

**PENGENALAN TULISAN AKSARA LAMPUNG DENGAN *LIBRARY
OPENCV* MENGGUNAKAN METODE *PROJECTION PROFILE* DAN
KLASIFIKASI *K-NN***

Skripsi

Oleh

REHTA YUNANI



**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2022**

ABSTRAK

Pengenalan Tulisan Aksara Lampung dengan *LIBRARY OPENCV* Menggunakan Metode *PROJECTION PROFILE* dan Klasifikasi *K-NN*

Oleh

REHTA YUNANI

Sebagai salah satu negara yang kaya akan budaya, Indonesia menghadapi tantangan yang cukup berat untuk melestarikan keragaman bahasa dan tulisannya. Provinsi Lampung merupakan salah satu daerah yang memiliki bahasa dan aksara. Namun seiring berjalannya waktu, ragam tulisan berupa aksara Lampung mulai dilupakan karena ragam tulisan ini tidak banyak digunakan sebagai sarana komunikasi sehari-hari. Salah satu upaya yang dapat dilakukan untuk mendigitalkan aksara Lampung adalah dengan metode pengenalan karakter menggunakan profil proyeksi sebagai metode ekstraksi fitur, dan algoritma K-Nearest Neighbors (KNN) untuk memprediksi bentuk karakter.

Kata kunci: Aksara Lampung, *Opencv*, *Projection profile*, *K-NN*.

ABSTRACT

RECOGNITION OF LAMPUNG SCRIPT WRITING WITH THE OPENCV LIBRARY USING THE PROJECTION PROFILE METHOD AND K-NN CLASSIFICATION

By

REHTA YUNANI

As one of the countries with a rich culture, Indonesia faces considerable challenges to preserve the diversity of its languages and writings. Lampung Province is one of the regions that has its language and script. However, over time, the variety of writings in the form of the Lampung script began to be forgotten because this variety of writings was not often used as a means of daily communication. One effort can be done to digitize the Lampung script, using the character recognition method using the projection profile as the feature extraction method, and K-Nearest Neighbors (KNN) algorithm to predict character shape. This study has two types of data: training data and testing data with 18-character labels.

Keywords: Lampung script, Opencv, Projection profile, K-NN.

**PENGENALAN TULISAN AKSARA LAMPUNG DENGAN *LIBRARY
OPENCV* MENGGUNAKAN METODE *PROJECTION PROFILE* DAN
KLASIFIKASI *K-NN***

Oleh

REHTA YUNANI

Skripsi

**Sebagai Salah Satu Syarat untuk Memperoleh Gelar
SARJANA KOMPUTER**

Pada

**Jurusan Ilmu Komputer
Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung**



**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2022**

Judul Skripsi : **PENGENALAN AKSARA LAMPUNG
DENGAN LIBRARY OPENCV
MENGUNAKAN METODE PROJECTION
PROFILE DAN KLASIFIKASI K-NN**

Nama Mahasiswa : **Rehta Yunani**

Nomor Pokok Mahasiswa : **1617051106**

Jurusan : **Ilmu Komputer**

Fakultas : **Matematika dan Ilmu Pengetahuan Alam**

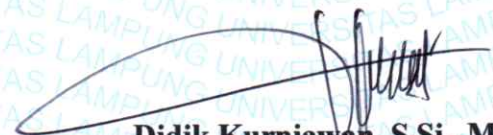


1. **Komisi Pembimbing**


Dr. rer. nat. Akmal Junaidi, M.Sc.
NIP. 19710129 199702 1 001


Ardiansyah, S.Kom., M.Kom.
NIP. 19870128 201803 1 001

2. **Ketua Jurusan Ilmu Komputer**


Didik Kurniawan, S.Si., M.T.
NIP. 19800419 200501 1 004

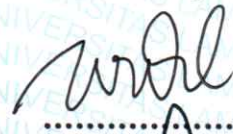
MENGESAHKAN

1. Tim Penguji

Ketua : Dr. rer. nat. Akmal Junaidi, M.Sc



Sekretaris : Ardiansyah, S.Kom., M.Kom



Penguji Utama : Dwi Sakethi, M.Kom.



2. Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam

Dr. Eng. Suripto Dwi Yuwono, S.Si., M.T.
NIP 197407052000031001



Tanggal Lulus Ujian Skripsi : 18 Agustus 2022



PERNYATAAN

Saya yang bertanda tangan di bawah ini, menyatakan bahwa skripsi saya dengan judul “Pengenalan Tulisan Aksara Lampung Dengan *Library Opencv* Menggunakan Metode *Projection Profile* Dan Klasifikasi *K-NN*” merupakan karya saya sendiri dan bukan karya orang lain. Semua tulisan yang tertuang dalam skripsi ini telah mengikuti kaidah penulisan karya ilmiah Universitas Lampung. Apabila dikemudian hari terbukti skripsi saya merupakan hasil penjiplakan atau dibuat oleh orang lain, maka bersedia menerima sanksi berupa pencabutan gelar yang telah saya terima.

Bandar Lampung, 18 Agustus 2022



Rehta Yunani

NPM. 1617051106

RIWAYAT HIDUP



Penulis dilahirkan di Kotaagung pada tanggal 13 Juli 1999, sebagai anak keempat dari empat bersaudara. Ayah bernama Tusiman dan Ibu bernama Yumi Saretha. Penulis menyelesaikan pendidikan pertama mulai dari Sekolah dasar (SD) di SDN 1 Kuripan pada tahun 2010, Sekolah Menengah Pertama (SMP) di SMPN 1 Kotaagung pada tahun 2013, dan Sekolah Menengah Atas (SMA) di SMAN 1 Kotaagung pada tahun 2016.

Pada tahun 2016, penulis terdaftar sebagai mahasiswa jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam melalui jalur SBMPTN. Pada bulan Juli – Agustus 2019/2020, penulis melaksanakan Kuliah Kerja Nyata (KKN Trilateral) di desa Sukaasem Banten. Selama menjadi mahasiswa, penulis aktif mengikuti beberapa kegiatan, antara lain:

1. Menjadi anggota Media Informasi Himpunan Mahasiswa Jurusan Ilmu Komputer pada tahun 2017.
2. Menjadi anggota divisi 1 Informasi dan Dasar Kepemimpinan Kristen UKM Kristen periode pada tahun 2017.
3. Menjadi anggota sie KK Perkantas Lampung pada tahun 2018.
4. Menjadi sekretaris divisi 1 Informasi dan Dasar Kepemimpinan Kristen UKM Kristen pada tahun 2019.
5. Asisten Dosen pada periode 2017/2018 dan 2018/2019.
6. Melaksanakan Kerja Praktek di Kantor OJK Lampung pada tahun 2019.

PERSEMBAHAN

Puji syukur kepada Tuhan Yesus Kristus atas kesempatan, berkat dan kasih karunia yang tak pernah habis sehingga penulis dapat menyelesaikan penelitian.

Kupersembahkan karya ini kepada :

Tuhan Yesus Kristus yang selalu ada sampai saat ini dan terus memberkati dalam banyak hal

Kedua orang tua yang telah mendoakan dan mendukung dalam segala hal.

Keluarga Ilmu Komputer 2016

Serta Almamater Tercinta Universitas Lampung

MOTTO

“Segala perkara dapat kutanggung di dalam Dia yang memberi kekuatan kepadaku,” -Filipi 4 : 13

“But he knows the way that I take; when he has tested me, I will come forth as gold,” -Job 23:10.

SANWACANA

Puji Syukur kepada Tuhan Yesus atas kasih karunia-Nya sehingga penulis dapat menyelesaikan penelitian dengan judul “Pengenalan Tulisan Aksara Lampung Dengan *Library Opencv* Menggunakan Metode *Projection Profile* Dan Klasifikasi *K-NN*” sebagai salah satu syarat memperoleh gelar sarjana Ilmu Komputer di Universitas Lampung.

Penelitian ini tidak lepas dari pihak yang telah membimbing, mendukung, dan membantu kelancaran penyelesaian penelitian ini. Ucapan terima kasih yang tulus penulis sampaikan, kepada:

1. Kedua orangtua, Bapak Tusiman dan Ibu Yumi Saretha, ketiga kakak, Reny margaretha, Rena agatha dan Tusiman kedua yang telah memberikan doa, dukungan, dan motivasi kepada penulis.
2. Bapak Didik Kurniawan, S.Si., MT. selaku Ketua Jurusan Ilmu Komputer.
3. Dr. rer. nat. Akmal Junaidi, M.Sc. sebagai Sekretaris Jurusan Ilmu Komputer FMIPA Universitas Lampung sekaligus pembimbing pertama yang telah memberikan ide, masukan, ilmu, dan nasihat dalam membantu penulis dalam menyelesaikan skripsi ini.
4. Ardiansyah, S.Kom.,M.Kom. selaku pembimbing kedua yang telah memberikan, masukan, ilmu, dan nasihat dalam membantu penulis dalam menyelesaikan skripsi ini.
5. Bapak Dwi Sakethi, S.Si., M.Kom selaku pembahas yang telah memberikan masukan, ide, dan nasihat yang bermanfaat dalam perbaikan pada skripsi ini.
6. Bapak Dr. Eng. Suropto Dwi Yuwono, S.Si., M.T sebagai Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam.
7. Tristiyanto, S.Kom., M.I.S., Ph.D. selaku pembimbing akademik penulis.

8. Seluruh Bapak dan Ibu Dosen yang telah memberikan ilmu pengetahuan yang sangat bermanfaat selama penulis menjadi mahasiswa di Jurusan Ilmu Komputer.
9. Ibu Ade Nora Maela, Kak Zainuddin dan seluruh staf di Jurusan Ilmu Komputer yang telah sabar membantu segala urusan administrasi selama masa perkuliahan.
10. Sahabat Bar bar (Naomy Tiara Dewi, Kavin Fransiskus) yang selalu menemani, mendukung dan mengasihi setiap hari nya
11. KRISKAT 2016 yang berjuang bersama dari awal perkuliahan sampai saat ini yaitu Josua Fernandes Nababan, Friska Daesy Simbolon, Rachelia Dita Anggraini, Yunita Rosalina Manurung, Hagai Aritonang, Putra Sinaga, Stefanus Alvin, Indah Martika Pasaribu, Yolanda Meilani Sirait
12. Taurus yang selalu menjadi semangat.
13. Teman – teman Ilmu Komputer 2016 yang tidak dapat disebutkan satu persatu dan telah berjuang bersama menjalankan studi di Jurusan Ilmu Komputer.

Penulis menyadari bahwa skripsi ini masih jauh dari kesempurnaan, akan tetapi penulis berharap skripsi ini dapat berguna dan bermanfaat bagi para pembaca.

Bandar Lampung, 18 Agustus 2022

Rehta Yunani
NPM. 1617051106

DAFTAR ISI

Halaman

DAFTAR GAMBAR	iii
DAFTAR TABEL	iv
I. PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	3
1.4 Tujuan Penelitian.....	3
1.5 Manfaat Penelitian.....	3
II. TINJAUAN PUSTAKA	4
2.1 Studi Literatur Pengenalan Pola Aksara Lampung	4
2.2 Bahasa dan Aksara Lampung	5
2.2.1 Induk Huruf Aksara Lampung	5
2.2.2 Anak Huruf Aksara Lampung	6
2.2.3 Tanda baca Aksara Lampung.....	8
2.3 Pengenalan Pola.....	9
2.3.1 <i>Preprocessing</i>	9
2.3.1.1 <i>Noise Reduction</i>	10
2.3.1.2 <i>Otsu's Binarization Thresholding</i>	10
2.3.1.3 <i>Dilating dan Eroding</i>	11
2.3.2 <i>Segmentation</i>	12
2.3.3 <i>Feature Extraction</i>	12
2.3.4 <i>Classification dan Recognition</i>	12
2.4 <i>Projection profile</i>	12
2.5 Klasifikasi <i>K-Nearest neighbors</i>	15
2.6 <i>Cross-Validation</i>	19
2.7 <i>Confusion Matrix</i>	19
III. METODOLOGI PENELITIAN	22
3.1 Waktu dan Tempat Pelaksanaan.....	22
3.2 Data dan Alat.....	22

3.2.1	Data	22
3.2.2	Alat	23
3.2.2.1	Perangkat Keras (<i>Hardware</i>)	23
3.2.2.2	Perangkat Lunak (<i>Software</i>).....	23
3.3	Tahapan Penelitian	25
3.3.1	Studi Literatur	26
3.3.2	<i>Data Preprocessing</i>	26
3.3.2.1	<i>Noise Reduction</i>	26
3.3.2.2	<i>Thresholding</i>	27
3.3.2.3	<i>Dilating dan Eroding</i>	27
3.3.2.4	<i>Feature Extraction</i>	27
3.3.2.5	Klasifikasi <i>K-Nearest neighbors (kNN)</i>	28
3.3.2.6	Evaluasi dan Penarikan Kesimpulan.....	34
IV.	HASIL DAN PEMBAHASAN	35
4.1	Hasil <i>Data Preprocessing</i>	35
4.2	Hasil <i>Feature Extraction</i>	41
4.3	Hasil Klasifikasi <i>K-Nearest neighbors (kNN)</i>	44
4.3.1	Hasil <i>Base Classification Model</i>	48
4.3.2	Hasil <i>GridSearchCV Classification Model</i>	49
4.4	Evaluasi Hasil Klasifikasi.....	51
4.4.1	Evaluasi Hasil <i>Base Classification Model</i>	51
4.4.2	Evaluasi <i>GridSearchCV Classification Model</i>	57
4.5	Pembahasan Hasil Klasifikasi	63
4.6	Analisis Kesalahan Klasifikasi	67
4.6.1	Kesalahan Klasifikasi <i>Base Classification Model</i>	68
4.6.2	Kesalahan Klasifikasi <i>GridSearchCV Classification Model</i>	70
V.	SIMPULAN DAN SARAN	72
5.1	Simpulan.....	72
5.2	Saran	73
	DAFTAR PUSTAKA	74

DAFTAR GAMBAR

Gambar	Halaman
1. Karakter Dasar Aksara Lampung.....	5
2. Ilustrasi Proyeksi Horizontal dan Vertikal (Amaliah dkk., 2011).....	14
3. Contoh Penghitungan Proyeksi Horizontal dan Vertikal.	14
4. Alur Pengerjaan Penelitian Pengenalan Aksara Lampung.....	25
5. Visualisasi Sebaran Data <i>Training</i> per Kelas.....	45
6. Visualisasi Sebaran Data <i>Testing</i> per Kelas.....	46
7. <i>Confusion Matrix Base Classification Model</i> dengan Fitur Gabungan.	52
8. <i>Confusion Matrix Base Classification Model</i> Menggunakan Fitur <i>Horizontal Projection Profile</i>	54
9. <i>Confusion Matrix Base Classification Model</i> Menggunakan Fitur <i>Vertical Projection Profile</i>	56
10. <i>Confusion Matrix GridSearchCV Classification Model</i> Menggunakan Fitur Gabungan.....	58
11. <i>Confusion Matrix GridSearchCV Classification Model</i> dengan Fitur <i>Horizontal Projection</i>	60
12. <i>Confusion Matrix GridSearchCV Classification Model</i> dengan Fitur <i>Vertical Projection Profile</i>	62

DAFTAR TABEL

Tabel	Halaman
1. Anak Huruf Terletak di atas Induk Huruf Aksara Lampung	6
2. Anak Huruf yang Terletak di Bawah Induk Huruf	7
3. Anak Huruf yang Terletak di Depan Induk Huruf	7
4. Tanda baca aksara Lampung	8
5. Bentuk Dasar <i>Confusion Matrix</i> (Bekkar et al., 2013).....	19
6. Metrik Evaluasi Berdasarkan <i>Confusion Matrix</i> (Bekkar et al., 2013).....	20
7. Pembagian <i>Dataset</i>	22
8. Daftar Label Target	23
9. Jumlah Fitur per Citra Berdasarkan Jenis Fitur.....	28
10. Konfigurasi <i>Train test split Data Training</i>	29
11. Konfigurasi <i>Base Classification Model</i>	30
12. Rentang Nilai Parameter Model <i>kNN</i>	32
13. Konfigurasi <i>Base model kNN</i> untuk <i>GridSearchCV</i>	33
14. Parameter <i>GridSearchCV</i>	33
15. Hasil <i>Noise Reduction</i> dan <i>Thresholding</i>	37
16. Hasil Citra Tahap <i>Data Preprocessing</i>	40
17. Fitur <i>Horizontal Projection Profile</i>	42
18. Fitur <i>Vertical Projection Profile</i>	43
19. Fitur Gabungan <i>Horizontal</i> dan <i>Vertical Projection Profile</i>	43
20. Jumlah Sebaran Data <i>Training</i> per Kelas.....	45
21. Jumlah Sebaran Data <i>Testing</i> per Kelas	47
22. Konfigurasi Akhir Model <i>kNN</i> Menggunakan <i>GridSearchCV</i>	50
23. Evaluasi <i>Base Classification Model</i> dengan Fitur Gabungan	53
24. Evaluasi <i>Base Classification Model</i> dengan Fitur <i>Horizontal</i> <i>Projection Profile</i>	55

25.	Evaluasi <i>Base Classification Model</i> dengan Fitur <i>Vertical Projection Profile</i>	56
26.	Evaluasi <i>GridSearchCV Classification Model</i> dengan Fitur Gabungan.	59
27.	Evaluasi <i>GridSearchCV Classification Model</i> dengan Fitur <i>Horizontal Projection Profile</i>	61
28.	Evaluasi <i>GridSearchCV Classification Model</i> dengan Fitur <i>Vertical Projection Profile</i>	63
29.	Komparasi Hasil Evaluasi <i>Base model Classification</i> dan <i>GridSearchCV Classification Model</i>	64
30.	Perbandingan Nilai K dan Metode Pembobotan	65
31.	Perbandingan Distribusi Data per Kelas	66
32.	Perbandingan Data Citra Sebelum dan Sesudah <i>Data Preprocessing</i>	67
33.	Jumlah Kesalahan Klasifikasi <i>Base Classification Model</i>	68
34.	Perbandingan Data Citra “a” dengan “la” pada Data <i>Training</i> dan <i>Data Testing</i>	69
35.	Kesalahan <i>Preprocessing</i> Data Citra “a”.	70
36.	Jumlah Kesalahan Klasifikasi <i>GridSearchCV Classification Model</i> .	70

I. PENDAHULUAN

1.1 Latar Belakang

Indonesia merupakan salah satu negara yang kaya akan kebudayaan di setiap daerah, seperti tulisan daerah dan bahasa daerah. Setiap daerah memiliki ragam tulisan dengan ciri khas dan keunikan tersendiri yang bisa digunakan sebagai pembeda dengan daerah lain. Bahasa daerah pun demikian, termasuk peninggalan dari para leluhur yang biasa digunakan untuk berkomunikasi. Tetapi pada saat ini minat mempelajari bahasa dan tulisan daerah sangatlah menurun karena adanya keinginan mengikuti perkembangan jaman daripada melestarikan kebudayaan.

Lampung adalah salah satu daerah yang masih memiliki tulisan dan bahasanya sendiri, tulisan Lampung disebut juga aksara Lampung. Lampung sendiri memiliki aksara Lampung yang menjadi mata pelajaran wajib dari sekolah dasar, ini merupakan langkah untuk melestarikan kebudayaan. Tetapi ketika sudah lulus dari sekolah biasanya masyarakat Lampung secara perlahan mulai lupa tentang aksara dan bahasa Lampung. Apabila berlanjut, hal seperti ini warisan budaya akan hilang dan tergantikan dengan hal-hal baru yang sesuai zamannya. Maka dari itu melestarikan kebudayaan agar masyarakat mencintai ragam bahasa dan Aksara Lampung.

Salah satu pendekatan ilmiah yang peneliti lakukan adalah dengan memanfaatkan kemajuan teknologi di bidang informasi. Bentuk perkembangan teknologi yang dapat digunakan adalah proses pengenalan pola. Pengenalan pola melibatkan mesin komputer untuk dapat melakukan

pengelompokkan citra berupa angka maupun simbol secara otomatis (Andana dkk., 2018).

Pengenalan pola yang digunakan adalah pengenalan karakter dikenal dengan *character recognition* (Mahto et al., 2015). *Character recognition* digunakan untuk mengenali tulisan dengan mesin komputer dan bisa dimasukkan dalam sistem (Gonzales and Woods, 2006). Sampai saat ini pengenalan pola yang sudah berkembang di Indonesia pengenalan pola tulisan tangan karakter angka arab (Nasri et al., 2012), pengenalan citra hasil tulisan tangan (Andana dkk., 2018), pengenalan citra huruf alfabet (Puspasari, 2013). Pengenalan pola merupakan salah satu bidang yang penting dalam membedakan pola satu dengan pola lainnya, dengan kemampuan untuk membedakan pola, pengenalan pola juga dapat menghasilkan tingkat keakuratan yang tinggi (Nasri et al., 2012).

Tujuan dari penelitian ini adalah mengembangkan strategi pengenalan pola terkhusus untuk aksara Lampung. Penelitian yang dilakukan juga berdasarkan penelitian sebelumnya dengan judul Lampung *Handwritten Character Recognition* (Junaidi, 2016). penelitian ini menggunakan metode *projection profile* dalam mendapatkan fitur yang akan digunakan untuk mengenali aksara Lampung.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang sudah dijelaskan, masalah yang akan diselesaikan dalam skripsi ini adalah sebagai berikut:

1. Bagaimana melakukan ekstraksi fitur tulisan aksara Lampung dengan *projection profile*.
2. Bagaimana tingkat keberhasilan *projection profile* dan *K-Nearest neighbors (kNN)* dalam pengenalan karakter aksara Lampung.
3. Bagaimana meningkatkan tingkat akurasi pengenalan karakter aksara Lampung dengan fitur *projection profile* berbasis metode klasifikasi *K-Nearest neighbors (kNN)* melalui optimalisasi nilai parameter.

1.3 Batasan Masalah

Batasan masalah diperlukan agar penelitian ini memaksimalkan dan fokus pada rumusan masalah. Adapun batasan masalah penelitian yakni :

1. Data yang tersedia adalah karakter aksara Lampung yang sudah di konversi menjadi grayscale image dengan ukuran 20 x 20 piksel.
2. Penelitian ini menggunakan *projection profile* sebagai metode ekstraksi fitur.
3. Penelitian ini menggunakan klasifikasi *K-Nearest neighbors (kNN)* sebagai metode klasifikasi karakter aksara Lampung.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah mengetahui level keberhasilan klasifikasi pengenalan karakter tulisan tangan aksara Lampung dengan menggunakan metode ekstraksi *Projection profile* dan metode klasifikasi *K-Nearest neighbors (kNN)*.

1.5 Manfaat Penelitian

Manfaat dari dilakukannya penelitian ini adalah :

1. Mengembangkan pendekatan *handwritten recognition* untuk aksara Lampung, berdasarkan penelitian yang telah dilakukan sebelumnya.
2. Mengetahui tingkat efektivitas *projection profile* dalam mengekstraksi fitur pada karakter aksara Lampung.
3. Mengetahui perbandingan hasil klasifikasi dengan menggunakan *Horizontal Projection Profile*, *Vertical Projection Profile*, dan gabungan kedua jenis *projection profile*.
4. Mengetahui keefektifan metode *K-Nearest neighbors (kNN)* untuk klasifikasi karakter aksara Lampung.

II. TINJAUAN PUSTAKA

2.1 Studi Literatur Pengenalan Pola Aksara Lampung

Upaya melakukan pelestarian aksara Lampung adalah dengan dibuatnya pengenalan aksara yang dapat membaca tulisan tangan aksara Lampung. Upaya yang telah dilakukan misalnya penyediaan *dataset* untuk penelitian Aksara Lampung (Junaidi et al., 2011). Dalam penelitian tersebut, selain menyediakan *dataset*, penelitian ini menggunakan *water reservoir*, *branch points*, *pixel density*, dan *end points* untuk pengenalan aksara Lampung. Akurasi pengenalan tulisan tangan aksara Lampung pada penelitian tersebut adalah 94,27%.

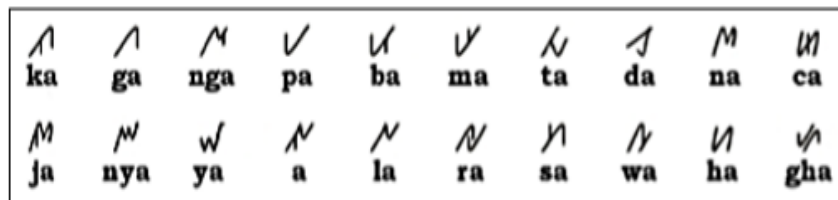
Selain itu, dilakukan juga penelitian untuk pelabelan aksara Lampung secara *semi-supervised*. Dalam penelitian tersebut Aksara Lampung dikelompokkan menggunakan metode *unsupervised learning* (Junaidi et al., 2011). Setelah didapatkan kelompoknya, kemudian dilabeli oleh seorang pakar. Setelahnya akan dilakukan pengenalan algoritma *K-Nearest neighbors*. Akurasi yang didapatkan adalah 86,21% dan untuk *dataset benchmark Modified National Institute of Standards and Technology (MNIST)* menghasilkan akurasi 94,81%. Penelitian lainnya adalah penelitian tentang pendekatan pemasangan diakritik dengan huruf induknya (Junaidi et al., 2011). Dalam penelitian tersebut, dilakukan dua kali pengenalan aksara yaitu pengenalan huruf induk dan dilanjutkan dengan pengenalan diakritik. Setelah keduanya diidentifikasi, kemudian dilakukan pemasangan diakritik terhadap huruf induk yang seharusnya. Pendekatan yang digunakan adalah *Gaussian Mixture Model* dengan optimalisasi melalui (*Expectation-Maximization Algorithm (EM)*). Hasil yang diperoleh melalui EM adalah 92.2%.

2.2 Bahasa dan Aksara Lampung

Aksara Lampung adalah pengembangan bentuk aksara Devanagari yang berasal dari India Selatan. Sistem aksara Lampung terbagi menjadi induk huruf, anak huruf, dan tanda baca. Induk huruf aksara Lampung terbagi menjadi 20 jenis, anak huruf terbagi menjadi 12 jenis, dan tanda baca terbagi menjadi 5 jenis. Kombinasi antara induk huruf dan anak huruf pada aksara Lampung menghasilkan cara baca yang berbeda untuk mengakomodir pembentukan kata dalam bahasa Lampung.

2.2.1 Induk Huruf Aksara Lampung

Aksara Lampung memiliki induk huruf yang terdiri dari 20 jenis karakter yang dibaca dengan awalan konsonan atau kombinasi konsonan yang diakhiri dengan huruf – “a”. Induk huruf dalam bahasa Lampung berfungsi untuk memberikan bunyi utama pembentuk kata dalam bahasa Lampung. Karakter Induk huruf bahasa Lampung disajikan dalam Gambar 1.









Gambar 1. Karakter Dasar Aksara Lampung.

Berdasarkan Gambar 1, terlihat bahwa karakter utama aksara Lampung atau induk huruf hanya berakhiran “a”, sehingga tidak mungkin untuk membentuk kata yang beragam dalam bahasa Lampung hanya dengan menggunakan induk huruf. Keterbatasan fungsi ini yang mengakibatkan karakter induk huruf harus di kombinasikan lebih lanjut dengan anak huruf untuk dapat memenuhi kebutuhan pembentukan kosakata dalam bahasa Lampung.

2.2.2 Anak Huruf Aksara Lampung

Anak huruf Aksara Lampung terdiri dari 12 karakter yang terletak di atas, bawah, dan depan (samping kanan) induk huruf. Anak huruf yang terletak di atas induk huruf disajikan dalam Tabel 1.

Tabel 1. Anak Huruf Terletak di atas Induk Huruf Aksara Lampung.




Nama	Simbol Aksara	Keterangan
<i>Bicek</i>		Simbol vokal “e”
<i>Ulan “i”</i>		Simbol vokal “i”
<i>Ulan “e”</i>		Simbol vokal “e”
<i>Datasan</i>		Simbol pengganti konsonan “n”
<i>Rejunjung</i>		Simbol pengganti konsonan “r”
<i>Tekelubang</i>		Simbol pengganti konsonan “ng”

Berdasarkan Tabel 1, berikut dijabarkan penjelasan dari masing-masing fungsi dari anak huruf yang terletak di atas induk huruf. *Bicek* adalah anak huruf dengan bentuk garis tegak yang terletak di atas induk huruf. *Bicek* melambangkan bunyi “e” seperti pada kata “pepaya”. *Ulan* adalah anak huruf yang berbentuk setengah lingkaran kecil yang terletak di atas induk huruf. *Ulan* terdiri atas dua macam, *Ulan* yang menghadap ke atas melambangkan bunyi “i”, sedangkan *Ulan* yang menghadap ke bawah melambangkan bunyi “e” seperti pada kata “bebek”. *Datasan* adalah anak huruf yang berbentuk dua garis mendatar (seperti simbol sama-dengan) yang terletak di atas induk huruf. *Datasan* melambangkan bunyi “n”.

Rejunjung adalah anak huruf yang berbentuk spiral yang terletak di atas induk huruf. *Rejunjung* melambangkan bunyi “r”. *Tekelubang* adalah anak huruf yang berbentuk garis mendatar (seperti tanda hubung dalam ejaan bahasa Indonesia) yang terletak di atas induk

huruf. *Tekelubang* melambangkan bunyi “ng”. Kemudian, anak huruf yang terletak dibawah induk huruf disajikan dalam Tabel 2.




Tabel 2. Anak Huruf yang Terletak di Bawah Induk Huruf.

Nama	Simbol Aksara	Keterangan
<i>Bitan “o”</i>		Simbol vokal “o”
<i>Bitan “u”</i>		Simbol vokal “u”
<i>Tekelungau</i>		Siimbol vokal “au”

Berdasarkan Tabel 2, dijelaskan fungsi anak huruf yang yang berada di bawah induk huruf. *Bitan* adalah anak huruf yang terletak di bawah induk huruf. *Bitan* terdiri atas dua macam. *Bitan* yang berupa garis pendek mendatar melambangkan bunyi “u” dan *Bitan* yang berupa garis tegak melambangkan bunyi “o”. *Tekelungau* adalah anak huruf yang berbentuk setengah lingkaran kecil yang terletak di bawah induk huruf. *Tekelungau* melambangkan bunyi “au”.

Selain anak huruf yang berada di atas dan bawah induk huruf, terdapat anak huruf yang terletak depan (samping kanan) induk huruf. Anak huruf yang terletak di depan (samping kanan) induk huruf disajikan dalam Tabel 3.

Tabel 3. Anak Huruf yang Terletak di Depan Induk Huruf.

Nama	Simbol Aksara	Keterangan
<i>Tekelingai</i>		Simbol vokal “ai”
<i>Keleniah</i>		Simbol pengganti konsonan “ah”
<i>Nengen</i>		Simbol mati aksara

Berdasarkan Tabel 3, *Tekelingai* adalah anak huruf yang berbentuk garis tegak yang terletak di depan induk huruf. *Tekelingai* melambangkan bunyi “ai”. *Keleniah* adalah anak huruf yang berbentuk seperti huruf ha, tetapi kecil yang terletak di depan induk huruf. *Keleniah* melambangkan bunyi “ah”. *Nengen* adalah anak huruf yang berbentuk garis miring yang terletak di depan induk huruf.

Nengen melambangkan huruf yang berada di belakang *Nengen* menjadi huruf mati. Akan tetapi, untuk melambangkan bunyi “ng”, “r”, “n”, “y”, “h” atau “w” *Nengen* tidak digunakan. Bunyi-bunyi itu dilambangkan dengan menggunakan anak huruf berikut, bunyi “ng” menggunakan *Tekelubang*, bunyi “r” menggunakan *Rejunjung*, bunyi “n” menggunakan *datas*, bunyi “y” menggunakan *Tekelingai*, bunyi “h” menggunakan *Keleniah*, Bunyi “w” menggunakan *Tekelungau*.

2.2.3 Tanda baca Aksara Lampung

Dalam sistem aksara Lampung, selain dari induk dan anak huruf, terdapat lima jenis tanda baca. Tanda baca yang terdapat dalam sistem aksara Lampung disajikan dalam Tabel 4.

Tabel 4. Tanda baca aksara Lampung.

Nama	<i>Ngemula</i>	<i>Beradu</i>	<i>Kuma</i>	<i>Ngulih</i>	Tanda Seru
Aksara	☼	○	✓	ㄣ	y

Penjelasan dari tanda baca aksara Lampung yang terdapat dalam Tabel 4 yakni, *Ngemula* berfungsi untuk memulai sebuah kalimat dalam aksara Lampung. *Beradu*, berfungsi tanda *Beradu* berlawanan dari tanda *Ngemula*. Tanda ini diletakkan di akhir kalimat untuk menyelesaikannya. *Kuma*, sebagai tanda baca, *Kuma* digunakan untuk menjeda kalimat atau untuk memisahkan elemen dalam serangkaian (tiga atau lebih) hal dalam satu kalimat. *Ngulih*, tanda yang diletakkan

di akhir kalimat sebagai tanda baca dalam kalimat yang berisi pertanyaan tentang sesuatu. Tanda Seru, adalah tanda baca untuk menyatakan bahwa kalimat mengandung perintah atau pernyataan tegas.

2.3 Pengenalan Pola

Bidang ilmu yang dimanfaatkan untuk melakukan klasifikasi atau pencitraan objek berdasarkan pengukuran kuantitatif fitur (ciri) sebuah citra disebut dengan pengenalan pola (pattern recognition) (Putra, 2010). Pengenalan tulisan tangan (*handwriting recognition*) adalah kemampuan komputer untuk dapat menginterpretasikan *input* citra tulisan tangan yang dapat dimengerti dari sumber seperti dokumen kertas, foto, layar sentuh dan perangkat lainnya.

Pola merupakan objek struktur dan dapat diidentifikasi melalui ciri-cirinya. Ciri-ciri tersebut digunakan untuk membedakan suatu pola dengan pola lainnya. Ciri yang baik adalah ciri yang memiliki tingkat pembeda yang tinggi, sehingga pengelompokan pola berdasarkan ciri yang dimiliki dapat dilakukan dengan keakuratan yang tinggi. Ciri pada suatu pola diperoleh dari hasil pengukuran terhadap objek uji. (Nasri et al., 2012). Proses *character recognition* yang akan digunakan terdiri dari beberapa fase. Urutan fase-fase tersebut adalah sebagai berikut.

2.3.1 Preprocessing

Tahap awal pengolahan citra yang umum dilakukan disebut dengan *Preprocessing*. *Preprocessing* juga merupakan proses peningkatan kualitas citra agar bias meningkatkan keberhasilan dalam proses pengolahan citra berikutnya. Data citra yang akan diambil harus diproses lebih lanjut untuk penyeragaman data citra yang digunakan sebagai *dataset*. Tahap *Preprocessing* melibatkan berbagai tahap operasi untuk meningkatkan kualitas citra agar dapat diproses lebih lanjut. *Preprocessing* melibatkan *noise reduction*, binerisasi, deteksi tepi, *thresholding*, *dilating*, dan *eroding*.

2.3.1.1 *Noise Reduction*

Noise reduction adalah proses untuk menghilangkan *noise* yang dapat menyebabkan kualitas citra menurun. *Noise reduction* merupakan hal yang penting untuk dilakukan mengingat semakin sedikit *noise* yang dimiliki oleh citra, akan semakin baik kualitas citra yang dihasilkan (Junaidi, 2016).

Noise dapat disebabkan oleh beberapa faktor. Pertama, kondisi suhu sensor kamera yang terlalu tinggi pada saat pengambilan citra. Kedua, kurang baiknya kondisi pencahayaan pada saat pengambilan citra. Ketiga, kondisi lingkungan yang dapat menyebabkan munculnya *noise* (Cattin, 2016).

Kemunculan *noise* yang paling sering ditemui pada kasus akuisisi data citra tulisan tangan disebabkan oleh buruknya kondisi cahaya, dan *noise* yang dihasilkan dari sirkuit elektronik kamera atau *scanner*, jenis *noise* ini disebut dengan *Gaussian noise* (Cattin, 2016).

Hal ini tentu berlaku pada *dataset* tulisan tangan aksara Lampung, di mana proses akuisisi data dilakukan dengan melakukan scan hasil tulisan tangan aksara Lampung dengan menggunakan *scanner* (Junaidi et al., 2011). Sehingga untuk menghilangkan *noise* yang terdapat dalam citra akan digunakan *Gaussian filter*, yang dapat menghilangkan atau mengurangi *Gaussian noise* yang terdapat di dalam citra.

2.3.1.2 *Otsu's Binarization Thresholding.*

Proses pengenalan pola karakter tulisan tangan sangat dipengaruhi oleh kemampuan untuk mengekstraksi pola citra yang ingin dikenali dari latar belakang citra. Oleh karena itu,

penting untuk memberikan batas yang tepat antara citra yang ingin dikenali dan latar belakang citra (Otsu, 1979).

Metode *Otsu's Binarization Thresholding*, bekerja secara otomatis untuk memberikan nilai threshold pada komponen piksel yang ada dalam citra. Hal ini dilakukan dengan memaksimalkan penghitungan nilai diskriminan untuk setiap piksel. Citra yang dihasilkan akan memiliki nilai piksel yang berbeda dikarenakan nilai threshold yang berbeda, dan membuat pola citra yang akan di ekstraksi memiliki perbedaan yang jelas dengan latar belakang citra (Otsu, 1979).

2.3.1.3 Dilating dan Eroding.

Dalam pengenalan pola, citra yang akan diolah perlu diubah ke dalam bentuk *binary image*. Untuk mengatasi citra biner yang memiliki bentuk yang kurang sempurna, maka dapat dilakukan serangkaian operasi pada citra dengan cara mengubah bentuk yang mendasari citra biner, hal ini dikenal dengan *morphological operations* (Ritter and Wilson, 2000).

Untuk melakukan operasi morfologi atau *morphological operations*, citra biner yang akan digunakan perlu digabungkan dengan *binary structuring element* yang berfungsi untuk memberikan standar bentuk terhadap setiap piksel pada citra biner (Szeliski, 2021), kemudian hasil operasi biner yang dilakukan akan dipengaruhi oleh nilai *kernel* dan *threshold* yang ditetapkan (Szeliski, 2021).

Terdapat dua jenis operasi morfologi yang sering dilakukan untuk memperbaiki bentuk citra biner yang kurang sempurna. Pertama *dilating*, yakni operasi morfologi pada citra biner untuk menambahkan nilai vektor pada setiap piksel berdasarkan nilai maksimal vektor biner yang telah

ada pada citra (Ritter and Wilson, 2000). Kedua, *eroding*, yakni operasi morfologi pada citra biner untuk menghilangkan nilai yang merupakan komplemen dari hasil output yang dihasilkan *dilating* (Ritter and Wilson, 2000).

2.3.2 Segmentation

Segmentasi digunakan agar karakter yang tergabung dapat terpisah satu sama lain. Metode segmentasi yang digunakan untuk memisahkan objek tanpa terpengaruh oleh kemiripan objek disebut dengan *connected component labeling* (Nasri et al., 2012). Tingkat akurasi pengenalan karakter ditentukan oleh keberhasilan segmentasi. Tujuan segmentasi adalah membagi serangkaian karakter menjadi karakter-karakter yang terpisah.

2.3.3 Feature Extraction

Feature extraction adalah metode yang digunakan untuk mengambil bahan dari data yang masih mentah untuk digunakan mengenali *image*. Data yang digunakan adalah representasi dari karakter terbentuk image yang selanjutnya diekstrak nilai projectionnya. Ekstraksi fitur merupakan fase penting dalam mengenali suatu karakter pada pengenalan pola karena dapat menentukan tingkat akurasinya.

2.3.4 Classification dan Recognition

Klasifikasi merupakan proses mengenali kelas – kelas berdasarkan format yang berbentuk fitur. Perbandingan fitur input dengan pola tersimpan kemudian dilakukan untuk menemukan pencocokan kelas yang sesuai dengan input citra (Mahto et al., 2015).

2.4 Projection profile

Projection profile adalah menambahkan jumlah piksel hitam per baris dan per kolom (Chacko and Dhanya, 2015). Secara umum, segmentasi baris dalam dokumen cetak (printed document) dilakukan menggunakan metode

projection profile, sehingga menghasilkan tulisan dengan baris yang terpisah dengan jelas. Namun, metode ini juga dapat diadaptasi untuk dokumen tulisan tangan (Likforman-Sulem et al., 2007).

Projection profile terbagi menjadi dua yaitu, *Horizontal Projection Profile* (HPP) dan *Vertical Projection Profile* (VPP). *Horizontal Projection Profile* sering digunakan untuk segmentasi baris dan *Vertical Projection Profile* sering digunakan untuk segmentasi karakter.

Pada penelitian ini akan digunakan dua jenis *projection profile*, *Horizontal Projection Profile* (HPP) dilakukan dengan cara menelusuri baris horizontal dan menghitung akumulasi nilai piksel hitam citra biner, sedangkan *Vertical Projection Profile* sebagai representasi piksel hitam yang diakumulasi sepanjang kolom vertikal. Adapun fungsi HPP dan VPP secara matematis dapat dilihat pada Persamaan 1. Piksel citra dirumuskan dengan fungsi $f(x,y)$ di mana x dan y secara berturut-turut mewakili baris dan kolom. Parameter “ n ” menjelaskan banyaknya kolom pada suatu citra. Pada citra dengan ukuran “ m ” baris dan “ n ” kolom, proyeksi horizontal dan vertikal direpresentasikan oleh persamaan berikut (Javed et al., 2013).

$$\begin{aligned}
 &HPP(x) \\
 &= \sum_{1 \leq y \leq n} f(x, y) \dots \dots \dots (1)
 \end{aligned}$$

Persamaan (1). *Horizontal Projection Profile*.

$$\begin{aligned}
 &VPP(x) \\
 &= \sum_{1 \leq x \leq m} f(x, y) \dots \dots \dots (2)
 \end{aligned}$$

Persamaan (2). *Vertical Projection Profile*.

Persamaan 1 dan Persamaan 2 dijabarkan sebagai berikut :

HPP = *Horizontal Projection Profile*

VPP = *Vertical Projection Profile*

x = Indeks Baris

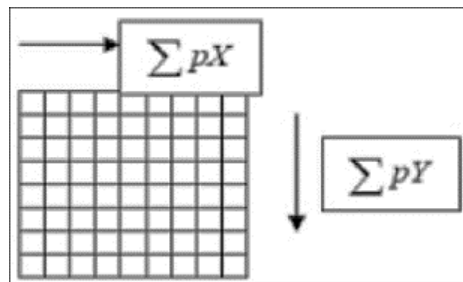
y = Indeks Kolom

m = Jumlah Baris

n = Jumlah Kolom

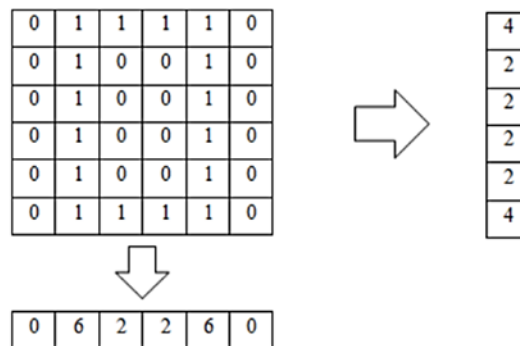
$f(x,y)$ = posisi baris dan kolom

Proyeksi horizontal dan vertikal dalam sebuah citra diilustrasikan pada Gambar 2.



Gambar 2. Ilustrasi Proyeksi Horizontal dan Vertikal (Amaliah dkk., 2011).

Ilustrasi penghitungan proyeksi horizontal dan vertikal yang ditunjukkan pada Gambar 2, disajikan lebih lanjut dalam Gambar 3.



Gambar 3. Contoh Penghitungan Proyeksi Horizontal dan Vertikal.

Gambar 3 menyajikan contoh penghitungan proyeksi horizontal dan vertikal pada data citra biner. Nilai projection untuk proyeksi horizontal terlihat pada bagian kanan dan proyeksi vertikal pada bagian bawah Gambar 3.

Hasil perhitungan fitur *projection profile* yang didapat pada Gambar 3 bernilai $\{0,6,2,2,6,0,4,2,2,2,2,4\}$. Nilai yang didapatkan akan digunakan sebagai nilai fitur *untuk* mengidentifikasi setiap data citra.

Projection profile horizontal dan *projection profile* vertikal akan membuat rangkaian nilai presentasi dari jumlah piksel *foreground* pada kolom dan baris pada sebuah citra. Nilai yang dipresentasikan akan dinormalisasi ke dalam range 0-1 dengan membagi setiap nilai *Projection profile* horizontal dengan jumlah kolom citra dan membagi setiap nilai *Projection profile* vertikal dengan jumlah baris citra (Shabbir and Siddiqi, 2016).

Hasil yang akan dibentuk dalam rangkaian nilai desimal antara 0 dan 1 adalah proses normalisasi berupa fitur proyeksi. Fitur yang dinormalisasi dapat dikombinasikan dengan nilai fitur lain, sehingga menghasilkan nilai fitur baru yang setara.

2.5 Klasifikasi *K-Nearest neighbors*

Klasifikasi adalah sebuah proses menemukan model yang mencitrakan perbedaan antara kelas data atau himpunan yang memiliki tujuan untuk dapat digunakan untuk memprediksi kelas dari objek label yang kelasnya tidak diketahui (Han et al., 2012). *K-Nearest neighbors* (*kNN*) adalah metode klasifikasi yang bekerja dengan cara menemukan sekelompok “*k*” objek dalam data training yang memiliki nilai jarak paling dekat dengan objek pada data testing, dan mendasarkan penetapan label pada dominasi kelas tertentu berdasarkan jumlah neighbor yang didapatkan (Wu and Kumar, 2009). *kNN* merupakan salah satu teknik *lazy learning*. *kNN* dilakukan dengan mencari kelompok *k* objek dalam data *Training* yang paling dekat (mirip) dengan objek pada data baru atau data (Wu and Kumar, 2009).

K-Nearest neighbors juga melakukan klasifikasi data dari objek yang berdasarkan jumlah “*k*” dari data *Training* dengan jarak terdekat (Wahyono dan Nugroho, 2018). Tujuan dari klasifikasi *kNN* adalah melakukan klasifikasi berdasarkan atribut dan training sample untuk pada objek baru. Diberikan titik *query* agar ditemukan sejumlah *k* objek atau titik *Training* yang paling dekat dengan *query*. Klasifikasi menggunakan voting yang paling

banyak di antara klasifikasi dari k objek. Algoritma kNN menggunakan klasifikasi ketetanggaan untuk menilai prediksi dari *query instance* yang baru.

Algoritma seperti *brute force*, *K-D Tree*, dan *Ball Tree* merupakan algoritma yang dapat digunakan dalam mendapatkan nilai *nearest neighbors*. *Brute force* bekerja dengan cara menghitung nilai jarak untuk setiap pasangan data yang akan *diklasifikasikan* dengan menggunakan kNN . Metode ini efektif untuk digunakan pada jumlah data yang sedikit, mengingat semakin banyaknya pasangan data yang akan dicari nilai jaraknya, maka daya komputasi yang dibutuhkan akan semakin besar dan mengakibatkan inefisiensi waktu komputasi.

Mengingat inefisiensi waktu yang diperlukan untuk mencari nilai *nearest neighbors* apabila menggunakan algoritma *brute force*, terdapat algoritma lain yang dapat digunakan untuk mencari nilai *nearest neighbors* yakni *K-D Tree*. Setiap data yang dimasukkan ke dalam bentuk *K-D Tree* disimpan dalam bentuk node dalam tree, di mana setiap node memiliki dua opsi nilai yakni null, atau pointer ke node lain yang ada dalam *K-D Tree* (Bentley, 1975). Apabila diilustrasikan dengan tiga titik data A, B, dan C, di mana nilai jarak titik A dengan B sangat jauh, dan titik B dekat dengan titik C, maka algoritma *K-D Tree* tidak akan menghitung keseluruhan jarak untuk setiap pasangan data, dan dapat mengambil kesimpulan bahwa titik A dengan titik C memiliki jarak yang sangat jauh (Bentley, 1975).

Dalam mencari jarak data antar kelas dalam kNN , terdapat beberapa metrik pengukuran jarak yang sering digunakan yakni *Manhattan Distance* dan *Euclidean Distance*. Perhitungan untuk mencari nilai *Manhattan Distance* disajikan dalam Persamaan (3).

$$d_{st} = \sum_{j=1} |x_{sj} - y_{tj}| \dots \dots \dots (3)$$

Persamaan (3). Persamaan *Manhattan Distance*.

Untuk melakukan klasifikasi menggunakan *kNN*, tahapan yang akan dilakukan adalah dengan menentukan parameter "*k*" atau jumlah kelompok data terdekat. Kemudian menghitung jarak antar pasangan data, dengan menggunakan metrik pengukuran jarak yang dapat menggunakan *Manhattan Distance* atau *Euclidean Distance*, sebagaimana ditunjukkan dalam Persamaan 5. Setelah jarak untuk setiap pasangan data dihitung dengan menggunakan salah satu metrik pengukuran jarak, langkah selanjutnya adalah dengan mengurutkan hasil pengukuran jarak dengan urutan menaik dari nilai jarak terkecil menuju nilai jarak terbesar. Setelah pengurutan untuk setiap nilai jarak yang didapatkan dari hasil penghitungan jarak antar pasangan data dilakukan, kemudian urutan data akan dikelompokkan berdasarkan kelas target klasifikasi. Kemudian kelas akan ditentukan menggunakan nilai k yang telah ditentukan sebelumnya, dan data mayoritas berdasarkan nilai k (*Nearest neighbors*) yang paling mayoritas akan menentukan kelas tersebut.

Seiring dengan berkembangnya algoritma *kNN*, permasalahan yang ditemukan adalah dalam menentukan nilai "*k*" untuk menentukan jumlah data yang termasuk dalam *nearest neighbors* (Gou et al., 2011). Metode yang umum digunakan untuk mengelompokkan data ke dalam *nearest neighbor*, adalah dengan mengelompokkan objek data berdasarkan nilai jarak data, dimana setiap titik data dianggap memiliki peluang yang sama untuk dapat dikelompokkan ke dalam *nearest neighbors (uniform kNN)*, namun hal ini memiliki kekurangan dimana radius pengelompokkan objek data sangat sensitif terhadap nilai "*k*" (Gou et al., 2011).

Mengingat klasifikasi menggunakan *kNN* sangat sensitif terhadap radius nilai "*k*", kemudian dirumuskan metode untuk menentukan apakah sebuah data dapat dimasukkan ke dalam *nearest neighbors*, dengan melakukan pembobotan data untuk setiap objek data yang akan di klasifikasikan. Pembobotan dilakukan dengan cara memberikan bobot menggunakan fungsi *distance*, dengan bobot lebih besar akan diberikan untuk objek data dengan nilai *distance* lebih dekat dibandingkan dengan objek data yang memiliki nilai

distance lebih jauh, hal ini disebut dengan weighted kNN (WkNN) (Gou et al., 2011).

2.6 *Cross-Validation*

Cross-Validation adalah metode statistik untuk melakukan evaluasi dan perbandingan hasil algoritma dengan cara membagi data menjadi dua bagian, bagian pertama digunakan untuk melatih model, bagian kedua digunakan untuk melakukan validasi pada model (Refaeilzadeh et al., 2009).

Bentuk *Cross-Validation* yang umum digunakan adalah k-fold *Cross-Validation*. K-fold *Cross-Validation* dilakukan dengan cara membagi data menjadi k-subbagian, kemudian dilakukan perulangan *Training* sebanyak “k” kali, di mana jumlah data yang berasal dari k-subbagian sebanyak k-1 menjadi data yang digunakan untuk *Training* (Refaeilzadeh et al., 2009). Dalam kasus data mining dan *machine learning*, 10-fold *Cross-Validation* dengan jumlah k = 10 merupakan metode yang paling sering digunakan (Refaeilzadeh et al., 2009).

2.7 *Confusion Matrix*

Performa model klasifikasi secara umum dapat dideskripsikan dengan menggunakan *Confusion Matrix* (Kuhn and Johnson, 2013). *Confusion Matrix* didapatkan dengan membandingkan komponen nilai kelas aktual dengan nilai kelas hasil prediksi.

Bentuk dasar dari *Confusion Matrix* adalah matriks 2x2, yang digunakan untuk kasus klasifikasi biner (Bekkar et al., 2013). Bentuk dasar dari *Confusion Matrix*, disajikan pada Tabel 5.

Tabel 5. Bentuk Dasar *Confusion Matrix* (Bekkar et al., 2013)

	Predicted Positive	Predicted Negative
<i>Actual Positive</i>	<i>True Positive</i> (TP)	<i>False Negative</i> (FN)
<i>Actual Negative</i>	<i>False Positive</i> (FP)	<i>True Negative</i> (TN)

Penjabaran dari komponen pada Tabel 5, dijabarkan sebagai berikut :

- a. *True Positive*, adalah jumlah kasus positif yang *diklasifikasikan* dengan benar ke dalam kasus positif.
- b. *False Negative*, adalah jumlah kasus positif yang *diklasifikasikan* dengan salah ke dalam kasus negatif.
- c. *False Positive*, adalah jumlah kasus negatif yang *diklasifikasikan* dengan salah ke dalam kasus positive.
- d. *True Negative (TN)*, adalah jumlah kasus negatif yang *diklasifikasikan* benar ke dalam kasus negatif.

Berdasarkan hasil *Confusion Matrix*, terdapat beberapa metrik evaluasi yang umum digunakan untuk mengukur performa model klasifikasi. Metrik evaluasi tersebut disajikan dalam Tabel 6.

Tabel 6. Metrik Evaluasi Berdasarkan *Confusion Matrix* (Bekkar et al., 2013).

Metrik Evaluasi	Persamaan
<i>Accuracy</i>	$\frac{TP + TN}{TP + TN + FP + FN}$
<i>Error rate = 1-Accuracy</i>	$\frac{FP + FN}{TP + TN + FP + FN}$
<i>Sensitivity (Recall)</i>	$\frac{TP}{TP + FN}$
<i>Specificity</i>	$\frac{TN}{TN + FP}$
<i>Precision</i>	$\frac{TP}{TP + FP}$

Berdasarkan Tabel 6, *Accuracy*/akurasi merupakan metrik evaluasi yang digunakan untuk menghitung tingkat *efektivitas* model klasifikasi, dengan cara menghitung tingkat peluang nilai positif. Kemudian, *Error rate* adalah nilai estimasi tingkat error berdasarkan hasil klasifikasi yang dihasilkan oleh model.

Recall menunjukkan nilai tingkat akurasi untuk hasil klasifikasi positif, sementara *precision* mengukur berapa banyak jumlah klasifikasi positif, yang merupakan kasus positif. Kemudian *Specificity* digunakan untuk mengukur

probabilitas klasifikasi salah adalah kasus positif berdasarkan nilai *True Negative* dan *False Positive*. Nilai *precision* menunjukkan berapa hasil prediksi positif yang benar merupakan kasus positif (Bekkar et al., 2013).

Selain metrik pengukuran yang telah disajikan dalam Tabel 6, terdapat metrik pengukuran lain yang merupakan metrik evaluasi kombinasi dari metrik evaluasi yang berada pada Tabel 6. Metrik pengukuran tersebut adalah :

a. *F-Measure*.

F-Measure adalah metrik yang mengukur rata-rata harmonik antara nilai *Precision* dan *Recall* (Bekkar et al., 2013). *F-Measure* disajikan dalam Persamaan (6).

$$F - Measure = \frac{2 \times Recall \times Precision}{Recall \times Precision} \dots \dots \dots (6)$$

Persamaan (6). Persamaan *F-Measure*.

b. *Matthew Correlation Coefficient* (MCC).

Matthew Correlation Coefficient (MCC) merupakan metrik pengukuran yang dianggap paling baik untuk evaluasi model klasifikasi (Jurman et al., 2012), dikarenakan MCC mampu memberikan nilai pengukuran yang konsisten baik untuk kasus klasifikasi biner atau *multiclass*.

Selain itu, MCC memberikan hasil yang baik dalam kondisi sebaran data yang seimbang maupun tidak seimbang (Jurman et al., 2012). Metrik evaluasi MCC disajikan dalam Persamaan 7.

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

Persamaan (7). Persamaan *Mathew Correlation Coefficient* (MCC)

Berdasarkan Persamaan 7, nilai yang dapat dihasilkan oleh MCC adalah -1 untuk model klasifikasi paling buruk, dan nilai 1 untuk model dengan klasifikasi paling baik (Jurman et al., 2012).

III. METODOLOGI PENELITIAN

3.1 Waktu dan Tempat Pelaksanaan

Penelitian ini akan dilaksanakan pada semester ganjil tahun ajaran 2019/2020, di Jurusan Ilmu Komputer Fakultas Ilmu Pengetahuan Alam Universitas Lampung.

3.2 Data dan Alat

Data dan alat yang digunakan dalam penelitian ini dijabarkan sebagai berikut.

3.2.1 Data

Data yang digunakan dalam penelitian ini adalah *dataset Lampung Handwritten Characters* (Junaidi et al., 2011) yang dapat diunduh secara terbuka melalui laman web berikut <http://patrec.cs.tu-dortmund.de/cms/en/home/Resources/>. *Dataset* karakter aksara Lampung tersebut sudah melalui proses *preprocessing* dan segmentasi sehingga bentuk karakter yang akan digunakan sudah dalam bentuk karakter tunggal dengan ukuran 20 x 20 piksel dalam format PGM (*gray image*). Pembagian *dataset* dilakukan berdasarkan penelitian oleh (Junaidi et al., 2011) yang digunakan dalam penelitian ini disajikan dalam Tabel 7.

Tabel 7. Pembagian *Dataset*

No	Pembagian Data	Jumlah Data
1	Data <i>Training</i>	24287
2	Data <i>Testing</i>	7853

Berdasarkan Tabel 7, komposisi data *Training* yang akan digunakan dalam penelitian ini adalah sebesar 75,56%, dimana data *Training*

akan digunakan juga sebagai data validasi dalam pembentukan model, sedangkan data *Testing* sebesar 24,43%, dari keseluruhan jumlah *dataset* yang akan digunakan dalam penelitian.

Dataset yang digunakan memiliki 18 label target yang berfungsi sebagai *target class*. Label target yang akan menjadi kelas yang digunakan untuk klasifikasi disajikan dalam Tabel 8.

Tabel 8. Daftar Label Target.

No	Nama Label Target
1	"a"
2	"pa"
3	"ca"
4	"da"
5	"ga"
6	"ha"
7	"ja"
8	"ka"
9	"la"
10	"ma"
11	"na"
12	"nga"
13	"nya"
14	"pa"
15	"sa"
16	"ta"
17	"wa"
18	"ya"

3.2.2 Alat

Alat yang digunakan dalam penelitian ini terbagi menjadi dua bagian, yakni perangkat keras dan perangkat lunak.

3.2.2.1 Perangkat Keras (*Hardware*)

Perangkat keras yang digunakan dalam penelitian ini yaitu :

- a. *Processor* : AMD A9-9420e RADEON R5 1.80 GHz.
- b. *Installed RAM* : 8.00 GB

3.2.2.2 Perangkat Lunak (*Software*)

Penelitian ini menggunakan perangkat lunak yakni :

- a. *Visual Studio Code* 1.69.0
Aplikasi *open source* yang digunakan sebagai *code-editor* dalam menuliskan program dalam bahasa *python*.
- b. *Jupyter* 2022.6. 1101950301
Ekstensi yang digunakan untuk dapat menjalankan *Jupyter notebook* pada *Visual Studio Code*.
- c. *Library OpenCV* 4.6.0.66
Library OpenCV adalah salah satu *library* yang digunakan pada bahasa *python* untuk rekayasa file citra dan *computer vision* secara real-time.
- d. *Library Scikit-learn* 0.23.2
Library Scikit-learn(*sklearn*) adalah salah satu *library machine learning* dan analisis data dalam bahasa *python*. Seperti untuk keperluan klasifikasi, regresi, dan klustering.
- e. *Library Numpy* 1.19.2
Library Numpy adalah salah satu *library* yang digunakan dalam *python* untuk melakukan pengolahan *array* dengan data numerik, objek, dan string.
- f. *Library Pandas* 1.1.13
Library Pandas adalah salah satu *library* untuk melakukan analisis data dan pengolahan struktur data dalam bahasa *python*.
- g. *Library Seaborn* 0.11.0
Library Seaborn adalah *library* visualisasi data statistikal dalam bahasa *python*.
- h. *Library Matplotlib* 3.3.2
Library Matplotlib adalah *library* visualisasi dalam bentuk diagram yang tersedia dalam bahasa *python*.
- i. *Library Joblib* 0.17.0
Library Joblib adalah *library* dengan fungsi untuk melakukan *pipelining job* seperti untuk melakukan

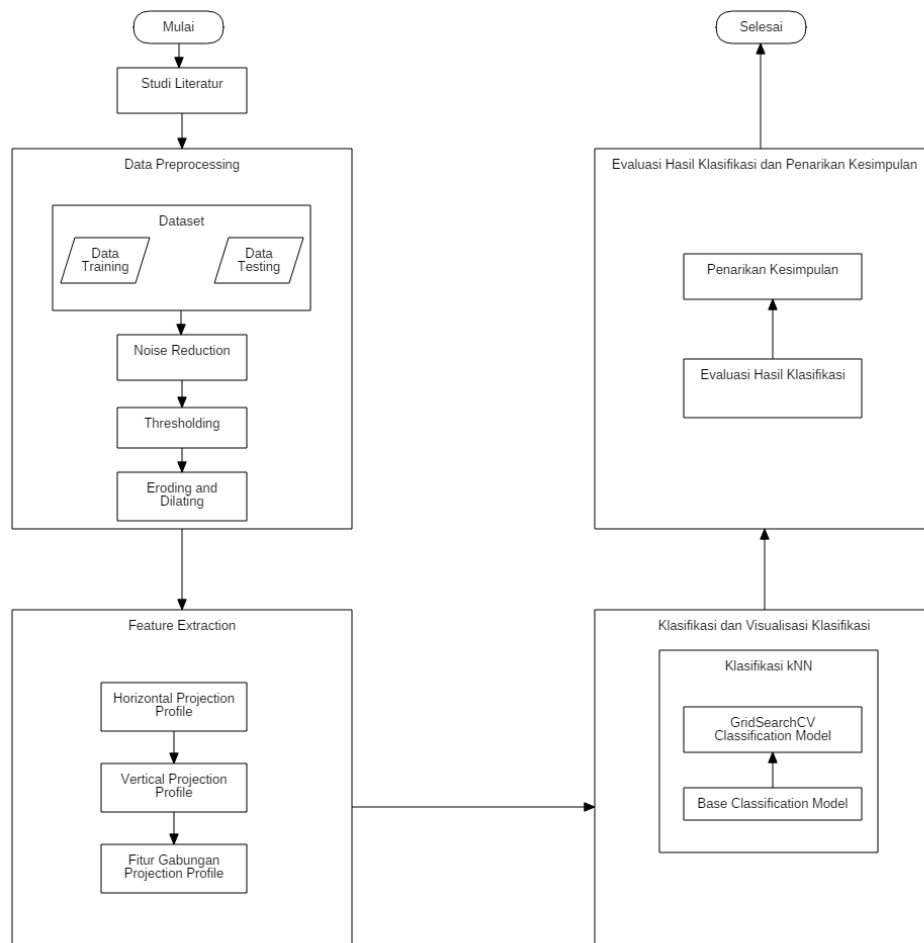
penyimpanan struktur data yang dihasilkan dalam sebuah proses analisis data.

j. *Library Imbalanced-learn 0.7.0*

Library Imbalanced-learn adalah salah satu library yang digunakan dalam *machine learning* untuk kasus distribusi data yang tidak seimbang.

3.3 Tahapan Penelitian

Penelitian ini memiliki tahapan – tahapan penelitian sebagaimana telah disajikan dalam Gambar 4.



Gambar 4. Alur Pengerjaan Penelitian Pengenalan Aksara Lampung.

Gambar 4 menyajikan alur pengerjaan penelitian untuk pengenalan pola karakter tulisan aksara Lampung. Tahapan penelitian secara garis besar terbagi menjadi *Data Preprocessing*, *Feature Extraction*, klasifikasi dan visualisasi hasil klasifikasi, dan evaluasi hasil klasifikasi serta penarikan kesimpulan.

3.3.1 Studi Literatur

Studi Literatur merupakan tahapan pengumpulan dasar materi untuk penelitian agar dapat relevan dengan penelitian yang dikerjakan serta memberikan gagasan yang kuat dalam penyusunan kerangka metodologi penelitian berdasarkan kajian literatur. Studi literatur juga membantu mengembangkan penelitian yang sesuai dengan latar belakang penelitian.

3.3.2 Data Preprocessing

Dataset yang digunakan berjumlah 32.140 data citra aksara Lampung berdasarkan penelitian *Lampung Handwritten Character Recognition* (Junaidi, 2016), dengan data citra sebanyak 24287 citra aksara Lampung digunakan sebagai data *Training* sekaligus data untuk melakukan validasi model yang dibentuk, dan 7853 data citra sebagai data *Testing*. Pembagian data sedemikian rupa dilakukan untuk menghindari ketidakseimbangan distribusi data citra, dimana karakter “ca” dan “wa” merupakan karakter yang jarang dipakai dalam bahasa Indonesia secara umum (Junaidi, 2016). Proses dalam tahap *data preprocessing* dijabarkan sebagai berikut :

3.3.2.1 Noise Reduction

Kualitas citra yang menjadi input data dalam klasifikasi dapat mempengaruhi tingkat klasifikasi yang dihasilkan. Citra yang memiliki kualitas buruk, atau memiliki banyak *noise* akan menghasilkan hasil klasifikasi yang buruk, sedangkan citra yang memiliki nilai *noise* yang minim dapat membantu hasil klasifikasi untuk mencapai nilai yang

maksimum. Citra yang akan dijadikan input untuk klasifikasi dalam penelitian ini direduksi tingkat *nois*nya dengan menggunakan modul *Gaussian Blur* dalam *Library OpenCV*.

3.3.2.2 *Thresholding*

Setelah citra telah melalui tahap reduksi *noise*, kemudian akan dilakukan *thresholding* untuk setiap citra pada *dataset*. Metode *thresholding* yang digunakan dalam penelitian ini adalah *Otsu's binarization*, menggunakan modul *THRESH_BINARY* dan *THRESH_OTSU* pada *Library OpenCV*.

3.3.2.3 *Dilating dan Eroding*

Citra yang sudah melalui tahap *thresholding* kemudian melalui tahap *eroding* dan *dilating*. *Eroding* bertujuan untuk mengikis citra original untuk mendapatkan citra yang sesuai dan bebas *noise*, sedangkan *dilating* bertujuan untuk menambah bagian dari citra original, sehingga apabila terdapat bagian dari citra yang tidak sempurna, maka piksel tersebut akan diisi. Pada penelitian ini, proses *eroding* dan *dilating* akan menggunakan bantuan modul *erode* dan *dilate* yang terdapat pada *Library OpenCV*.

3.3.2.4 *Feature Extraction*

Setelah citra melalui tahap *Data Preprocessing*, kemudian akan dilakukan ekstraksi fitur menggunakan metode *projection profile*. Ekstraksi fitur menggunakan *projection profile* dilakukan terhadap garis horizontal (*Horizontal Projection Profile*) dan garis vertikal (*Vertical Projection Profile*). Setiap data citra akan didapatkan 40 fitur, yang terdiri dari 20 fitur *Horizontal Projection*, dan 20 fitur *Vertical Projection*. Hasil ekstraksi fitur yang didapatkan dari hasil ekstraksi pada garis horizontal dan vertikal dapat

digunakan secara terpisah, atau menjadi gabungan keseluruhan nilai *projection profile* yang mencakup nilai *Horizontal Projection Profile* dan *Vertical Projection Profile*.

3.3.2.5 Klasifikasi *K-Nearest neighbors (kNN)*

Tahap klasifikasi menggunakan *K-Nearest neighbors (kNN)* akan dilakukan dengan tiga jenis fitur yang digunakan. Pertama, menggunakan fitur *Horizontal Projection*. Kedua, menggunakan fitur *Vertical Projection*. Ketiga, menggunakan gabungan fitur *Horizontal Projection* dan *Vertical Projection*. Detail jumlah fitur per citra yang akan digunakan untuk klasifikasi disajikan dalam Tabel 9.

Tabel 9. Jumlah Fitur per Citra Berdasarkan Jenis Fitur.

No	Jenis Fitur	Jumlah	Keterangan
1	<i>Horizontal Projection</i>	20	Label data <i>hr_px1</i> s.d <i>hr_px20</i>
2	<i>Vertical Projection</i>	20	Label data <i>vr_px1</i> s.d <i>vr_px20</i>
3	<i>Horizontal dan Vertical Projection</i>	40	Label data <i>hr_px1</i> s.d <i>hr_px20</i> dan <i>vr_px1</i> s.d <i>vr_px20</i>

Klasifikasi dilakukan menggunakan pembagian fitur seperti disajikan pada Tabel 9. Hal ini dilakukan untuk membandingkan jenis *projection profile* terbaik yang dapat digunakan dalam melakukan klasifikasi karakter aksara Lampung, dan melihat tingkat keberhasilan klasifikasi apabila menggunakan gabungan fitur *Horizontal Projection Profile* dan *Vertical Projection Profile*. Tahapan untuk melakukan klasifikasi menggunakan *K-Nearest neighbors* dijabarkan sebagaimana berikut :

a. *Train test split.*

Tahap *Train test split* dilakukan dengan cara membagi data *Training* menjadi dua bagian. Bagian pertama adalah, data yang berisi nilai *projection profile* yang digunakan untuk melatih algoritma *kNN* dan evaluasi model *kNN* yang dibentuk. Bagian kedua adalah data target yang merupakan label klasifikasi untuk setiap fitur *projection profile*. Kemudian untuk menghindari bias yang dapat disebabkan nilai fitur yang tidak merata, terlebih dahulu data akan dinormalisasi menggunakan modul *MinMaxScaler* dengan rentang nilai 0-1. Selanjutnya untuk label target kelas yang berupa nama aksara seperti “ka”, “ga”, “nga” akan diubah menjadi data label numerik dengan menggunakan modul *LabelEncoder*, hal ini dilakukan untuk meminimalisir kesalahan klasifikasi dengan menggunakan data label berupa string. Pembagian data *Training* untuk *Train test split* dilakukan berdasarkan konfigurasi yang disajikan pada Tabel 10.

Tabel 10. Konfigurasi *Train test split* Data *Training*.

No	Nama Konfigurasi	Nilai Konfigurasi
1	<i>test_size</i>	0.33
2	<i>random_state</i>	42
3	<i>stratify</i>	Label target

Tabel 10 menyajikan konfigurasi yang digunakan untuk membagi data *Training* menggunakan metode *Train test split*. Nilai *test_size* sebesar 0.33 menunjukkan jumlah data *Training* yang akan digunakan sebagai data untuk validasi model yang dibentuk adalah sebesar 33%.

Kemudian nilai *random_state* sebesar 42, menunjukkan bahwa jumlah data *Training* yang akan diacak sebelum dibagi menjadi data yang akan digunakan sebagai data *Training* dan data untuk validasi adalah sebesar 42 data. Nilai *stratify* yang digunakan adalah label target, hal ini dilakukan untuk menghindari ketidakseimbangan sebaran data yang disebabkan oleh jumlah data yang tidak merata untuk setiap target kelas yang terdapat pada data *training* dan data *testing*.

b. *Base Classification Model*.

Berdasarkan hasil pembagian data *Training* pada tahap *Train test split*, kemudian data untuk setiap jenis fitur, akan digunakan untuk membentuk model klasifikasi *kNN* dengan konfigurasi dasar sesuai dengan nilai parameter default untuk modul *KNeighborsClassifier* yang ada dalam *library Scikit-learn*. Konfigurasi *Base Classification Model* disajikan dalam Tabel 11.

Tabel 11. Konfigurasi *Base Classification Model*.

No	Nama Konfigurasi	Nilai Konfigurasi
1	<i>algorithm</i>	“auto”
2	<i>metric</i>	“ <i>minkowski</i> ”
3	<i>n_neighbors</i>	5
4	<i>P</i>	2
5	<i>weights</i>	“ <i>Uniform</i> ”

Berdasarkan Tabel 11, parameter “algorithm” dengan nilai “auto” menunjukkan bahwa *library Scikit-learn* akan menentukan algoritma untuk menghitung jarak yang digunakan untuk melakukan klasifikasi *kNN* dengan otomatis. Algoritma yang dapat digunakan akan brute force, dan K-D Tree. Parameter *metric* dengan nilai

“minkowski” dan parameter $p = 2$, menunjukkan bahwa penghitungan nilai jarak antar data menggunakan euclidean distance. Parameter `n_neighbors` dengan nilai 5 berarti jumlah *nearest neighbors* adalah 5 data terdekat. Parameter `weights` dengan nilai “Uniform” digunakan untuk menentukan data dalam *nearest neighbors* dimana setiap objek data mendapatkan peluang setara untuk dikelompokkan menjadi *nearest neighbors* berdasarkan jarak data.

c. *GridSearchCV Classification Model.*

Menggunakan modul *GridSearchCV* yang terdapat dalam *library Scikit-learn*, peneliti akan melakukan optimasi hasil klasifikasi dengan cara mencari kombinasi nilai parameter dalam rentang nilai tertentu, yang memiliki hasil klasifikasi paling baik.

Untuk mencari nilai parameter yang menghasilkan klasifikasi terbaik, *GridSearchCV* bekerja dengan konsep *exhaustive searching*, di mana untuk setiap rentang nilai parameter yang dideklarasikan, nilai parameter akan digunakan sebagai pembentuk model *kNN* yang digunakan untuk klasifikasi, kemudian akan dipilih kombinasi nilai parameter yang menghasilkan klasifikasi paling baik berdasarkan metrik evaluasi model.

Dalam melakukan hal ini, modul *GridSearchCV* membutuhkan dua komponen utama. Pertama, deklarasi rentang nilai parameter sebagai nilai dasar pembentukan model *kNN*. Kedua, objek model *kNN* yang akan dicari hasil klasifikasi terbaik berdasarkan metrik evaluasi model. Rentang nilai parameter yang akan digunakan

untuk menemukan model *kNN* dengan hasil klasifikasi terbaik disajikan pada Tabel 12.

Tabel 12. Rentang Nilai Parameter Model *kNN*.

No	Nama Parameter	Nilai Parameter
1	<i>n_neighbors</i>	List(range(5,3000,65))
2	<i>weights</i>	["Uniform", "Distance"]

Parameter *n_neighbors* berfungsi sebagai nilai "*k*" yang merupakan jumlah objek tetangga terdekat yang akan digunakan untuk penentuan kelas, sehingga nilai parameter "*k*" paling awal adalah 5, dengan interval sebesar 65, dengan nilai maksimum yang dapat dicapai adalah 3000. Nilai interval adalah nilai jarak yang akan terus ditambahkan hingga nilai "*k*" mencapai nilai maksimal yakni 3000.

Parameter *weights* dengan nilai ["Uniform", "Distance"] berfungsi untuk menentukan apakah pembobotan untuk setiap nilai jarak antar pasangan data. Nilai "*Uniform*" berarti setiap nilai jarak pasangan data memiliki bobot yang sama untuk dimasukkan ke dalam sebuah kelas, sedangkan nilai "*weights*" akan memberikan hasil pembobotan yang berbeda untuk setiap nilai jarak antar pasangan data, di mana data dengan nilai *Distance* yang besar akan lebih kecil pengaruh nya terhadap klasifikasi yang dilakukan.

Nilai parameter sebagaimana disajikan pada Tabel 11, kemudian akan dipasangkan dengan *base model* dengan konfigurasi yang disajikan pada Tabel 13.

Tabel 13. Konfigurasi *Base model kNN* untuk *GridSearchCV*.

No	Nama Konfigurasi	Nilai Konfigurasi
1	<i>algorithm</i>	“kd_tree”
2	<i>leaf_size</i>	40
3	<i>p</i>	2
4	<i>metric</i>	“minkowski”

Konfigurasi *algorithm* dengan nilai “kd_tree” menunjukkan algoritma yang akan digunakan untuk mendapatkan nilai *kNN*. Algoritma *K-D Tree* digunakan karena nilai efisiensi waktu yang lebih baik apabila dibandingkan dengan algoritma *brute force* yang menghitung nilai jarak untuk setiap pasangan data, dengan nilai *leaf_size* sebesar 40, di mana nilai *leaf_size* akan berpengaruh untuk kecepatan pencarian jarak menggunakan algoritma *K-D Tree* (Bentley, 1975). Konfigurasi *metric* menggunakan nilai “minkowski” dan $p = 2$, berarti pengukuran jarak antar pasangan data dilakukan menggunakan *Euclidean Distance*.

Untuk memastikan bahwa hasil klasifikasi yang didapatkan efektif untuk setiap *target class*, nilai parameter *GridSearchCV* disesuaikan dengan nilai yang disajikan pada Tabel 14.

Tabel 14. Parameter *GridSearchCV*.

No	Nama Parameter	Nilai Parameter
1	<i>n_jobs</i>	-1
2	<i>refit</i>	True
3	<i>cv</i>	10

Nilai parameter $n_jobs = -1$ digunakan untuk mengaktifkan semua *core processor* untuk melakukan penghitungan. Nilai $refit = True$ menandakan untuk setiap nilai terbaik yang dihasilkan sebuah kombinasi parameter, kombinasi parameter tersebut akan digunakan untuk pembentukan model klasifikasi yang baru. Parameter cv dengan nilai 10, berfungsi untuk membentuk model dengan menggunakan *k-fold cross validation*, dengan nilai $k = 10$.

3.3.2.6 Evaluasi dan Penarikan Kesimpulan

Tahap evaluasi dilakukan untuk hasil klasifikasi yang dihasilkan oleh *Base Classification Model* dan *GridSearchCV Classification Model*. Evaluasi hasil klasifikasi dilakukan untuk klasifikasi yang menggunakan fitur *Horizontal Projection Profile*, *Vertical Projection Profile*, dan gabungan fitur horizontal dan *Vertical Projection Profile*.

Untuk setiap jenis fitur yang digunakan untuk melakukan klasifikasi, akan dilakukan klasifikasi menggunakan *Base Classification Model* dengan konfigurasi yang disajikan pada Tabel 11, dan *GridSearchCV* dengan konfigurasi pada Tabel 12,13, dan 14.

Evaluasi model klasifikasi akan dilakukan menggunakan metrik *Accuracy*, *Precision*, *Sensitivity*, *Specificity*, *F-Measure*, dan *MCC*.

V. SIMPULAN DAN SARAN

5.1 Simpulan

Berdasarkan hasil dan pembahasan mengenai klasifikasi karakter citra aksara Lampung pada bab empat menggunakan dua jenis metode pembentukan model klasifikasi *kNN*, dapat disimpulkan hasil penelitian sebagai berikut :

1. Data citra aksara Lampung yang berasal dari penelitian *Lampung – A new Handwritten Character Benchmark : Database, Labeling, and Recognition*, memiliki distribusi data citra per kelas yang tidak seimbang, sehingga proses *Cross-Validation* dan stratifikasi perlu dilakukan untuk menghasilkan model klasifikasi *kNN* yang memiliki performa baik untuk setiap kelas target.
2. Hasil klasifikasi terbaik didapatkan menggunakan *GridSearchCV Classification Model* dengan fitur gabungan horizontal dan *Vertical Projection Profile*.
3. Model klasifikasi dengan fitur *Horizontal Projection Profile* mendapatkan nilai performa klasifikasi paling buruk pada semua metode pembentukan model klasifikasi. Namun, fitur *Horizontal Projection Profile* memberikan peningkatan sebesar 10 – 13% dalam performa model klasifikasi jika dibandingkan dengan model klasifikasi yang dibentuk hanya dengan menggunakan fitur *Vertical Projection Profile*.
4. Metrik evaluasi MCC menghasilkan nilai yang objektif untuk klasifikasi yang dilakukan menggunakan data dengan distribusi data per kelas yang tidak merata.

5.2 Saran

Berikut dijabarkan saran terkait dengan hasil penelitian ini maupun penelitian lanjutan yang menggunakan data citra yang sama :

1. Penelitian ini dapat dikembangkan dengan metode ekstraksi fitur lain, yang dapat memberikan hasil ekstraksi fitur terhadap citra aksara Lampung yang lebih baik.
2. Penelitian ini dapat ditingkatkan lebih lanjut dengan cara melakukan kombinasi metode klasifikasi *kNN* dengan *Bagging Predictors* (Breiman, 1996), untuk mendapatkan nilai gabungan terbaik dari kombinasi nilai yang dihasilkan oleh model klasifikasi *kNN*.
3. Penelitian ini dapat menggunakan metode klasifikasi lainnya seperti *Naïve Bayes*, *Decision Tree*, *Gradient Boosting*, dan *Random Forest*.
4. Untuk mengatasi ketidakseimbangan distribusi data citra per kelas, dapat menggunakan metode sampling lain seperti *SMOTE Oversampling*, dan *Random Undersampling*.

DAFTAR PUSTAKA

- Andana, A., Widyati, R., and Irzal, M. (2018). Pengenalan Citra Tulisan Tangan Dengan Metode Backpropagation. *Jurnal Matematika Terapan*, 2(1), 36–44.
- Bekkar, M., Djemaa, H. K., and Alitouche, T. A. (2013). Evaluation Measures for Models Assessment over Imbalanced Data Sets. *Journal of Information Engineering and Applications*, 3(10), 27–38.
- Bentley, J. L. (1975). Multidimensional Binary Search Trees Used for Associative Searching. *Communications of the ACM*, 19.
- Cattin, P. (2016). Image Restoration Introduction to Signal and Image Processing. *MIAC, University of Base*, 9(19), 1–70.
- Chacko, A. M. M. O., and Dhanya, P. M. (2015). A Comparative Study of Different Feature Extraction Techniques for Offline Malayalam Character Recognition. In *Smart Innovation, Systems and Technologies* (Vol. 32, pp. 9–18).
- Gonzales, R. C., and Woods, R. E. (2006). Digital Image Processing. In *SANITAS : Jurnal Teknologi dan Seni Kesehatan* (Vol. 3). Prentice-Hall, Inc.
- Gou, J., Du, L., Zhang, Y., and Xiong, T. (2011). A New Distance-weighted k -nearest Neighbor Classifier A New Distance-weighted k -nearest Neighbor Classifier. *Journal of Information and Computational Science*, November.
- Han, J., Kamber, M., and Pei, J. (2012). Data Mining: Concepts and Techniques. In *Morgan Kaufmann Series in Data Management Systems*. Elsevier.

- Javed, M., Nagabhushan, P., and Chaudhuri, B. B. (2013). Extraction of projection profile, run-histogram and entropy features straight from run-length compressed text-documents. *Proceedings - 2nd IAPR Asian Conference on Pattern Recognition, ACPR 2013*, 813–817.
- Junaidi, A. (2016). *Lampung Handwritten Character Recognition*.
- Junaidi, A., Vajda, S., and Fink, G. A. (2011). Lampung - A new handwritten character benchmark: Database, labeling and recognition. *ACM International Conference Proceeding Series*.
- Jurman, G., Riccadonna, S., and Furlanello, C. (2012). A comparison of MCC and CEN error measures in multi-class prediction. *PLoS ONE*, 7(8), 1–8.
- Kuhn, M., and Johnson, K. (2013). Measuring Performance in Classification Models. In *Applied Predictive Modeling* (pp. 247–273). Springer New York.
- Likforman-Sulem, L., Zahour, A., and Taconet, B. (2007). Text line segmentation of historical documents: A survey. *International Journal on Document Analysis and Recognition*, 9(2–4), 123–138.
- Mahto, M. K., Bhatia, K., and Sharma, R. K. (2015). Combined horizontal and vertical projection feature extraction technique for Gurmukhi handwritten character recognition. *Conference Proceeding - 2015 International Conference on Advances in Computer Engineering and Applications, ICACEA 2015*, 59–65.
- Nasri, N., Samet, M., Wei, A., and Technology, I. (2012). Analysis Study of Time Synchronization Protocols in WSN. *Networking and Internet Architecture*.
- Otsu, N. (1979). A Threshold Selection Method from Gray-Level Histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1), 62–66.
- Puspasari, B. D. (2013). Aplikasi Pengenalan Huruf Alfabet Bagi Anak Usia Dini Menggunakan Metode Pengolahan Citra Berbasis Data Suara. *Jurnal ELTEK*, 11(01), 1693–4024.

- Putra, D. (2010). *Pengolahan Citra Digital* (Westriningsih (ed.); 1st ed.). Penerbit Andi.
- Refaeilzadeh, P., Tang, L., and Liu, H. (2009). Cross-Validation. In *Encyclopedia of Database Systems* (pp. 532–538). Springer US.
- Ritter, G. X., and Wilson, J. N. (2000). Handbook of computer vision algorithms in image algebra: Second edition. In *Handbook of Computer Vision Algorithms in Image Algebra: Second Edition*.
- Szeliski, R. (2021). *Computer Vision : Algorithms and Applications 2nd Edition*. Springer, 1196.
- Wahyono, W., and Nugroho, A. (2018). Penerapan Algoritma C4.5 Untuk Prediksi Tingkat Kompetensi Karyawan Pt Multistrada Arah Sarana. *Joutica*, 3(1), 145.
- Wu, X., and Kumar, V. (2009). The Top Ten Algorithms in Data Mining, Chapman & Hall/CRC Data Mining and Knowledge Discovery. In *Chapman & Hall/CRC Data Mining and Knowledge Discovery Series* (Vol. 53, Issue 9).