat diprediksi, dengan seluruh kasus data kenyataan yang tidak sesuai dengan prediksi. *Recall* digunakan karena dapat memberikan nilai toleransi kesalahan yang diterima. Perhitungan *recall* dapat dilihat pada persamaan berikut (2-4) [20]:

$$recall = \frac{TP}{TP+FN} \text{ atau } recall = \frac{TP}{TAP} \quad (2-4)$$

Dengan:

TAP : total seluruh kasus kenyataan sesuai dengan prediksi dan tidak sesuai dengan prediksi

BAB III

METODOLOGI PENELITIAN

3.1 Alat dan Bahan Penelitian

Alat dan bahan yang digunakan dalam penelitian ini berupa perangkat keras, perangkat lunak, dan data tulisan tangan aksara Bima dalam bentuk digital.

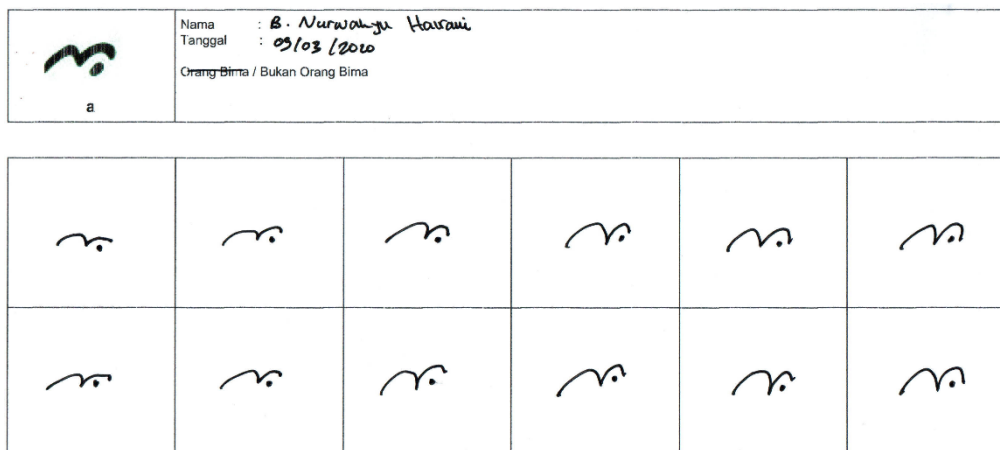
1. Alat Penelitian

Alat-alat yang digunakan pada penelitian ini adalah sebagai berikut :

- a. Laptop ASUS ROG GL553VE.
- b. Processor Intel® Core™ i7-7700HQ CPU @ 2.80GHz (8 CPUs), ~2.8GHz
- c. Random Access Memory 8GB
- d. Graphics Processing Unit NVIDIA® GeForce® GTX 1050TI 4GB
- e. Sistem Operasi Windows 10 Home Single Language 64-bit
- f. Jupyter Lab
- g. Google Colab
- h. Microsoft Office 2016

2. Bahan Penelitian

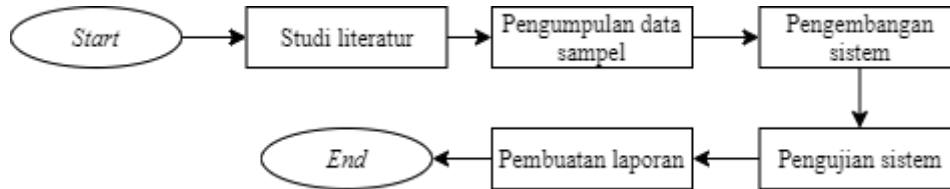
Bahan penelitian yang digunakan adalah citra digital tulisan tangan aksara Bima yang didapatkan dari 20 responden. Data yang didapatkan sebanyak 5280 citra untuk 22 karakter dari aksara Bima. Citra yang dihasilkan pada proses pengambilan data tulisan tangan aksara Bima dapat dilihat pada Gambar 3.1.



Gambar 3.1 Citra pengambilan data tulisan tangan aksara Bima

3.2 Rancangan Penelitian

Penelitian yang dilakukan dapat digambarkan pada diagram alir berikut ini. diagram alir penelitian dimulai dari pengumpulan data hingga pembuatan laporan dapat dilihat pada Gambar 3.2.



Gambar 3.2 Diagram alir penelitian

Langkah pertama yang dilakukan dalam penelitian ini yaitu dengan mengumpulkan data sampel tulisan tangan aksara Bima yang telah diubah ke dalam bentuk citra digital. Langkah kedua yakni melakukan studi literatur untuk mempelajari metode yang digunakan. Selanjutnya adalah pengembangan sistem sesuai dengan rancangan yang telah dibuat. Tahap pengujian dilakukan untuk menghitung performa dari algoritma yang diterapkan. Tahap terakhir yang dilakukan adalah pembuatan laporan hasil penelitian yang telah dilakukan.

3.3 Studi Literatur

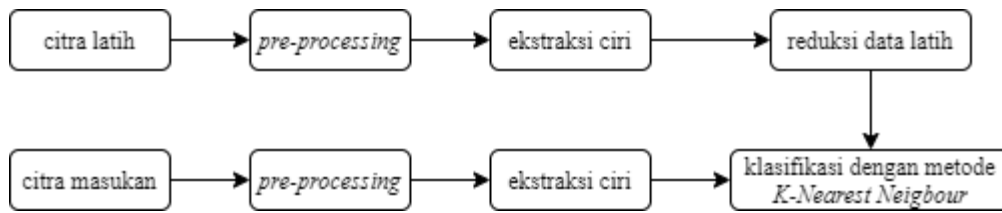
Studi literatur yang dilakukan pada penelitian ini adalah dengan mempelajari buku, jurnal penelitian, serta sumber lain yang berkaitan dengan permasalahan dan menjadi topik pada penelitian ini. materi yang dipelajari dalam studi literatur berkaitan dengan aksara Nusantara, ekstraksi ciri menggunakan metode LBP (*Local Binary Pattern*), reduksi data latih menggunakan metode K-SVNN (*K-Support Vector Nearest Neighbour*), dan klasifikasi citra menggunakan metode K-NN (*K-Nearest Neighbour*).

3.4 Pengumpulan Data

Pengumpulan data dilakukan dengan cara memberikan lembaran kertas ukuran F4 yang berisi 22 huruf aksara Bima kepada 20 responden dengan menggunakan bolpoin gel ukuran 1.0 mm. Untuk masing-masing huruf, responden menulis sebanyak 12 kali. Kertas yang telah diisi oleh responden akan melalui proses *scanning* untuk dijadikan data digital. Data digital tersebut akan melalui proses *cropping* oleh penulis dengan ukuran yang sama berdasarkan huruf aksara Bima dengan format file .JPG dan ukuran 512x512 piksel. Sehingga data yang dihasilkan sejumlah 5.280 citra.

3.5 Rancangan Algoritma

Rancangan algoritma pada penelitian ini secara garis besar dapat di gambarkan pada empat proses utama yaitu *pre-processing* ekstraksi ciri, reduksi data latih, dan klasifikasi yang dapat dilihat pada Gambar 3.3.



Gambar 3.3 Blok diagram rancangan algoritma

3.5.1 Tahap *Pre-processing*

Pada tahap ini dilakukan konversi citra menjadi citra mode *grayscale* 8-bit. Proses ini dibutuhkan untuk dapat melakukan proses ekstraksi ciri dengan metode LBP(*Local Binary Pattern*)[21]. Proses konversi ruang citra dilakukan dengan menggunakan persamaan (3-1).

$$Gray = (0.2125 \times R) + (0.7154 \times G) + (0.0721 \times B) \quad (3-1)$$

Dimana: R : intensitas *red*

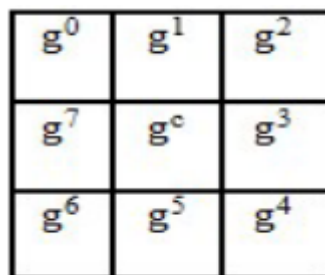
G : intensitas *green*

B : intensitas *blue*

Pada tahap ini tidak dilakukan proses normalisasi citra karena hasil pencahayaan citra yang diperoleh dari proses scanning sama.

3.5.2 Tahap Ekstraksi Ciri

Pada tahap ekstraksi ciri digunakan metode LBP (*Local Binary Pattern*). Cara kerja metode ini dengan melibatkan piksel tetangga (g_p) terhadap piksel pusat (g_c) secara melingkar (*circular*) dengan radius (R) tertentu sehingga sejumlah piksel tetangga (P) terlibat. Metode ini bekerja pada citra digital dengan mode *grayscale*. Jumlah piksel tetangga (P) yang terlibat merupakan kelipatan 8 dari radius (R) yang digunakan. Visualisasi piksel pusat dan piksel tetangga dapat dilihat pada Gambar 3.3.



Gambar 3.3 Visualisasi piksel pusat dan piksel tetangga[16]

Setelah menentukan piksel pusat dan piksel tetangga, langkah selanjutnya adalah nilai intensitas pada piksel tetangga ditransformasikan menjadi bilangan biner 1 jika nilai intensitas piksel pusat lebih besar dari nilai intensitas piksel tetangga, jika tidak maka ditransformasikan menjadi bilangan biner 0. Setelah seluruh piksel selesai ditransformasikan menjadi bilangan biner, langkah selanjutnya adalah menyusun bilangan biner tersebut dan merubahnya menjadi bilangan desimal[8]. Proses tersebut dapat dinyatakan dalam persamaan(3-2), (3-3), (3-4):

$$LBP_{P,R}(x_c, y_c) = \sum_{p=0}^{P-1} s(F)2^p \quad (3-2)$$

$$F = g_p - g_c \quad (3-3)$$

$$s(F) = \begin{cases} 1 & F \geq 0 \\ 0 & F < 0 \end{cases} \quad (3-4)$$

Dimana :

x_c dan y_c : Koordinat piksel pusat

p : indeks piksel tetangga

F : perbedaan nilai intensitas piksel tetangga dengan piksel pusat

$s(F)$: *thresholding*

P : jumlah piksel tetangga yang terlibat

R : besaran radius ketetanggaan

g_p : nilai intensitas piksel tetangga p

g_c : nilai intensitas piksel pusat

Berikut adalah contoh perhitungannya.

Diketahui matriks citra sebagai berikut:

5	9	1
4	4	6
7	2	3

Dengan nilai $g_c = 4$, Maka nilai LBP adalah:

$$LBP_{P,R}(1,1) = \sum_{p=0}^{P-1} s(F)2^p$$

$$LBP_{8,1}(1,1) = (s(5-4)2^0) + (s(9-4)2^1) + (s(1-4)2^2) + (s(6-4)2^3) \\ + (s(3-4)2^4) + (s(2-4)2^5) + (s(7-4)2^6) + (s(4-4)2^7)$$

$$LBP_{8,1}(1,1) = (s(1)2^0) + (s(5)2^1) + (s(-3)2^2) + (s(2)2^3) + (s(-1)2^4) + (s(-2)2^5) \\ + (s(3)2^6) + (s(0)2^7)$$

$$LBP_{8,1}(1,1) = (1 \times 1) + (1 \times 2) + (0 \times 4) + (1 \times 8) + (0 \times 16) + (0 \times 32) + (1 \times 64) \\ + (1 \times 128)$$

$$LBP_{8,1}(1,1) = 1 + 2 + 8 + 64 + 128$$

$$LBP_{8,1}(1,1) = 203$$

Jumlah piksel tetangga (P) yang terlibat merupakan kelipatan 8 dari radius (R) yang digunakan.

Hubungan antara P dan R dapat dilihat pada Tabel 3.1.

Tabel 3.1 Hubungan antara jumlah piksel tetangga dan radius

Radius (R)	Jumlah piksel tetangga (P)
1	8
2	16
3	24
n	$8 \times n$

3.5.3 Tahap Reduksi Data Latih

Pada tahap klasifikasi digunakan metode K-SVNN (*K-Nearest Neighbors*). Dengan metode tersebut, data latih dapat direduksi berdasarkan nilai SD (*Significant Degree*) setiap data. Data yang memiliki nilai SD nol adalah data yang tidak memiliki pengaruh sama sekali terhadap garis batas keputusan klasifikasi. Semakin besar nilai SD yang dimiliki sebuah data, maka data tersebut semakin tinggi pengaruh data tersebut terhadap garis batas keputusan klasifikasi. Pada penelitian ini, nilai SD dihitung menggunakan *Entropy*. Terdapat tiga proses utama untuk menentukan nilai SD. Proses pertama adalah menghitung nilai $V_i(k)$, dimana $V_i(k)$ adalah jumlah pemanggilan sebagai tetangga terdekat kelas k pada data ke- i dimana $k = 1, \dots, C$. C adalah jumlah kelas. $V_i(k)$ direpresentasikan sebagai persamaan (3-5).

$$V_i(k) = \sum_{j=1}^N I(i, k, j) \quad (3-5)$$

Dimana:

N : jumlah pemanggilan tetangga terdekat

Berdasarkan persamaan (3-5), $I(i, k, j)$ adalah hasil pemeriksaan data ke- j terpanggil sebagai K tetangga terdekat data ke- i yang memiliki kelas k . jika kelas data ke- j sama dengan kelas data ke- i , maka nilai $V_i(k)$ pada kelas tersebut dinaikkan 1, jika tidak sama maka $V_i(k)$ pada semua kelas selain k dinaikkan 1. Hal tersebut dapat disajikan sebagai persamaan (3-6) dan (3-7).

$$I(i, k, j) = \begin{cases} 1, & C_i(k) = C(j) \\ 0, & \text{lainnya} \end{cases} \quad (3-6)$$

$$I(i, \sim k, j) = \begin{cases} 1, & C_i(\sim k) = C(j) \\ 0, & \text{lainnya} \end{cases} \quad (3-7)$$

Dimana:

$C_i(k)$: kelas data ke- i

$C(j)$: kelas data ke- j

$I(i, \sim k, j)$: hasil pemeriksaan data ke- j untuk kelas selain k ketika terpilih sebagai tetangga terdekat data ke- i

Proses kedua adalah melakukan normalisasi pada nilai $V_i(k)$. Proses normalisasi direpresentasikan sebagai persamaan (3-7).

$$V_i^{norm}(k) = \frac{V_i(k)}{\sum_{k=1}^C V_i(k)} \quad (3-8)$$

Proses ketiga adalah menghitung nilai *Entropy* sebagai nilai SD. *Entropy* direpresentasikan sebagai persamaan (3-8)[4].

$$SD_i = Entropy = - \sum_{k=1}^C V_i^{norm}(k) \times \log_2(V_i^{norm}(k)) \quad (3-9)$$

Berikut adalah contoh perhitungan.

Terdapat 5 data latih yang memiliki 3 kelas yang berbeda yang direpresentasikan pada Tabel 3.2.

Tabel 3.2 Tabel contoh data latih

index	0	1	2	3	4
Data	25	19	22	40	16
kelas	A	B	D	C	A

Dengan menghitung tetangga terdekat sejumlah 3, maka didapatkan hasil sebagai berikut.

Index 0 :

Data = 25

Kelas data = A

Tetangga = 22,19,16

Kelas tetangga = D, B, A

Proses menghitung $V_i(k)$:

Iterasi 1 :

kelas data 25 (A) \neq kelas tetangga 22(D)

$V_0(A) = 0$; $V_0(B) = 1$; $V_0(C) = 1$; $V_0(D) = 1$

Iterasi 2 :

kelas data 25 (A) \neq kelas tetangga 19(B)

$V_0(A) = 0$; $V_0(B) = 2$; $V_0(C) = 2$; $V_0(D) = 2$

Iterasi 3 :

kelas data 25 (A) = kelas tetangga 16(A)

$V_0(A) = 1$; $V_0(B) = 2$; $V_0(C) = 2$; $V_0(D) = 2$

Proses normalisasi $V_i(k)$:

$$V_0^{norm}(A) = \frac{V_0(A)}{V_0(A) + V_0(B) + V_0(C) + V_0(D)}$$

$$V_0^{norm}(A) = \frac{1}{1 + 2 + 2 + 2}$$

$$V_0^{norm}(A) = \mathbf{0.1429}$$

$$V_0^{norm}(B) = \frac{V_0(B)}{V_0(A) + V_0(B) + V_0(C) + V_0(D)}$$

$$V_0^{norm}(B) = \frac{2}{1 + 2 + 2 + 2}$$

$$V_0^{norm}(B) = \mathbf{0.2857}$$

$$V_0^{norm}(C) = \frac{V_0(C)}{V_0(A) + V_0(B) + V_0(C) + V_0(D)}$$

$$V_0^{norm}(C) = \frac{2}{1 + 2 + 2 + 2}$$

$$V_0^{norm}(C) = \mathbf{0.2857}$$

$$V_0^{norm}(D) = \frac{V_0(D)}{V_0(A) + V_0(B) + V_0(C) + V_0(D)}$$

$$V_0^{norm}(D) = \frac{2}{1 + 2 + 2 + 2}$$

$$V_0^{norm}(D) = \mathbf{0.2857}$$

Proses menghitung nilai SD :

$$SD_0 = -((V_0^{norm}(A) \times \log_2 V_0^{norm}(A)) + (V_0^{norm}(B) \times \log_2 V_0^{norm}(B)) \\ + (V_0^{norm}(C) \times \log_2 V_0^{norm}(C)) + (V_0^{norm}(D) \times \log_2 V_0^{norm}(D)))$$

$$SD_0 = -((0.1429 \times \log_2 0.1429) + (0.2857 \times \log_2 0.2857) + (0.2857 \times \log_2 0.2857) \\ + (0.2857 \times \log_2 0.2857))$$

$$SD_0 = \mathbf{3.7549}$$

3.5.4 Tahap Klasifikasi

Pada tahap klasifikasi digunakan metode K-NN (*K-Nearest Neighbors*). K-NN memiliki prinsip kerja pencarian jarak data terdekat antara data latih dengan data yang akan dievaluasi[22]. Ketika proses pelatihan dilakukan, data latih akan dikelompokkan berdasarkan label yang telah diberikan. Setelah itu pada tahap pengujian dilakukan, data uji akan diklasifikasikan berdasarkan jumlah data uji terdekat yang ditentukan berdasarkan nilai k [23]. Dekat jauhnya data uji terhadap data latih dihitung berdasarkan *Euclidean Distance*[24]. *Euclidean Distance* direpresentasikan sebagai persamaan (3-9).

$$d(X, Y) = \sqrt{\sum_{i=1}^p (dl_i - du_i)^2} \quad (3-10)$$

Dimana:

$d(X, Y)$: jarak skalar dari vektor X dan Y

p : jumlah data citra

dl_i : nilai citra latih

du_i : nilai citra uji

3.6 Pengujian Sistem

Pengujian sistem pada penelitian ini dilakukan dengan menggunakan metode *confusion matriks*. Berdasarkan *confusion matriks* tersebut, kemudian dilakukan perhitungan akurasi, presisi, dan *recall*[19]. Perhitungan akurasi, presisi dan *recall* dilakukan dengan menggunakan persamaan (2-2), (2-3), dan (2-4). Untuk menghitung akurasi, presisi, dan *recall* terhadap data yang memiliki 3 kelas, dapat dilihat pada contoh berikut:

Tabel 2.2 Model *confusion matriks* 3 kelas

		Prediksi kelas		
		Kelas A	Kelas B	Kelas C
Kelas pada kenyataan	Kelas A	50 (AA)	25 (AB)	15 (AC)
	Kelas B	10 (BA)	60 (BB)	20 (BC)
	Kelas C	26 (CA)	6 (CB)	58 (CC)

Sehingga:

$$akurasi = \frac{AA + BB + CC}{AA + AB + AC + BA + BB + BC + CA + CB + CC}$$

$$akurasi = \frac{50 + 60 + 58}{50 + 25 + 15 + 10 + 60 + 20 + 26 + 6 + 58}$$

$$akurasi = \frac{168}{270}$$

$$akurasi = 0.62 = 62\%$$

$$presisi A = \frac{AA}{AA + AB + AC}$$

$$presisi A = \frac{50}{50 + 25 + 15}$$

$$presisi A = \frac{50}{90}$$

$$presisi A = 0.56$$

$$recall A = \frac{AA}{AA + BA + CA}$$

$$recall A = \frac{50}{50 + 10 + 26}$$

$$recall A = \frac{50}{86}$$

$$recall A = 0.58$$

3.7 Skenario Pengujian

Skenario pengujian pada penelitian ini menggunakan *k-fold cross validation* dengan nilai iterasi (*k*) 10. Dengan menggunakan *k-fold cross validation* data akan dibagi menjadi 75% data latih dan 25% data uji. Pada setiap iterasi, pengujian dilakukan dengan mengukur kinerja kombinasi pada tahap ekstraksi ciri, reduksi data latih, dan klasifikasi. Skenario lebih lengkapnya dapat dilihat sebagai berikut:






































1. Pada tahap ekstraksi ciri dengan menggunakan metode LBP (*Local Binary Pattern*), skenario yang diusulkan adalah variasi penggunaan radius dari 1 hingga 3.
2. Pada tahap reduksi data latih dengan menggunakan metode K-SVNN (*K-Support Vector Nearest Neighbors*), skenario yang diusulkan adalah digunakannya metode tersebut dan tidak digunakan. Pada penggunaan metode tersebut, skenario yang diusulkan adalah variasi penggunaan nilai ketetanggaan (*k*) 3, 5, dan 7.
3. Pada tahap klasifikasi dengan menggunakan metode K-NN (*K-Nearest Neighbors*), skenario yang diusulkan adalah variasi penggunaan nilai ketetanggaan (*k*) 3, 5, dan 7.

BAB IV

HASIL DAN PEMBAHASAN

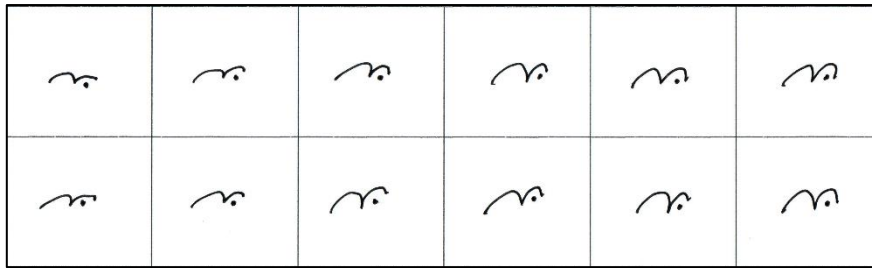
4.1 Akuisisi Data

Pada penelitian ini dilakukan pengumpulan dataset dari 20 orang, namun pada pelaksanaannya hanya didapatkan dataset dari 10 orang saja. Dataset tersebut terdiri dari 2640 citra tulisan tangan karakter Aksara Bima, dimana setiap orang menuliskan 22 jenis karakter Aksara Bima sebanyak 12 kali untuk tiap karakter. Pengambilan dataset tulisan tangan dilakukan menggunakan media kertas berukuran F4 dengan bolpoin gel berukuran 1.0 mm. Adapun cara untuk mengakuisisi citra ke dalam bentuk digital dengan menggunakan scanner, dimana citra yang diperoleh memiliki resolusi 600 dpi. Contoh hasil scan dari tulisan tangan yang didapatkan dapat dilihat pada Gambar 4.1.

 a	<div style="display: flex; justify-content: space-between;"><div>Nama : Alif Sabraji Tanggal : 20/03/2020 Orang Bima / Bukan Orang Bima</div></div>												
<table border="1" style="width: 100%; border-collapse: collapse;"><tr><td style="width: 16.6%; text-align: center;"></td><td style="width: 16.6%; text-align: center;"></td><td style="width: 16.6%; text-align: center;"></td><td style="width: 16.6%; text-align: center;"></td><td style="width: 16.6%; text-align: center;"></td><td style="width: 16.6%; text-align: center;"></td></tr><tr><td style="width: 16.6%; text-align: center;"></td><td style="width: 16.6%; text-align: center;"></td><td style="width: 16.6%; text-align: center;"></td><td style="width: 16.6%; text-align: center;"></td><td style="width: 16.6%; text-align: center;"></td><td style="width: 16.6%; text-align: center;"></td></tr></table>													
													
													

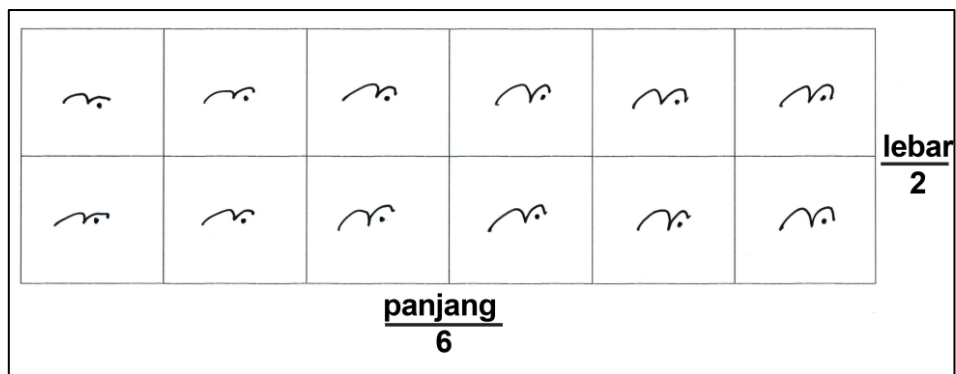
Gambar 4.1 Hasil *scan* formular pengambilan dataset

Setelah didapatkan citra utuh melalui proses *scan*, selanjutnya untuk masing-masing citra form dataset, dilakukan proses *cropping* untuk mendapatkan citra masing-masing karakter. Langkah pertama yang dilakukan dalam mempermudah proses *cropping* adalah melakukan *cropping* pada daerah 12 kotak aksara saja sehingga dihasilkan gambar seperti pada



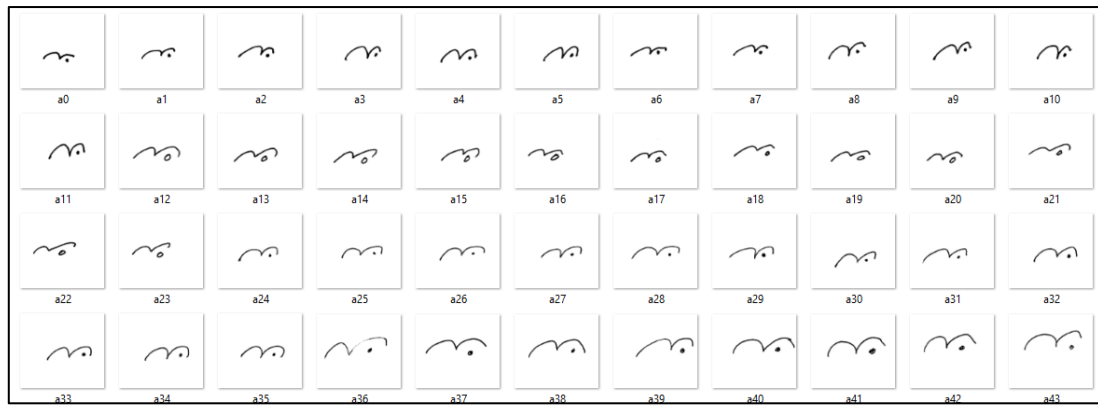
Gambar 4.2 *Cropping* awal dataset

Selanjutnya dilakukan proses cropping yang menghasilkan 12 buah citra karakter, dimana proses cropping ini dilakukan menggunakan code dengan Bahasa pemrograman Python. Langkah yang dilakukan dalam melakukan cropping untuk per-kotak karakter adalah mendapatkan ukuran panjang dan lebar, dimana ukuran panjang didapatkan dengan membagi panjang citra untuk 12 kotak karakter dengan pembagi bernilai 6 dan nilai lebar didapatkan dengan membagi lebar citra untuk 12 kotak karakter dengan pembagi bernilai 2 seperti ilustrasi pada **Gambar 4.3**.



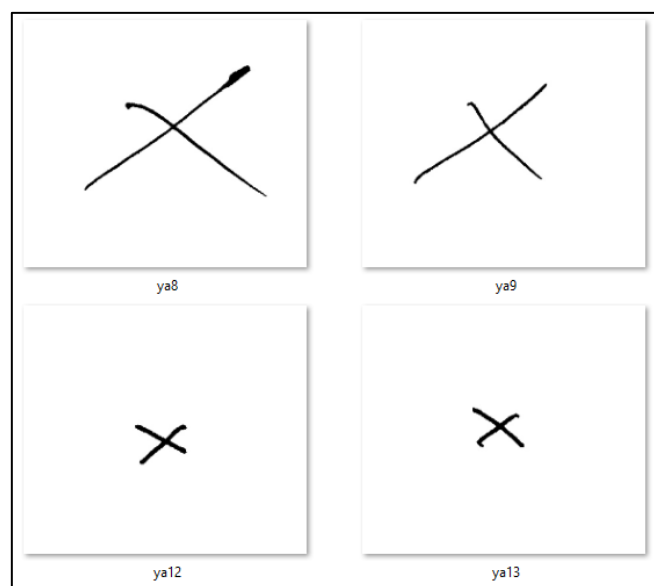
Gambar 4.3 Pembagian Panjang dan lebar citra per karakter

Langkah selanjutnya dilakukan proses slicing terhadap gambar untuk mendapatkan citra per karakter, proses slicing dimulai dari koordinat (0, 0) atau pojok kiri atas citra untuk kotak karakter pertama hingga pojok kanan bawah atau kotak karakter terakhir pada 1 kertas, dengan slicing berukuran nilai panjang per kotak yang didapatkan dan lebar per kotak yang didapatkan pada perhitungan sebelumnya pada Gambar 4.3. Pada Gambar 4.3 dapat dilihat terdapat border atau garis yang membatasi tempat penulisan tiap karakter. Untuk menghilangkan border atau garis tersebut dari cropping, dilakukan slicing kembali dengan padding nilai 10 untuk masing-masing citra sehingga dihasilkan sekumpulan citra per karakter seperti pada **Gambar 4.4**



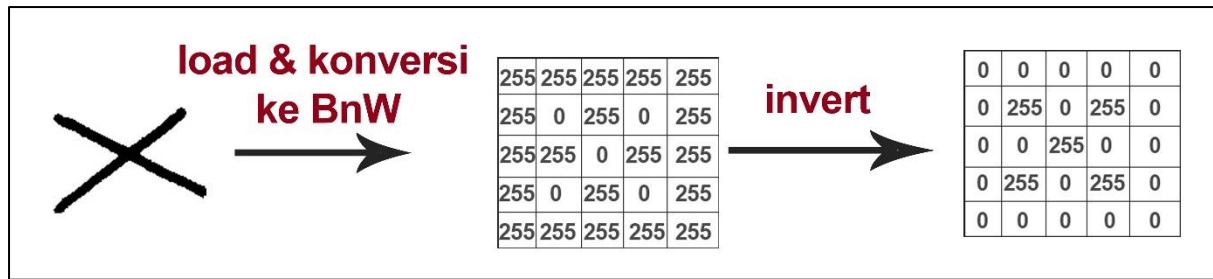
Gambar 4.4 Hasil *cropping* gambar

Proses cropping dilanjutkan dengan melakukan cropping are Region of Interest (ROI) untuk tiap tulisan agar didapatkan standarisasi ukuran tulisan yang sama. Hal tersebut perlu dilakukan karena terdapat beberapa tulisan yang memiliki ukuran tulisan yang berbeda pada karakter “ya” saat ditulis seperti pada ilustrasi di **Gambar 4.5**. Terdapat 4 karakter yang sama dengan ukuran penulisan yang berbeda. Perlu dilakukan proses cropping area tulisannya saja dengan mencari ROI dari masing-masing karakter.



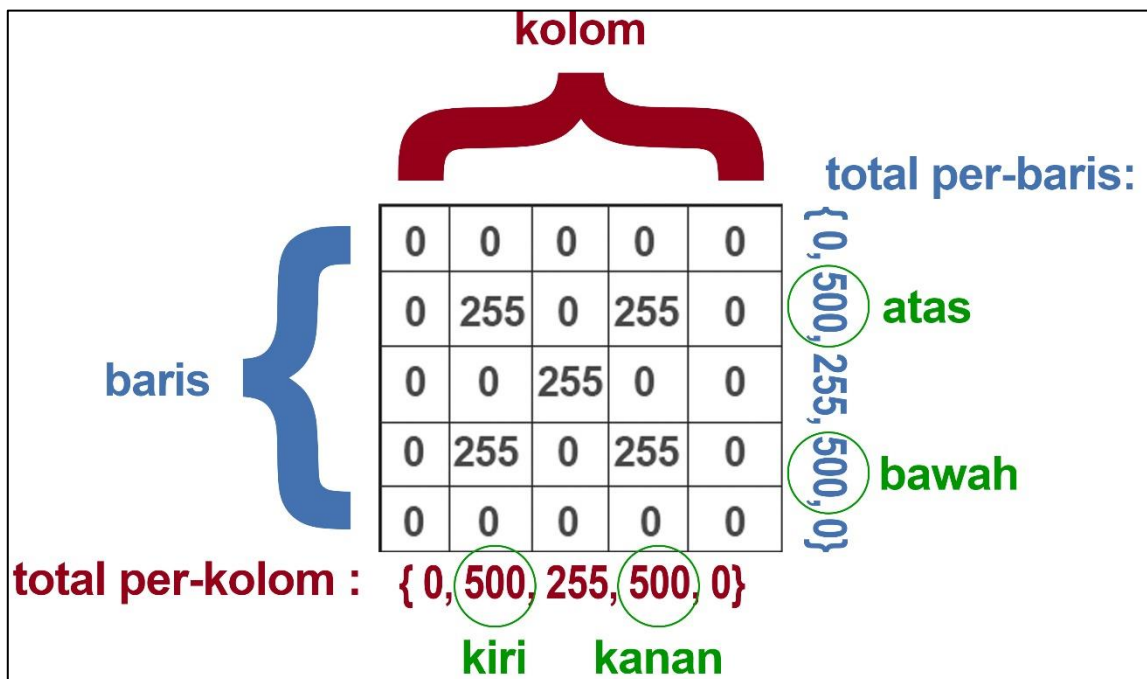
Gambar 4.5 Hasil *cropping* sebelum dicari area ROI

Proses pencarian ROI dan cropping dilakukan dengan menggunakan code. Tahapan yang dilakukan dalam pencarian ROI dilakukan dengan memuat gambar dan dikonversi ke dalam ruang warna Black and White. Adapun konversi ke ruang warna Black and White untuk memudahkan mencari area di mana letak awal dan akhir tulisan secara vertikal dan horizontal. Kemudian dilakukan proses invert, dimana nilai piksel 0 yang menandakan warna hitam ditukar dengan nilai piksel 255 yang menandakan warna putih, dan sebaliknya nilai 255 ditukar menjadi nilai 0 untuk memudahkan proses pencarian ROI seperti pada ilustrasi di **Gambar 4.6**.



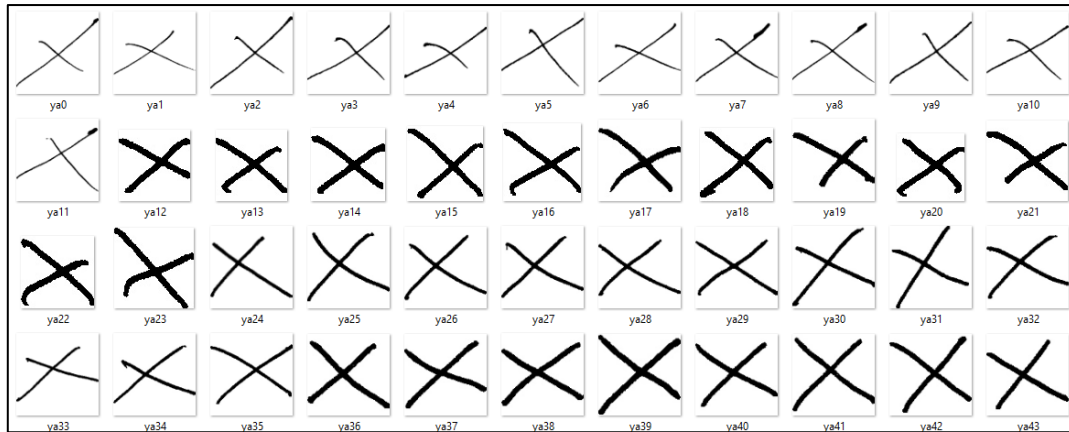
Gambar 4.6 Hasil *invert* citra asli

Langkah selanjutnya dalam mencari ROI adalah mencari nilai total piksel pada baris awal yang bernilai lebih dari nol sebagai batas atas dari ROI dan nilai baris akhir yang bernilai lebih dari nol sebagai batas bawah dari ROI. Kemudian dilakukan pencarian nilai total piksel kolom awal yang bernilai lebih dari nol sebagai batas kiri dari ROI dan nilai kolom akhir yang bernilai lebih dari nol sebagai batas kanan dari ROI seperti ilustrasi pada **Gambar 4.7**.



Gambar 4.7 Ilustrasi pencarian batas ROI

Setelah didapatkan posisi baris atau piksel yang bernilai lebih dari nol untuk batas atas, bawah, kiri, dan kanan, dilakukan proses slicing sesuai dengan batas ROI yang didapat, sehingga dihasilkan sekumpulan citra yang memiliki ukuran yang hampir sama seperti pada Gambar 4.8.



Gambar 4.8 Hasil *crop* berdasarkan ROI

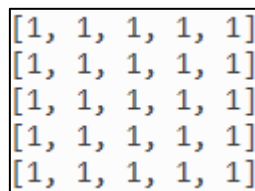
4.2 Data Pre-processing

Pada penelitian ini, tahapan pre-processing yang dilakukan terhadap seluruh data citra yang telah di-scan dan di-crop adalah melakukan proses konversi ruang warna menjadi ruang warna *greyscale* untuk syarat melakukan proses ekstraksi ciri *Local Binary Pattern*. Setelah itu citra dikonversi menjadi citra hitam dan putih, dimana pada warna latar diubah menjadi hitam dan tulisan diubah menjadi putih sehingga citra yang dihasilkan seperti pada Gambar 4.9.



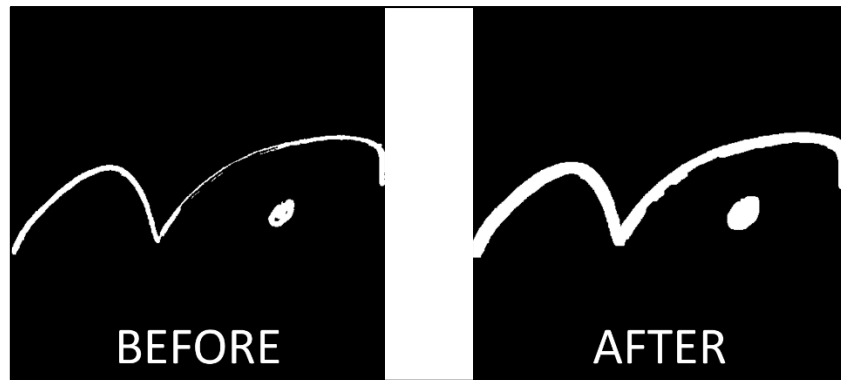
Gambar 4.9 Proses konversi hitam dan putih

Citra tersebut digunakan untuk melakukan proses dilasi dan opening secara berurutan. Kernel yang digunakan pada proses tersebut seperti pada Gambar 4.10.



Gambar 4.10 Kernel proses dilasi dan opening

Pada proses dilasi, menggunakan nilai *iterations* 2. Bertujuan untuk menciptakan citra bergaris lebih tebal seperti pada gambar Gambar 4.11.



Gambar 4.11 Proses dilasi

Citra hasil proses opening dengan latar yang semula hitam diubah menjadi putih dan tulisan yang semula putih diubah menjadi hitam. (4 bit image)

4.3 Ekstraksi Ciri

Pada penelitian ini dilakukan ekstraksi ciri LBP, hasil dari proses LBP tersebut diproses dijadikan data histogram sehingga bisa dijadikan data ciri citra. Ekstraksi ciri dilakukan dengan menggunakan *library* Skimage pada Python. Sebelum proses ekstraksi dilakukan proses *zoning* image dengan ukuran 2x2, 4x4, 8x8, dan 16x16. Lalu masing-masing citra hasil *zoning* tersebut diproses ekstraksi ciri dengan kombinasi radius 1, 2, 3, 4, 5 dan jumlah tetangga 4. Setiap hasil kombinasi ekstraksi ciri disimpan dalam *file* dengan ekstensi .csv untuk mempermudah tahapan selanjutnya.

4.4 Mekanisme Pengujian

Pada penelitian ini terdapat beberapa skenario pengujian yang dilakukan, antara lain sebagai berikut.

1. Nilai radius pada proses ekstraksi ciri menggunakan metode *Local Binary Pattern*. Nilai radius yang digunakan adalah bilangan bulat dari 1 hingga 5.
2. Mengecilkan citra kedalam beberapa ukuran, yaitu 32x32 piksel, 64x64 piksel, 128x128 piksel, dan 256x256 piksel.
3. Penggunaan metode *zoning* pada citra yang sudah dilakukan pengecilan citra. *Zoning* dibagi menjadi 4 daerah, 16 daerah, 64 daerah, 256 daerah.
4. Nilai tetangga pada proses klasifikasi menggunakan metode *K-Nearest Neighbor*. Nilai tetangga yang digunakan adalah 1, 3, 5, 7, 9.
5. Melakukan proses reduksi data latih menggunakan metode *K-Support Vector Nearest Neighbor* sebelum melakukan proses klasifikasi dan tidak melakukan proses reduksi data latih.

Skenario pengujian dilakukan berdasarkan kombinasi skenario di atas. Total skenario yang dilakukan untuk melakukan proses ekstraksi ciri adalah 80 skenario, dan total skenario yang

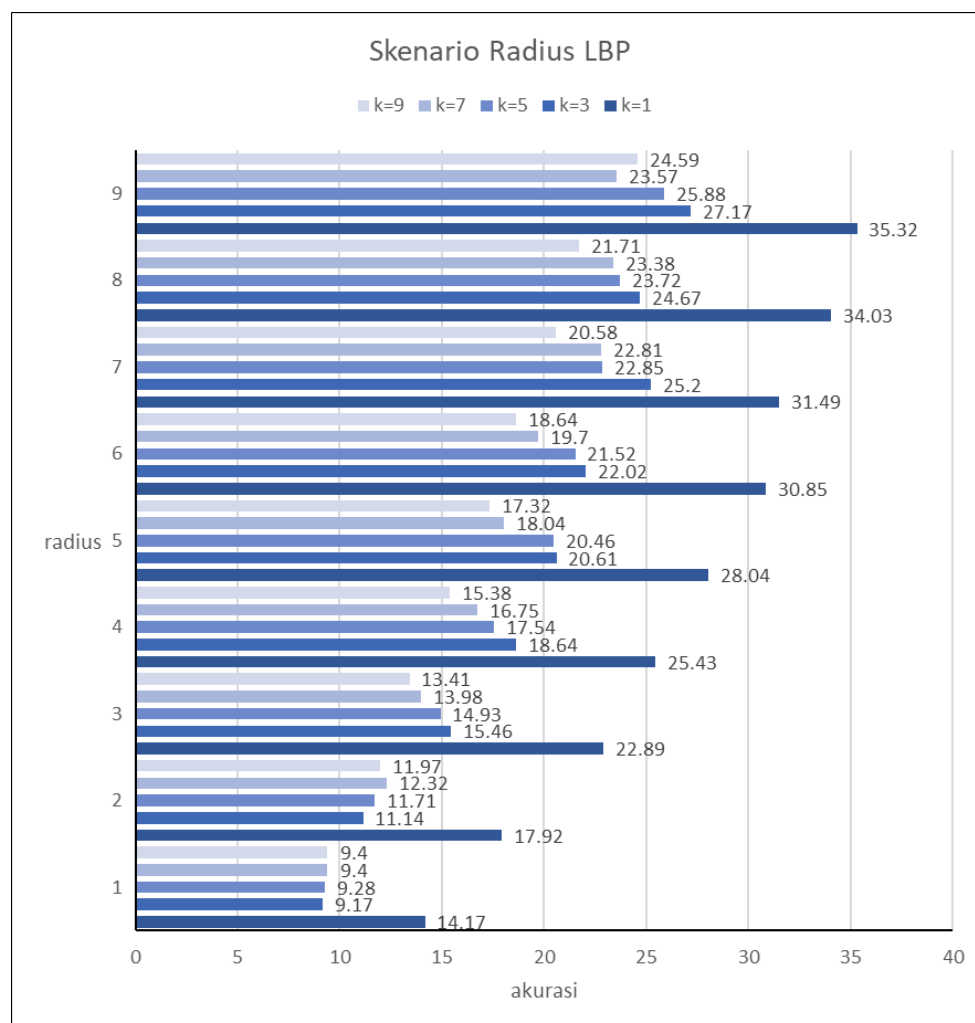
dilakukan untuk melakukan klasifikasi adalah 400 skenario. Adapun hasil terbaik pada masing-masing skenario mengacu pada nilai akurasi, presisi dan *recall* terbaik. Dan juga pada skenario proses reduksi, hasil terbaik mengacu juga pada waktu komputasi yang dilakukan pada proses klasifikasi.

Untuk menghindari bias dalam pembagian data, maka pada setiap pengujian skenario digunakan K-Fold Cross Validation dengan menggunakan nilai fold 5, pembagian dilakukan dengan membagi data latih : data uji dengan perbandingan 8:2.

4.5 Hasil Pengujian

4.5.1 Pengujian Skenario Radius LBP

Pada pengujian skenario radius LBP, variasi nilai radius yang diujikan yaitu rentang nilai 1 sampai 9 dan variasi nilai tetangga klasifikasi yaitu 1, 3, 5, 7, dan 9. Adapun pengujian dilakukan dengan menggunakan K-Fold Cross Validation dengan nilai split senilai 5 yaitu perbandingan data latih : data uji sebesar 8:2. Hasil dari pengujian variasi radius LBP antara lain sebagai berikut.

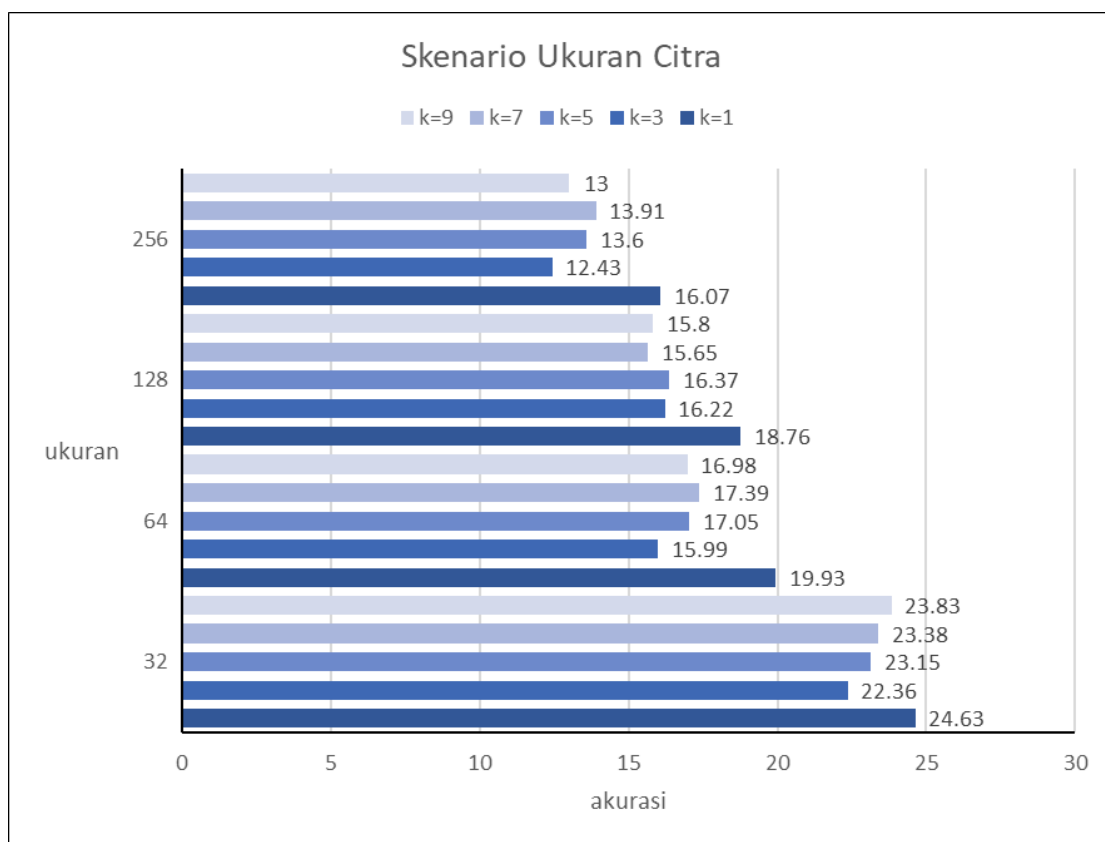


Gambar 4.12 Hasil pengujian skenario radius LBP

Berdasarkan hasil pengujian skenario radius LBP di atas, diperoleh nilai akurasi tertinggi pada nilai radius LBP 9 dengan akurasi 35,32%. Nilai radius LBP mempengaruhi akurasi, semakin tinggi nilai radius LBP, maka akurasi semakin meningkat. Nilai radius LBP 9 akan digunakan untuk skenario selanjutnya.

4.5.2 Pengujian Skenario Ukuran Citra

Pada pengujian skenario ukuran citra, variasi nilai ukuran yang diujikan yaitu rentang nilai 32x32, 64x64, 128x128, dan 256x256 dan variasi nilai tetangga klasifikasi yaitu 1, 3, 5, 7, dan 9. Adapun pengujian dilakukan dengan menggunakan *K-Fold Cross Validation* dengan nilai split senilai 5 yaitu perbandingan data latih : data uji sebesar 8:2. Hasil dari pengujian variasi ukuran citra antara lain sebagai berikut.



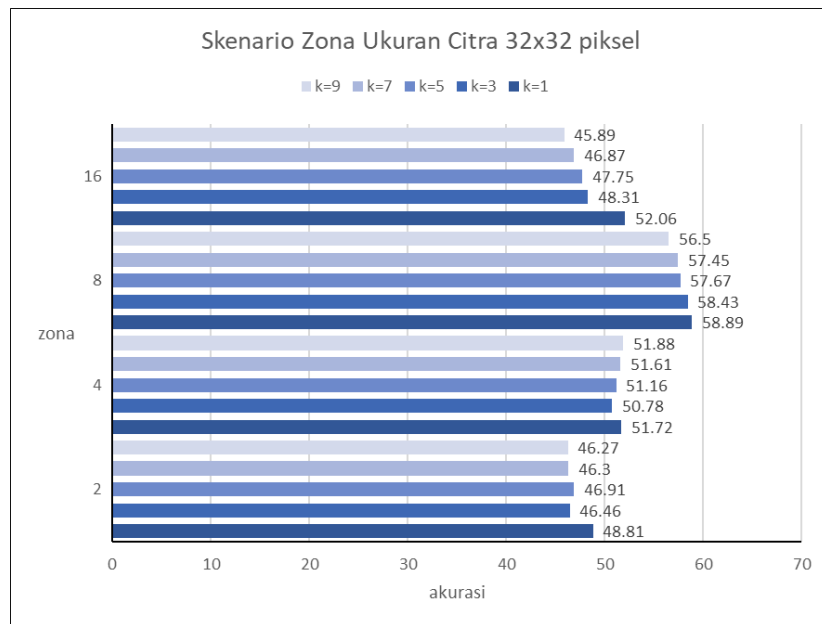
Gambar 4.13 Hasil pengujian skenario ukuran citra

Berdasarkan hasil pengujian skenario ukuran citra di atas, diperoleh nilai akurasi tertinggi pada nilai ukuran 32x32 piksel dengan akurasi 24,63%. Nilai ukuran citra mempengaruhi akurasi, semakin tinggi besar ukuran citra, maka akurasi semakin menurun.

4.5.3 Pengujian Skenario Zona Citra

Pada pengujian skenario zona citra, data yang digunakan adalah variasi ukuran citra 32x32, 64x64, 128x128, dan 256x256 dan ekstraksi ciri LBP dengan radius 9. Variasi nilai zona yang digunakan adalah 2x2, 4x4, 8x8, 16x16. Berikut adalah hasil pengujian skenario zona citra.

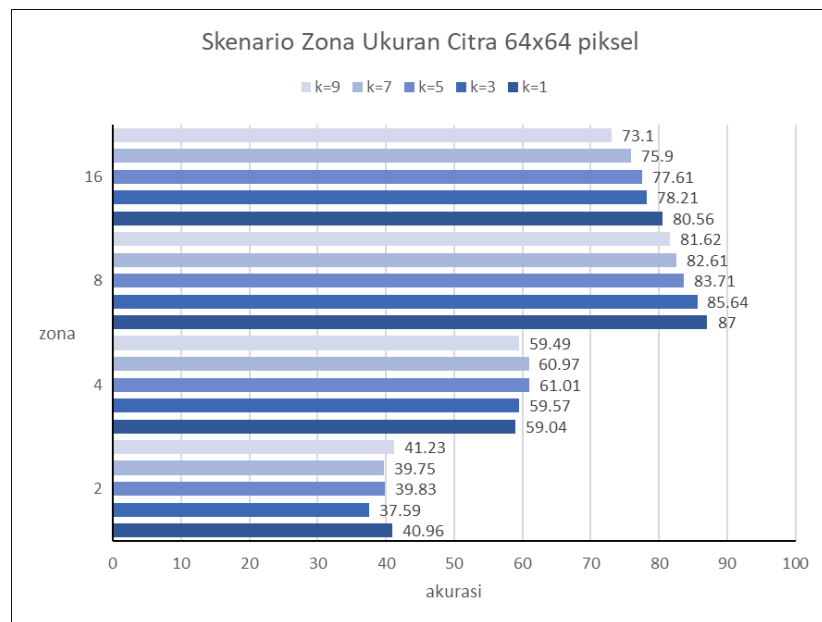
1. Skenario zona ukuran citra 32x32 piksel.



Hasil skenario zona ukuran citra 32x32 piksel

Berdasarkan hasil pengujian skenario zona ukuran citra 32x32 piksel, diperoleh akurasi tertinggi pada skenario zona 8x8 dengan akurasi 58,89%. Jumlah zona tidak mempengaruhi akurasi.

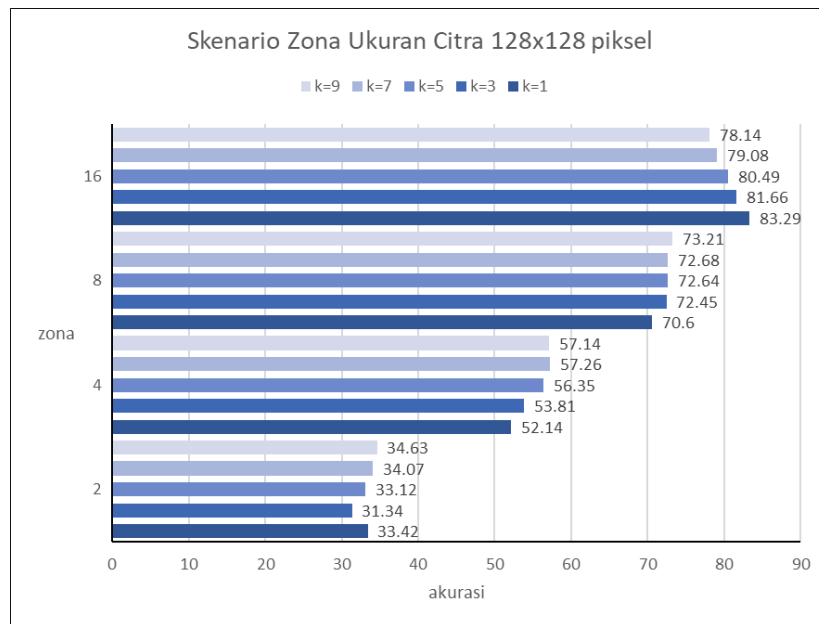
2. Skenario zona ukuran citra 64x64 piksel.



Hasil skenario zona ukuran citra 64x64 piksel

Berdasarkan hasil pengujian skenario zona ukuran citra 64x64 piksel, diperoleh akurasi tertinggi pada skenario zona 8x8 dengan akurasi 87%. Jumlah zona tidak mempengaruhi akurasi.

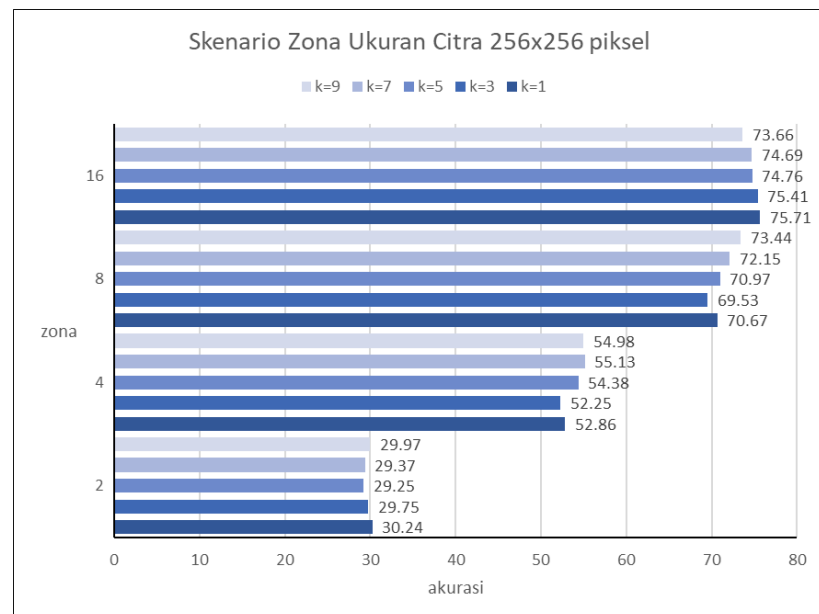
3. Skenairo zona ukuran citra 128x128 piksel.



Hasil skenario zona ukuran citra 128x128 piksel

Berdasarkan hasil pengujian skenario zona ukuran citra 128x128 piksel, diperoleh akurasi tertinggi pada skenario zona 8x8 dengan akurasi 83,29%. Jumlah zona mempengaruhi akurasi, semakin banyak jumlah zona yang digunakan maka akurasi akan semakin meningkat.

4. Skenairo zona ukuran citra 256x256 piksel.

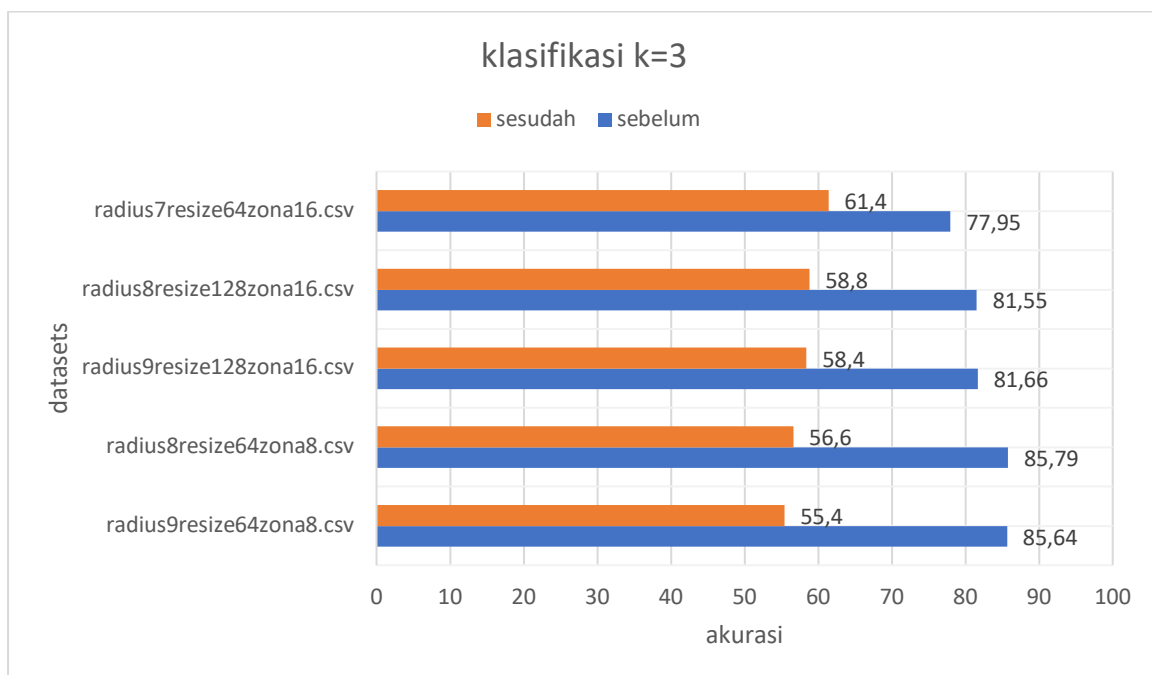
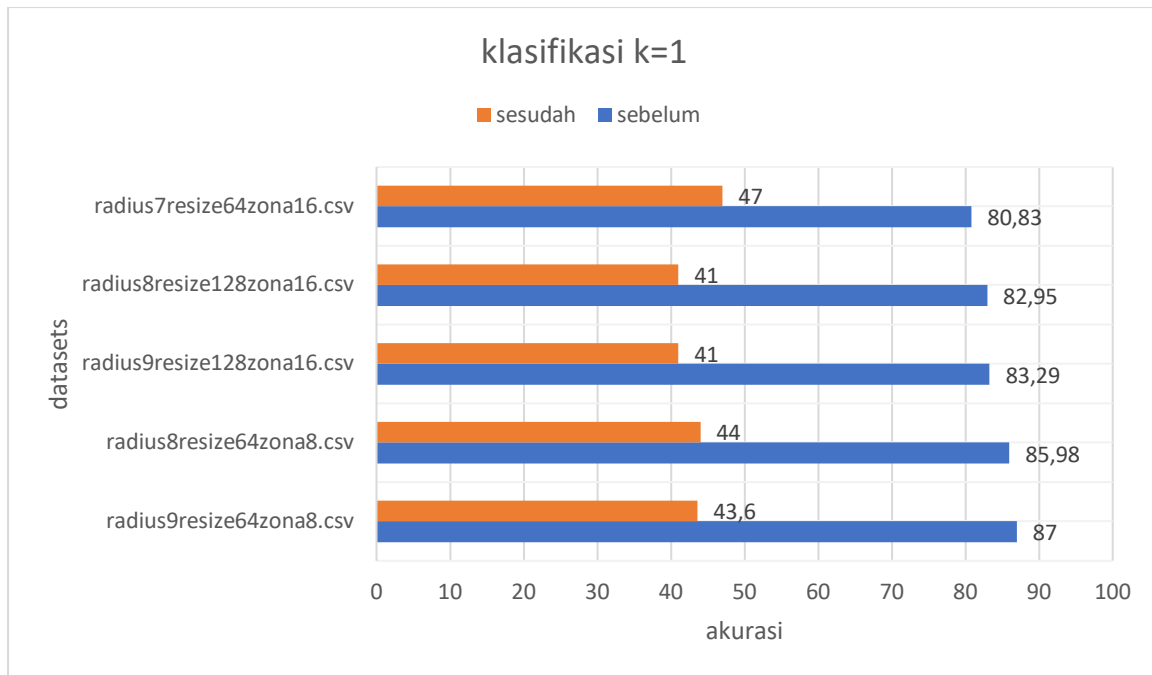


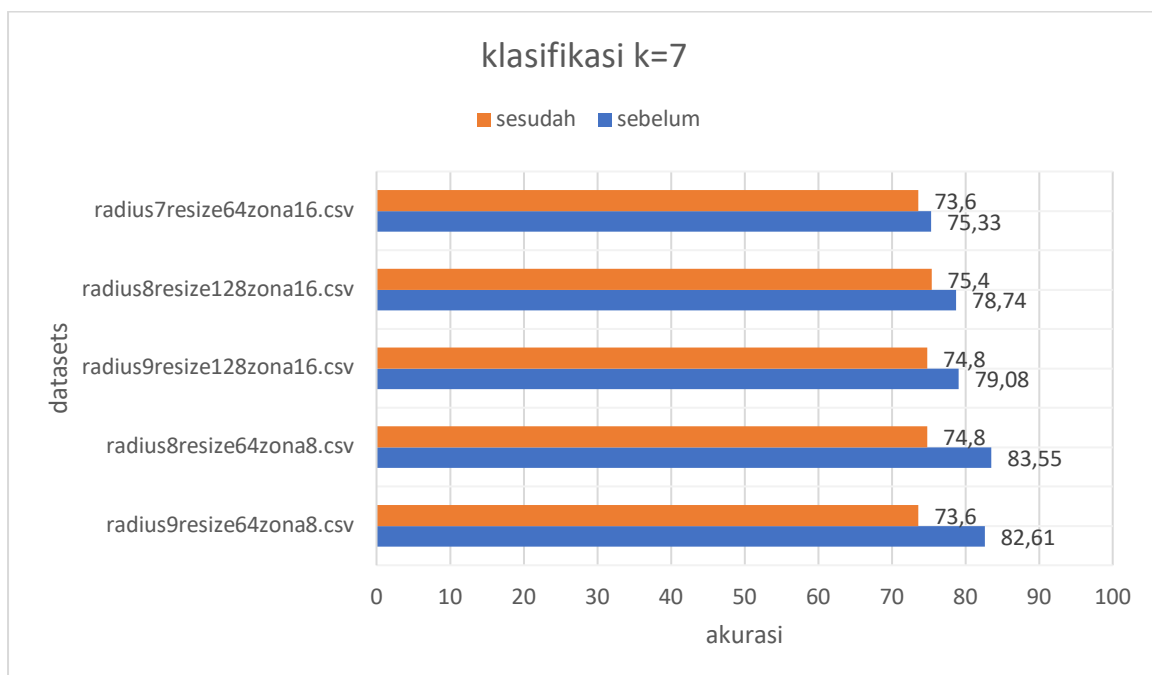
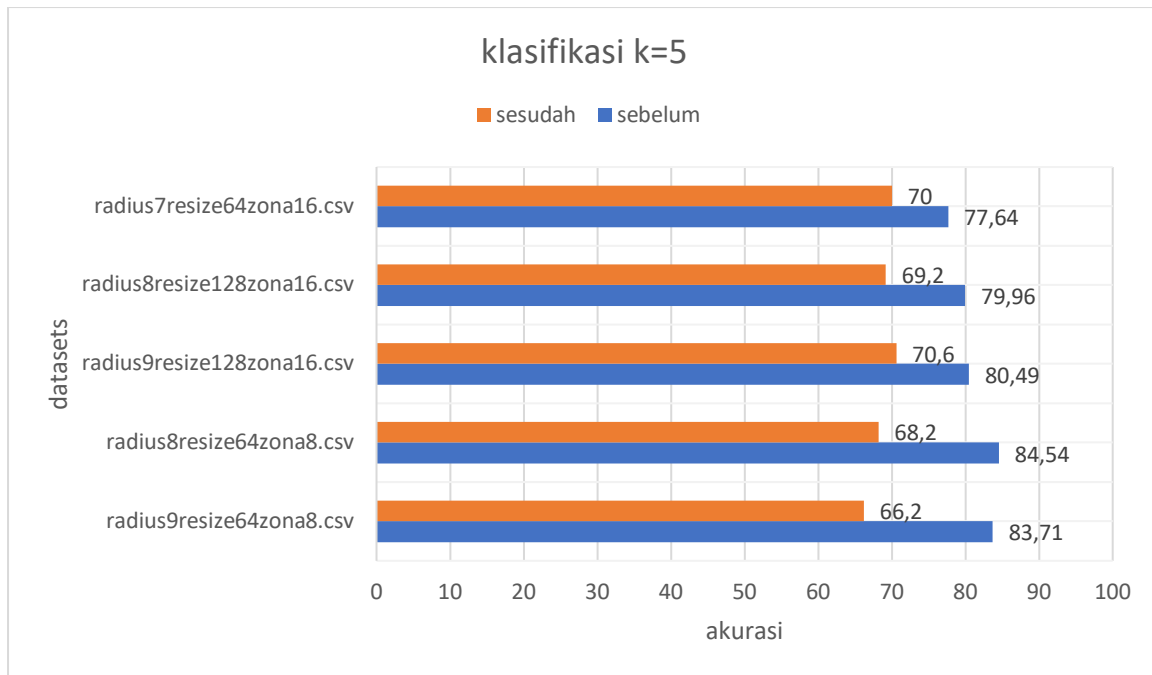
Hasil skenario zona ukuran citra 256x256 piksel

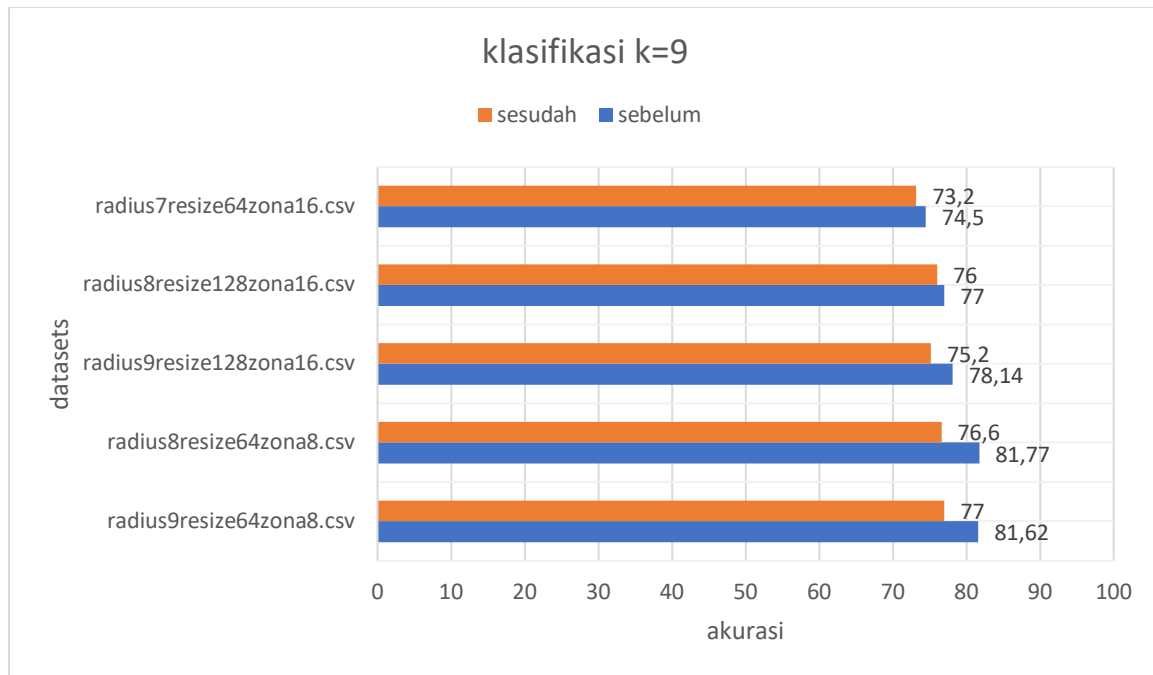
Berdasarkan hasil pengujian skenario zona ukuran citra 256x256 piksel, diperoleh akurasi tertinggi pada skenario zona 8x8 dengan akurasi 75,71%. Jumlah zona mempengaruhi akurasi, semakin banyak jumlah zona yang digunakan maka akurasi akan semakin meningkat.

4.5.4 Pengujian Skenario Reduksi

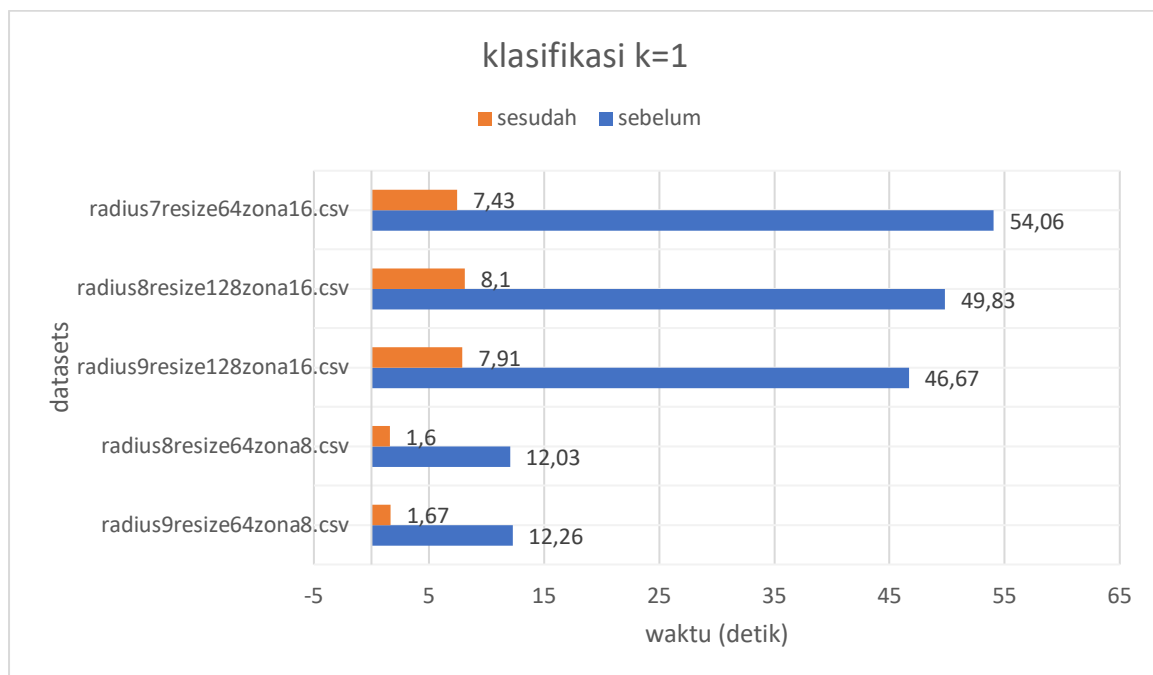
akurasi

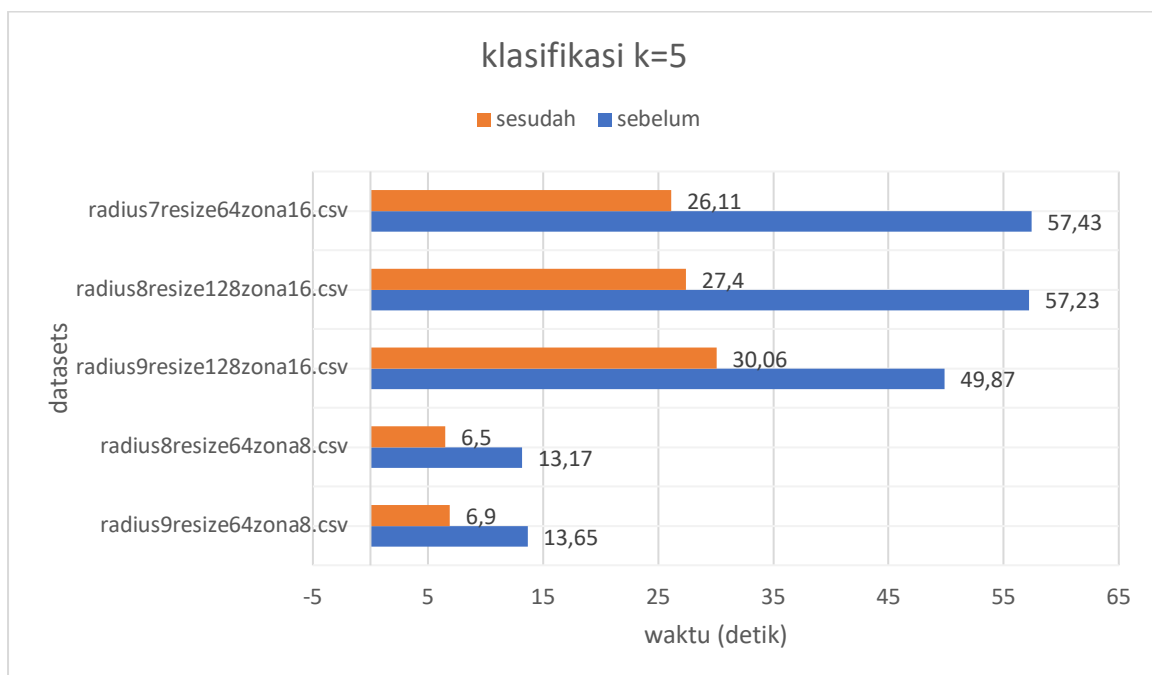
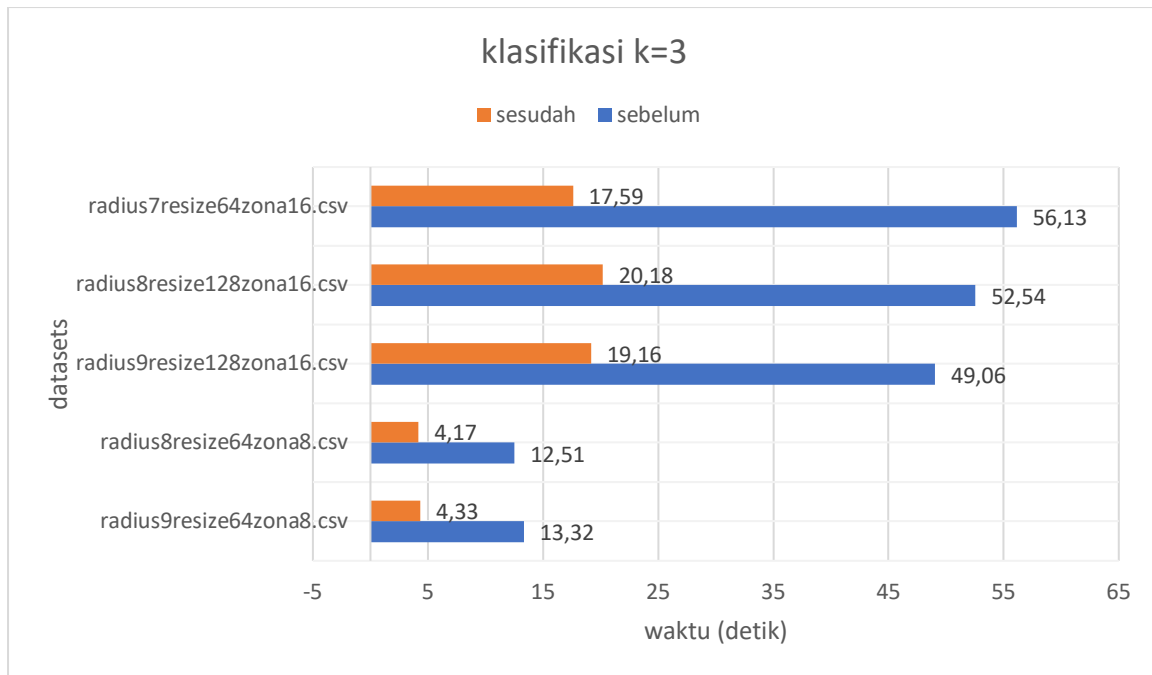


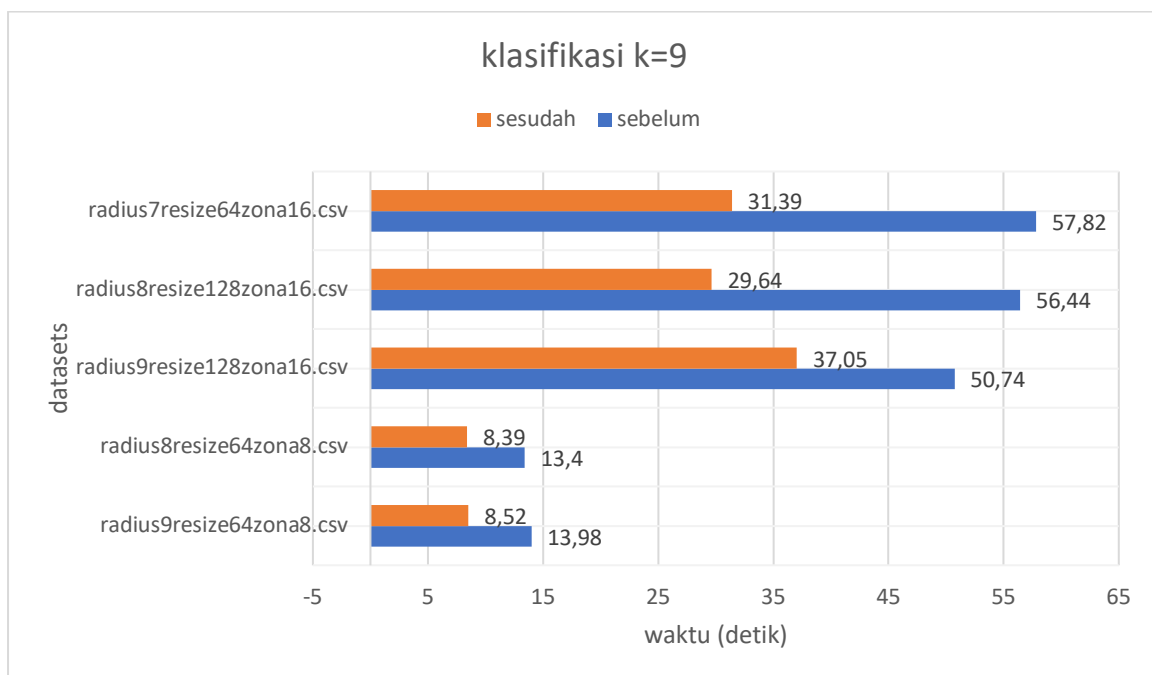
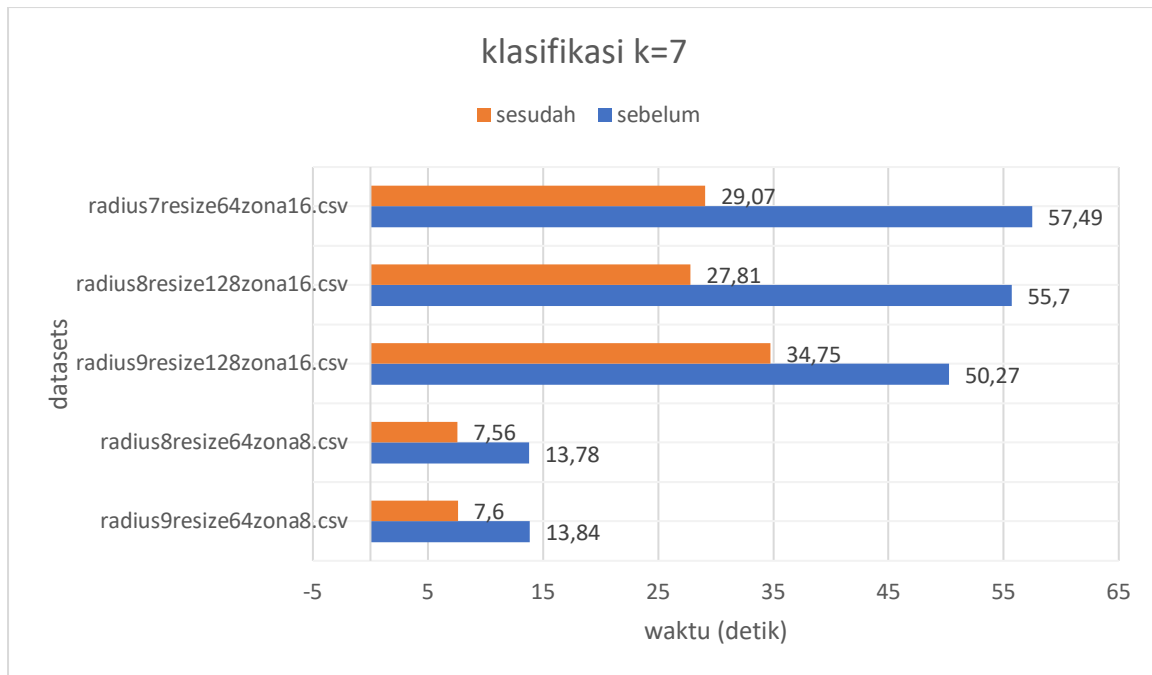




waktu







4.6 Pembahasan

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

5.2 Saran

DAFTAR PUSTAKA

- [1] Nasruddin, *Sejarah Dan Budaya Lokal Dari Sulawesi Sampai Bima*. Jakarta Pusat: Gunadarma Ilmu, 2016.
- [2] A. Susanto, D. Sinaga, E. H. Rachmawanto, and D. R. I. M. Setiadi, "Unjuk Kerja K-Nearest Neighbors Pada Pengengalan Karakter Jawa Berbasis Local Binary Pattern," in *Prosiding SNATIF Ke -5 Tahun 2018*, 2018, pp. 25–30, doi: 10.2298/PAN0903301G.
- [3] I. Ayu Dian Purnama Sari, "Pengengalan aksara Bali Dengan Metode Local Binary Pattern," in *e-Proceeding of Engineering*, 2015, vol. 2, no. 2, pp. 2697–2704.
- [4] E. Prasetyo, R. D. Adityo, N. Suciati, and C. Fatichah, "Reduksi Data Latih pada K-Support Vector Nearest Neighbor Menggunakan Entropy," in *Seminar Nasional Aplikasi Teknologi Informasi (SNATi) 2018*, 2018.
- [5] A. S. Wijaya, N. Chamidah, and M. M. Santoni, "Pengengalan Karakter Tulisan Tangan Dengan K-Support Vector Nearest Neighbor," *IJEIS (Indonesian J. Electron. Instrum. Syst.)*, vol. 9, no. 1, p. 33, 2019, doi: 10.22146/ijeis.38729.
- [6] E. Prasetyo, R. A. D. Rahajoe, and A. Arizal, "Perbandingan K-Support Vector Nearest Neighbour Terhadap Decision Tree dan Naive Bayes," in *Seminar Nasional Teknik Informatika*, 2013.
- [7] N. Ilmi, "Pengengalan Angka Tulisan Tangan dengan Menggunakan Local Binary Pattern Variance dan Klasifikasi K-Nearest Neighbour Handwriting Digit Recognition with Use Local Binary Pattern Variance and K-Nearest Neighbour Classification," 2015. [Online]. Available: <https://libraryproceeding.telkomuniversity.ac.id/index.php/engineering/article/view/1271>.
- [8] R. Purnama, "Pengengalan Tulisan Tangan aksara Lampung Menggunakan Fitur Local Binary Pattern," *Skripsi*, 2018.
- [9] A. Hidayat and R. N. Shofa, "Self Organizing Maps (SOM) Suatu Metode Untuk Pengengalan aksara Jawa," *J. Siliwangi*, vol. 2, no. 1, pp. 64–70, 2016.
- [10] T. Hassan and H. A. Khan, "Handwritten Bangla numeral recognition using Local Binary Pattern," in *2nd International Conference on Electrical Engineering and Information and Communication Technology, iCEEiCT 2015*, 2015, doi: 10.1109/ICEEICT.2015.7307371.
- [11] R. Paredes, J. S. Cardoso, and X. M. Pardo, "Arabic Writer Identification Using Local Binary Patterns (LBP) of Handwritten Fragments," *Springer Int. Publ. Switz. 2015*, vol.

- 9117, pp. 237–244, 2015, doi: 10.1007/978-3-319-19390-8.
- [12] A. Susanto, D. Sinaga, C. A. Sari, E. H. Rachmawanto, and D. R. I. M. Setiadi, “A High Performace of Local Binary Pattern on Classify Javanese Character Classification,” *Sci. J. Informatics*, vol. 5, no. 1, pp. 1–8, 2018, doi: 10.15294/sji.v5i1.14017.
 - [13] E. Prasetyo, “K-Support Vector Nearest Neighbor Untuk Klasifikasi Berbasis K-NN,” *SESINDO 2012*, 2012, doi: 10.13140/RG.2.1.4298.3768.
 - [14] E. Prasetyo, “Reduksi Data Latih Dengan K-Svnn Sebagai Pemrosesan Awal Pada Ann Back-Propagation Untuk Pengurangan Waktu Pelatihan,” *Simetris J. Tek. Mesin, Elektro dan Ilmu Komput.*, vol. 6, no. 2, pp. 223–230, 2015, doi: 10.24176/simet.v6i2.456.
 - [15] E. Prasetyo, “K-Support Vector Nearest Neighbor: Classification Method, Data Reduction, and Performance Comparison,” *J. Electr. Eng. Comput. Sci.*, vol. 1, no. 1, 2016.
 - [16] N. Hayaty *et al.*, “Jurnal Sustainable : Jurnal Hasil Penelitian dan Industri Terapan Penerapan Algoritma Local Binary Pattern untuk Pengenalan Pola Sidik Jari,” *J. Sustain. J. Has. Penelit. dan Ind. Terap.*, vol. 06, no. 02, pp. 6–11, 2017.
 - [17] E. Prasetyo, R. D. Adityo, N. Suciati, and C. Fatichah, “Multi-class K-support vector nearest neighbor for mango leaf classification,” *Telkomnika (Telecommunication Comput. Electron. Control.*, vol. 16, no. 4, pp. 1826–1837, 2018, doi: 10.12928/TELKOMNIKA.v16i4.8482.
 - [18] M. R. Hidayah, I. Akhlis, and E. Sugiharti, “Recognition Number of The Vehicle Plate Using Otsu Method and K-Nearest Neighbour Classification,” *Sci. J. Informatics*, vol. 4, no. 1, pp. 66–75, 2017, doi: 10.15294/sji.v4i1.9503.
 - [19] Indriyanti, D. Sugianti, and M. A. Al Karomi, “Peningkatan Akurasi Algoritma KNN dengan Seleksi Fitur Gain Ratio untuk Klasifikasi Penyakit Diabetes Mellitus,” *IC-Tech*, vol. 7, no. 2, pp. 1–6, 2017, [Online]. Available: <https://ejournal.stmik-wp.ac.id/index.php/ictech/article/view/3>.
 - [20] P. Mayadewi and E. Rosely, “Prediksi Nilai Proyek Akhir Mahasiswa Menggunakan Algoritma Klasifikasi Data Mining,” *SESINDO 2015*, 2015.
 - [21] M. Biglari, F. Mirzaei, and J. G. Neycharan, “Persian/Arabic Handwritten Digit Recognition Using Local Binary Pattern,” *Int. J. Digit. Inf. Wirel. Commun.*, vol. 4, no. 4, pp. 486–492, 2014, doi: 10.1017/CBO9781107415324.004.
 - [22] A. Pariyandani, “Klasifikasi Citra Ikan Berformalin Menggunakan Metode K-NN dan GLCM,” in *Prosiding Seminar Nasional Teknologi Informatika*, 2019, vol. 2, no. 1, pp.

42–47.

- [23] F. Fandiansyah, J. Y. Sari, and I. P. Ningrum, “Pengenalan Wajah Menggunakan Metode Linear Discriminant Analysis dan k Nearest Neighbor,” *J. Ultim.*, vol. 9, no. 1, pp. 1–9, 2017, doi: 10.31937/ti.v9i1.557.
- [24] Y. Rahman and H. Wijayanto, “Klasifikasi Batik Menggunakan Metode K-Nearest Neighbour Berdasarkan Gray Level Co-Occurrence Matrices (GLCM),” in *Seminar Nasional Aplikasi Teknologi Informasi*, 2015, doi: 10.1109/ISIT.2018.8437785.