PENDEKATAN *OVERSAMPLING* SMOTE UNTUK *IMBALANCED DATASET* AKSARA LAMPUNG DAN KLASIFIKASI MENGGUNAKAN SVM

(Skripsi)

Oleh:

EDO PRIYONO MUALIM
1617051113

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG

2022

PENDEKATAN *OVERSAMPLING* SMOTE UNTUK *IMBALANCED DATASET* AKSARA LAMPUNG DAN KLASIFIKASI MENGGUNAKAN SVM

Oleh:

EDO PRIYONO MUALIM

(Skripsi)

Sebagai Salah Satu Syarat untuk Mencapai
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

ABSTRAK

PENDEKATAN *OVERSAMPLING* SMOTE UNTUK *IMBALANCED DATASET* AKSARA LAMPUNG DAN KLASIFIKASI MENGGUNAKAN SVM

oleh

EDO PRIYONO MUALIM

Klasifikasi gambar adalah pembelajaran *supervised* dengan mendifinisikan kelas gambar, kemudian melatih sebuah model menggunakan data training, validasi, dan testing untuk memprediksi label gambar tulisan tangan Aksara Lampung. Tujuan penelitian ini adalah untuk mengembangkan model *Support Vector Machine* (SVM) yang mampu mengklasifikasi kelas gambar secara akurat. Teknik klasifikasi SVM dipilih karena dapat memproses data dalam bentuk tulisan dan gambar, memiliki implementasi *kernel* sehingga memungkinkan data yang tidak dapat dibagi secara *linear* untuk dikonversikan ke data dimensi lebih tinggi. Hasil penelitian dapat digunakan sebagai panduan untuk penelitian selanjutnya dengan fokus topik yang lebih mendalam, terutama untuk subjek mengenai klasifikasi gambar. Penelitian ini menggunakan karakter tradisional dari Kota Lampung di Indonesia. Berbeda dengan alfabet latin umumnya, Bahasa Lampung memiliki 20 huruf induk, 12 anak huruf, dan tanda baca khusus. Dataset Lampung yang digunakan terdiri dari 18 huruf induk dan tersebar di 32140 gambar *grayscale* tanpa anak huruf dan tanda baca. Akan tetapi data tidak memiliki pesebaran data yang seimbang. Penelitian ini menggunakan pendekatan oversampling *Synthetic Minority Oversampling Technique* (SMOTE) sebagai solusi masalah data tidak seimbang. Fitur gambar diekstraksi ke dalam *NumPy array,* dimanipulasi dan dianalisis menggunakan *library* dari bahasa pemrograman Python. Penelitian ini menghasilkan model SVM dengan akurasi terendah yaitu 92.03 dan akurasi tertinggi sebesar 95.90.

Kata Kunci : *Dataset* Lampung, Klasifikasi Gambar*, SMOTE* , Support Vector Machine

ABSTRACT

OVERSAMPLING USING SMOTE ON IMBALANCED DATASET OF LAMPUNG HANDWRITTEN IMAGES FOR SVM CLASSIFICATION

by

EDO PRIYONO MUALIM

Image classification is supervised learning by defining a set of target classes, then training a model using one partitioned dataset of training, validation, and testing which contains labeled data of Lampung handwritten images. The goal is to develop a model that could accurately classify these data with the right label using Support Vector Machine (SVM). SVM classification technique was chosen because it could process data in a form of text and images, and with the implementation of kernel, data does not have to be linearly separable since input data can be converted into high dimensional data. Information derived from this endeavor to develop an optimized classification model can also be used as references for future studies with a more specialized focus point, especially on similar subjects such as image classification. This research uses traditional characters from a city in Indonesia called Lampung. Unlike the conventional 26 Latin alphabets, Lampung has 20 characters, 12 diacritics, and unique punctuations. Lampung dataset consists of 18 Lampung characters, in the form of 32140 grayscale images without diacritics and punctuations. These data however are not evenly distributed among the 18 characters. This research makes use of oversampling approach using Synthetic Minority Oversampling Technique (SMOTE) as a solution for class imbalance found in this dataset. Lampung images features were extracted into NumPy array and are manipulated and analyzed using various libraries from Python programming language. This research yields an SVM model with the lowest accuracy of 92.03 and the highest accuracy of 95.90.

Keywords : Image Classification, Lampung Dataset, SMOTE, Support Vector Machine

Judul Skripsi	:	**PENDEKATAN OVERSAMPLING SMOTE UNTUK IMBALANCED DATASET AKSARA LAMPUNG DAN KLASIFIKASI MENGGUNAKAN SVM**
Nama Mahasiswa	:	*Edo Priyono Mualim*
Nomor Pokok Mahasiswa	:	**1617051113**
Program Studi	:	**S1 Ilmu Komputer**
Fakultas	:	**Matematika dan Ilmu Pengetahuan Alam**

MENYETUJUI

1. Komisi Pembimbing

Dr. rer. nat. Akmal Junaidi, M.Sc.	**Dewi Asiah Shofiana, S.Komp., M.Kom.**
NIP 19710129 199702 1 001	NIP 19950929 202012 2 030

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 : **Dewi Asiah Shofiana, S.Komp., M.Kom.**

   Anggota : **Favorisen R. Lumbanraja S.Kom., M.Si., Ph.D.**

2. Dekan Fakultas Matematika Ilmu Pengetahuan Alam

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

Tanggal Lulus Ujian Skripsi: **01 April 2022**

PERNYATAAN SKRIPSI MAHASISWA

Saya yang bertanda tangan di bawah ini:

Nama	: Edo Priyono Mualim
Nomor Pokok Mahasiswa	: 1617051113
Program Studi	: S1 Ilmu Komputer
Fakultas	: Matematika dan Ilmu Pengetahuan Alam
Perguruan Tinggi	: Universitas Lampung

Dengan ini menyatakan bahwa skripsi saya yang berjudul:

"Pendekatan Oversampling SMOTE untuk Imbalanced Dataset Aksara Lampung dan Klasifikasi menggunakan SVM" merupakan hasil pekerjaan saya sendiri dan apabila di kemudian hari terbukti bahwa skripsi ini merupakan hasil salinan atau dibuat oleh orang lain, maka saya bersedia menerima sanksi sesuai dengan ketentuan akademik yang berlaku.

Bandar Lampung, Juni 2022

Edo Priyono Mualim

NPM 1617051113

RIWAYAT HIDUP

Penulis memiliki nama lengkap Edo Priyono Mualim, dilahirkan di Bandar Lampung pada tanggal 1 Mei 1998. Penulis merupakan anak ketiga dari empat bersaudara pasangan Bapak Lim Hendra dan Ibu Thung Charmeylina Herawaty. Penulis menempuh pendidikan pertama di SD Xaverius 1, Teluk Betung, Bandar Lampung dan diselesaikan pada tahun 2009. Pendidikan Sekolah Menengah Pertama ditempuh di SMP Xaverius 1, Teluk Betung, Bandar Lampung dan diselesaikan pada tahun 2012, kemudian dilanjutkan pendidikan Sekolah Menengah Atas di Bishan Park School, Singapura dan diselesaikan pada tahun 2016. Pada tahun yang sama, penulis diterima sebagai mahasiswa Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung pada tahun 2016 melalui jalur Seleksi Bersama Masuk Perguruan Tinggi Negeri (SBMPTN). Selama menjadi mahasiswa penulis pernah dipercaya menjadi asisten dosen mata kuliah Pemrograman Terstruktur pada semester genap tahun ajaran 2017/2018 dan mata kuliah Bahasa Inggris pada semester ganjil tahun ajaran 2018/2019. Pada bulan Januari-Februari 2019, penulis melaksanakan Kerja Praktik (KP) di Bank BTN KC Jalan Wolter Monginsidi, No. 80 – 88, Teluk Betung Selatan, Bandar Lampung. Pada bulan Juli-Agustus 2019, penulis melaksanakan kegiatan Kuliah Kerja Nyata (KKN) Kebangsaan Universitas Lampung di Desa Belimbing Sari, Kecamatan Jabung, Kabupaten Lampung Timur.

MOTTO

“Anything worth doing is worth doing badly, the first time.”

(G. K. Chesterton)

“Be kind to yourself by forgiving yourself.”

(Olivia Remes)

“Doing something with someone else in mind can carry you

through the toughest time”

(Olivia Remes)

PERSEMBAHAN

Puji dan syukur saya ucapkan kepada Tuhan Yang Maha Esa atas segala Rahmat dan Karunia-Nya sehingga saya dapat menyelesaikan skripsi ini

Teruntuk Ayah dan Ibu yang sangat saya cintai kupersembahkan skripsi ini

Terima kasih atas semua doa, perhatian, pengorbanan, perjuangan, kesabaran, serta kasih saying dan dukungan yang telah kalian berikan untukku.

Teruntuk sahabat dan teman-teman seperjuanganku, terima kasih telah berjuang bersama dan memberikan banyak cerita,

Keluarga Ilmu Komputer 2016

Serta Almamater tercinta yaitu UNIVERSITAS LAMPUNG

SANWACANA

Puji syukur kepada Tuhan yang Maha Esa karena telah melimpahkan rahmat, sehingga penulis dapat menyelesaikan skripsi dengan judul "Pendekatan Oversampling SMOTE untuk Imbalanced Dataset Aksara Lampung dan Klasifikasi Menggunakan SVM".

Skripsi ini dapat penulis selesaikan dengan bantuan dari berbagai pihak yang terkait. Oleh karena itu, pada kesempatan ini penulis menyampaikan rasa terimakasih kepada:

1. Orangtua tercinta, Bapak Lim Hendra, Ibu Thung Charmeylina Herawaty yang telah memberi dukungan moril, nasihat, doa dan kasih sayang yang tak pernah putus diberikan selama ini.
2. Bapak Dr. rer. nat. Akmal Junaidi, M.Sc. selaku dosen pembimbing utama yang telah membimbing dan memberikan pembelajaran, ilmu, serta bimbingan penulisan skripsi ini dapat terselesaikan.
3. Ibu Dewi Asiah Shofiana, S.Komp., M.Kom.selaku dosen pembimbing pembantu yang telah memberikan pengarahan dan saean dalam penyusunan skripsi ini.
4. Bapak Favorisen R. Lumbanraja S.Kom., M.Si., Ph.D selaku dosen pembahas yang telah memberikan masukan dan saran dalam perbaikan pada skripsi ini.
5. Bapak Dr. Ir. Kurnia Muludi, M.S.Sc. selaku Dosen Pembimbing akademik Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung.
6. Bapak Didik Kurniawan, S.Si., M.T. selaku Ketua Jurusan Ilmu Komputer Universitas Lampung
7. Bapak Dr. rer. nat. Akmal Junaidi, M.Sc. selaku Sekretaris Jurusan Ilmu Komputer Universitas Lampung.
8. Bapak Dr. Eng. Suripto Dwi Yuwono, M.T., selaku Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung.
9. Bapak dan Ibu Dosen Jurusan Ilmu Komputer yang telah memberikan ilmu dan pengalaman hidup selama penulis menjadi mahasiswa.

10. Seluruh Staff dan Karyawan di Jurusan Ilmu Komputer FMIPA Universitas Lampung yang turut membantu selama penulis menjadi mahasiswa
11. Kakak dan adik yang penulis sayangi atas doa, dukungan, motivasi, semangat dan kasih sayang yang diberikan selama ini.
12. Teman-teman Ilmu Komputer angkatan 2016 yang secara langsung maupun tidak langsung membantu penulis dalam skripsi ini.
13. Almamater Tercinta, Universitas Lampung yang telah memberikan penulis kesempatan untuk menempuh pendidikan perkuliahan S1.

Bandar Lampung

Penulis,

Edo Priyono Mualim

DAFTAR ISI

DAFTAR GAMBAR ..... **iii**

DAFTAR TABEL ..... **v**

BAB I PENDAHULUAN **1**
A. Latar Belakang 1
B. Rumusan Masalah 3
C. Tujuan Penelitian 3
D. Batasan Masalah 3
E. Manfaat Penelitian 4

BAB II TINJAUAN PUSTAKA **5**
A. Penelitian Terdahulu 5
B. *Machine Learning* 7
C. Python 10
D. *Oversampling* 11
E. SMOTE 12
1. *Borderline* SMOTE 13
2. *K-Means* SMOTE 13
3. SVM SMOTE 13
F. *Image Classification* 14
G. SVM 14
H. Pengujian Model 16

BAB III METODE PENELITIAN **18**
A. Tempat dan Waktu Penelitian 18
B. Alat dan Pendukung Penelitian 18
C. Tahapan Penelitian 19
1. Pengumpulan dan Analisis Data 20
2. *Preprocessing* 24
3. Ekstraksi Fitur Gambar 25
4. Membagi Data Menjadi *Training*, *Validation*, dan *Testing* 25
5. *Oversampling* SMOTE 26
6. Merancang Model Klasifikasi SVM 26
7. Evaluasi Performa Model 27
D. Penulisan Laporan 27
E. Jadwal Penelitian 27

BAB IV HASIL DAN PEMBAHASAN 28
A. Hasil 28
B. Pembahasan 30
1. *Preprocessing* 30
2. Ekstraksi Fitur Gambar 33
3. Membagi Data Menjadi *Training*, *Validation*, dan *Testing* 36
4. *Oversampling* SMOTE 40
5. *Hypertuning Borderline* SMOTE 44
6. *Hypertuning* SVM 45
7. Pengujian Model Klasifikasi Dengan dan Tanpa SMOTE 47

BAB V SIMPULAN DAN SARAN 50
A. Simpulan 50
B. Saran 51

DAFTAR PUSTAKA 52

DAFTAR GAMBAR

Gambar Halaman

1. Alur Klasifikasi Gambar 6
2. *Linear Probabilistic Modeling* 8
3. Ilustrasi Kernel (Xie, 2010) 9
4. Perbandingan *Syntax* Java, C++, dan Python. 10
5. Proses *Oversampling* dan *Undersampling* 12
6. *Hyperplane* 2 Dimensi 15
7. *Hyperplane* 3 Dimensi 15
8. *Flow Chart* Tahapan Penelitian 19
9. Gambar dengan Ukuran Terbesar 20
10. Gambar dengan Ukuran Terkecil 21
11. *Subfolder Dataset* Aksara Lampung 22
12. Data Gambar dengan Label Sumber Gambar 22
13. Anotasi Data Gambar 23
14. Gambar dari *Dataset* Aksara Lampung 23
15. Melabelkan Data Gambar 24
16. Citra Aksara Ba dengan Ukuran 73x47 *Pixel* 25
17. Citra Aksara Ba dengan Ukuran 48x48 *Pixel* 25
18. *Oversampling Training Dataset* Menggunakan SMOTE 26
19. *Source Code* Parameter Model Awal 29
20. *Source Code* Parameter Model Akhir 29
21. *File* Berisi Informasi Label Data Gambar Aksara Lampung 31
22. *Drop Column* Untuk Menghapus Data yang Tidak Relevan 31
23. *Shutil Move* Untuk Memindahkan Gambar ke *Subfolder* Kelas 32
24. Mengubah Ukuran Semua Gambar Menjadi 48x48 *Pixel* 33
25. Ekstraksi Fitur Gambar Aksara Lampung 33
26. *Import Pickle* 18 Kelas Aksara Lampung 34
27. Normalisasi dan Mengubah Jenis Data Menjadi *Numpy Array* 35
28. Visualisasi *Dataset* Mentah Menggunakan *Matlpotlib* 35
29. Perbandingan Performa Rasio Data dengan Nilai 60:20:20 36
30. Grafik Perbandingan Akurasi SVM Menggunakan Berbagai Rasio 38
31. Citra Grafik Distribusi Data Menggunakan *Matplotlib* 39
32. Citra Menggambarkan Performa k-NN SMOTE 40
33. Citra Menggambarkan Performa *Borderline* SMOTE 40
34. Citra Menggambarkan Performa *K-Means* SMOTE 41
35. Citra Menggambarkan Performa SVM SMOTE 42
36. Visualisasi Data Mentah 43
37. Visualisasi Data SMOTE 43

38. Akurasi dan F1 untuk SMOTE k=1 44
39. Model SVM dengan *Kernel Polynomial* 45
40. SVM *Kernel* RBF dengan Nilai C=1 46
41. Akurasi, F1 Klasifikasi SVM tanpa SMOTE 48
42. Misklasifikasi Gambar 49

DAFTAR TABEL

Tabel Halaman

1. Kelebihan *Oversampling* dan *Undersampling* 11
2. *Confusion Matrix* 17
3. Distribusi Data Aksara Lampung dan Persentase 21
4. Jadwal Penelitian 27
5. Parameter Model Awal 28
6. Parameter Model Klasifikasi SVM yang Sudah Dioptimalkan 28
7. Tabel Perbandingan Akurasi SVM Menggunakan Berbagai Rasio 38
8. Tabel Perbandingan Akurasi dan Waktu Hitung Varian SMOTE 42
9. Tabel Perbandingan Akurasi dan F1 dengan Nilai k Berbeda 45
10. Tabel Perbandingan *Kernel* SVM 46
11. Tabel Perbandingan Performa *Kernel* RBF Berdasarkan Nilai C 47
12. Tabel Perbandingan Akurasi dan F1 Menggunakan dan Tanpa SMOTE 48

BAB I PENDAHULUAN

A. Latar Belakang

Pembelajaran mesin (*machine learning*) telah menjadi bagian penting di masyarakat, dan berbagai jenis algoritme sudah diimplementasikan dan digunakan, seperti *filter spam* otomatis yang digunakan oleh Gmail, video rekomendasi Youtube di akhir video maupun *sidebar*, dan jenis iklan yang ditampilkan saat menelusuri sosial media. Ini hanya beberapa contoh dari banyak pengaplikasian pembelajaran mesin yang sering dijumpai sehari-hari.

Algoritme adalah rangkaian instruksi yang akan dilakukan komputer untuk mengubah input menjadi *output*. Sejak komputer pertama kali ditemukan, ahli komputer telah mampu mengembangkan berbagai macam algoritme dengan kegunaan yang berbeda-beda. Di pembelajaran mesin, ilmu mengenai cara membuat algoritme dipelajari dengan menggunakan parameter sebagai kontrol untuk menentukan bagaimana cara algoritme itu bekerja. Pembelajaran mesin menggunakan teori statistika dalam membangun model matematika karena pada dasarnya tujuan dari pembelajaran mesin adalah untuk melakukan prediksi berdasarkan sampel data yang dimiliki. Data perlu dilatih, disimpan, dan diproses secara efisien dan dioptimalisasi bila diperlukan agar dapat merepresentasikan data yang dimiliki secara baik, kemudian menggunakan model matematika untuk menguji efisiensi pembelajaran mesin menggunakan algoritme tersebut berdasarkan kompleksitas waktu dan ruang, kemudian diukur akurasinya (Alpaydin, 2020).

Algoritme diperlukan untuk mengidentifikasi aksara Lampung dan mengklasifikasi aksara tersebut dengan tepat. Pada umumnya, pembacaan tulisan tangan masih

dilakukan secara manual oleh manusia. Namun dengan menggunakan algoritme, pembacaan dapat dilakukan secara otomatis dengan mengekstrak fitur gambar aksara Lampung berdasarkan distribusi warna, kemudian data tersebut diproses menggunakan algoritme yang sesuai untuk mendapatkan akurasi yang tinggi.

Kasus di dunia nyata yang menjadi masalah adalah banyak data yang ditemui memiliki distribusi tidak seimbang di setiap kelasnya. Jenis data ini dikenal sebagai *imbalanced data*. Jenis data seperti ini menyebabkan degradasi performa di banyak algoritme yang sering dipakai. Oleh karena itu tujuan dilakukan pendekatan *oversampling* adalah untuk meningkatkan akurasi pengklasifikasian kelas minoritas. Pada kasus pengklasifikasian biner (*binary classification*), data hanya dibagi menjadi dua kelas, proses *oversampling* cukup mudah, yaitu dengan meningkatkan jumlah data minoritas agar seimbang dengan data mayoritas. Akan tetapi, untuk klasifikasi multi kelas (*multiclass classification*), prosesnya akan menjadi lebih kompleks karena ada beberapa situasi yang harus diperhatikan seperti hubungan antar kelas menjadi tidak jelas dan batas ruang (*boundary*) antar kelas mungkin saling tumpang tindih. Oleh karena itu, metode *oversampling* dapat menyebabkan penurunan akurasi apabila tidak digunakan secara tepat. Strategi umum yang digunakan adalah *one-versus-one* (OVO) dan *one-versus-all* (OVA) (Xuebing et al, 2017).

Salah satu pendekatan yang banyak digunakan adalah *Synthetic Minority Oversampling Technique* (SMOTE). Pendekatan ini tidak mereplikasi data, melainkan membuat data baru dalam menyeimbangkan distribusi data. Untuk setiap data minoritas, sejumlah data tetangga yang telah ditentukan di parameter akan dihitung, kemudian sebagian data di kelas minoritas dipilih secara acak sebagai titik sumber proses pembuatan data baru. Setelah itu observasi dilakukan secara artifisial di garis yang menghubungkan data minoritas dengan tetangga terdekatnya (Mukherjee & Khushi, 2021).

Dalam mengklasifikasi aksara Lampung diperlukan algoritme yang mampu memproses data fitur gambar dengan baik. Paradigma SVM mampu mengklasifikasi data dengan dimensi tinggi. Kemudian model SVM dibangun

menggunakan *kernel* yang sesuai seperti *linear*, *polynomial*, *Gaussian Radial Basis Function* (RBF), dan *Sigmoid*. Data kemudian dibagi menggunakan *hyperplane* (Kecman, 2005). Dengan menggunakan pendekatan *oversampling* SMOTE dan mengklasifikasikan aksara Lampung menggunakan SVM diharapkan proses pembacaan aksara Lampung dapat dilakukan secara otomatis untuk mempermudah pembacaan baik bagi para ahli maupun orang yang ingin membaca sastra Lampung.

B. Rumusan Masalah

Adapun masalah yang ditemui pada klasifikasi aksara Lampung adalah bagaimana merancang model klasifikasi SVM yang sesuai dan mengaplikasikan pendekatan SMOTE untuk meningkatkan hasil akurasi klasifikasi tulisan tangan aksara lampung

C. Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah sebagai berikut.

1. Melakukan pendekatan *oversampling* SMOTE pada *dataset* aksara Lampung.
2. Merancang dan mengembangkan model SVM berbahasa Python untuk mengklasifikasi *dataset* tulisan tangan aksara Lampung dengan akurasi cukup tinggi (di atas 90%)

D. Batasan Masalah

Adapun batasan masalah pada penelitian ini adalah:

1. *Dataset* berisi 32140 gambar dengan jenis *file portable graymap* (PGM).
2. *Dataset* aksara Lampung memiliki 18 label yang merepresentasikan 18 jenis aksara Lampung.
3. Pemodelan yang dibuat masih berbasis *executable source code*, dan belum mengimplementasi *User Interface* (UI) dan *User Experience* (UX).
4. Pengklasifikasian tidak mencakup tanda baca aksara Lampung.

E. Manfaat Penelitian

Adapun manfaat dari penelitian ini adalah sebagai berikut:

1. Menambah keilmuan statistika di bidang pembelajaran mesin terutama klasifikasi yang menggunakan algoritme SVM.
2. Memberi informasi mengenai penerapan *oversampling* SMOTE sebagai solusi permasalahan distribusi data yang tidak seimbang pada *dataset* tulisan tangan Aksara Lampung.
3. Penelitian ini dapat dijadikan rujukan untuk penelitian selanjutnya mengenai klasifikasi tulisan tangan dengan memanipulasi atau memodifikasi pendekatan *oversampling* standar terhadap *dataset* yang kelas-kelasnya tidak merata.

BAB II TINJAUAN PUSTAKA

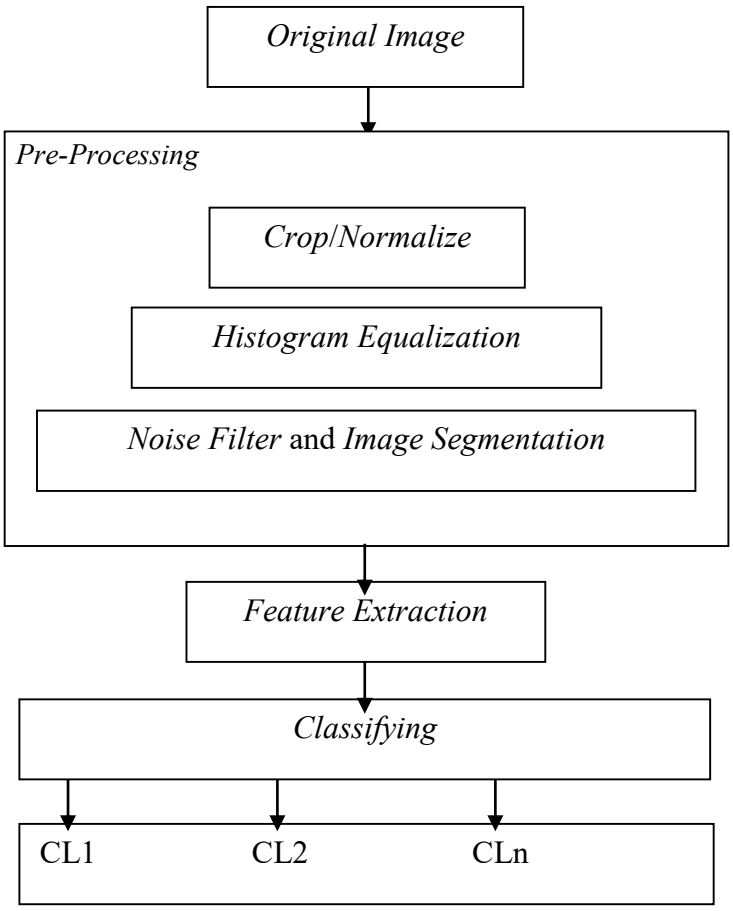
A. Penelitian Terdahulu

Penelitian (Bojja et al, 2019) dengan judul "*Handwritten Text Recognition using Machine Learning Techniques in Application of NLP*" membahas tentang metode yang dapat digunakan dalam pembacaan tulisan tangan berbasis *visual*. Penelitian dilakukan dengan tujuan untuk mengklasifikasi tulisan tangan menjadi beberapa kategori. Setiap tulisan tangan memiliki bentuk tertentu yang merepresentasikan masing-masing huruf alfabet. Tahapan penelitian yang digunakan adalah pengumpulan data gambar, digitalisasi gambar, *preprocessing*, ekstraksi fitur/informasi gambar, kemudian mengeluarkan *output*. Penelitian dilakukan menggunakan bahasa Python versi 3 dengan menggunakan beragam *library* seperti Pytesseract, OS, Gtts (*Google Text To Speech*), dan lainnya. Metode pengklasifikasian yang digunakan adalah pohon keputusan (*Decision Tree*) dan pemodelan jaringan syaraf (*neural network model*). Penelitian ini membaca gambar menggunakan *library* gTTs. Jenis karakter abjad berjumlah 17 dengan akurasi klasifikasi bernilai 92.7%.

Penelitian (Barro et al, 2013) dengan judul "Penerapan *Synthetic Minority Oversampling Technique* (SMOTE) Terhadap Data Tidak Seimbang Pada Pembuatan Model Komposisi Jamu" membahas tentang pemodelan algoritme regresi logistik untuk mencari formulasi tumbuhan yang tepat dalam proses pembuatan jamu memiliki khasiat dalam mengatasi gangguan suasana hati dan perilaku. Data mentah yang digunakan memiliki distribusi yang tidak seimbang, dan pendekatan SMOTE dipilih sebagai solusi dari permasalahan tersebut. Jumlah data minoritas diperbanyak agar setara dengan kelas mayoritas dengan cara melakukan sintesis data buatan. Data tersebut dibuat berdasarkan k-tetangga

terdekat (k-NN). Perbandingan klasifikasi dengan SMOTE dan tanpa SMOTE untuk penelitian ini adalah 97.6% dan 90.8%.

Penelitian (Le et al, 2012) dengan judul "*Image Classification using Support Vector Machine and Artificial Neural Network*" membahas tentang metode klasifikasi aksara romawi dengan jumlah 10 kelas. Penulis menggunakan klasifikasi SVM, salah satu pendekatan terbaik dalam klasifikasi pola dan gambar. Variabel X menyimpan informasi mengenai *dimensional feature space,* dan variabel y menyimpan informasi mengenai *class label*. SVM membuat *hyperplane* optimal dalam pembagian daerah klasifikasi berdasarkan fungsi *kernel* (K). Semua gambar yang berada di daerah tersebut dikategorikan sebagai data pada kelas tersebut. Tahapan utama yang digunakan dalam proses pengklasifikasian adalah pengumpulan data, *pre-processing*, ekstraksi fitur, dan klasifikasi seperti di Gambar 1. Hasil akhir klasifikasi adalah 86%.



Gambar 1. Alur Klasifikasi Gambar

B. *Machine Learning*

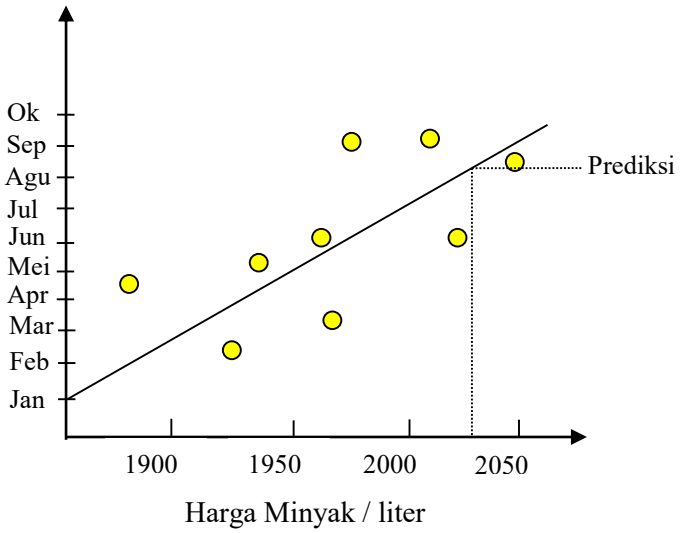
Pembelajaran mesin (*machine learning*) adalah salah satu bidang ilmu komputer yang mempelajari cara pembangunan algoritme yang bermanfaat berdasarkan kumpulan data yang dimiliki. Jenis data yang dapat digunakan antara lain data mentah, data yang sudah dimodifikasi oleh manusia maupun data yang didapatkan dari perhitungan algoritme lain. Pembelajaran mesin juga dapat diartikan sebagai proses penyelesaian masalah dengan cara mengumpulkan data menjadi *dataset* kemudian secara algoritme membangun model statistika berdasarkan *dataset* tersebut. Model statistika diasumsikan dapat menyelesaikan permasalahan sejenis yang diajukan (Burkov, 2019).

Berdasarkan (Chollet, 2017) pembelajaran mesin memiliki beberapa pendekatan antara lain:

1. *Probabilistic Modeling*

Probabilistic modeling adalah algoritme yang mengaplikasikan prinsip statistika dalam menganalisis data. Pendekatan ini digunakan di masa awal pembelajaran mesin. Meskipun metode ini cukup tua, sampai saat ini *probabilistic modeling* masih digunakan banyak orang. Salah satu algoritme yang paling dikenal masyarakat adalah *Naive Bayes*. Algoritme ini mengklasifikasi data dengan mengaplikasikan teori Bayes dengan mengasumsikan data memiliki fitur yang tidak bergantung dengan data lain (independen). Oleh karena itu algoritme ini dinamakan *Naive Bayes*, karena di dunia nyata jarang ditemukan data yang memiliki karakteristik seperti ini. Algoritme ini telah digunakan sebelum tahun 1950 dan tidak menggunakan komputer, melainkan secara manual oleh ahli statistika.

Algoritme lain yang juga lumayan terkenal adalah *logistic regresion*. Algoritme ini mengestimasi parameter model dengan memberikan nilai probabilitas biner (0 atau 1) ke data yang dimiliki sebagai metode klasifikasi. Pendekatan ini memakan waktu komputasi yang sangat lama dan tidak efisien dibanding algoritme lain yang tersedia saat ini. Namun karena sifatnya yang sederhana dan mudah digunakan untuk berbagai macam data, masih banyak ahli yang menggunakan metode ini untuk memberikan gambaran pengklasifikasian data yang ada.



Gambar 2. *Linear Probabilistic Modeling*

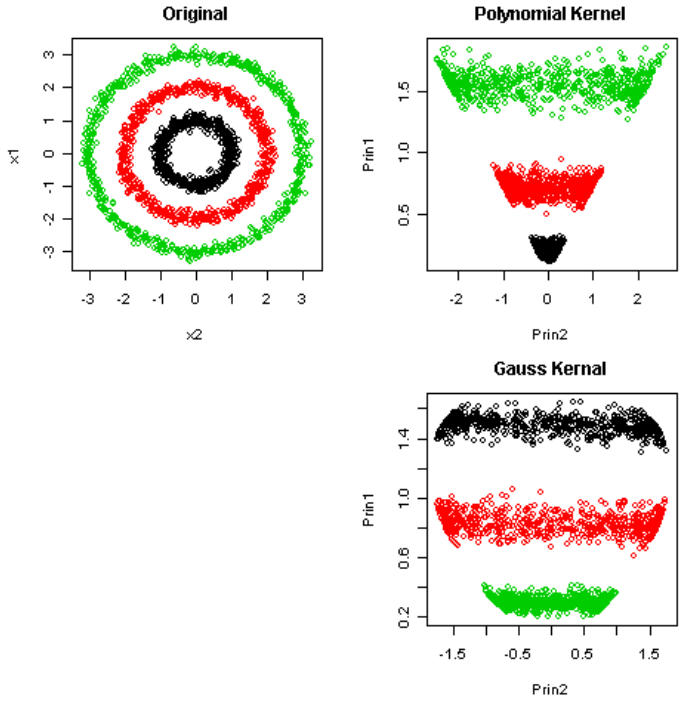
Dapat dilihat di Gambar 2 salah satu contoh dari *Linear Probabilistic Modeling*. Titik kuning merupakan *data point*, dan berdasarkan distribusi data tersebut dapat ditarik asumsi probabilitas *linear* yang digambarkan menggunakan garis lurus. Model ini dapat memprediksi harga minyak per liter di tahun tersebut dengan menarik garis *vertical* dan *horizontal* dari titik garis probabilitas linear.

2. *Early Neural Network*

Iterasi awal jaringan syaraf sudah hampir digantikan sepenuhnya dengan varian yang lebih baru. Jaringan syaraf ini memanfaatkan algoritme *backpropagation*, yang merupakan metode untuk melatih operasi parameter menggunakan *gradient-descent optimization*. Pengaplikasian pertama yang berhasil dilakukan oleh Bell Labs pada tahun 1989 dan Yann LeCun (Cun et al, 1989) menggunakan metode ini untuk mengklasifikasikan tulisan tangan yang kemudian digunakan oleh USPS, jasa pengiriman paket di Amerika Serikat di tahun 1990 untuk mengotomatisasi pembacaan kode pos di surat yang dikirim.

3. Metode *Kernel*

Metode kernel adalah kumpulan algoritme klasifikasi yang digunakan untuk menganalisis pola data dengan mencari dan mempelajari hubungan yang dimiliki data secara general (seperti *clusters, rankings, principal components, correlations, classifications*) pada *dataset*. Salah satu algoritme terkenal yang memanfaatkan metode kernel adalah *Support Vector Machine* (SVM).



Gambar 3. Ilustrasi Kernel (Xie, 2010)

Gambar 3 menunjukkan perubahan distribusi data setelah diproses menggunakan kernel *polynomial* dan *Gauss*. Metode kernel merupakan salah satu teknik penting dalam *data mining* yang dapat diaplikasikan ke banyak algoritme.

Metode pembelajaran data dibagi menjadi dua, yaitu *supervised* dan *unsupervised*. Menurut (Berry & Mohamed, 2020), kedua metode cocok digunakan untuk mengekstraksi informasi dari *dataset* besar. Pembelajaran secara *supervised* dilakukan dengan menggeneralisasi informasi yang didapat dari data yang memiliki label dan kemudian menggunakannya untuk memprediksi data baru yang tidak memiliki label. Untuk mengukur performa pendekatan ini, data yang memiliki label diprediksi terlebih dahulu, kemudian diukur akurasi dan misklasifikasi menggunakan berbagai satuan pengukur. Berdasarkan (Libbrecht & Noble, 2015), algoritme *supervised* melakukan tugas analisis menggunakan data training, dan kemudian membangun fungsi *contingent* untuk digunakan dalam mengklasifikasi atribut yang ada. Sementara pembelajaran *unsupervised* merujuk kepada proses pengelompokan data menjadi *cluster* dengan metode atau algoritme otomatis terhadap data yang belum diklasifikasi atau dikelompokkan. Pada situasi ini, algoritme harus mempelajari hubungan atau fitur data secara *implicit* dan kemudian

mengelompokkan data yang memiliki fitur atau karakteristik serupa. Apabila sebagian data memiliki label, pembelajaran menggunakan data ini dapat dikatakan *semi-supervised*.

C. Python

Python adalah salah satu bahasa yang digunakan dalam pembuatan program. Python memanfaatkan *command-line interpreter* untuk menghasilkan output dari *syntax* yang ditulis dan dapat dilihat secara *realtime* oleh pengguna. Dibandingkan bahasa pemrograman lain, penulisan kode Python sangat sederhana seperti ditampilkan di Gambar 4.

```
// JAVA "HELLO WORLD"

public class Hello{
    public static void main(String args[]){
        System.out.println("HELLO WORLD");
    }
}
```

```
// C++ "HELLO WORLD"

#include <iostream>
using namespace std;
int main(){
    cout<<"HELLO WORLD"<<endl;
}
```

```
// PYTHON "HELLO WORLD"

print "HELLO WORLD"
```

Gambar 4. Perbandingan *Syntax* Java, C++, dan Python.

Python juga tidak memiliki atribut seperti *public, private, protected*, jadi program menjadi semakin sederhana, pendek dan lebih mudah dipahami. Atribut yang dimiliki Python bersifat dinamis sehingga dapat dibuat di mana pun dan kapan pun, yang tidak dapat dilakukan oleh Java dan juga C++. Fungsi dan kelas yang dimiliki Python bersifat *polymorphism*, yang berarti satu *interface* dapat mewakili entitas dengan tipe berbeda. Salah satu contohnya adalah *operator overloading*. Pada proses ini, satu jenis operator dapat digunakan untuk lebih dari satu jenis fungsi tergantung argumen yang diberikan. Kelebihan Python lainnya adalah penggunaan indentasi yang sangat bermanfaat dalam membantu penulisan kode agar menjadi lebih mudah dipahami. Indentasi memungkinkan pembuat program untuk membagi satu rangkaian kode menjadi ke banyak baris tanpa diputus. Selain itu Python juga memiliki banyak *library* bawaan yang memiliki perhitungan algoritme, jadi pembuat program tidak perlu mempelajari teori perhitungan dan hanya harus

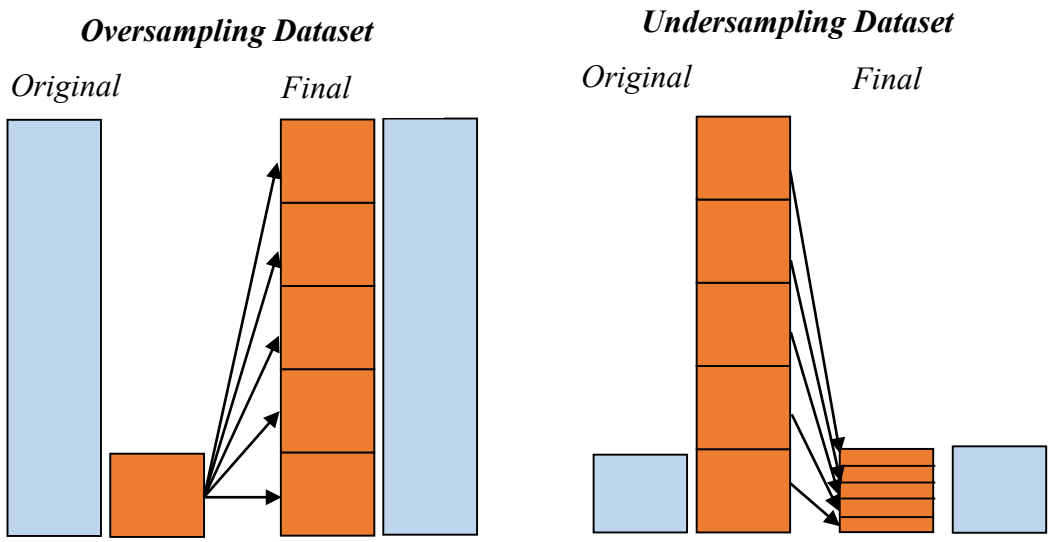
mengerti parameter yang dibutuhkan dan gambaran cara kerja algoritme tersebut (Bogdanchikov et al, 2013). Oleh karena itu Python sangat cocok digunakan untuk proses pengklasifikasian data.

D. *Oversampling*

Oversampling adalah salah satu metode yang digunakan untuk menyeimbangkan distribusi data dengan meningkatkan jumlah data minoritas. Metode ini dapat mempengaruhi hasil perhitungan algoritme pembelajaran mesin, karena di beberapa kasus algoritme tidak mementingkan data kelas minoritas bahkan ada yang mengabaikan data tersebut. Salah satu pendekatan yang dapat dilakukan adalah dengan menggunakan *random sampling*. Pendekatan ini menduplikasi data minoritas untuk meratakan jumlah keseluruhan data di semua kelas. Data dipilih secara acak sampai distribusi data yang diinginkan tercapai (He & Ma, 2013). Metode seperti ini tidak menambahkan fitur baru terhadap *dataset* sehingga cocok diimplementasikan pada data yang besar dan kompleks karena waktu eksekusi yang cepat. Perubahan distribusi data juga hanya diterapkan ke *training dataset*. Metode lain yang dapat dilakukan sebagai solusi distribusi data yang tidak seimbang adalah *undersampling*. Berbeda dari *oversampling*, *undersampling* menghapus data dari kelas mayoritas sampai jumlah data disetiap kelas sama. Kedua metode memiliki kelebihan seperti ditampilkan di Tabel 1.

Tabel 1. Kelebihan *Oversampling* dan *Undersampling*

	Oversampling	*Undersampling*
Kelebihan	Tidak ada informasi yang dibuang dari data mentah.	Waktu pelatihan data menjadi lebih singkat.
	Cocok digunakan untuk dataset yang berjumlah sedikit.	Memiliki performa baik untuk data dengan dimensi tinggi.



Gambar 5. Proses *Oversampling* dan *Undersampling*

Gambar 5 menujukkan proses *oversampling* dan *undersampling* pada umumnya. Proses *oversampling* memperbanyak data minoritas dari *dataset* mentah lewat berbagai macam pendekatan sampai jumlah data minoritas tersebut sama dengan jumlah data mayoritas. Sedangkan proses *undersampling* membuat sebagian data mayoritas sampai jumlahnya setara dengan data minoritas.

E. SMOTE

Synthetic Minority Oversampling Technique (SMOTE) adalah teknik statistika untuk menambahkan jumlah data di kelas minoritas agar seimbang dengan kelas mayoritas. Berbeda dengan teknik *random oversampling*, SMOTE tidak hanya menduplikasi data, melainkan menggunakan algoritme k-NN untuk mensintesis data baru berdasarkan data yang telah dimiliki. Cara kerja SMOTE yaitu sebuah data dari kelas minoritas dipilih secara acak, kemudian menggunakan algoritme k-NN untuk mencari data tetangga dan menghubungkan kedua data tersebut menggunakan garis. Data sintesis diambil dari kombinasi *convex* kedua data yang telah dipilih (He & Ma, 2013).

SMOTE memiliki beberapa varian dengan proses ekstrapolasi data tidak hanya memanfaatkan algoritme k-NN saja, tetapi juga memanfaat algoritme lainnya.

Adapun varian tersebut antara lain *Borderline* SMOTE, *K-Means* SMOTE, dan SVM SMOTE.

1. *Borderline* SMOTE

   Metode *Borderline* SMOTE hanya mengekstrapolasi data yang memiliki data tetangga berasal dari kelas mayoritas dan minoritas (Nguyen et al, 2011). Data tersebut dapat dikatakan berada di area "*borderline*". Metode ini mengklasifikan data di kelas minoritas menjadi dua, yaitu *noise point*, dan *border point*. Pendekatan SMOTE ini tidak mempertimbangkan *noise point* dan hanya menghasilkan data berdasarkan *border point*.

2. *K-Means* SMOTE

   K-Means SMOTE adalah metode *oversampling imbalanced data* yang membantu proses klasifikasi dengan menghasilkan data kelas minoritas baru secara aman dan di area yang penting sebagai *input*. Metode ini berusaha menghindari sintesis data *noise* (yang tidak terlalu berguna). Tahapan yang dilakukan antara lain :

   - Melakukan *clustering* terhadap semua data menggunakan algoritme *K-Means Clustering*.
   - Memilih *cluster* yang memiliki jumlah data dari kelas minoritas terbanyak.
   - Membuat data baru berdasarkan sampel yang didapat ke *cluster* yang memiliki data kelas minoritas sedikit.

3. SVM SMOTE

   Metode ini menggunakan algoritme SVM untuk mengidentifikasi contoh misklasifikasi di batas keputusan (*decision boundary*). Batas keputusan didefinisikan menggunakan support *vector* dan data yang berada disekitar *support vector* tersebut menjadi fokus dalam pembuatan data sintesis baru. Selain itu metode ini juga lebih mengutamakan area dengan data dari kelas minoritas yang lebih sedikit dan kemudian mengekstrapolasi ke arah batas keputusan.

F. *Image Classification*

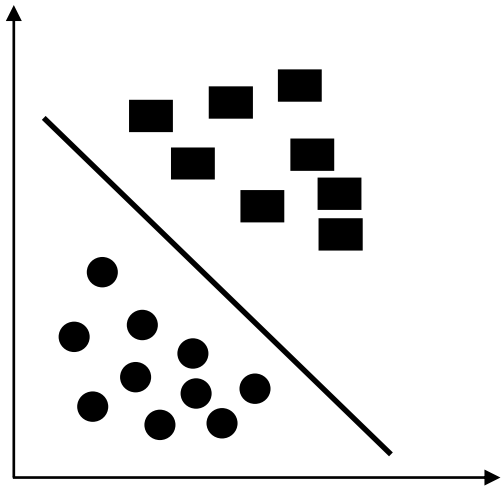
Berdasarkan (Anand, 2017), klasifikasi gambar adalah proses pengubahan data *spectral* menjadi informasi. Data *spectral* adalah sekelompok *pixel* yang memiliki informasi fitur gambar dan terdapat di berbagai *channel* warna. Klasifikasi dilakukan dengan mengidentifikasi distribusi pixel dengan nilai serupa dan kemudian mengkelompokkan data tersebut menjadi satu kelas. Salah satu penerapan klasifikasi gambar adalah *remote sensing*. Seperti klasifikasi lainnya, *image classification* juga dapat dilakukan dengan dua pendekatan yaitu *supervised* dan *unsupervised*. Teknik supervised antara lain *maximum likelihood*, *minimum distance*, dan *parallelpiped classification*. Di *remote sensing*, pendekatan *unsupervised* yang dapat digunakan antara lain ISODATA dan *K-Means clusterring*. Klasifikasi gambar bersifat diskriminan, yaitu mencari nilai tertinggi untuk klasifikasi *multiclass*, dan nilai positif untuk klasifikasi biner. Contoh klasifikasi *multiclass* adalah pembacaan karakter dan angka dari sastra yang di *scan*.

G. SVM

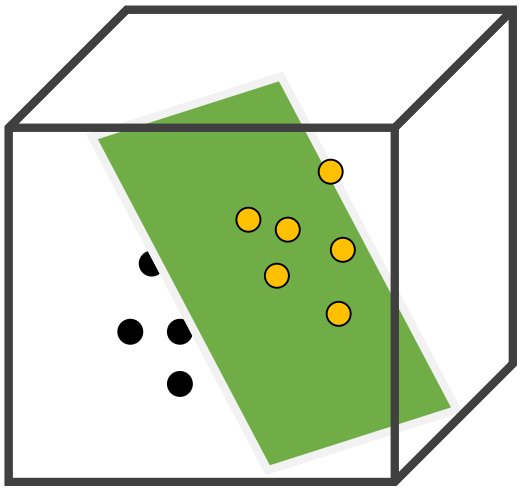
Pendekatan *Support Vector Machine* (SVM) dilakukan dengan mencari *hyperplane* optimal yang dapat membagi daerah antar kelas. Untuk kasus yang hanya memiliki dua kelas, klasifikasi dapat dilakukan secara linear. SVM memilih dari banyak kemungkinan garis keputusan berada dan memilih posisi yang memiliki nilai *margin* antar kelas paling besar. Margin didefinisikan sebagai jumlah jarak antar *hyperplane* dan kedua kelas terdekat. Perhitungan ini dapat diselesaikan dengan teknik standar *quadratic programming optimization* (Pal & Foody, 2010).

SVM merupakan algoritme pembelajaran mesin *supervised* yang dapat digunakan untuk masalah klasifikasi dan regresi, tetapi algoritme ini biasanya digunakan untuk proses klasifikasi. Pada algoritme ini ada beberapa data yang dijadikan *support vector*, yang kemudian digunakan untuk membantu optimalisasi letak *hyperplane*. Biasanya *support vector* berada paling dekat dengan *hyperplane* dan yang paling sulit untuk diklasifikasikan. Jika data ini dihilangkan maka posisi *hyperplane* tentunya akan berubah. Untuk data yang dapat dibagi secara linear, *hyperplane*

berbentuk garis untuk 2 dimensi seperti di Gambar 6, dan persegi/persegi panjang untuk 3 dimensi seperti di Gambar 7.



Gambar 6. *Hyperplane* 2 Dimensi



Gambar 7. *Hyperplane* 3 Dimensi

Untuk permasalahan yang tidak dapat diselesaikan secara *linear*, SVM menggunakan *kernel* untuk mengoptimalisasikan letak *hyperplane*. Fungsi *kernel* yang sering digunakan antara lain *polynomial*, *Radial Basis Function* (RBF), dan *Sigmoid*.

H. Pengujian Model

Pengujian model dilakukan dengan menggunakan *confusion matrix*. *Confusion Matrix* merepresentasikan informasi tentang seberapa banyak suatu pola dideteksi dengan benar (Ruuska et al, 2018). Pengujian model ini dapat digunakan untuk mengevaluasi performa sebuah model klasifikasi melalui metrik perhitungan performa seperti akurasi, *precision*, *recall,* dan nilai F1.

Pengukuran performa model menggunakan metrik pengukuran akurasi dan F1 dapat dicari dengan memanfaatkan fungsi *accuracy_score*, *f1_score* dari *library sklearn*.

$$Accuracy = \frac{True\ Positive + True\ Negative}{\text{Total}}$$ Persamaan 1. Perhitungan Nilai Akurasi

Nilai akurasi didapatkan berdasarkan rata-rata data klasifikasi yang benar dibagi jumlah data seperti di Persamaan 1. Cara kerjanya yaitu dengan menggunakan 2 parameter y_test dan poly_pred. Untuk setiap elemen X di matriks poly_pred, nilai tersebut dibandingkan dengan elemen X di matriks y_test.

Contoh kasus:
y_pred = [0, 2, 1, 3, 0].
y_true = [0, 1, 2, 3, 0].

Elemen yang sama (*True Positive* + *True Negative*) = 3.
Total Elemen = 5.
Nilai akurasi = 3/5 = 0.6 atau 60% (dikalikan konstan 100).

Untuk mencari nilai F1, diperlukan nilai *precision* dan *recall*. Nilai tersebut dapat dicari menggunakan Persamaan 2 dan Persamaan 3.

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive}$$ Persamaan 2. Perhitungan Nilai *Precision*

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative}$$ Persamaan 3. Perhitungan Nilai *Recall*

Nilai F1 didapatkan dari rata-rata *precision* dan *recall* seperti pada Pesamaan 4.

$$F1 = \frac{Precision * Recall}{Precision + Recall}$$ Persamaan 4. Perhitungan Nilai *F1*

Variabel perhitungan dicari menggunakan *confusion matrix* seperti di Tabel 2. Adapun pengertian dari variabel tersebut antara lain:

- *True Positive* (TP) : Data positif diklasifikasikan positif
- *True Negative* (TN) : Data negatif diklasifikasikan negatif
- *False Positive* (FP) : Data negatif diklasifikasikan positif
- *False Negative* (FN) : Data positif diklasifikasikan negatif

Tabel 2. *Confusion Matrix*

	Positive	*Negative*
Positive	*True Positive* (TP)	*False Positive* (FP)
Negative	*False Negative* (FN)	*True Negative* (TN)

BAB III METODE PENELITIAN

A. Tempat dan Waktu Penelitian

Penelitian ini dilakukan di Fakultas Matematika dan Ilmu Komputer Universitas Lampung yang berada di Jalan Prof. Dr. Soemantri Brojonegoro No.1, Kelurahan Gedong Meneng, Kecamatan Rajabasa, Bandar Lampung. Waktu penelitian dilaksanakan pada bulan November 2021 sampai Maret 2022.

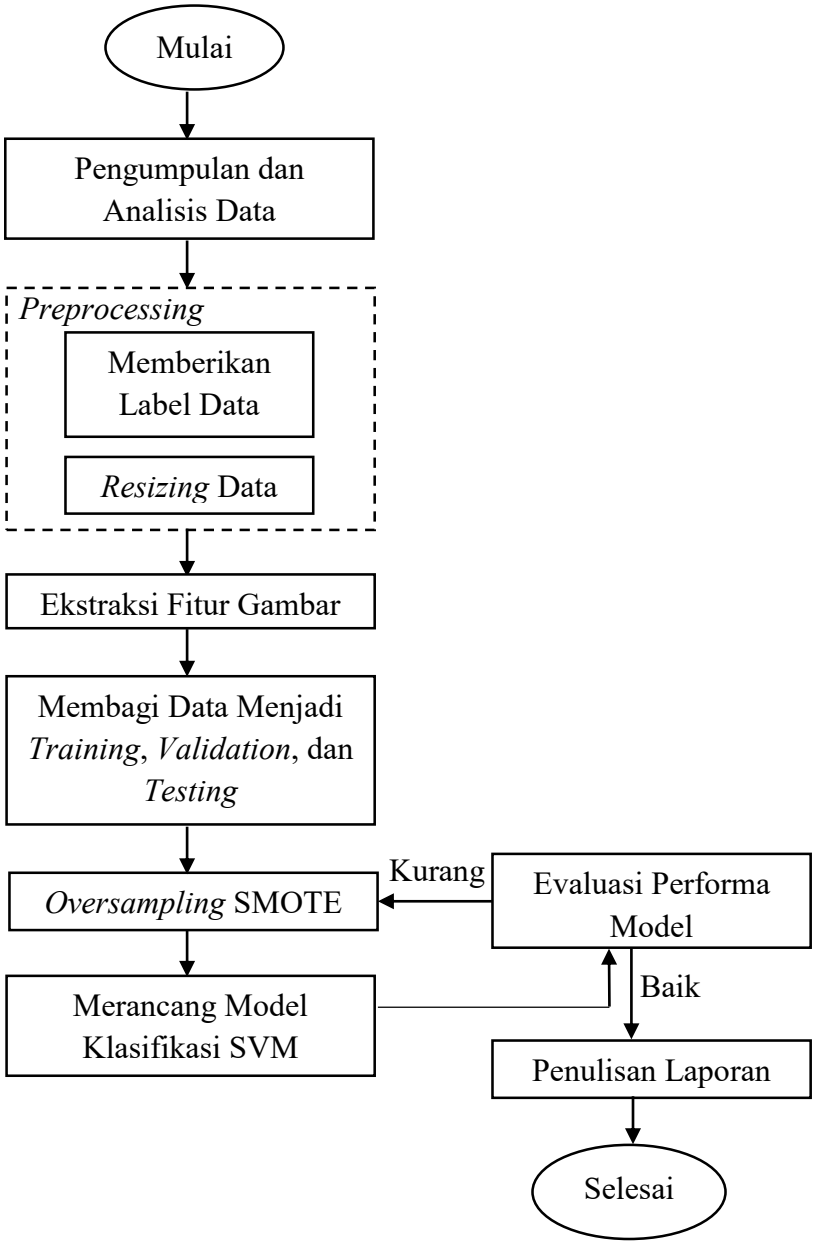
B. Alat dan Pendukung Penelitian

Peralatan pendukung dalam menunjang penelitian berupa perangkat keras (*hardware*) dan perangkat lunak (*software*).

1. Perangkat keras yang digunakan
   a. *Processor* : Intel Core i7-6700K CPU @4.00GHz.
   b. RAM : 2 x 8 GB DDR4.
   c. *System Type* : *64-bit Operating System*, x64-*based processor*.
   d. *Storage* : 1 TB.
2. Perangkat lunak yang digunakan
   a. Sistem Operasi : Windows 10.
   b. *Text Editor* : Jupyter Notebook 6.4.5.
   c. IDE : Jupyter Lab 3.2.2.
   c. Python 3.9.7.
   d. *Browser* : Chrome.
   e. Python *Libraries* :
      - Matplotlib 3.4.3 : visualisasi *dataset*.
      - Numpy 1.21.4 : numpy.unique (mengambil nilai label data).
      - Joblib 1.1.0 : joblib.dump (menyimpan data hasil komputasi ke *pickle* agar dapat dibaca tanpa harus mengulang proses perhitungan).

- skimage 0.18.3 : skimage.io.imread (mengekstrak fitur gambar aksara lampung dari file pgm), skimage.transform.resize (mengubah dimensi resolusi gambar).
- sklearn 1.0.1 : sklearn.model_selection (membagi data menjadi *training*, *testing*, dan *validation*), sklearn.datasets.make_classification (melakukan klasifikasi terhadap *dataset*), sklearn.metrics (mengevaluasi performa model klasifikasi).
- imblearn 0.8.1 : imblearn.oversampling (melakukan pendekatan *oversampling* seperti SMOTE).

C. Tahapan Penelitian

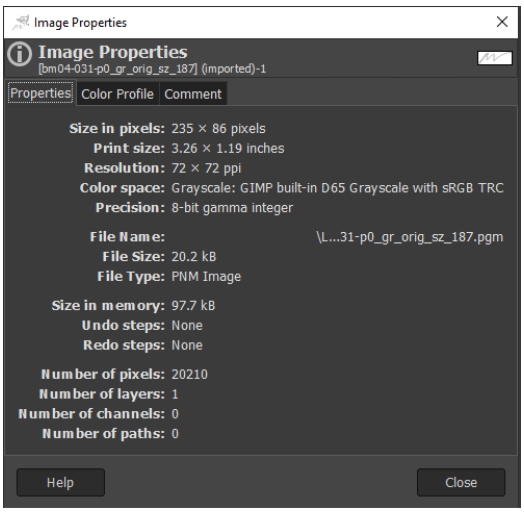


Gambar 8. *Flow Chart* Tahapan Penelitian

Penelitian "Pendekatan *Oversampling* SMOTE untuk *Imbalanced Dataset* Aksara Lampung dan Klasifikasi Menggunakan SVM" diawali dengan pengumpulan data dan studi literatur, kemudian dilanjutkan dengan *preprocessing* data dengan menormalisasikan ukuran gambar menjadi 48x48, kemudian gambar tersebut diekstrak fiturnya dengan menggunakan *library* skimage. Setelah data siap untuk dimanipulasi, *dataset* dibagi menjadi tiga yaitu *training*, *validation*, dan *testing*. Setelah itu dilakukan pendekatan *oversampling* SMOTE terhadap data *training*. Data kemudian diklasifikasikan menggunakan algoritme *Support Vector Machine* (SVM). Performa model kemudian dievaluasi dengan *confusion matrix* menggunakan metrik pengukuran *accuracy* dan F1. Bila hasil evaluasi belum memuaskan, *parameter* SMOTE dan SVM diubah lewat proses *hypertuning parameter* sampai akurasi dan nilai F1 cukup tinggi. Setelah itu hasil dan proses penelitian didokumentasikan ke dalam laporan penelitian. Alur tahapan penelitian diperlihatkan pada Gambar 8.

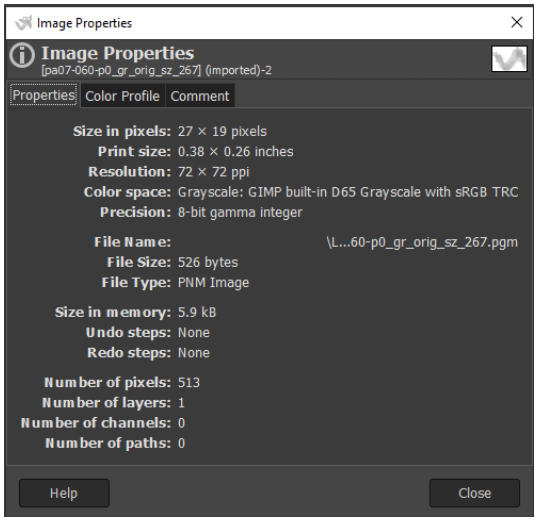
1. Pengumpulan dan Analisis Data

Data mentah didapat dari penelitian sebelumnya (Junaidi et al, 2011). Data gambar berbentuk pgm sehingga gambar hanya memiliki satu *channel* warna yaitu *grayscale*. Ukuran dan dimensi gambar data bervariasi. Gambar 9 menunjukkan karakteristik citra dengan ukuran terbesar yaitu 20.2 kB. Gambar tersebut memiliki resolusi sebesar 235x86 *pixel*.



Gambar 9. Gambar dengan Ukuran Terbesar

Gambar 10 menunjukkan karakteristik citra dengan ukuran terkecil yaitu 0.526 kB. Gambar tersebut memiliki resolusi sebesar 27x19 *pixel*.



Gambar 10. Gambar dengan Ukuran Terkecil

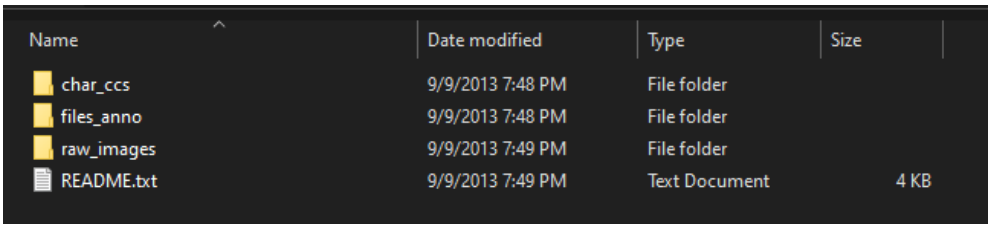
Seperti pada Tabel 3, persentase distribusi data aksara Lampung adalah tidak seimbang. Pesentase distribusi terendah memiliki nilai sebesar 0.74% untuk aksara ca dengan jumlah 238 data, dan memiliki persentase distribusi terbesar bernilai 11.83% untuk aksara pa dengan jumlah 3802 data.

Tabel 3. Distribusi Data Aksara Lampung dan Persentase

Aksara	Jumlah	Persentase (%)
ka	3131	9.74
ga	2633	8.19
nga	695	2.16
pa	3802	11.83
ba	1957	6.09
ma	2874	8.89
ta	3093	9.62
da	2164	6.73
na	1201	3.74
ca	238	0.74
ja	563	1.75
nya	772	2.40
ya	660	2.05
a	2928	9.11
la	1715	5.34

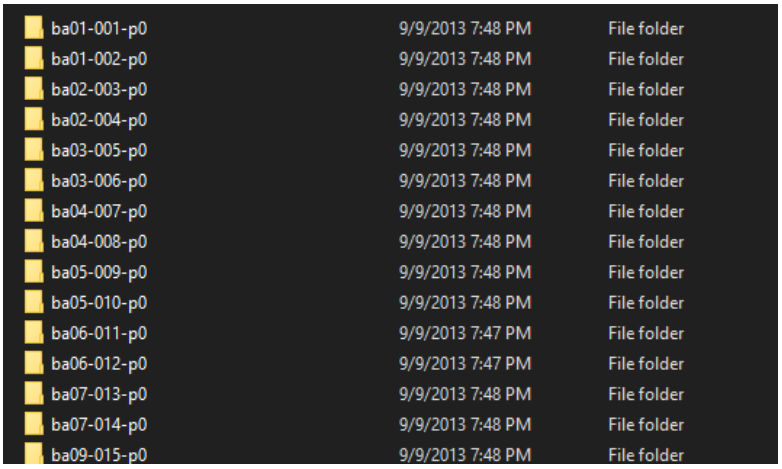
Aksara	Jumlah	Persentase (%)
sa	2305	7.17
wa	254	0.79
ha	660	2.05

Dataset memiliki tiga *subfolder* yaitu char_css, files_anno, dan raw_images seperti di Gambar 11. Char_css berisikan data gambar aksara Lampung, files_anno berisikan anotasi sumber sastra Lampung tempat aksara tersebut diambil, dan raw_images berisikan data gambar sastra Lampung.



Gambar 11. *Subfolder Dataset* Aksara Lampung

Data mentah gambar yang disimpan di folder char_ccs masih dilabelkan berdasarkan nama *file* sumber gambar tersebut. Oleh karena itu diperlukan proses melabelkan data berdasarkan jenis aksara gambar sebelum data dapat dimanipulasi untuk klasifikasi. Gambar 12 menunjukkan folder gambar dengan format penamaan berdasarkan masing-masing sumber gambar. Jadi setiap folder merepresentasikan satu gambar sastra Lampung yang berisikan banyak aksara Lampung yang diekstrak menjadi *file* pgm dan dijadikan bahan penelitian ini.



Gambar 12. Data Gambar dengan Label Sumber Gambar

Informasi label data gambar dapat dirujuk dari file .anno di folder dengan nama files_anno yang berisikan informasi nama gambar di kolom pertama, dan label gambar di kolom terakhir. Gambar 14 menujukkan informasi data gambar dalam bentuk kolom. Kolom pertama berisikan nama *file* gambar aksara Lampung, kolom kedua sampai kelima berisikan informasi letak gambar aksara berada, dan kolom keenam berisikan jenis aksara Lampung.

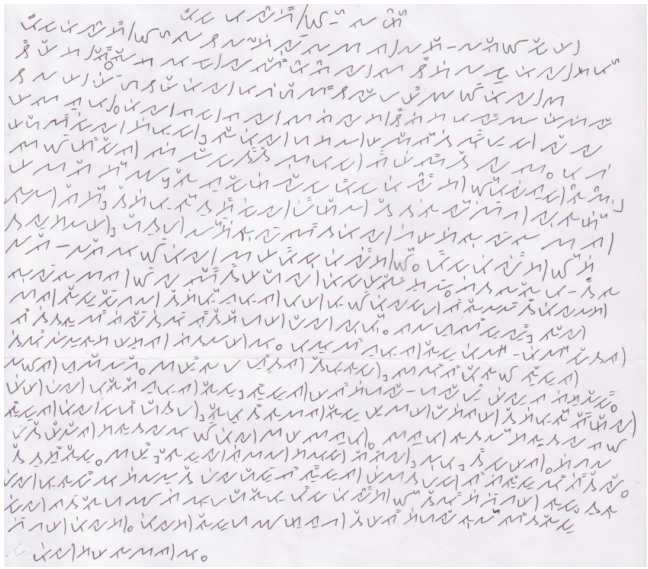
ba01-001-p0_lampung_char.anno - Notepad

File Edit Format View Help

ba01-001-p0_gr_orig_sz_0.pgm	701	24	73	47	ba
ba01-001-p0_gr_orig_sz_1.pgm	878	29	79	49	ba
ba01-001-p0_gr_orig_sz_2.pgm	960	32	74	43	ga
ba01-001-p0_gr_orig_sz_3.pgm	1191	34	91	70	ya
ba01-001-p0_gr_orig_sz_4.pgm	756	36	79	49	ta
ba01-001-p0_gr_orig_sz_5.pgm	1010	37	64	43	pa
ba01-001-p0_gr_orig_sz_6.pgm	1065	42	77	37	ga
ba01-001-p0_gr_orig_sz_7.pgm	1365	57	109	44	a
ba01-001-p0_gr_orig_sz_8.pgm	1497	58	73	45	ca
ba01-001-p0_gr_orig_sz_9.pgm	72	62	83	47	ba
ba01-001-p0_gr_orig_sz_10.pgm	141	73	72	44	ta
ba01-001-p0_gr_orig_sz_11.pgm	228	81	69	40	ba
ba01-001-p0_gr_orig_sz_12.pgm	291	85	72	41	ga
ba01-001-p0_gr_orig_sz_13.pgm	337	94	71	44	pa
ba01-001-p0_gr_orig_sz_14.pgm	407	94	71	43	sa
ba01-001-p0_gr_orig_sz_15.pgm	520	101	98	60	ya
ba01-001-p0_gr_orig_sz_16.pgm	619	109	43	31	ha
ba01-001-p0_gr_orig_sz_17.pgm	665	112	122	51	a
ba01-001-p0_gr_orig_sz_18.pgm	774	122	77	60	da
ba01-001-p0_gr_orig_sz_19.pgm	861	141	94	40	la

Gambar 13. Anotasi Data Gambar

Subfolder raw_images berisikan gambar hasil *scan* sastra Lampung. Gambar 13 menunjukkan salah satu contoh file didalamnya.

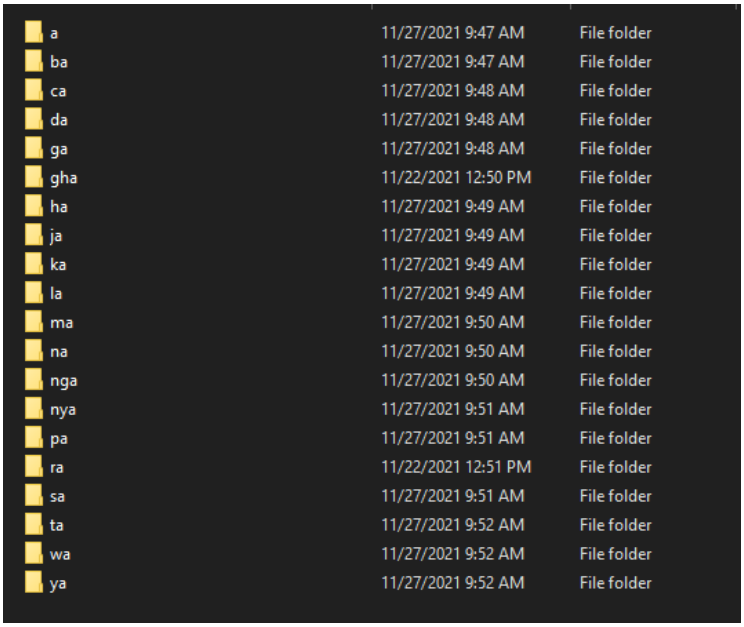


Gambar 14. Gambar dari *Dataset* Aksara Lampung

2. *Preprocessing*

a. Melabelkan Data

Data dipindahkan ke *folder* kelas data gambar tersebut berada seperti pada Gambar 15. Kemudian fitur gambar diekstraksi menggunakan fungsi *imread* dari *library skimage* dan disimpan dalam bentuk *pickle* untuk mempermudah pembacaan data gambar meskipun kernel Python telah di *reset* ulang. Kemudian data diubah menjadi *numpy array* untuk mempermudah proses pengolahan data gambar.

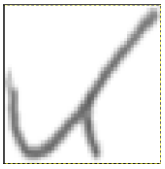


Gambar 15. Melabelkan Data Gambar

b. *Resizing* Data

Data gambar memiliki latar belakang berwarna putih dan goresan penulisan yang jelas. Data juga sudah dalam bentuk *grayscale*, yang berarti *channel input* warna sudah bernilai 1 dan cocok untuk dimanipulasi dalam bentuk *numpy array*. Variasi data gambar terletak di dimensi karena ukuran dimensi berkisar dari 27x19 sampai 235x86. Oleh karena itu untuk meningkatkan integritas data, dimensi semua gambar dinormalisasikan menjadi 48x48. Ukuran ini dipilih karena bernilai cukup kecil demi mempersingkat waktu komputasi tanpa mengorbankan kualitas secara berlebihan.

Gambar 16. Citra Aksara Ba dengan Ukuran 73x47 *Pixel*



Gambar 17. Citra Aksara Ba dengan Ukuran 48x48 *Pixel*

Contoh pengubahan dimensi gambar dapat dilihat seperti di Gambar 16 untuk data mentahnya, yang kemudian menghasilkan citra seperti di Gambar 17 dengan ukuran 48x48 *pixel*.

3. Ekstraksi Fitur Gambar

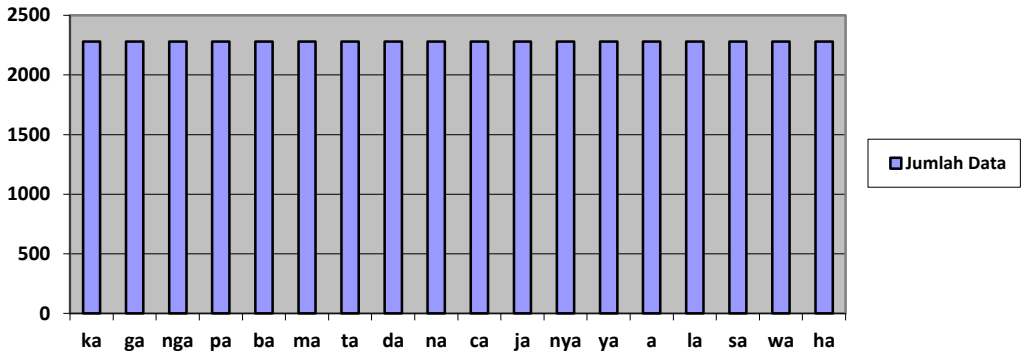
Data mentah aksara Lampung memiliki distribusi warna *grayscale*, yang berarti hanya memiliki satu *channel* warna, yang ditampilkan dalam warna varian abu-abu, dan warna paling gelap ditampilkan dengan warna hitam. Aksara Lampung direpresentasikan dengan variasi warna tersebut dalam bentuk *pixel* di kanvas gambar. Berdasarkan distribusi *pixel*, informasi gambar dapat diekstraksi mengunakan pendekatan *feature extraction*.

4. Membagi Data Menjadi *Training*, *Validation*, dan *Testing*

Data dibagi menjadi *training*, *validation* dan *testing* dengan perbandingan data paling berdasarkan hasil experimen rasio 60:20:20, 70:10:20, 50:20:30, dan 60:10:30. Dibutuhkannya data validasi untuk menghindari terjadinya *overfitting* pada model klasifikasi. *Overfitting* terjadi jika performa model lebih baik pada data *training* dibandingkan pada data *testing*. Selain itu, data validasi juga dibutuhkan agar proses *hypertuning* parameter tidak bias yang berarti parameter berfungsi baik tidak hanya untuk data *testing* saja, tetapi juga memiliki performa serupa jika model digunakan pada *dataset* lainnya. Pembagian data Aksara Lampung dilakukan menggunakan *library sklearn*, salah satu *library scikit* terpopuler dalam mengklasifikasi data.

5. *Oversampling* SMOTE

Data *training* di *resample* menggunakan *oversampling* SMOTE agar jumlah data di semua kelas minoritas setara dengan jumlah data di kelas mayoritas. Contoh: Di data *training*, kelas "pa" merupakan kelas mayoritas dan memiliki data berjumlah 2281. Maka data di kelas minoritas disetarakan dengan jumlah kelas pa yaitu 2281 seperti di Gambar 18.



Gambar 18. *Oversampling Training Dataset* Menggunakan SMOTE

Di penelitian ini metode *oversampling* lebih cocok digunakan dibandingkan *undersampling* karena *dataset* aksara Lampung keseluruhannya hanya berjumlah 32140. Oversampling menyimpan seluruh informasi dari data mentah sementara *undersampling* mengharuskan penghapusan sebagian informasi dari data mentah. Informasi tersebut memiliki peran penting dalam pemodelan klasifikasi walaupun berasal dari kelas mayoritas. Jadi secara umum, untuk *dataset* yang berjumlah sedikit *oversampling* lebih cocok dibandingkan *undersampling*.

6. Merancang Model Klasifikasi SVM

Model klasifikasi dirancang dengan parameter jenis kernel, nilai C yang merupakan parameter *regularization*, dan nilai derajat (*degree*) jika menggunakan kernel *polynomial* dengan memanfaatkan fungsi SVC dari library sklearn.svm. Model dirancang menggunakan nilai C=1.0, *degree*=3 kemudian dibandingkan dengan performa model menggunakan parameter bernilai berbeda lewat proses *hypertuning* parameter.

7. Evaluasi Performa Model

Performa model diukur berdasarkan *confusion matrix* dan menggunakan nilai akurasi dan F1 sebagai metrik pengukuran performa model klasifikasi dengan memanfaatkan *library* sklearn.metrics. Nilai tersebut ditampilkan dalam satuan persen, yang juga dijadikan acuan dalam proses *hypertuning parameter* model supaya hasil klasifikasi yang didapat memiliki nilai cukup tinggi (di atas 90%).

D. Penulisan Laporan

Laporan tertulis merupakan bentuk dokumentasi dari serangkaian penelitian pengembangan "Pendekatan *Oversampling* SMOTE untuk *Imbalanced Dataset* Aksara Lampung dan Klasifikasi Menggunakan SVM" secara bertahap dan mencakup semua kegiatan yang telah dilakukan sehingga menjadi bukti nyata bahwa penelitian ini telah berlangsung.

E. Jadwal Penelitian

Penelitian dilakukan sesuai dengan jadwal yang telah ditentukan dalam bentuk *gantt chart*. Jadwal penelitian ditampilkan pada Tabel 4.

Tabel 4. Jadwal Penelitian

No	Nama	Nov 2021				Des 2021				Jan 2022				Feb 2022				Mar 2022			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	Pengumpulan dan Analisis Data																				
2	Seminar Usul																				
3	*Preprocessing*																				
4	*Oversampling* SMOTE																				
5	Merancang Model Klasifikasi SVM																				
6	Evaluasi Performa Model																				
7	Seminar Hasil																				
8	Seminar Komprehensif																				

BAB V SIMPULAN DAN SARAN

A. Simpulan

Berdasarkan hasil penelitian Pendekatan *Oversampling* SMOTE untuk *Imbalanced Dataset* Aksara Lampung dan Klasifikasi Menggunakan SVM maka didapatkan simpulan sebagai berikut:

1. Berhasil menggunakan pendekatan *oversampling* SMOTE sebagai solusi dari distribusi data yang tidak seimbang.
2. Berhasil membangun model SVM untuk mengklasifikasi data aksara Lampung dengan akurasi cukup tinggi (di atas 90%).
3. Manipulasi data gambar dalam bentuk *numpy array* mampu menjaga integritas data sekaligus memungkinkan penggunaan berbagai macam fungsi bersangkutan dengan pembelajaran mesin.
4. Klasifikasi menggunakan *dataset* dengan rasio 60:10:30 paling cocok digunakan untuk penelitian ini. Dataset dengan rasio testing tertinggi diantara keempat rasio yang dicoba memiliki performa lebih baik.
5. Pendekatan *Borderline* SMOTE dengan nilai k=3 cocok digunakan untuk *dataset* aksara Lampung karena menghasilkan akurasi tinggi dan waktu komputasi yang cukup rendah. Dapat dilihat juga bahwa pendekatan SVM SMOTE tidak cocok digunakan karena waktu komputasi yang sangat lama dibandingkan varian SMOTE lainnya.
6. Klasifikasi SVM *kernel* RBF dengan nilai C=7 menghasilkan akurasi tertinggi, dan kernel *Sigmoid* menghasilkan akurasi yang jauh lebih rendah dibanding kernel RBF dan *polynomial*.
7. Pendekatan SMOTE menghasilkan sebagian data baru yang berbayang, ini disebabkan karena data dibuat berdasarkan 2 data yang cukup berbeda.

B. Saran

Berdasarkan penelitian Pendekatan *Oversampling* SMOTE untuk *Imbalanced Dataset* Aksara Lampung Dan Klasifikasi Menggunakan SVM yang telah dilakukan, maka didapatkan saran sebagai berikut:

1. Mengembangkan model klasifikasi dengan menggunakan *dataset* yang mencakup seluruh 20 aksara Lampung dan membandingkan performa beberapa algoritme lainnya.
2. Mengembangkan model yang mampu mengklasifikasi aksara dengan tanda baca.
3. Memanfaatkan pendekatan *undersampling* kelas mayoritas bersamaan dengan *oversampling* menggunakan algoritme lain.
4. Mengembangkan aplikasi berbasis android maupun *website* yang mengintegrasikan model klasifikasi ini untuk membaca sastra Lampung dan telah menerapkan konsep *User Interface* (UI) dan *User Experience* (UI).

DAFTAR PUSTAKA

Alpaydin E. 2020. *Introduction to machine learning*. Cambridge: MIT Press Academic.

Anand A. 2017. Unit 13 Image classification

Barro R.A., Sulvianti I.D., Afendi F.M. 2013. Penerapan Synthetic Minority Oversampling Technique (SMOTE) Terhadap Data Tidak Seimbang Pada Pembuatan Model Komposisi Jamu. *Xplore* 1: 1-6

Ben-Hur A., Weston J. 2010. A User's Guide to Support Vector Machines. *Methods in molecular biology (Clifton, N.J.)* 609: 223-39

Berry M., Mohamed A. 2020. *Supervised and Unsupervised Learning for Data Science*.

Bogdanchikov A., Zhaparov M., Suliyev R. 2013. Python to learn programming. *Journal of Physics Conference Series* 423: 2027

Bojja P., Velpuri N.S.S.T., Pandala G.K., Polavarapu S.D.L.R.S., Kumari P.R. 2019. Handwritten Text Recognition using Machine Learning Techniques in Application of NLP. *International Journal of Innovative Technology and Exploring Engineering (IJITEE)* 9: 1394-97

Burkov A. 2019. *The Hundred-Page Machine Learning Book*. pp. 3. Andriy Burkov.

Chollet F. 2017. *Deep Learning with Python*. pp. 14-16. New York, NY: Manning Publications.

Cun Y.L., Jackel L.D., Boser B., Denker J.S., Graf H.P., et al. 1989. Handwritten digit recognition: applications of neural network chips and automatic learning. *IEEE Communications Magazine* 27: 41-46

He H., Ma Y. 2013. Imbalanced Learning: Foundations, Algorithms, and Applications.

Junaidi A., Vajda S., Fink G. 2011. Lampung-A new handwritten character benchmark: Database, labeling and recognition. *Int. Workshop on Multilingual OCR. ACM*

Kecman V. 2005. Support Vector Machines – An Introduction, pp. 605-05. Cambridge: MIT Press

Le H., Le T., Tran S., Tran H., Thuy N. 2012. Image Classification using Support Vector Machine and Artificial Neural Network. *International Journal of Information Technology and Computer Science* 4

Libbrecht M., Noble W. 2015. Machine learning applications in genetics and genomics. *Nature reviews. Genetics* 16

Mukherjee M., Khushi M. 2021. SMOTE-ENC: A Novel SMOTE-Based Method to Generate Synthetic Data for Nominal and Continuous Features. *Applied System Innovation* 4: 18

Nguyen H., Cooper E., Kamei K. 2011. Borderline over-sampling for imbalanced data classification. *International Journal of Knowledge Engineering and Soft Data Paradigms* 3: 4-21

Pal M., Foody G. 2010. Feature Selection for Classification of Hyperspectral Data by SVM. *Geoscience and Remote Sensing, IEEE Transactions on* 48: 2297-307

Ruuska S., Hamalainen W., Kajava S., Mughal M., Matilainen P., Mononen J. 2018. Evaluation of the confusion matrix method in the validation of an automated system for measuring feeding behaviour of cattle. *Behav Processes* 148: 56-62

Xie L. 2010. SAS implementation of Kernel PCA. https://programmingsas.wordpress.com/2010/02/06/sas-implementation-of-kernel-pca/. Diakses pada 28 Februari 2022 pukul 12.54

Xuebing Y., Kuang Q., Zhang W., Zhang G. 2017. AMDO: an Over-Sampling Technique for Multi-Class Imbalanced Problems. *IEEE Transactions on Knowledge and Data Engineering* PP: 1-1