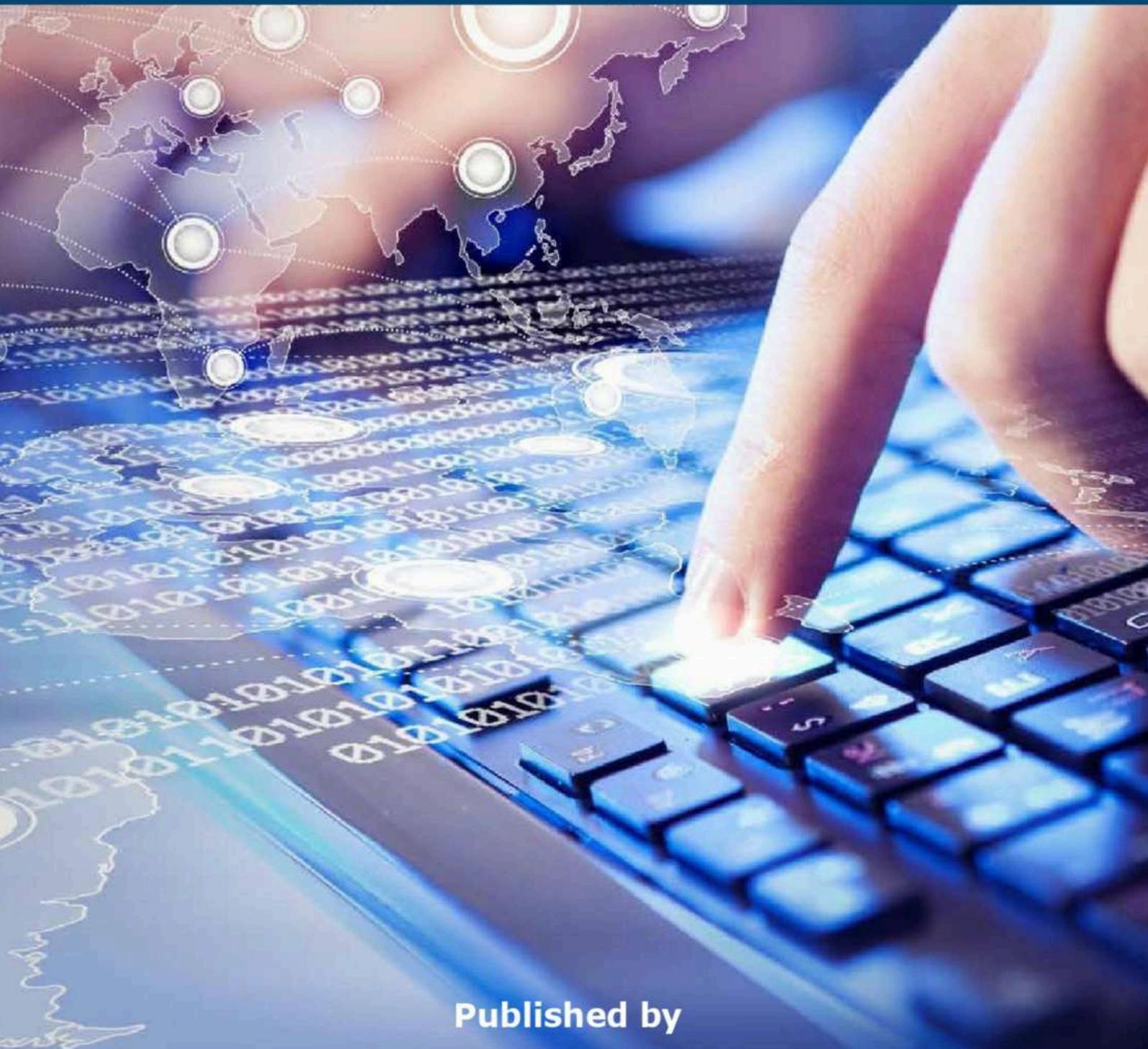


ISSN 2477-0043
E-ISSN 2460-7908

Jurnal Teknologi Terpadu

Volume 10 No. 1, Juli 2024



Published by
LPPM STT TERPADU NURUL FIKRI

Jurnal Teknologi Terpadu

Jurnal Teknologi Terpadu memuat jurnal ilmiah di bidang Ilmu Komputer, Sistem Informasi dan Teknik Informatika. Jurnal Teknologi Terpadu diterbitkan oleh LPPM STT Nurul Fikri dengan periode dua kali dalam setahun, yakni pada bulan Juli dan Desember. Jurnal Teknologi Terpadu telah terakreditasi nasional Sinta 4 sesuai dengan Surat Keputusan No. 5/E/KPT/2022 tanggal 7 Desember 2022 yang dikeluarkan oleh Direktorat Jenderal Pendidikan Tinggi, Riset, dan Teknologi.

Ketua Penyunting (Editor-in-chief)

Drs. Rusmanto, M.M,
Sistem Informasi,
STT Terpadu Nurul Fikri

Anggota Penyunting (Managing Editor)

Yekti Wirani, S.T., M.T.I.,
Sistem Informasi,
STT Terpadu Nurul Fikri

Dewan Penyunting (Editorial Board Member)

Dr. Yan Riyanto,
Lembaga Ilmu Pengetahuan Indonesia
Indonesia

Dr. Lukman Rosyidi, S.T., M.M., M.T.,
Teknik Informatika,
STT Terpadu Nurul Fikri

Dr. Amalia Rahmah, S.T., M.T.,
Sistem Informasi,
STT Terpadu Nurul Fikri

Dr. Sigit Puspito Wigati Jarot,
Teknik Informatika,
STT Terpadu Nurul Fikri

Pristi Sukmasetya, S.Komp., M.kom.,
Universitas Muhammadiyah Magelang
Indonesia

Mitra Bestari (Reviewer)

Indra Hermawan, S.Kom., M.Kom,
Politeknik Negeri Jakarta, Indonesia

Dr. Sirojul Munir, S.Si., M.Kom.,
STT Terpadu Nurul Fikri, Indonesia

Tirsa Ninia Lina, S.Kom., M.Cs.,
Universitas Victory Sorong, Indonesia

Oman Somantri, S.Kom., M.Kom.,
Politeknik Negeri Cilacap, Indonesia

Kelik Sussolaikah, S.Kom., M.Kom.,
Universitas PGRI Madiun, Indonesia

Candro Kartiko, S.Kom., M.T.I.,
Institut Teknologi Telkom Purwokerto,
Indonesia

Matheus Supriyanto Rumetna, S.Kom., M.Cs.,
Universitas Victory Sorong, Indonesia

Tifani Nabarian, S.Kom., M.T.I.,
STT Terpadu Nurul Fikri, Indonesia

Ninik Sri Lestari, S.T., M.Kom.,
STT Mandala, Indonesia

Afif Zuhri Arfianto, S.T, M.T.,
Politeknik Perkapalan Negeri Surabaya,
Indonesia

Edy Victor Haryanto, M.Kom.,
Universitas Potensi Utama, Indonesia

Rismayani, S.Kom., M.T.,
STMIK Dipanegara Makassar, Indonesia

Arnisa Stefanie, S.T., M.T.,
Universitas Singaperbangsa Karawang,
Indonesia

Taufik Hidayat, S.Kom., M.T.,
Universitas Wiralodra, Indonesia

Rahmad Hidayat, S.T., M.T.,
Sekolah Tinggi Teknologi Mandala,
Indonesia

Ahmad Rio Adriansyah, S.Si., M.Si.
STT Terpadu Nurul Fikri, Indonesia

Joko Kuswanto, M.Kom.
Universitas Baturaja, Indonesia

Penyunting Pelaksana (Assistant Editors)

Nurul Janah, S.IIP, M.Hum.,
STT Terpadu Nurul Fikri

Muh Syaiful Romadhon, S.Kom.,
STT Terpadu Nurul Fikri

Miftahussa'adah Putri Siddiq, S.Kom.,
STT Terpadu Nurul Fikri

Hilmia Zahra, S.T.,
STT Terpadu Nurul Fikri

Jurnal Teknologi Terpadu telah terindeks oleh Google Scholar, Index Copernicus International, Garuda, Neliti, dan Sinta. Tanggung jawab isi artikel berada di penulis bukan pada penerbit atau editor.

Diterbitkan oleh:

LPPM STT Terpadu Nurul Fikri

Alamat Redaksi dan Distribusi:

Kampus B STT Terpadu Nurul Fikri Lantai 3

Jl. Lenteng Agung Raya 20, Jakarta Selatan, DKI Jakarta, 12640

Telp. 021 – 786 3191 dan WhatsApp. 0851 7444 3360

Email: journal@nurulfikri.ac.id

Website: <https://journal.nurulfikri.ac.id/index.php/jtt/> dan lppm.nurulfikri.ac.id

Daftar Isi

Klasifikasi Penyakit Daun Pisang menggunakan <i>Convolutional Neural Network (CNN)</i>	1
M. Duta Pratama, Rendra Gustriansyah, Evi Purnamasari	
Implementasi Metode <i>Hybrid Filtering Technique</i> pada Penentuan <i>Rating</i> Pestisida	7
Ardimansyah, Husain, Herlinda, Kasmawaru, Nurdiansah, Marsa	
Analisis Prediksi Kata Kunci Situs Web MonsterMAC dengan Metode <i>Long Short-Term Memory (LSTM)</i>	14
Fityan Hanif Assalmi, Wahyu Syaifullah Jauharis Saputra, Amri Muhaimin	
Optimasi Parameter DBSCAN menggunakan Metode <i>Differential Evolution</i> untuk Deteksi Anomali pada Data Transaksi Bank	22
Rifqi Karunia Ibadirachman, Yulison Herry Chrisnanto, Puspita Nurul Sabrina	
Penerapan Metode <i>Combined Compromise Solution (CoCoSo)</i> dalam Pemilihan <i>Franchise Minuman</i>	32
Julita Marito, Wahyuni Betris Nainggolan, Gede Surya Mahendra	
Perbandingan Klasifikasi Label Tunggal untuk Soal Ujian Fisika menggunakan <i>Naïve Bayes</i> dan <i>K-Fold Cross Validation</i>	40
Christopher Kevin Herijanto, Yulia Wahyuningsih	
Implementasi Bi-LSTM dengan Ekstraksi Fitur Word2Vec untuk Pengembangan Analisis Sentimen Aplikasi Identitas Kependudukan Digital	46
Romario Onsu, Daniel Febrian Sengkey, Feisy Diane Kambey	
Deteksi Citra Daun untuk Klasifikasi Penyakit Padi menggunakan Pendekatan <i>Deep Learning</i> dengan Model CNN	56
Muhammad Rijal, Andi Muhammad Yani, Abdul Rahman	
Pengaruh Jarak Objek Citra pada Model Deteksi dan Klasifikasi Botol Plastik menggunakan YOLO	63
Nurvelly Rosanti, Retnani Latifah, Sirojul Munir, Izzuddin Al Qossam Maududi	
Pemanfaatan Data Ulasan Pengguna untuk Membangun Sistem Klasterisasi berdasarkan <i>Pain Points</i> menggunakan Algoritma <i>K-Means</i>	70
Ikhyia Ulummuddin, Anggraini Puspita Sari, Made Hanindia Prami Swari	



KLASIFIKASI PENYAKIT DAUN PISANG MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK (CNN)

M. Duta Pratama¹, Rendra Gustriansyah², Evi Purnamasari³

^{1,2,3}Fakultas Ilmu Komputer dan Sains, Universitas Indo Global Mandiri
Palembang, Sumatera Selatan, Indonesia 30129
2020110033@students.uigm.ac.id, rendra@uigm.ac.id, evi.ps@uigm.ac.id

Abstract

Bananas are a fruit with promising economic value in Indonesia. They are an essential commodity for farmers, but diseases affecting banana plants can harm their livelihoods. Banana diseases initially attack the leaves, and in the early stages, they are difficult to differentiate with the naked eye due to farmers' limited knowledge of pathogens. This research utilized the Convolutional Neural Network (CNN) method with transfer learning assistance using Google Colab to facilitate the classification of banana leaf diseases. The trained model experienced overfitting, so regularization was applied using dropout. The best model achieved an accuracy of 92%, precision of 92%, sensitivity of 91%, and an F1-score of 91% at a 70:20:10 ratio on epoch 80, as evaluated and validated using a confusion matrix. This study produced a reliable model for classifying banana leaf disease.

Keywords: Banana leaf disease, classification, CNN, dropout, transfer learning

Abstrak

Pisang merupakan buah yang memiliki nilai ekonomi yang menjanjikan di Indonesia. Pisang merupakan komoditas yang penting bagi para petani dan penyakit yang menyerang pisang bisa merugikan petani. Penyakit pada pisang awalnya akan menyerang daun pisang, tahap awal pada penyakit susah untuk dibedakan menggunakan mata telanjang karena kurangnya pengetahuan petani akan *pathogen*. Penelitian ini menggunakan metode *Convolutional Neural Network* (CNN) dengan bantuan *transfer learning* menggunakan *Google Colab* dengan tujuan untuk memudahkan klasifikasi penyakit daun pisang. Model yang sudah dilatih mengalami *overfitting* sehingga dilakukan regularisasi menggunakan *dropout*. Hasil model terbaik didapatkan dengan rasio 70:20:10 pada *epoch* 80 dengan akurasi 92%, presisi 92%, sensitivitas 91%, dan *f1-score* 91% yang diuji dan divalidasi menggunakan *confusion matrix*. Penelitian ini menghasilkan model yang mampu mengklasifikasi penyakit daun pisang dengan baik.

Kata kunci: CNN, *dropout*, klasifikasi, penyakit daun pisang, *transfer learning*

1. PENDAHULUAN

Pisang adalah salah satu buah yang dibudidayakan secara luas di berbagai negara tropis dan subtropis [1]. Informasi yang dimuat pada Badan Pusat Statistik (BPS) di Indonesia tahun 2022 menjelaskan total produksi pisang mencapai 9,2 ton dan merupakan buah dengan produksi paling banyak di Indonesia. Pisang merupakan komoditas penting di Indonesia, namun penyakit pada pisang sering menginfeksi yang menyebabkan kerugian bagi petani. Ada beberapa jenis penyakit pada buah pisang yang sering dijumpai di Indonesia seperti bercak daun *cordana*, gugur daun *pestaliopsis*, *sigatoka*, dan layu jamur dan bakteri.

Petani di Indonesia menghadapi masalah dalam mengidentifikasi jenis penyakit pada daun pisang karena

kurangnya pengetahuan tentang jenis *pathogen* yang menyerang pisang sehingga menyebabkan kesalahan dalam penanganan penyakit. Petani di Indonesia menghadapi masalah dalam mengidentifikasi jenis penyakit pada daun pisang karena kurangnya pengetahuan tentang jenis *pathogen* yang menyerang pisang sehingga menyebabkan kesalahan dalam penanganan penyakit yang membuat pertumbuhan pisang melambat atau gagal sehingga menyebabkan kerugian pada petani akibat banyak pisang yang gagal panen [2] sehingga diperlukan sebuah sistem untuk mengklasifikasi penyakit pisang. *Convolutional Neural Network* (CNN) dipilih untuk penelitian klasifikasi penyakit pada daun pisang yang dapat membantu petani. Penelitian terkait klasifikasi daun pisang diterapkan

arsitektur VGG19 dalam CNN pada penelitian ini untuk melihat apakah hasil klasifikasi akan lebih optimal.

Penelitian terdahulu terkait klasifikasi pernah dilakukan seperti klasifikasi kematangan buah pisang menggunakan CNN sederhana [3] dengan akurasi 92%, penelitian lain untuk klasifikasi penyakit daun pisang menggunakan CNN *mobile* dengan akurasi 95,13% [4]. Penelitian tentang klasifikasi juga dilakukan dengan objek berbeda seperti tomat, makanan tradisional dan masker [5], [6], [7]. Penelitian untuk klasifikasi menggunakan metode lain juga pernah dilakukan seperti metode *Support Vector Machine* [8], *Naïve Bayes* [9], *Decision tree* [10], *K-means* [11], dan KNN [12]. Penelitian dilakukan menggunakan CNN karena metode CNN memiliki kedalaman jaringan yang tinggi dan banyak diaplikasikan pada data citra [13], [14]. Penelitian juga dilakukan menggunakan *transfer learning* dan arsitektur *Visual Geometry Group* (VGG) lebih tepatnya VGG19 karena memiliki akurasi tinggi untuk masalah identifikasi citra [15]. Penelitian ini menggunakan CNN dengan VGG19 sebagai arsitektur karena CNN cocok untuk mengolah data citra dan VGG19 memiliki akurasi yang tinggi untuk identifikasi citra [15]. Penelitian ini juga menambahkan *transfer learning* untuk menaikkan lagi akurasi pada model CNN.

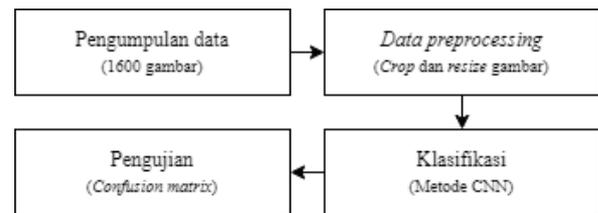
Klasifikasi merupakan kegiatan untuk menyusun sesuatu berdasarkan ciri dan sifat-sifatnya [16]. Klasifikasi menggunakan komputer dilakukan menggunakan *deep learning* dengan mencari kesamaan ciri pada suatu gambar, *deep learning* dapat mengekstraksi fitur dan pola yang akan dikelompokkan. Klasifikasi pada penyakit daun pisang dapat diambil dari ciri bentuk, warna dan tekstur pada daun pisang.

Tanaman buah pisang memiliki potensi serangan hama yang tinggi pada daerah tropis. Petani di Indonesia saat ini masih belum *familiar* dengan gejala-gejala penyakit yang menyerang pisang. Penyakit yang menyerang pisang ini menyebabkan kerugian karena produksi pisang yang menurun bahkan berhenti [2]. Penyakit berbahaya yang sering menyerang adalah *cordana*, *pestalotiopsis*, dan *sigatoka*.

CNN merupakan salah satu algoritma pada *deep learning* yang merupakan versi lebih baik dari jaringan saraf tiruan. CNN dikembangkan dengan tujuan untuk mengolah data dua dimensi seperti data citra [13]. CNN populer digunakan karena menggunakan kedalaman jaringan yang tinggi untuk mengelola data citra.

2. METODE PENELITIAN

Tahap-tahap penelitian dijelaskan pada Gambar 1 sebagai panduan penelitian yang akan dilakukan.



Gambar 1. Alur penelitian

Penelitian dimulai dengan melakukan studi literatur terlebih dahulu untuk memahami lebih pasti dasar-dasar teori yang berkaitan pada penelitian. Masalah ditemukan dengan membaca berbagai penelitian tentang penyakit pisang di Indonesia. Tahap selanjutnya dilakukan pengumpulan data berjumlah 1600 gambar yang didapat pada *mendeley data*. Data yang telah dikumpulkan akan melalui tahap pra-proses menggunakan augmentasi data dan akan dibagi menjadi data latih dan validasi. Tahap selanjutnya adalah melakukan implementasi model CNN dengan mengatur arsitektur VGG19 melalui *Google Colab*. Tahap terakhir setelah model selesai dilatih adalah melakukan pengujian pada model dan menarik kesimpulan apakah model yang dihasilkan sudah sesuai dengan kriteria yang diinginkan.

2.1 Pengumpulan data

Sumber data pada penelitian ini diambil pada *platform mendeley data*. Data diambil dari *Bangadundu Sheikh Mujibur Rahman Agricultural University* (BSMRAU) [4]. Data terdiri dari 400 gambar masing-masing 4 kelas (*cordana*, *healthy*, *pestalotiopsis*, dan *sigatoka*) dalam format jpeg.

2.2 Data preprocessing

Data preprocessing merupakan langkah yang dilakukan untuk menyiapkan data mentah agar data yang dihasilkan dapat lebih mudah untuk dilakukan *training* dan hasil model lebih baik. *Preprocessing* pada penelitian akan dilakukan *resize* dan *crop* pada gambar untuk menonjolkan fitur gambar yang ingin diklasifikasi. Pembagian data juga dilakukan pada tahap ini, data yang dihasilkan akan dibagi menjadi data latih (*train*), data validasi (*validation*) dan data *test*. Data dibagi menjadi dua rasio, rasio pertama menggunakan 70% data latih, 20% data validasi dan 10% data *test* sedangkan rasio kedua menggunakan 80% data latih, 10% data validasi dan 10% data *test*. Contoh gambar hasil data *preprocessing* bisa dilihat pada Gambar 2.

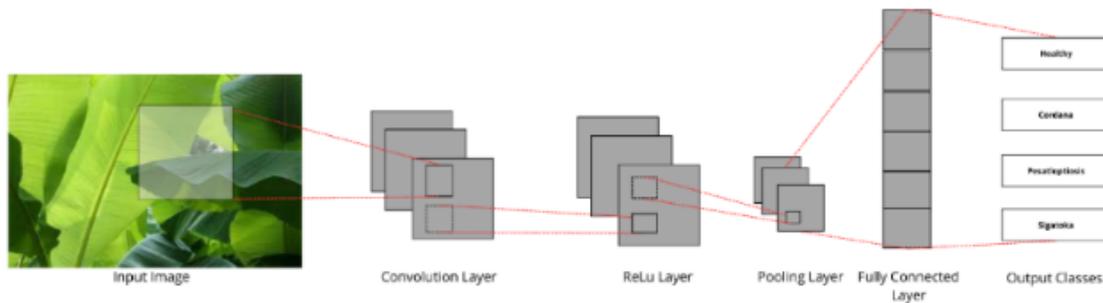


Gambar 2. Hasil *preprocessing*

2.3 Klasifikasi menggunakan CNN

CNN adalah salah satu algoritma dari *deep learning* dan merupakan versi lebih baik dari jaringan syaraf tiruan (JST) yang dirancang khusus untuk data gambar. Struktur dari CNN terinspirasi dari syaraf pada tubuh manusia lebih tepatnya syaraf otak pada manusia[17]. CNN secara penuh memanfaatkan *input* data dua dimensi seperti sebuah sinyal.

Proses ini memaksimalkan jumlah angka dalam parameter menjadi bagian-bagian yang lebih kecil yang berakibat pada simplifikasi proses *data training* dan mempercepat prosesnya[13], [18]. CNN biasanya menggunakan model serupa dengan *Multi Layer Perception* (MLP), yang tersusun dari banyak lapisan konvolusi dan *pooling* diakhiri dengan *fully connected layer*. Contoh arsitektur CNN untuk mengklasifikasi gambar bisa dilihat pada Gambar 3.



Gambar 3. Contoh CNN [13]

2.4 Pengujian

Confusion matrix merupakan metode yang bisa dipakai untuk mengetahui tingkat keakuratan suatu model. Pada *confusion matrix*, terdapat empat hasil dari perhitungan menggunakan metode ini yaitu *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN) [19]. Nilai akurasi menunjukkan kemampuan sistem untuk dapat mengklasifikasikan data dengan benar. Nilai akurasi didapat melalui perbandingan antara data yang terklarifikasi benar dan seluruh data. Nilai akurasi dapat dilihat pada persamaan 1. Nilai presisi merupakan total data positif yang diklasifikasi dengan benar lalu dibagi jumlah data positif dan dapat dilihat pada persamaan 2. Sensivitas menunjukkan ketepatan hasil positif yang diklasifikasikan oleh sistem dan dapat dilihat pada persamaan 3. *F1-score* merupakan data negatif yang diprediksi dengan benar dibagi seluruh data negatif dan dapat dilihat pada persamaan 4.

Akurasi

$$= \frac{TP + TN}{TP + TN + FP + FN} \times 100\% \quad (1)$$

Presisi

$$= \frac{TP}{TP + FP} \times 100\% \quad (2)$$

Sensivitas

$$= \frac{TP}{TP + FN} \times 100\% \quad (3)$$

F1 – Score

$$= 2 \times \frac{\text{Presisi} \times \text{Sensitivitas}}{\text{Presisi} + \text{Sensitivitas}} \quad (4)$$

Keterangan :

- TP adalah *True Positive* yaitu nilai data positif yang terklarifikasi benar.
- TN adalah *True Negative* yaitu nilai data negatif yang terklarifikasi benar.
- FP adalah *False Positive* yaitu nilai data positif yang terklarifikasi salah.
- FN adalah *False Negative* yaitu nilai data negatif yang terklarifikasi salah.

2.5 Skenario pengujian

Pada bagian ini akan diuraikan skenario pengujian yang akan dilakukan. Skenario penelitian dapat dilihat pada Tabel 1.

Tabel 1. Skenario pengujian

Skenario	Keterangan
Pembagian data	70% latih : 20% validasi : 10% uji, 80% latih : 10% validasi : 10% uji
Kelas	<i>Cordana, healthy, pestalotiopsis, sigatoka</i>
Ukuran data	224 x 224 pixel
<i>Batch size</i>	32
<i>Epochs</i>	20, 50, 80, 100

3. HASIL DAN PEMBAHASAN

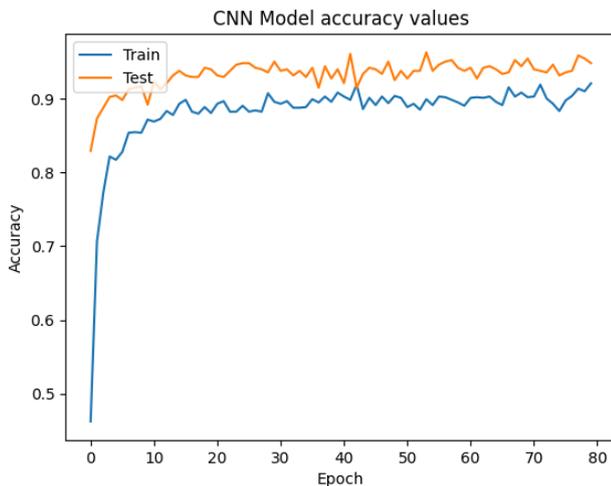
3.1 Hasil Training model

Proses selanjutnya adalah melakukan *training* pada model. Sebelum melakukan *training* pada model, ditentukan terlebih dahulu *epochs* yang ingin digunakan. Penelitian terdahulu yang menjadi acuan pada penelitian ini, tidak ada ketentuan yang pasti untuk menentukan berapa *epochs* yang diperlukan karena penentuan *epochs* bisa bervariasi tergantung jumlah data dan arsitektur yang digunakan. Tabel 2 menyajikan percobaan *training* model dengan beberapa rasio pembagian data dan *epochs*.

Tabel 2 Percobaan model

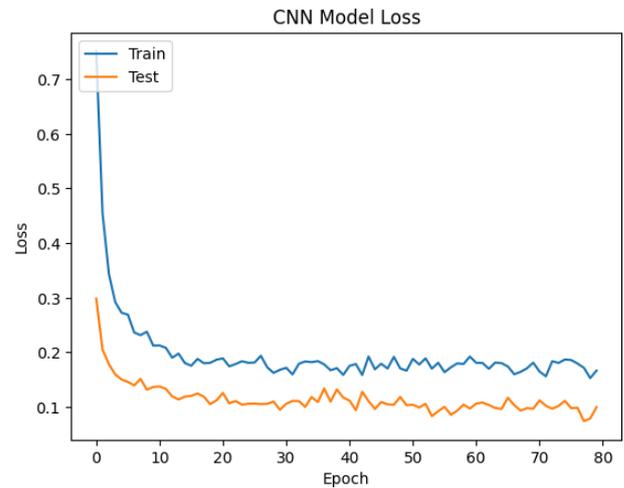
Percobaan ke	Rasio	Epochs	Latih		Validasi	
			Akurasi	Loss	Akurasi	Loss
1	70:30	20	0,86	0,19	0,91	0,11
2	70:30	50	0,90	0,15	0,93	0,11
3	70:30	80	0,92	0,16	0,94	0,10
4	70:30	100	0,90	0,18	0,93	0,11
5	80:20	20	0,87	0,18	0,94	0,09
6	80:20	50	0,90	0,17	0,93	0,10
7	80:20	80	0,89	0,18	0,91	0,12
8	80:20	100	0,90	0,18	0,93	0,07

Berbagai percobaan *training* pada model dipilih model pada percobaan ke-3 yang menghasilkan akurasi *training* dengan nilai 0,92 dan akurasi validasi bernilai 0,94. Hasil akurasi *training* dan validasi dapat dilihat pada Gambar 4.



Gambar 4. Akurasi training dan validasi

Ditampilkan juga *loss* pada *training* bernilai 0,16 dan *loss* validasi dengan nilai 0,10. Hasil *loss* pada *training* dan validasi dapat dilihat pada Gambar 5.

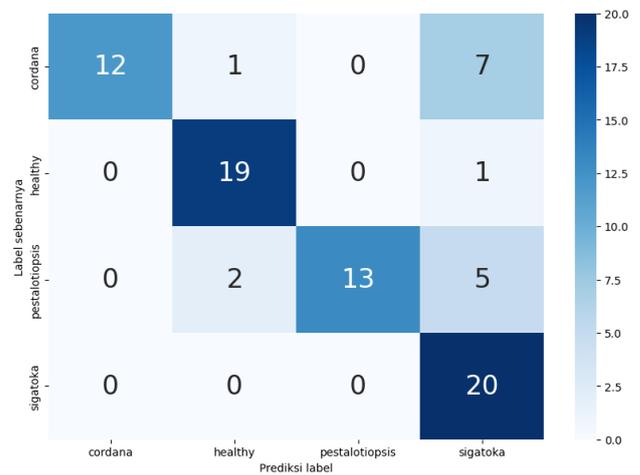


Gambar 5. Loss pada training dan validasi

Setelah mendapatkan hasil pada model. Tahap selanjutnya dalam penelitian ini adalah menguji model yang telah dibuat.

3.2 Pengujian model

Pengujian adalah tahapan yang selalu dilakukan ketika selesai membuat sistem, pengujian dilakukan untuk mengukur keberhasilan yang dicapai suatu penelitian. Pengujian pada penelitian ini dilakukan dengan *confusion matrix* pada *Google Colab* untuk mengetahui akurasi, presisi, *recall*, dan *f1-score* pada model. Model diuji menggunakan dataset *test* dengan jumlah 80 gambar yang terdiri dari 20 gambar *cordana*, 20 gambar sehat, 20 gambar *pestalotiopsis*, dan 20 gambar *sigatoka*. Hasil *confusion matrix* dapat dilihat pada Gambar 6.



Gambar 6. Confusion matrix

Menggunakan persamaan (1-4) diketahui akurasi *test* sebesar 82%, presisi sebesar 85%, *recall* sebesar 83% dan *f1-score* sebesar 82%. Secara keseluruhan model yang dihasilkan baik dan hasil *test* identik dengan hasil pelatihan dan validasi. Model yang telah dihasilkan bisa diterapkan untuk melakukan klasifikasi penyakit dengan cara mengambil gambar daun pisang dari kebun petani dan

melakukan *test* klasifikasi menggunakan model yang telah dibuat.

4. KESIMPULAN

Model yang dihasilkan telah dianggap layak digunakan karena hasil *test* identik dengan hasil *training*. Evaluasi kinerja model dilakukan dengan menilai beberapa metrik untuk mengetahui nilai presisi, *recall*, *f1-score*, dan akurasi. Presisi mengukur seberapa akurat prediksi nilai positif, *recall* mengukur kemampuan model untuk menemukan semua nilai positif yang sebenarnya, sedangkan *f1-score* adalah gambaran keseluruhan keseimbangan antara presisi dan *recall*. Model klasifikasi yang dihasilkan, penelitian ini telah sukses mengembangkan sistem klasifikasi penyakit daun pisang menggunakan metode CNN. Model dengan kinerja terbaik bisa mencapai Tingkat akurasi sebesar 82%, presisi sebesar 85%, *recall* sebesar 83% dan *f1-score* sebesar 82%. Penelitian yang telah dilakukan berhasil membuat suatu model yang bisa membantu petani mengklasifikasi penyakit pada daun pisang dan model penelitian dapat diterapkan menggunakan *drone* ataupun kamera untuk mendapat gambar daun pisang yang akan diklasifikasi.

Penelitian menemui limitasi yaitu kurangnya data yang ada untuk pelatihan model. Untuk penelitian ini bisa semakin berkembang, beberapa saran perlu ditambahkan. Pertama, *dataset* pada model bisa ditambahkan untuk memperluas cakupan variasi situasi pada daun pisang. Kelas pada model juga bisa ditambahkan dengan memasukkan penyakit lain pada daun pisang. Pengujian model juga lebih baik dilakukan pada situasi nyata pada kebun pisang dan juga lebih baik lagi jika mencoba menggunakan arsitektur lain pada model untuk mencari tahu apakah ada arsitektur dengan kinerja lebih baik untuk melakukan klasifikasi penyakit pada daun pisang.

DAFTAR PUSTAKA

- [1] I. Ryan and S. Pigai, "Morfologi Tanaman Pisang Jiigikago Berdasarkan Kearifan Lokal Suku Mee Di Kampung Idaiyo Distrik Obano Kabupaten Paniai," 2020.
- [2] F. Riana and D. Primasari, "Diagnosis Penyakit Utama Pisang karena Jamur Patogen dengan Dempster-Shafer," *Krea-TIF*, vol. 7, no. 2, p. 72, Nov. 2020, doi: 10.32832/kreatif.v7i2.2649.
- [3] A. Upadhyay, S. Singh, and S. Kanojiya, "Segregation of Ripe and Raw Bananas Using Convolutional Neural Network," in *Procedia Computer Science*, Elsevier B.V., 2022, pp. 461–468. doi: 10.1016/j.procs.2023.01.028.
- [4] M. A. B. Bhuiyan, H. M. Abdullah, S. E. Arman, S. Saminur Rahman, and K. Al Mahmud, "BananaSqueezeNet: A very fast, lightweight convolutional neural network for the diagnosis of three prominent banana leaf diseases," *Smart Agricultural Technology*, vol. 4, pp. 1–13, Aug. 2023, doi: 10.1016/j.atech.2023.100214.
- [5] R. Soekarta, N. Nurdjan, and A. Syah, "Klasifikasi Penyakit Tanaman Tomat Menggunakan Metode Convolutional Neural Network (CNN)," *INSECT*, vol. 8, no. 2, pp. 143–151, 2023.
- [6] A. Rohim and Y. Arum Sari, "Convolution Neural Network (CNN) Untuk Pengklasifikasian Citra Makanan Tradisional," 2019. [Online]. Available: <http://j-ptiik.ub.ac.id>
- [7] Y. Arvio, I. Sangadji, and K. Dewantara, "Penerapan Metode Convolution Neural Network (CNN) Dalam Proses Pengolahan Citra Untuk Mendeteksi Cacat Produksi Pada Produk Masker," vol. 16, no. 4, pp. 340–350, 2023, doi: 10.30998/faktorexacta.v16i4.20073.
- [8] R. Gustriansyah, N. Suhandi, S. Puspasari, and A. Sanmorino, "Machine Learning Method to Predict the Toddlers' Nutritional Status," vol. 16, no. 1, pp. 1–12, 2024, doi: 10.20895/INFOTEL.V15I4.988.
- [9] R. Gustriansyah, N. Suhandi, S. Puspasari, A. Sanmorino, and D. Sartika, "Toddlers' Nutritional Status Prediction Using the Multinomial Logistics Regression Method," *Journal of Computer Networks, Architecture and High Performance Computing*, vol. 6, no. 1, pp. 25–33, Dec. 2023, doi: 10.47709/cnahpc.v6i1.3372.
- [10] D. Sartika and R. Gustriansyah, "Comparison of naive Bayes and decision tree algorithms to assess the performance of Palembang City fire and Disaster management employees," *TEKNOSAINS: Jurnal Sains, Teknologi dan Informatika*, vol. 11, no. 1, pp. 132–138, Jan. 2024, doi: 10.37373/tekno.v11i1.843.
- [11] R. Gustriansyah, J. Alie, A. Sanmorino, R. Heriansyah, and M. M. Noor, "Machine Learning for Regencies-Cities Clustering Based on Inflation and Poverty Rates in Indonesia," 2022.
- [12] H. Azis, P. Purnawansyah, F. Fattah, and I. P. Putri, "Performa Klasifikasi K-NN dan Cross Validation Pada Data Pasien Pengidap Penyakit Jantung," *ILKOM Jurnal Ilmiah*, vol. 12, no. 2, pp. 81–86, Aug. 2020, doi: 10.33096/ilkom.v12i2.507.81-86.
- [13] L. Alzubaidi *et al.*, "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions," *J Big Data*, vol. 8, no. 54, pp. 1–74, Dec. 2021, doi: 10.1186/s40537-021-00444-8.
- [14] M. Zaid Romegar, C. Widya, Y. Evi, M. Dona, Theresiawati, and I. Ika Nurlaili, "Convolutional Neural Network Analysis on Handwriting Patterns

- and Its Relationship to Personality: A Systematical Review*". IEEE, 2023.
- [15] W. Setiawan, "Perbandingan Arsitektur Convolutional Neural Network Untuk Klasifikasi Fundus," *Jurnal Ilmiah SimanteC*, vol. 7, no. 2, pp. 49–54, 2019.
- [16] Ò. Lorente, I. Riera, and A. Rana, "Image Classification with Classic and Deep Learning Techniques," May 2021, [Online]. Available: <http://arxiv.org/abs/2105.04895>
- [17] Z. R. Mair and M. H. Irfani, "Permainan INGBAS (Gunting, Batu, Kertas) Menggunakan Arsitektur Convolutional Neural Network," *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 10, no. 1, pp. 1019–1026, 2023.
- [18] A. Anshu, "Review Paper on Data Mining Techniques and Applications," *International Journal of Innovative Research in Computer Science & Technology*, vol. 7, no. 2, pp. 22–26, Mar. 2019, doi: 10.21276/ijrcst.2019.7.2.4.
- [19] I. Düntsch and G. Gediga, "Confusion matrices and rough set data analysis," 2019.



IMPLEMENTASI METODE *HYBRID FILTERING TECHNIQUE* PADA PENENTUAN *RATING PESTISIDA*

Ardimansyah¹, Husain², Herlinda³, Kasmawaru⁴, Nurdiansah⁵, Marsa⁶

^{1,2,4,5,6}Teknik Informatika, Universitas Dipa Makassar

³Sistem Informasi, Universitas Dipa Makassar
Makassar, Sulawesi Selatan, Indonesia 90245

ardiman@undipa.ac.id, husain@undipa.ac.id, herlinda@undipa.ac.id, kasmawaru@undipa.ac.id,
nurdiansah@undipa.ac.id, marshaarie@undipa.ac.id

Abstract

Pesticides result from mixing organic chemicals that farmers use to protect their rice plants from disease. Farmers find it difficult to determine pesticide selection due to insufficient information. So many pesticide products are available on the market, and their various advantages make it increasingly difficult for farmers to choose pesticides suitable for certain rice diseases. This research aims to provide farmers with recommendations on determining the best pesticide to eradicate rice diseases. The wrong choice of pesticide used can harm or reduce farmers' crop yields. This research used the Hybrid Filtering Technique combined with Content Based Filtering and Collaborative Filtering methods to search for weight values and rating prediction values using price criteria, pesticide ingredients, and form (liquid, solid, powder). The results of the calculation analysis of implementing the hybrid filtering technique method for each alternative criterion can simulate a ranking to recommend the best pesticide to eradicate the causes of rice disease. This research has concluded that the rating carried out by farmers who have used pesticides influences the determination of the rating value for each pesticide product. The system test results showed that the type of pesticide with the highest rating value was the equity pesticide, with a value of 2,256.

Keywords: *Hybrid Filtering Technique, Pesticides, Rating, Recommendation System, Rice*

Abstrak

Pestisida adalah hasil pencampuran bahan kimia organik yang digunakan petani agar dapat menghindarkan tanaman padi mereka dari penyakit. Petani merasa kesulitan pada penentuan pemilihan pestisida disebabkan kurangnya informasi tentang pestisida. Produk - produk pestisida begitu banyak yang tersedia di pasaran serta berbagai kelebihan yang ditawarkan, sehingga makin membuat kesulitan para petani untuk menentukan pilihan pestisida yang cocok dengan penyakit padi tertentu. Tujuan pada penelitian ini adalah sebagai pemberi rekomendasi kepada petani dalam penentuan jenis pestisida terbaik untuk dapat memusnahkan penyakit padi. Kesalahan pemilihan pestisida yang digunakan dapat merugikan atau mengurangi hasil panen petani. Pada penelitian ini digunakan metode *Hybrid Filtering Technique* yang mengombinasikan metode *Content Based Filtering* dan *Collaborative Filtering* dengan melakukan pencarian nilai bobot dan pencarian nilai prediksi *rating* menggunakan kriteria harga, bahan pestisida, dan bentuk (cair, padat, serbuk). Hasil analisa perhitungan implementasi metode *Hybrid Filtering Technique* pada setiap kriteria dari alternatif dapat menyimulasi peringkat untuk merekomendasikan jenis pestisida yang terbaik untuk membasmi penyebab penyakit padi. Penelitian ini telah diperoleh kesimpulan bahwa pemberian *rating* yang dilakukan oleh petani yang telah menggunakan pestisida berpengaruh terhadap penentuan besarnya nilai *rating* setiap produk pestisida. Hasil pengujian sistem diperoleh jenis pestisida yang memiliki nilai *rating* tertinggi adalah pestisida equity dengan nilai *rating* 2,256.

Kata kunci: *Hybrid Filtering Technique, Padi, Pestisida, Rating, Sistem Rekomendasi*

1. PENDAHULUAN

Sektor pertanian merupakan yang sangat penting dalam pemenuhan kebutuhan pangan masyarakat. Salah satu tanaman pangan utama adalah tanaman padi, Penyebab

kerugian besar pada kurangnya hasil panen padi salah satunya adalah adanya penyakit yang menghinggapi tanaman padi. Untuk mengendalikan penyakit ini, penggunaan pestisida menjadi salah satu solusi yang umum

digunakan oleh para petani [1]. Kemajuan teknologi informasi mengalami perkembangan sangat pesat, penggunaan pestisida di dunia pertanian makin meningkat [2]. Kemajuan serta perkembangan teknologi informasi dan komunikasi yang begitu pesat dengan penggunaan teknologi komputer menjadikan pekerjaan diselesaikan dengan mudah, cepat, efektif dan efisien [3]. Perekomendasi adalah salah satu kemajuan teknologi yang disebut sebagai Sistem rekomendasi [4]. Untuk melindungi tanaman padi maka pestisida menjadi alternatif untuk memberantas atau mencegah hama-hama dan penyakit-penyakit yang merusak tanaman atau hasil-hasil pertanian.. Penentuan memilih pestisida sering kali terjadi kesalahan pilihan, untuk mengatasi kesalahan tersebut perlu dibuatkan sebuah sistem. Sistem rekomendasi merupakan cara yang tepat memberikan informasi mengenai *item* pestisida yang ampuh untuk mencegah dan memberantas hama padi [5].

Penelitian terkait mengenai penentuan *rating* telah banyak dilakukan. Berdasarkan berbagai cara pengumpulan data dengan menggunakan metode kolaboratif *filtering*, yang memanfaatkan data media sosial, *metadata* film dan memanfaatkan data pengguna [6]. Penelitian sebelumnya juga yang dilakukan oleh simbolan [1] penggunaan metode *Multi Attribute Utility Theory* untuk merekomendasikan pestisida dengan kriteria cara kerja dari pestisida, pengaruh pada tanaman padi, harga pestisida, daya tahan penyimpanan, pengaruh terhadap manusia, hewan, dan tahan terhadap cuaca/iklim. Dengan menggunakan metode *Content Based Filtering* [7] analisis penelitiannya menyatakan penentuan lahan pertanian yang memiliki potensi mendapatkan komoditi tanaman pangan dan rekomendasi adanya kesamaan *profil* antara pedagang dan petani.

Berbagai metode yang telah dikembangkan untuk menghasilkan sistem rekomendasi pestisida. Akan tetapi belum dapat memberikan rekomendasi yang sesuai harapan petani yang berlokasi di kabupaten Bulukumba khususnya wilayah Batukarope . Sehingga, penelitian ini lebih fokus pada implementasi metode *Hybrid Filtering Technique* artinya memadukan dua metode pada penentuan nilai *rating* untuk merekomendasikan jenis pestisida membasmi penyakit tanaman padi. Perekomendasi ini menggunakan metode *hybrid filtering* [8]. Metode *hybrid filtering* dapat menghasilkan *item* rekomendasi menangani masalah akurasi nilai prediksi dan nilai *sparsity* [9]. Kinerja metode *Collaborative Filtering* ada memfilter data terdapat kesamaan nilai karakteristik, sehingga memberikan layanan informasi terbaru kepada *user* [10]. Kinerja sistem akan memilah dan mengurutkan nilai peringkat *item* berdasar kesamaan profil *item* dan profil pengguna. Tujuan penelitian ini menghasilkan aplikasi web yang akan membantu para petani untuk menentukan pestisida yang diinginkan [11]. Hasil penelitian ini diharapkan mampu berkontribusi secara signifikan terhadap efektivitas dan efisiensi penggunaan

pestisida untuk membasmi penyakit tanaman padi. Informasi dari sistem rekomendasi yang akurat, memungkinkan dapat mengurangi penggunaan pestisida berlebih dan meminimalkan risiko kerugian akibat dari penyakit tanaman padi bagi petani [12]. Selain itu, penggunaan pestisida yang lebih tepat juga akan berdampak positif terhadap keberlanjutan lingkungan dan kesehatan manusia.

Berdasarkan uraian dan manfaat yang dijelaskan di atas, penelitian kami bertujuan mengimplementasikan metode *Hybrid Filtering Technique* yaitu kolaborasi metode *Content Based Filtering* dan *Collaborative Filtering*, untuk merekomendasikan jenis pestisida untuk tanaman padi. Diharapkan dapat memberikan kontribusi positif dan signifikan mengenai perekomendasi jenis pestisida yang dibutuhkan para petani untuk mengendalikan dan membasmi penyakit pada tanaman padi serta meningkatkan produktivitas pertanian secara berkelanjutan.

2. METODE PENELITIAN

Terdapat dua bagian pada penelitian ini yaitu metode analisa data serta metode perancangan sistem [13]. Dalam proses analisa data, menggunakan perpaduan metode *Content Based Filtering* dan *Collaborative Filtering*, sehingga menghasilkan metode *Hybrid Filtering Technique* . Adapun data penelitian yang digunakan bersumber dari data primer hasil wawancara yang dilakukan penulis kepada para petani dan pemilik toko penjual produk pestisida di Bulukumba sebagai obyek penelitian. Alat yang digunakan untuk mengolah data laptop (perangkat keras), sistem operasi *windows 10*, Bahasa pemrograman *PHP Native, MySql, CSS, software Ms. Word, Ms excel, visual studio code, Xampp, googel chorme* (perangkat lunak). Analisis penentuan *rating* menggunakan persamaan atau rumus *Content Based Filtering, Collaborative Filtering* , dan *Hybrid Filtering Technique*

2.1 Content Based Filtering

Cara kerja dari *Content Based Filtering* berdasarkan *item* dan deskripsinya [14]. Metode *Content Based Filtering* ini, untuk mendeskripsikan *item* digunakan kata kunci [15]. Dalam penelitian ini deskripsi *item* yang kami gunakan adalah bahan aktif, harga dan bentuk pestisida, yang digunakan. Terdapat dua formula yang akan penulis gunakan untuk metode *Content Based Filtering* yaitu pencarian nilai bobot dan pencarian nilai prediksi *rating*:

- a) Pencarian nilai bobot fitur dengan formula berikut:

$$w_{f_n} = \frac{R_{f_n}}{\sum R_{f_n}} \quad (2.1)$$

Di mana :

- w_{f_n} merupakan bobot fitur ke-n.
- R_{f_n} merupakan perkalian *rating* dengan fitur
- R_{f_n} merupakan jumlah hasil perkalian *rating* dengan fitur.

- b) Pencarian nilai prediksi dengan formula berikut:

$$R = \sum((wf_1 \times F1) + (wf_2 \times F2) + \dots + (wf_n \times F_n)) \quad (2.2)$$

Di mana:

- a. R merupakan prediksi *rating content based filtering*
- b. F merupakan nilai fitur dari suatu *item*.

2.2 Collaborative Filtering

Metode *colaborative filtering* pada penelitian ini memiliki dua formula yang akan penulis gunakan yaitu pencarian nilai *similarity* dan pencarian nilai prediksi *rating* [16].

- a) Pencarian nilai *similarity* menggunakan formula :

$$Similarity(k,l) = \frac{\sum_{u=1}^m (R_{u,k} - \bar{R}_k) (R_{u,l} - \bar{R}_l)}{\sqrt{\sum_{u=1}^m (R_{u,k} - \bar{R}_k)^2} \sqrt{\sum_{u=1}^m (R_{u,l} - \bar{R}_l)^2}} \quad (2.3)$$

Di mana :

- a. *Similarity(k,l)* merupakan nilai kemiripan *user x* juga *user y*
 - b. $R_{u,k}$ dan R_k adalah nilai *rating* pada pestisida yang sama
- b) Pencarian nilai prediksi *rating* dengan formula berikut :

$$R_{u,k} = R_k + \frac{\sum_{u=1}^m (R_{u,l} - \bar{R}_l) \times sim(k,l)}{\sum_{i=1}^n |sim(k,l)|} \quad (2.4)$$

Di mana :

- a. $R_{u,k}$ adalah nilai prediksi pada film yang akan *dirating*
- b. R_k adalah hasil nilai kemiripan
- c. *sim(k,l)* adalah nilai kemiripan antara 2 *user*
- d. $R_{u,l}$ dan R_l adalah nilai *rating* pada data

2.3 Hybrid Filtering Technique

Metode *Hybrid Filtering Technique* adalah cara kerjanya yaitu menggabungkan lebih dari satu teknik rekomendasi supaya meningkatkan performansi kinerja rekomendasi yang dapat digunakan untuk memberikan solusi pada masalah yang terjadi di metode yang digunakan [17]. Pada penelitian ini, Metode *hybrid filtering* mengkombinasikan metode *Content Based Filtering* dan *Collaborative Filtering* yang diharapkan dapat menjadi solusi pada permasalahan yang ada pada petani. Cara yang digunakan pada pengembangan sistem ini berupa pendekatan *Content Based Filtering* dengan pengukuran nilai kemiripan antara kemiripan *item* dan deskripsinya dalam hal ini bahan aktif, harga dan bentuk. Terdapat dua formula yang akan penulis gunakan untuk metode hybrid filtering yaitu pencarian nilai bobot dan pencarian nilai prediksi *rating* [18].

- a) Pencarian nilai bobot dengan formula berikut:

$$wf = \frac{1}{|I_u|} \sum_{i \in I_u} x(i, j) r(u, i) \quad (2.5)$$

Di mana :

- a. $w(u,j)$ adalah variabel bobot untuk *user u* terhadap fitur *jk*.
 - b. I_u adalah *item* yang telah di *rating* oleh *user u*.
 - c. $x(i,j)$ adalah nilai kehadiran sebuah fitur (angka 1 atau 0)
 - d. $r(u,i)$ adalah nilai *rating* yang diberikan *user u* terhadap *item i*.
- b) Pencarian nilai prediksi *rating* dengan formula berikut :

$$R_{Hybrid} = w_1 R_1 + w_2 R_2 + \dots + w_n R_n \quad (2.6)$$

Di mana :

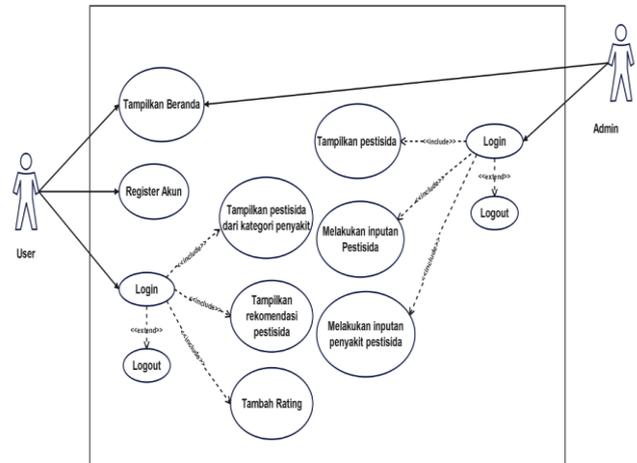
- a. R_{Hybrid} adalah nilai prediksi *rating hybrid*.
- b. w_n adalah nilai bobot prediksi *rating* dengan metode ke-n.
- c. R_n adalah nilai prediksi *rating* dengan metode ke-n.

2.4 Perancangan Sistem

Perancangan sistem pada penelitian ini dimaksudkan untuk memudahkan pembuatan sistem secara terstruktur dengan urutan-urutan fungsional sistem.

- a. *Diagram Use case*

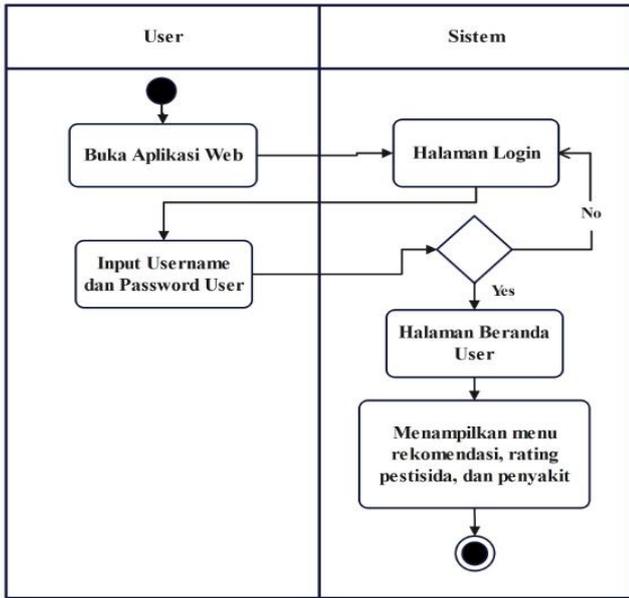
Diagram Use Case merupakan penjelasan mengenai urutan kinerja *user* dan kinerja sistem mencapai tujuan. Gambar 1 merupakan diagram *Use Case* sistem rekomendasi



Gambar 1. Diagram Use Case

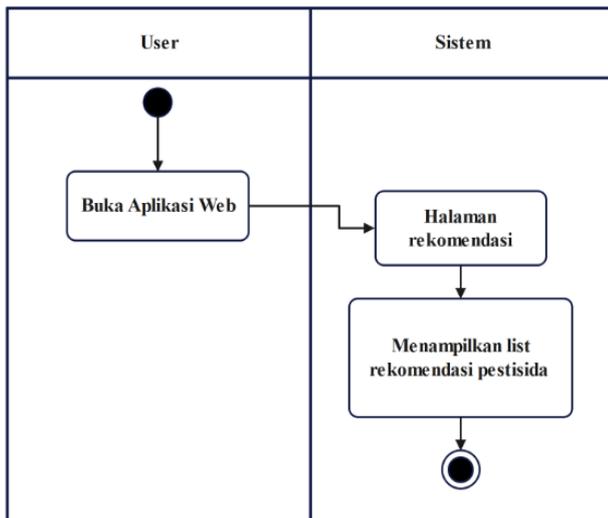
- b. *Diagram Activity*

Activity diagram merupakan keseluruhan tahapan aktivitas kerja yang direpresentasi secara grafis. secara grafis terlihat tahapan aktivitas, pemilihan tindakan, pengulangan dan hasil kinerja dari aktivitas. Penjelasan pada *Activity diagram* mengenai *user* dan sistem. Pada gambar 2 dapat dilihat *activity menu* rekomendasi.



Gambar 2. Activity Diagram Cari Rekomendasi

Selanjutnya untuk melihat *activity* diagram hasil rekomendasi dapat dilihat pada gambar 3.



Gambar 3. Activity Diagram Lihat Rekomendasi

3. HASIL DAN PEMBAHASAN

3.1 Proses Perhitungan *Content Based Filtering*

Langkah pertama yang dilakukan adalah menghitung bobot nilai dengan menggunakan rumus 2.1, di mana *user* memberikan *rating* pada jenis pestisida yang digunakan petani (jenis pestisida sebanyak 29 yaitu Enquity, Nativo, Topsin, Torbino, Biopatek, Blasgone, Filia, Kamikaze, Nopatek, Tillo, Antracol, Eco Sida, Indokor, Score, Threestar, Amistar Top, Arashi, Baro-k, Danvil, Agrept, Klensect, Kuproxat, Nordox, Plantomycin, Antilat, Biomoxa, Biowasil, Phefoc, Plenum) lalu jumlah fitur sebanyak 10, dan 14 petani memberi *rating* dengan skor *rating* 1-5. Berupa harga, bahan aktif dan bentuk. Tabel 1 Penentuan kriteria.

Tabel 1. Tabel Kriteria Pestisida

No	Kriteria	Fitur(F)
1	Harga Murah	F1
2	Sedang	F2
3	Mahal	F3
4	Bahan Aktif Sintetik	F4
5	<i>Organic Organo Khlorin</i>	F5
6	<i>Heterosiklik</i>	F6
7	<i>Karbamat</i>	F7
8	Bentuk Cair	F8
9	Butiran	F9
10	Serbuk	F10

Apabila benar maka kolom fitur dari pestisida tersebut bernilai 1 dan jika salah bernilai 0. Selanjutnya tabel 2 Nilai fitur pestisida.

Tabel 2. Tabel Fitur Pestisida

Item	Fitur (F)										Rating
	1	2	3	4	5	6	7	8	9	10	
Enquity	0	1	0	1	0	1	0	0	0	1	0
Nativo	1	0	0	1	1	1	0	0	0	1	0
Topsin	0	1	0	0	0	1	1	1	0	0	2
Torbino	0	0	1	1	1	0	0	1	0	0	4
Biopatek	1	0	0	1	1	0	0	1	0	0	2
Blasgone	1	0	0	1	0	1	0	0	1	0	5
Filia	0	0	1	1	0	0	0	1	0	0	0
Kamikaze	0	0	1	1	0	1	0	1	0	0	0
Nopatek	1	0	0	1	1	0	0	1	0	0	0
Tillo	1	0	0	1	1	0	1	1	0	0	0
Antracol	0	0	1	1	1	0	0	1	0	0	0
Eco Sida	1	0	0	1	1	0	0	0	1	0	1
Indokor	0	0	1	1	1	1	0	1	0	0	0
Score	0	0	1	0	1	1	0	1	0	0	5
Threestar	0	0	1	1	1	0	1	1	0	0	1
Amistar	0	1	0	1	0	0	0	1	0	0	0
Arashi	0	1	0	1	1	0	0	0	0	1	1
Baro-k	0	1	0	1	0	1	0	1	0	0	5
Danvil	0	1	0	1	1	1	0	1	0	0	0
Agrept	0	1	0	1	1	0	0	0	0	1	4
Klensect	0	1	0	1	1	1	0	1	0	0	0
Kuproxat	0	1	0	1	0	0	1	1	0	0	0
Nordox	0	1	0	1	1	0	0	0	0	1	5
Plantomycin	0	1	0	1	1	0	1	0	0	1	0
Antilat	0	1	0	1	1	0	0	1	0	0	1
Biomoxa	0	1	0	1	0	0	0	1	0	0	2
Biowasil	0	1	0	0	1	0	1	1	0	0	5
Phefoc	0	1	0	0	1	0	0	1	0	0	0
Plenum	1	0	0	1	1	0	0	0	1	0	0

Berdasarkan hasil perhitungan diperoleh nilai bobot setiap fitur adalah $F1 = 0,047$, $F2 = 0,146$, $F3 = 0,058$, $F4 = 0,181$, $F5 = 0,170$, $F6 = 0,099$, $F7 = 0,047$, $F8 = 0,158$, $F9 = 0,035$, $F10 = 0,058$. Selanjutnya langkah kedua, menghitung nilai prediksi *rating* dengan menggunakan rumus 2.2 dan diperoleh nilai prediksi *rating* setiap jenis penyakit yaitu Enquity = 0,485, Nativo = 0,555, Topsin = 0,450, Torbino = 0,567, Biopatek = 0,555, Blasgone = 0,363, Filia = 0,398,

Kamikaze = 0,497, Nopatek = 0,555, Tillo = 0,602, Antracol = 0,567, Eco Sida = 0,433, Indokor = 0,667, Score = 0,485, Threestar = 0,614, Amistar Top = 0,485, Arashi = 0,555, Baro-k = 0,584, Danvil = 0,754, Agrept = 0,555, Klensect = 0,754, Kuproxat = 0,532, Nordox = 0,556, Plantomycin = 0,602, Antilat = 0,655, Biomoxa = 0,485, Biowasil = 0,520, Phefoc = 0,474, Plenum 0,433.

3.2 Proses Perhitungan Collaborative Filtering

Proses *Collaborative Filtering* memiliki dua langkah perhitungan yaitu pertama, menghitung *similarity* atau nilai kemiripan antara *user x* dan *user y*. Hasil perhitungan yang diperoleh dengan menggunakan rumus 2.3 di mana telah dimasukkan hasil pilihan *user* (14 *user*) dengan 29 jenis pestisida yaitu nilai kemiripannya Enquity = 6,633, Nativo = 6,481, Topsin = 8,440, Torbino = 9,045, Biopatek = 7,394, Blasgone = 9,582, Filia = 6,914, Kamikaze = 8,552, Nopatek = 8,440, Tillo = 7,277, Antracol = 7,603, Eco Sida = 8,423, Indokor = 8,619, Score = 8,474, Threestar = 6,914, Amistar Top = 8,783, Arashi = 8,264, Baro-k = 8,304, Danvil = 7,659, Agrept = 8,018, Klensect = 7,934, Kuproxat = 8,558, Nordox = 9,502, Plantomycin = 5,707, Antilat = 8,772, Biomoxa = 7,502, Biowasil = 8,440, Phefoc = 10,128, Plenum 8,772. Selanjutnya langkah kedua dengan menghitung nilai ranting dengan penerapan rumus 2.4 diperoleh nilai prediksi *rating* Enquity = 0,338, Nativo = 0,284, Topsin = 2,283, Torbino = 2,483, Biopatek = 1,330, Blasgone = 3,806, Filia = 1,035, Kamikaze = 1,270, Nopatek = 0,550, Tillo = 0,230, Antracol = 0,966, Eco Sida = 1,509, Indokor = 1,094, Score = 2,641, Threestar = 0,854, Amistar Top = 0,555, Arashi = 0,732, Baro-k = 2,293, Danvil = 0,568, Agrept = 2,155, Klensect = 1,240, Kuproxat = 0,875, Nordox = 3,956, Plantomycin = 0,318, Antilat = 1,212, Biomoxa = 1,250, Biowasil = 4,206, Phefoc = 1,404, Plenum 0,475.

3.3 Proses Perhitungan Hybrid Filtering Technique

Perhitungan *Hybrid Filtering Technique* juga menggunakan dua langkah perhitungan yaitu langkah pertama dengan menggunakan rumus 2.5 menentukan nilai bobot formula yakni jumlah total nilai fitur dibagi dengan jumlah jenis pestisida diperoleh nilai bobot = 5,897 dan langkah kedua menghitung nilai *rating* hibrid menggunakan rumus 2.6 yaitu (nilai *rating* content * nilai bobot formula) + (nilai *rating* collaborative * nilai bobot formula) nilai *rating* selanjutnya pada tabel 3.

Tabel 3. Tabel *Rating* Pestisida

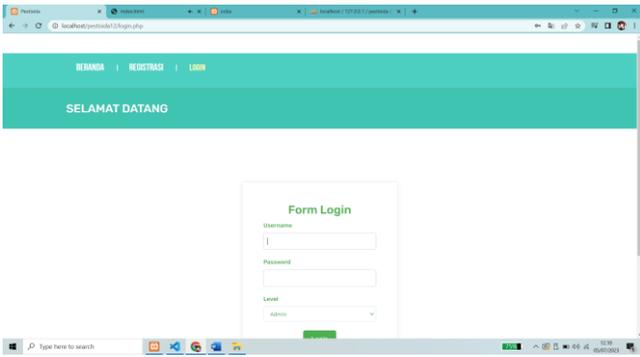
ITEM	RATING		
	CONTENT	COLLAB	HYBRID
enquity	0.485	0.338	4.853
Nativo	0.556	0.284	4.949
Topsin	0.450	2.283	16.119
Torbino	0.567	2.483	17.988

ITEM	RATING		
	CONTENT	COLLAB	HYBRID
Biopatek	0.556	1.330	11.116
Blasgone	0.363	3.806	24.579
Filia	0.398	1.035	8.448
Kamikaze	0.497	1.270	10.418
Nopatek	0.556	0.550	6.517
Tillo	0.602	0.230	4.910
Antracol	0.567	0.966	9.043
Eco Sida	0.433	1.509	11.452
Indokor	0.667	1.094	10.383
Score	0.485	2.641	18.437
Threestar	0.614	0.854	8.658
Amistar Top	0.485	0.555	6.134
Arashi	0.556	0.732	7.590
Baro-k	0.585	2.293	16.969
Danvil	0.754	0.568	7.798
Agrept	0.556	2.155	15.984
Klensect	0.754	1.240	11.759
Kuproxat	0.532	0.875	8.299
Nordox	0.556	3.956	26.604
Plantomycin	0.602	0.318	5.425
Antilat	0.655	1.212	11.006
Biomoxa	0.485	1.250	10.235
Biowasil	0.520	4.206	27.870
Phefoc	0.474	1.403	11.068

3.4 Implementasi Metode Content Based Filtering, Collaborative Filtering, dan Hybrid Filtering Technique untuk Rating Pestisida

a. Form Login

Setelah mengetikkan alamat *website* aplikasi maka *user* diminta membuat akun pada menu registrasi lalu Halaman *login* digunakan oleh *user* dan admin agar dapat masuk ke sistem dengan menggunakan akun yang telah dibuat pada menu registrasi. Di halaman ini mengisi data pengguna dan kata sandi yang telah terdaftar, setelah itu kita memilih masuk menggunakan *level user* atau admin. gambar 4. Menampilkan *form login* untuk mengakses aplikasi.



Gambar 4. Form Login

b. Halaman beranda

Halaman beranda akan menampilkan jenis pestisida pembasmi hama padi dapat dilihat pada gambar 5.



Gambar 5. Halaman Beranda

c. Menu Rekomendasi

Menu rekomendasi terlihat pada gambar 6 yang menampilkan semua pestisida dengan rating tertinggi sampai terendah yang telah diproses oleh sistem.

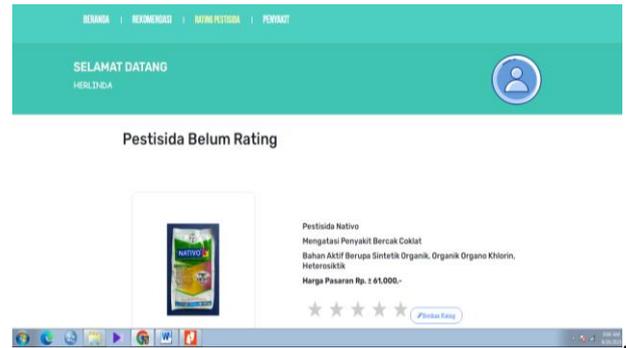


Gambar 6. Menu Rekomendasi

d. Menu Rating Pestisida

Halaman ini menampilkan semua pestisida yang belum di rating oleh user. Menu rating terlihat pada gambar 7 di mana user diminta untuk memberikan rating pada pestisida

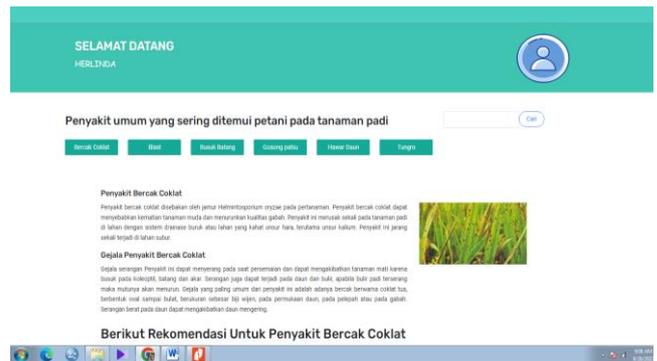
yang telah digunakan sehingga sistem dapat memproses rating tersebut, user hanya dapat memberikan penilaian sekali dengan memberi bintang 1-5 pada item pestisida



Gambar 7. Menu Rating Pestisida

e. Menu pestisida

Halaman ini menampilkan tombol menu untuk penyakit umum yang sering ditemui oleh petani. Di halaman ini setelah tombol ditekan user, sistem akan menampilkan penjelasan mengenai penyakit penyakit tersebut juga rekomendasi pestisida untuk penyakit tersebut. Tampilannya tertera pada gambar 8.



Gambar 8. Menu Pestisida

Penyakit bercak coklat terdiri dari 4 item pestisida, penyakit blast 6 item, penyakit busuk batang 5 item, penyakit gosong palsu 4 item, penyakit hawar daun 5 item, dan penyakit tungro 5 item. Sehingga total pestisida pada 6 jenis penyakit adalah 29 jenis pestisida pembasmi penyakit tanaman padi. Aplikasi akan menampilkan rating tertinggi pada setiap jenis penyakit tanaman padi.

4. KESIMPULAN

Berdasarkan hasil penelitian yang kami lakukan dapat disimpulkan bahwa Kolaborasi Metode Content Based Filtering, Collaborative Filtering, dan Hybrid Filtering Technique pada Menentukan Rating Pestisida dapat dijadikan sebagai rujukan untuk mendapatkan rekomendasi mengenai penentuan pestisida yang cocok digunakan membasmi hama tanaman padi. Dengan menggunakan metode ini dapat menentukan rating pestisida yang paling banyak digunakan petani pada jenis penyakit bercak coklat, penyakit blast, penyakit busuk batang, penyakit

gosong palsu, penyakit hawar daun, dan penyakit tungro. Hasil pengujian sistem diperoleh jenis pestisida yang memiliki nilai *rating* tertinggi adalah pestisida *enquity* dengan nilai *rating* 2,256. Saran untuk penelitian selanjut untuk melakukan kolaborasi metode *hybrid filtering* dengan metode *K-mens* pada penentuan *rating*.

Ucapan Terima Kasih

Penulis mengucapkan terima kasih kepada rekan-rekan dosen yang telah memberikan dukungan yang berharga sehingga penelitian ini dapat dilaksanakan dengan baik. Penulis juga berterima kasih kepada adik-adik mahasiswa yang telah membantu dalam pengumpulan data serta seluruh teman-teman yang membantu penelitian ini.

DAFTAR PUSTAKA

- [1] H. J. T. Simbolan Marta Etty, Saifullah, "SPK dalam merekomendasikan pestisida terbaik untuk membunuh hama pada tanaman padi menggunakan metode maut," *KOMIK*, vol. 3 no. 1, 2019.
- [2] and Y. R. B. Prasetyo, H. Haryanto, S. Astuti, E. Z. Astuti, "Implementasi Metode *Item-Based Collaborative Filtering* dalam Pemberian Rekomendasi Calon Pembeli Aksesoris Smartphone.," *Eksplorasi Inf.*, vol. 9 no., 2019.
- [3] and Y. Y. D. B. Muslimin, D. Kusmanto, K. F. Amilia, M. S. Ariffin, S. Mardiana, "Pengujian Black box pada Aplikasi Sistem Informasi Akademik Menggunakan Teknik Equivalence Partitioning," *J. Inform. Univ. Pamulang*, vol. 5 no. 1, 2020.
- [4] I. F. and A. M. S. Uyun, "*Item Collaborative Filtering* untuk Rekomendasi Pembelian Buku secara Online," *JUSI*, vol. 1 no. 1, 2018.
- [5] S. Deshpande, M.G., Muddebihalkar, A.V., Jadhav, A.B. & Kokate, "Hybrid Content-Based Filtering Recommendation Algorithm on Hadoop.," *Int. J. Adv. Res. Comput. Eng. Technol.*, vol. 5, 2018.
- [6] A. Abarja, R.A. dan Wibowo, "Movie *rating* prediction using convolutional neural network based on historical values," *Int. Journal Emerg. trends Eng. Res.*, 2020.
- [7] N. Putri, "Penerapan metode *Content Based Filtering* dalam Implementasi Sistem Rekomendasi Tanaman Pangan," *Teknika*, vol. 8 no. 1, 2019.
- [8] S. N. S. Ningrum, "Content based dan *Collaborative Filtering* pada Rekomendasi Tujuan Pariwisata Di Daerah Yogyakarta," *Telematika*, vol. 16 no. 1, 2023.
- [9] Y. I. Lubis, D. J. Napitupulu, and A. S. Dharma, "Implementasi Metode Hybrid Filtering (Collaborative dan Content-based) untuk Sistem Rekomendasi Pariwisata," *12th Conf. Inf. Technol. Electr. Eng.*, pp. 6–8, 2020.
- [10] S. and W. Rio, Oktora, "Perancangan Aplikasi E-Commerce Dengan Sistem Rekomendasi *Item-Based Collaborative Filtering*," *Manaj. Sist. Inf. Dan Teknol.*, 2018.
- [11] I. B. N. Swacita, "Pestisida dan Dampaknya Terhadap Lingkungan," *Kesehatan. Lingkung.*, 2017.
- [12] D. D. and Purwanto, "Rekomendasi Paket Pembelian Barang pada Toko Online dengan Collaborative Filing," in *Institut Teknologi Adhi Tama*, surabaya, 2019.
- [13] and S. H. Rifqy Rosyidah Ilmi, Fachrul Kurniawan, "Prediksi *Rating* Film IMDb Menggunakan Decision Tree," *JTIK*, vol. 10 no. 4, 2023.
- [14] A. A. P. W. E. Satria, N. Atina, M. E. Simbolon, "Spk : Algoritma Multi -Attribute Utility Theory (Maut) Padadestinasii Tujuan Wisata Lokal Di Kota Sidamanik," *Cess(Journal Comput. Eng. Syst. Sci.*, vol. 3 no. 2, 2018.
- [15] U. M. S. Februariyanti Herny, Laksono Aryo dewi, Wibiwo Jati Sasongko, "Implementasi metode *Collaborative Filtering* Untuk sistem Rekomendasi Penjualan pada Toko Mebel," *J. Khatulistiwa Inform.*, vol. IX no. 1, 2021.
- [16] I. W. J. and R. Wardoyo, "Algoritme Genetika untuk Mengurangi Galat Prediksi Metode *Item-based Collaborative Filtering* Genetic Algorithms for Reducing the Prediction Error of *Item-based Collaborative Filtering* Methods," *J. Math. Nat. Sci.*, vol. 25 no. 2, 2019.
- [17] R. Vahidi Farashah, M. Etebarian, A., Azmi, "A Hybrid recommender System based-on link prediction for movie baskets analysis," *J. big data*, vol. 8 no. 1, 2021.
- [18] R. M. Veronika Susanti, Suhartiningsih Dwi Nurcahyanti, "PERKEMBANGAN PENYAKIT DAN PERTUMBUHAN LIMA VARIETAS PADI (ORYZA SATIVA L.) DENGAN SISTEM TANAM BLOK," *J. Agroteknologi Trop.*, vol. 7 no. 1, 2018.



ANALISIS PREDIKSI KATA KUNCI SITUS WEB MONSTERMAC DENGAN METODE *LONG SHORT-TERM MEMORY* (LSTM)

Fityan Hanif Assalmi¹, Wahyu Syaifullah Jauharis Saputra², Amri Muhaimin³

^{1,2,3}Sains Data, Universitas Pembangunan Nasional Veteran Jawa Timur

Surabaya, Jawa Timur, Indonesia 60294

20083010028@student.upnjatim.ac.id, wahyu.s.j.saputra.if@upnjatim.ac.id, amri.muhamin.stat@upnjatim.ac.id

Abstract

Amid increasingly fierce competition in the digital realm, many companies are striving to increase the number of visitors to their websites. One such competing company is MonsterMAC, a startup. This research aims to provide early warnings and analyze relevant keywords on the MonsterMAC website using the Long Short-Term Memory (LSTM) method. Visitor data from Google Analytics and keyword data from Google Trends for the period July 22, 2022, to July 15, 2023, have been collected and processed through several stages, such as preprocessing, model design, LSTM training, and testing, as well as visualization and interpretation of results. The modeling results show satisfactory performance, with MAE Train Real User = 0.0615, Vending Machine = 0.0218, IoT = 0.0284, Machine Learning = 0.0365, Digital Business = 0.0186, Business Intelligence = 0.0296. Furthermore, this research indicates that the number of visitors is predicted to increase but will also experience a sharp decline in the coming days. The use of the keyword "IoT" shows a significant increasing trend. Implementing the keyword "IoT" in SEO strategies has increased the number of visitors over the next seven days from 250 to 350. This research guides website owners in optimizing their content and SEO strategies to increase their visibility and competitiveness in a highly competitive digital environment. This research also emphasizes the importance of the LSTM method in keyword analysis and prediction to create more targeted SEO strategies.

Keywords: google analytics, google trends, keywords, long short-term memory (LSTM), Search Engine Optimization (SEO)

Abstrak

Dalam tengah persaingan bisnis yang semakin ketat di ranah digital, banyak perusahaan yang berlomba-lomba untuk meningkatkan jumlah pengunjung pada situs web mereka. Salah satu perusahaan yang bersaing adalah MonsterMAC, sebuah perusahaan *startup*. Penelitian ini bertujuan untuk memberikan peringatan dini dan menganalisis kata kunci yang relevan di situs web MonsterMAC menggunakan metode *Long Short-Term Memory* (LSTM). Data pengunjung dari *Google Analytics* dan data kata kunci dari *Google Trends* untuk rentang waktu 22 Juli 2022 hingga 15 Juli 2023 telah dikumpulkan dan diproses melalui beberapa tahapan, seperti *preprocessing*, perancangan model, pelatihan dan pengujian LSTM, serta visualisasi dan interpretasi hasil. Hasil pemodelan menunjukkan kinerja yang memuaskan, dengan MAE Train Real User = 0.0615, Vending Machine = 0.0218, IoT = 0.0284, Machine Learning = 0.0365, Bisnis Digital = 0.0186, Kecerdasan Bisnis = 0.0296. Selain itu, penelitian ini juga menunjukkan bahwa jumlah pengunjung diprediksi akan mengalami peningkatan, namun juga akan mengalami penurunan yang tajam dalam beberapa hari ke depan. Adapun penggunaan kata kunci "IoT" menunjukkan tren peningkatan yang signifikan. Implementasi kata kunci "IoT" dalam strategi SEO terbukti meningkatkan jumlah pengunjung selama tujuh hari berikutnya dari 250 hingga 350. Penelitian ini memberikan panduan bagi pemilik situs web untuk mengoptimalkan strategi konten dan SEO demi meningkatkan visibilitas serta daya saing mereka di dalam lingkungan digital yang sangat kompetitif. Penelitian ini juga menekankan pentingnya metode LSTM dalam analisis dan prediksi kata kunci guna menciptakan strategi SEO yang lebih tepat sasaran.

Kata kunci: google analytics, google trends, kata kunci, long short-term memory (LSTM), Search Engine Optimization (SEO)

1. PENDAHULUAN

Jurnal Transformasi digital telah mengubah cara perusahaan beroperasi dan berinteraksi dengan pelanggan. Perusahaan-perusahaan saat ini semakin mengandalkan teknologi untuk

meningkatkan efisiensi dan memperluas jangkauan mereka. Penelitian [1] bertujuan memahami sentimen pelanggan menggunakan pembelajaran mesin. Sebagai bagian dari transformasi ini, perusahaan *startup* teknologi seperti

MonsterMAC yang berfokus pada layanan *vending machine*, memainkan peran penting dalam menghadirkan inovasi baru dan memperkaya pengalaman konsumen [2]. Dalam penelitian [3] metode *Long Short-Term Memory* (LSTM) digunakan untuk menganalisis dan memprediksi tren COVID-19 berdasarkan data dari 22 Januari 2020 hingga 12 Februari 2022, menghasilkan hasil yang memuaskan. Oleh karena itu, peneliti saat ini berusaha menerapkan metode LSTM pada *dataset* yang lebih pendek, yaitu satu tahun, untuk kasus yang berbeda, yaitu pola kunjungan pengguna MonsterMAC.

Bagaimana penggunaan Metode *Long Short-Term Memory* (LSTM) dapat membantu MonsterMAC meningkatkan lalu lintas situs webnya, apa saja faktor-faktor yang mempengaruhi visibilitas situs web MonsterMAC, bagaimana cara mengatasinya dan bagaimana strategi SEO yang dapat diterapkan berdasarkan hasil analisis pola kunjungan pengguna menggunakan LSTM.

Penelitian sebelumnya yang dilakukan oleh [4] menunjukkan bahwa peringatan dini menggunakan LSTM dapat digunakan untuk mengambil keputusan yang akurat. Hasil penelitian tersebut sesuai dengan tujuan penelitian ini yaitu memberikan peringatan dini kepada MonsterMAC dan menganalisis serta memprediksi kata kunci yang paling relevan pada situs web MonsterMAC menggunakan Metode *Long Short-Term Memory* (LSTM). Selain itu, penelitian ini bertujuan untuk menunjukkan bahwa metode LSTM dapat digunakan untuk memprediksi data deret waktu dalam jangka waktu satu tahun dan memberikan rekomendasi strategi SEO yang lebih tepat sasaran berdasarkan hasil analisis.

Penelitian [5] bertujuan untuk memprediksi 7 hari ke depan, serupa dengan penelitian ini yang juga berfokus pada peramalan 7 hari ke depan, namun berbeda dalam hal data dan metode yang digunakan. Penelitian ini menitikberatkan pada analisis dan prediksi pola kunjungan pengguna serta kata kunci yang sering digunakan pada situs web MonsterMAC menggunakan Metode *Long Short-Term Memory* (LSTM). Pembatasan penelitian mencakup penggunaan data historis kunjungan pengguna dari periode 22 Juli 2022 hingga 22 Juli 2023 sebagai dasar analisis, tanpa mempertimbangkan faktor eksternal seperti perubahan industri atau tren pasar yang tidak langsung berhubungan dengan aktivitas *online* situs web. Selain itu, data kata kunci dari *Google Trends* untuk periode 22 Juli 2022 hingga 2023 yang sering digunakan oleh MonsterMAC meliputi *vending machine*, IoT, *machine learning*, bisnis digital, dan kecerdasan buatan.

2. METODE PENELITIAN

2.1 Metode pengumpulan data

Pengumpulan data adalah langkah awal dalam penelitian ini di mana data terkait kunjungan pengguna pada situs web MonsterMAC dikumpulkan. Sumber data utamanya berasal

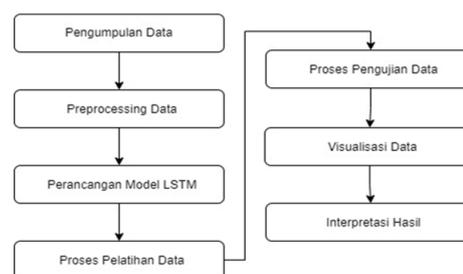
dari *google analytics* dan tujuannya adalah untuk memperoleh dataset yang mencakup informasi relevan seperti jumlah pengunjung, durasi kunjungan, dan pola perilaku pengguna lainnya. Dan untuk pengumpulan data kata kunci diperoleh dari *google trends* [6]. Berikut adalah tabel 1, yakni *Date* adalah Tanggal data tersebut didapatkan, di mana *User* adalah seseorang yang mengunjungi situs MonsterMAC beberapa kali selama periode tersebut, mereka dihitung sebagai satu pengguna, *New user* adalah pengunjung baru yang belum pernah mengunjungi situs MonsterMAC sebelumnya, *Real user* adalah jumlah dari *user* dan *new user*, *Keyword* adalah kata kunci yang digunakan pada tanggal tersebut, *Vending Machine*, IoT, *Machine Learning*, Kecerdasan Buatan Bisnis Digital adalah data kata kunci yang sering digunakan situs MonsterMAC dan akan digunakan sebagai batasan masalah untuk dianalisis

Tabel 1. Deskripsi data

No	Nama Variabel	Deskripsi variabel
1	<i>Date</i>	Tanggal data didapatkan
2	<i>User</i>	Pengunjung yang sudah pernah mengunjungi situs web MonsterMAC
3	<i>New User</i>	Pengunjung yang baru pertama kali ke situs web MonsterMAC
4	<i>Real User</i>	Jumlah dari <i>user</i> dan <i>new user</i>
5	<i>Keyword</i>	Kata kunci yang digunakan MonsterAC pada tanggal tersebut
6	<i>Vending Machine</i> , <i>Machine Learning</i> , IoT, Kecerdasan Buatan, Bisnis Digital.	Data kata kunci yang akan dianalisis

2.2 Tahapan penelitian

Secara garis besar metode penelitian adalah tahapan atau langkah kegiatan dalam penelitian sehingga data bisa diolah dan dianalisis. Tahap-tahap tersebut dapat dilihat pada gambar 1.



Gambar 1. Flowchart Penelitian

2.2.1. Pengumpulan Data

Pengumpulan data merupakan langkah awal dalam penelitian ini di mana pengumpulan data tersebut menggunakan dari dua sumber.

a) Data dari *Google Analytics* MonsterMAC

Data terkait kunjungan pengguna pada situs web MonsterMAC dikumpulkan. Sumber data berasal dari *Google Analytics* yang sudah disambungkan dengan situs web MonsterMAC. Tujuannya adalah untuk memperoleh *dataset* yang mencakup informasi relevan seperti jumlah pengunjung, durasi kunjungan, dan pola perilaku pengguna lainnya. Pengumpulan data yang komprehensif dan akurat sangat penting untuk memastikan analisis yang tepat dan hasil yang valid. Penelitian [7] memanfaatkan alat analisis web untuk manajemen pengetahuan yang efektif

b) Data kata kunci dari *Google Trends*

Data terkait pencarian kata kunci oleh seluruh pengguna *google*. Tujuannya untuk memperoleh *dataset* yang mencakup informasi berapa banyak pengguna yang mencari kata kunci pada hari tertentu. Penelitian [8] menggunakan *Google Trends* untuk menyajikan data mengenai seberapa sering kata kunci digunakan oleh pengguna dalam pencarian web di mesin pencari Google.

2.2.2 Preprocessing

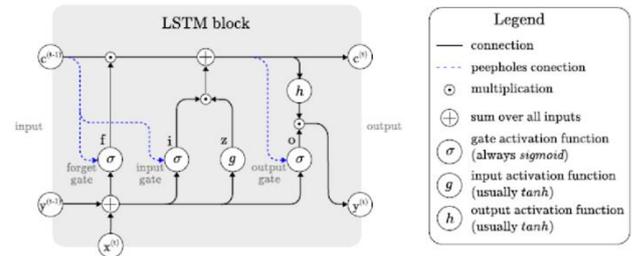
Preprocessing data melibatkan serangkaian langkah untuk membersihkan, memformat, dan mempersiapkan data sebelum digunakan untuk analisis [9]. Langkah-langkah ini mungkin mencakup penanganan nilai-nilai yang hilang, normalisasi data, atau transformasi data agar sesuai dengan persyaratan model LSTM. *Preprocessing* data yang baik akan membantu memastikan bahwa data yang digunakan untuk analisis adalah data yang berkualitas dan dapat diandalkan. [10]

2.2.3 Perancangan Model LSTM

Perancangan model LSTM melibatkan pembuatan struktur model *Long Short-Term Memory* (LSTM). Ini termasuk penentuan jumlah layer, konfigurasi *node*, dan pengaturan parameter lainnya. Model LSTM digunakan untuk memodelkan dan memahami pola dalam data deret waktu yang dihasilkan oleh kunjungan pengguna. Seperti penelitian [11] menggunakan model LSTM untuk meramalkan data seret waktu. Perancangan model yang tepat akan mempengaruhi kinerja dan akurasi model dalam memprediksi pola kunjungan pengguna.

Gambar 2 mengilustrasikan struktur pemrosesan data dalam LSTM, yang merupakan pengembangan dari algoritma RNN. Berbeda dengan arsitektur RNN standar yang terdiri dari jaringan modul berulang dengan fungsi layer sederhana, jaringan LSTM dideskripsikan sebagai beberapa modul berantai yang berulang, di mana setiap modul memiliki sub-modul yang berisi tiga fungsi gerbang

sigmoid. Fungsi *sigmoid* ini mengatur seberapa banyak informasi yang diteruskan. Terdapat tiga gerbang utama dalam sub-modul LSTM, yaitu *forget gate*, *input gate*, dan *output gate*. Penelitian [12] juga menjelaskan fungsi penting dari ketiga gerbang tersebut. Beberapa arsitektur LSTM yang dikembangkan, seperti yang terlihat pada gambar 2 biasanya memiliki tambahan satu gerbang lagi, yaitu *input modulation gate*. Secara umum, arsitektur LSTM dapat dibagi menjadi empat modul utama, yaitu *input gate* untuk membawa data masukan dari luar dan memprosesnya, *input modulation gate*, *forget gate* untuk menentukan data yang akan dihapus dan menentukan jeda waktu optimal untuk masukan berikutnya, dan *output gate* untuk memproses semua perhitungan masukan dan menghasilkan keluaran dari sel LSTM.



Gambar 2. Arsitektur LSTM

Dalam proses perhitungan LSTM, *input* data yang berbentuk *time-series* dimisalkan sebagai

$x = (x_1, x_2, \dots, x_n)$ dan *output* $y = (y_1, y_2, \dots, y_n)$ kemudian selanjutnya proses komputasi dilakukan dengan beberapa rumus sebagai berikut :

$$h_t = H (W_{hx} X_t + W_{hh} h_{t-1} + b_h) \quad (1)$$

$$P_t = W_{hy} Y_{t-1} + b_y \quad (2)$$

Di mana W adalah bobot dari matriks dan b adalah bias. Selanjutnya perhitungan *hidden state* dilakukan dengan rumus:

$$i_t = \sigma(W_{ix}x_t + W_{ih}h_{t-1} + W_{ic}c_{t-1} + b_i) \quad (3)$$

$$f_t = \sigma(W_{fx}x_t + W_{fh}h_{t-1} + W_{fc}c_{t-1} + b_f) \quad (4)$$

$$c_t = f_t * c_{t-1} + i_t * g(W_{cx}x_t + W_{ch}h_{t-1} + W_{cc}c_{t-1} + b_c) \quad (5)$$

$$o_t = \sigma(W_{ox}x_t + W_{oh}h_{t-1} + W_{oc}c_{t-1} + b_o) \quad (6)$$

$$h_t = o_t * h(c_t) \quad (7)$$

Di mana * merupakan hasil perkalian antara 2 matriks, sedangkan g dan h merupakan fungsi *sigmoid* dengan rentang nilai antara [-2,2] dan [-1,1]. σ merupakan fungsi *sigmoid* standar dengan rumus :

$$\sigma(x) = \frac{1}{1+e^x} \quad (9)$$

$$= \sum_{t=1}^n (y_t - p_t)^2 \quad (10)$$

2.2.4 Proses Pelatihan Data

Proses pelatihan data adalah langkah di mana model LSTM diberikan *dataset* untuk belajar pola dan hubungan dalam data. Dengan menggunakan metode pembelajaran mesin, model di latih dengan data historis sehingga dapat membuat prediksi yang akurat pada data yang belum pernah dilihat sebelumnya. Penelitian [13] menyarankan agar bentuk sebaran data asli tetap dipertahankan dan informasi yang ada dalam data tersebut tidak diubah. Proses pelatihan yang baik akan menghasilkan model yang lebih baik dalam memprediksi pola kunjungan pengguna di masa mendatang.

2.2.5 Proses Pengujian Data

Setelah model LSTM dilatih, proses pengujian data melibatkan penggunaan *dataset* terpisah yang tidak digunakan selama pelatihan untuk mengevaluasi kinerja model seperti yang disarankan oleh penelitian [14]. Hal ini membantu memastikan bahwa model dapat menggeneralisasi pola dengan baik pada data baru dan tidak hanya mempelajari data pelatihan. Pengujian data yang baik akan memberikan kepercayaan yang lebih besar terhadap model dalam memprediksi pola kunjungan pengguna secara akurat [15].

2.2.6 Visualisasi Data

Visualisasi data melibatkan pembuatan grafik, plot, atau visualisasi lainnya untuk menggambarkan hasil analisis dan prediksi. Dalam konteks ini, mungkin termasuk grafik waktu nyata kunjungan pengguna, perbandingan antara

prediksi model dan data aktual, atau visualisasi lainnya yang dapat membantu pemahaman dan interpretasi hasil. Visualisasi data yang baik akan memudahkan pemahaman terhadap hasil analisis dan memperkuat kesimpulan yang diambil dari data.

2.2.7 Interpretasi Hasil

Interpretasi hasil merupakan tahap penting dalam penelitian yang melibatkan proses analisis, penjelasan, dan pemberian makna pada data yang telah dikumpulkan selama penelitian. Tahap ini membantu peneliti untuk memahami implikasi temuan, mengaitkannya dengan pertanyaan penelitian, dan menyusun kesimpulan yang didasarkan pada data yang telah dianalisis. Interpretasi hasil yang baik akan memberikan wawasan yang berharga dan dapat dijadikan dasar untuk membuat keputusan atau tindakan selanjutnya.

3. HASIL DAN PEMBAHASAN

3.1 Pengumpulan Data

Tabel 2 berisi informasi mengenai kunjungan pengguna ke situs web MonsterMAC, yang didapatkan dari data *Google Analytics*. Data ini juga mencakup kata kunci yang sering digunakan dalam strategi SEO MonsterMAC, seperti '*vending machine*', 'IoT', '*machine learning*', 'bisnis digital', dan 'kecerdasan buatan'. Data kata kunci diperoleh dari *Google Trends*. Terdapat total 369 entri dengan 9 kolom, dan rentang waktu pengamatan data adalah dari 20 Juli 2022 hingga 15 Juli 2023.

Tabel 2 Data pengunjung dan data kata kunci yang sering digunakan MonsterMAC

Date	User	New User	Real User	Vending Machine	IoT	Machine Learning	Bisnis Digital	Kecerdasan Buatan
07/20/2022	104	96	200	40	69	75	46	35
07/21/2022	73	66	139	50	68	75	39	30
07/22/2022	63	61	124	40	70	75	35	25
...
07/14/2023	190	169	359	0	62	58	48	38
07/15/2023	164	150	314	0	62	54	49	39

Tabel 2 menggambarkan data harian untuk sejumlah variabel yang relevan dalam konteks bisnis atau aktivitas teknologi. Setiap baris dalam tabel mewakili satu hari, dimulai dari 20 Juli 2022 hingga 15 Juli 2023. Variabel "*User*" mencatat jumlah total pengguna yang menggunakan layanan atau produk pada hari itu, sedangkan "*New User*" mencatat jumlah pengguna baru yang menggunakan layanan atau produk pada hari tersebut. Variabel "*Real User*" mengindikasikan jumlah pengguna "*User*" dengan "*New User*" pada hari itu. Selain itu, tabel juga mencatat jumlah pencarian untuk kata kunci '*vending machine*', 'IoT', '*machine learning*', 'bisnis digital', dan 'kecerdasan buatan'

di mesin pencari *Google*. Dengan data ini, dapat dianalisis tren harian dalam penggunaan layanan, pertumbuhan pengguna baru, dan adopsi kata kunci yang sering dipakai pengguna *google*.

3.2 Preprocessing

Tahapan *preprocessing* melibatkan fungsi `str_to_datetime` untuk mengubah format string 'bulan/tanggal/tahun' menjadi objek `datetime`. Selain itu, dilakukan konversi tanggal '7/20/2022' menjadi objek `datetime`. Selanjutnya, data dalam kolom 'Date_int' dari DataFrame data diubah formatnya menjadi objek `datetime` menggunakan fungsi

str_to_datetime. Setelah itu, kolom 'Date_int' dijadikan sebagai indeks DataFrame, dan DataFrame data direset indeksnya menjadi indeks integer default. Tahapan ini mempersiapkan data untuk analisis dan pemodelan lebih lanjut. Pada gambar 3 adalah hasil *preprocessing* data yang sudah di lakukan.

```

0 2022-07-20
1 2022-07-21
2 2022-07-22
3 2022-07-23
4 2022-07-24
...
363 2023-07-18
364 2023-07-19
365 2023-07-20
366 2023-07-21
367 2023-07-22
Name: Date_int, Length: 368, dtype: datetime64[ns]
```

Gambar 3. Hasil *Preprocessing*

Selanjutnya, data dinormalisasi menggunakan metode *MinMaxScaler* dari pustaka *scikit-learn* sehingga nilai-nilai berada dalam rentang 0 hingga 1. Pertama, kelas *MinMaxScaler* diimpor dari modul *sklearn.preprocessing*. Kemudian, objek *MinMaxScaler* diinisialisasi dengan menetapkan parameter *feature_range* ke (0, 1) untuk menentukan rentang nilai setelah normalisasi. Proses normalisasi dilakukan dengan metode *fit_transform* yang diterapkan pada *dataset* *dataset_train*. Metode ini menghitung statistik yang diperlukan, seperti nilai minimum dan *maksimum*, dan kemudian menerapkan transformasi skala pada dataset sehingga semua fitur berada dalam rentang 0 hingga 1.

3.3 Perancangan Model LSTM

Penulis merancang dan mengimplementasikan model *Long Short-Term Memory* (LSTM) untuk menganalisis dan memprediksi jumlah pengguna aktual (*Real User*) pada situs web *MonsterMAC*, serta menganalisis kata kunci yang sering digunakan oleh *MonsterMAC*, seperti '*vending machine*', '*iot*', '*machine learning*', '*bisnis digital*', dan '*kecerdasan buatan*'. Proses pengembangan model dimulai dengan inisialisasi model LSTM sebagai model *sequential*, yang memungkinkan penambahan layer secara berurutan. Dua layer LSTM ditambahkan ke dalam model, masing-masing dengan 50 unit neuron. Layer pertama menggunakan *return_sequences=True* karena diikuti oleh layer LSTM lainnya, sedangkan layer kedua menggunakan *return_sequences=False* karena menjadi layer terakhir LSTM dalam rangkaian. Sebuah layer *output* dengan satu neuron ditambahkan untuk melakukan prediksi.

Model LSTM dikompilasi menggunakan *Adam optimizer* dengan *learning rate* 0.001. *Mean Squared Error* (MSE) digunakan sebagai fungsi loss, sementara *Mean Absolute Error* (MAE) dan *Mean Absolute Percentage Error* (MAPE) digunakan sebagai metrik evaluasi. Model dilatih menggunakan data pelatihan, dengan 3% dari data pelatihan

digunakan sebagai data validasi. *Batch size* sebesar 32 digunakan dan model dilatih selama 250 *epochs*.

Hasilnya, model LSTM berhasil dibangun, dikompilasi, dan dilatih untuk menganalisis dan memprediksi jumlah pengguna aktual serta kata kunci yang sering digunakan pada situs web *MonsterMAC*. Ringkasan arsitektur model menunjukkan jumlah parameter yang dapat di-*train*, memberikan gambaran tentang kompleksitas model.

3.4 Proses Pelatihan Data dan Pengujian Data

Dalam penelitian ini, data pelatihan telah dipersiapkan untuk digunakan dalam model *Long Short-Term Memory* (LSTM). Data tersebut awalnya diubah menjadi *array NumPy* untuk memungkinkan manipulasi dan pemrosesan lebih lanjut menggunakan fungsi-fungsi *NumPy*. Setelah itu, data tersebut diubah bentuk agar sesuai dengan format yang diperlukan oleh model LSTM.

Proses konversi dan *reshaping* data pelatihan menghasilkan bentuk *X_train* sebesar (348, 5, 1), menunjukkan terdapat 348 sampel dalam data pelatihan, masing-masing dengan 5 fitur atau langkah waktu, dan setiap langkah waktu hanya memiliki satu fitur. Bentuk dari *y_train* adalah (348, 1), yang berarti terdapat 348 sampel dalam data pelatihan, masing-masing dengan satu nilai target. Transformasi ini penting agar data pelatihan sesuai dengan format yang diperlukan oleh model LSTM, memungkinkan model untuk memahami urutan temporal data dan hubungan antar fitur dengan lebih efektif.

Selanjutnya, data pengujian diubah bentuk agar sesuai dengan format yang dibutuhkan oleh model LSTM. Hasilnya, bentuk dari *X_test* adalah (7, 1, 1), menunjukkan bahwa terdapat 7 sampel dalam data pengujian, setiap sampel hanya memiliki 1 fitur atau langkah waktu, dan setiap langkah waktu hanya memiliki satu fitur. Bentuk dari *y_test* adalah (7, 1), yang mengindikasikan terdapat 7 sampel dalam data pengujian, masing-masing dengan satu nilai target. Transformasi ini memastikan bahwa data pengujian memenuhi format yang dibutuhkan oleh model LSTM, memungkinkan model untuk memahami pola temporal dalam data dan memberikan prediksi yang akurat. Persiapan data pengujian dengan benar sangat penting dalam pengembangan model LSTM untuk prediksi deret waktu, karena dapat mempengaruhi kualitas hasil prediksi.

Model LSTM kemudian dikompilasi menggunakan *optimizer Adam* dan fungsi *loss MAE* (*Mean Absolute Error*). Pada tabel 3 merupakan hasil prediksi model yang didapatkan dari *MAE train*, *RMSE train*.

Tabel 3. Hasil Prediksi Model

No	Objek Penelitian	MAE Train	RMSE Train
1	<i>Real User</i>	0.0615	0.00748
2	<i>Vending Machine</i>	0.0218	0.04488

No	Objek Penelitian	MAE Train	RMSE Train
3	IoT	0.0284	0.03862
4	Machine Learning	0.0365	0.05401
5	Bisnis Digital	0.0186	0.03040
6	Kecerdasan Bisnis	0.0296	0.04205

Berdasarkan analisis performa model menggunakan *Mean Absolute Error* (MAE) dan *Root Mean Square Error* (RMSE) pada data pelatihan dengan berbagai jenis data, ditemukan variasi tingkat akurasi yang signifikan di antara *dataset* yang berbeda. Model yang diterapkan pada data Bisnis Digital menunjukkan performa terbaik dengan nilai MAE sebesar 0.0186 dan RMSE sebesar 0.03040, mengindikasikan tingkat kesalahan prediksi yang sangat rendah. Sebaliknya, model yang diterapkan pada data *Real User* memiliki nilai MAE tertinggi sebesar 0.0615, meskipun nilai RMSE-nya sangat rendah pada angka 0.00748, menunjukkan kesalahan absolut rata-rata yang lebih besar tetapi jarang menyimpang jauh dari nilai sebenarnya.

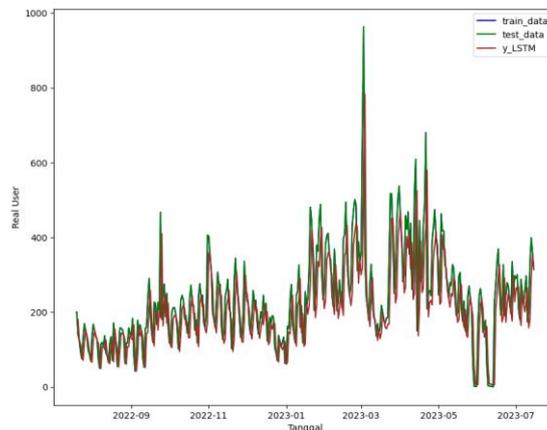
Model yang diterapkan pada data *Vending Machine* dan *Machine Learning* memiliki nilai RMSE yang lebih tinggi dibandingkan MAE, yang mengindikasikan adanya beberapa kesalahan prediksi yang signifikan. Model yang diterapkan pada data IoT dan Kecerdasan Bisnis menunjukkan performa yang cukup baik dengan nilai MAE dan RMSE yang relatif rendah, menandakan akurasi prediksi yang baik dan konsisten. Secara keseluruhan, model yang diterapkan pada data Bisnis Digital menunjukkan performa terbaik, sementara model yang diterapkan pada data *Real User* menunjukkan kesalahan absolut rata-rata terbesar, meskipun konsistensi kesalahan pada model ini memerlukan perhatian lebih lanjut.

Penelitian ini menunjukkan bahwa meskipun model yang digunakan sama, karakteristik kesalahan yang berbeda tergantung pada jenis data yang digunakan. Pemilihan *dataset* yang tepat sangat tergantung pada konteks dan kebutuhan spesifik dari aplikasi yang dimaksud. Analisis lebih lanjut diperlukan untuk memahami distribusi kesalahan dan faktor-faktor yang mempengaruhi performa model pada berbagai jenis data.

3.5 Visualisasi Data

Setelah menyelesaikan semua tahapan analisis menggunakan model *Long Short-Term Memory* (LSTM), langkah berikutnya adalah melakukan visualisasi data guna mempermudah pemahaman. Visualisasi tersebut berguna untuk meningkatkan strategi SEO (*Search Engine Optimization*) dengan memberikan pemahaman yang lebih baik mengenai tren penggunaan kata kunci dan perilaku pengunjung. Dengan pemahaman yang lebih baik ini, strategi SEO dapat dirancang lebih efektif sesuai dengan

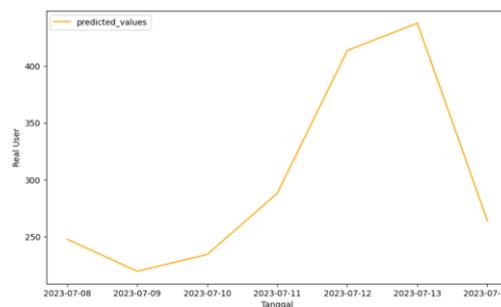
preferensi pengguna, sehingga meningkatkan visibilitas dan peringkat situs web MonsterMAC di mesin pencari. Gambar 4 menampilkan visualisasi data prediksi untuk memverifikasi kesesuaian prediksi dengan data asli.



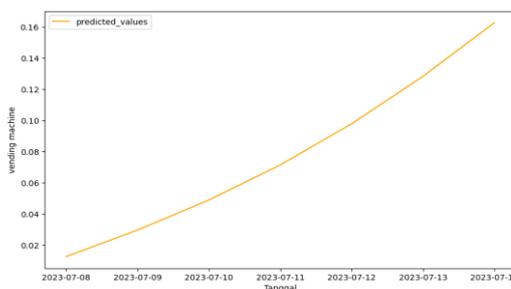
Gambar 4 Visualisasi Prediksi Data

Dari grafik tersebut, dapat disimpulkan bahwa pemodelan LSTM sudah mendekati data asli.

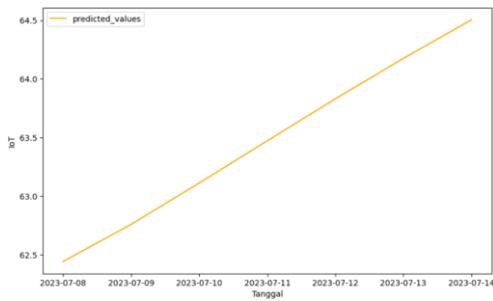
Gambar 5 menampilkan prediksi data kunjungan dan penggunaan kata kunci selama periode tujuh hari, yakni dari tanggal 8 Juli 2023 hingga 14 Juli 2023. Hasil prediksi menunjukkan bahwa jumlah kunjungan awalnya diestimasi akan meningkat, tetapi kemudian mengalami penurunan drastis dari 500 menjadi 275 pada tanggal 13 Juli hingga 14 Juli 2023. Sementara itu, penggunaan kata kunci "IoT" mengalami lonjakan signifikan dari 62 menjadi 64 selama periode yang sama. Hal ini mengindikasikan bahwa topik atau produk yang berkaitan dengan *Internet of Things* (IoT) kemungkinan akan menjadi sorotan utama atau tren dalam pencarian *online* minggu depan.



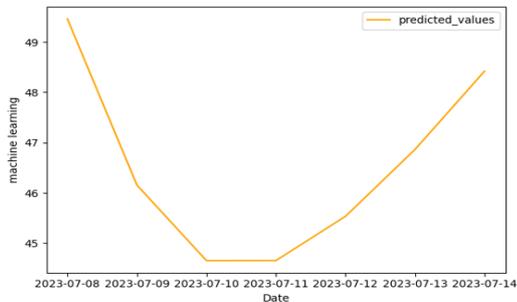
Gambar 5. (a) Visualisasi Prediksi Data Kata Kunci "Real User"



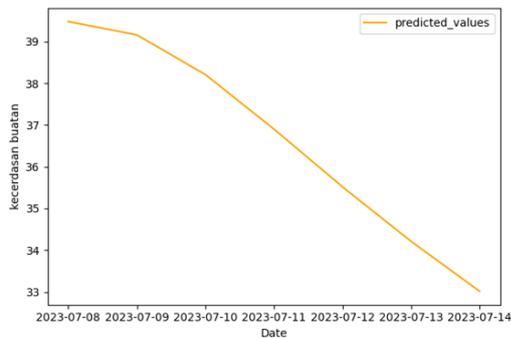
Gambar 5. (b) Visualisasi Prediksi Data Kata Kunci "vending machine"



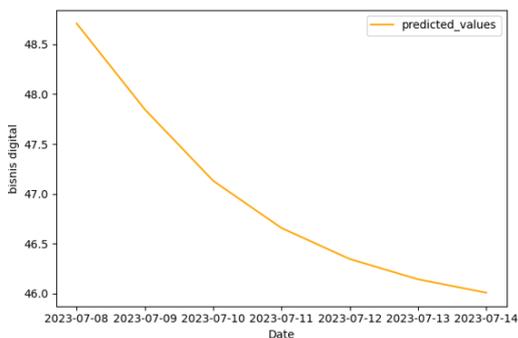
Gambar 5. (c) Visualisasi Prediksi Data Kata Kunci “IoT”



Gambar 5. (d) Visualisasi Prediksi Data Kata Kunci “machine learning”



Gambar 5. (e) Visualisasi Prediksi Data Kata Kunci “kecerdasan buatan”



Gambar 5. (f) Visualisasi Prediksi Data Kata Kunci “bisnis digital”

3.6 Pembuktian Model

Untuk menguji analisis kata kunci dengan model LSTM, peneliti memasukkan kata kunci "IoT" selama periode 7 hari sebelumnya, lalu memvisualisasikannya setelah 7 hari. Tujuannya adalah untuk mengetahui apakah prediksi peningkatan kata kunci "IoT" dalam 7 hari ke depan akan terbukti tepat. Penggunaan model LSTM dalam analisis ini memungkinkan pengamatan pola dan tren yang mungkin sulit dideteksi dengan metode analisis konvensional.

Dengan mempertimbangkan data historis, peneliti berharap dapat mengenali potensi kenaikan kata kunci "IoT" dan menggunakan informasi tersebut untuk pengambilan keputusan yang lebih baik dalam strategi pemasaran atau penelitian. Gambar 6 adalah visualisasi data Real User dari hasil pemasukan kata kunci IoT selama 7 hari di situs web MonsterMAC



Gambar 6. Visualisasi Real User

4. KESIMPULAN

Hasil penelitian menggunakan model *Long Short-Term Memory* (LSTM) menunjukkan bahwa jumlah pengunjung situs web MonsterMAC mengalami kenaikan namun kemudian turun drastis dari 500 menjadi 275 pada tanggal 13 hingga 14 Juli 2023. Sebaliknya, penggunaan kata kunci "IoT" mengalami lonjakan dari 62 menjadi 64 selama periode yang sama. Model LSTM berhasil digunakan untuk menganalisis data selama satu tahun dan menunjukkan variasi performa tergantung pada jenis data. Evaluasi model dengan *Mean Absolute Error* (MAE) dan *Root Mean Square Error* (RMSE) menunjukkan bahwa model pada data

Bisnis Digital memiliki performa terbaik dengan MAE sebesar 0.0186 dan RMSE sebesar 0.03040, sementara model pada data *Real User* memiliki nilai MAE tertinggi sebesar 0.0615, meskipun RMSE-nya rendah pada 0.00748. Kesimpulannya, model LSTM memberikan peringatan dini kepada situs web MonsterMAC untuk segera mengganti kata kunci, yang diharapkan dapat membantu dalam mengatasi penurunan pengunjung. Diharapkan hasil analisis ini dapat membantu MonsterMAC dalam meningkatkan visibilitas dan daya saing situs web mereka.

Ucapan Terima Kasih

Penulis F.H.A mengucapkan terima kasih kepada dosen – dosen Sains Data Universitas Pembangunan Nasional veteran Jawa Timur yang telah memberikan masukan untuk penulisan jurnal ini.

DAFTAR PUSTAKA.

- [1] Prabakaran.N, A.Anbarasi, N.Deepa, and Pandiaraja.P, “Enabling an On-demand Access to Community Sentiments using LSTM RNNs Web Service Architecture,” *ScienceDirect*, vol. 230, p. 584, 2023.
- [2] S. T. F. Sirojul Munir, Anifatul Aufah, Kuati Septiani, “ANALISIS DAN RANCANG BANGUN PROTOTYPE WEBMARKET PLACE

- UMKM JUARA,” *J. Teknol. Terpadu*, vol. 6, 2020, [Online]. Available: <https://journal.nurulfikri.ac.id/index.php/JTT/article/view/265/173>
- [3] S. Hansun, I. Charles, and T. Gherman, “The role of the mass vaccination programme in combating the COVID-19 pandemic: An LSTM-based analysis of COVID-19 confirmed cases,” *Heliyon*, vol. 9, no. 14397, p. 3, 2023.
- [4] Z. Lin *et al.*, “Early warning method for power supply service quality based on three-way decision theory and LSTM neural network,” *ScienceDirect*, vol. 8, p. 538, 2022, [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2352484722004917>
- [5] A. T. Damaliana, A. Muhaimin, and P. A. Riyantoko, “Peramalan Lonjakan Kasus Harian Covid-19 Di Indonesia Dengan Model Arima,” *SENADA*, vol. 3, 2023, [Online]. Available: <https://prosiding-senada.upnjatim.ac.id/index.php/senada/article/view/112>
- [6] F. T. Anggraeny, D. F. Rosida, W. S. J. Saputra, and H. Prasetyo, “Kolaborasi Pemasaran Digital Menggunakan Media Sosial dan Marketplace untuk Meningkatkan Produktivitas UMKM,” *J. Nas. Pengabd. Masy.*, vol. 2, 2021, [Online]. Available: <https://journal.jis-institute.org/index.php/jnpm/article/view/486>
- [7] T. Naprawski, “The Impact of Web Analytics Tools on Knowledge Management,” *ScienceDirect*, vol. 225, p. 3407, 2023, [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S187705092301493X>
- [8] A. Rovetta, “An integrated infoveillance approach using google trends and Talkwalker: Listening to web concerns about COVID-19 vaccines in Italy,” *ScienceDirect*, vol. 4, no. 100272, p. 2, 2023, [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2772442523001399?ref=pdf_download&fr=RR-2&rr=866fdcfb4bdf40e0
- [9] S. García and J. M. B. and F. H. Sergio Ramírez-Gallego, Julián Luengo, “Big data preprocessing: methods and prospects,” *Springer*, p. 3, 2016, doi: 10.1186/s41044-016-0014-0.
- [10] R. Maulana, A. Voutama, and T. Ridwan, “Analisis Sentimen Ulasan Aplikasi MyPertamina pada Google Play Store menggunakan Algoritma NBC,” *J. Teknol. Terpadu*, vol. 9, p. 44, 2023, [Online]. Available: <https://journal.nurulfikri.ac.id/index.php/JTT/article/view/609/306>
- [11] K. E. ArunKumar, D. V. Kalaga, C. M. S. Kumar, M. Kawaji, and T. M. Brenza, “Forecasting of COVID-19 using deep layer Recurrent Neural Networks (RNNs) with Gated Recurrent Units (GRUs) and Long Short-Term Memory (LSTM) cells,” *ScienceDirect*, vol. 146, p. 3, 2021, [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0960077921002149>
- [12] S. V. Belavadia, S. Rajagopala, R. Ra, and R. Mohan, “Air Quality Forecasting using LSTM RNN and Wireless Sensor Networks,” *ScienceDirect*, vol. 170, 2020.
- [13] K. E. ArunKumar, D. V. Kalaga, C. M. S. Kumar, Masahiro, Kawaji, and T. M. Brenza, “Comparative analysis of Gated Recurrent Units (GRU), long Short-Term memory (LSTM) cells, autoregressive Integrated moving average (ARIMA), seasonal autoregressive Integrated moving average (SARIMA) for forecasting COVID-19 trends,” *ScienceDirect*, vol. 61, 2022.
- [14] A. T. A *et al.*, “Reconstruction of missing streamflow series in human-regulated catchments using a data integration LSTM model,” *ScienceDirect*, vol. 52, 2024.
- [15] D. Fitriati, “PERBANDINGAN KINERJA CNN LeNet5 DANEXTREME LEARNING MACHINE PADA PENGENALAN CITRA TULISAN TANGAN ANGKA,” *J. Teknol. Terpadu*, vol. 2, 2016, [Online]. Available: <https://journal.nurulfikri.ac.id/index.php/JTT/article/view/45/33>



OPTIMASI PARAMETER DBSCAN MENGGUNAKAN METODE *DIFFERENTIAL EVOLUTION* UNTUK DETEKSI ANOMALI PADA DATA TRANSAKSI BANK

Rifqi Karunia Ibadirachman¹, Yulison Herry Chrisnanto², Puspita Nurul Sabrina³

^{1,2,3} Teknik Informatika, Universitas Jendral Achmad Yani
Cimahi, Jawa Barat, Indonesia 40525

rifqikarunia20@if.unjani.ac.id, yhc@if.unjani.ac.id, puspita.sabrina@lecture.unjani.ac.id

Abstract

Anomalies in bank transaction data often indicate fraudulent activity or errors. This research aims to detect anomalies in bank transaction data by optimizing DBSCAN parameters using the Differential Evolution (DE) method because there are shortcomings, namely the difficulty of determining the right parameters to create the right cluster in order to detect anomalies in bank transaction data properly. The data used is transaction data from Bank XYZ with more than 1011 data records. The research stages include data collection, data preprocessing (data cleaning, normalization, and transformation), system design, algorithm implementation, and analysis and testing using the Silhouette score and Z-score methods. The DE method is used to automatically determine the optimal parameters of MinPts and Epsilon. The results show that the use of DE can produce optimal parameters, with increased anomaly detection accuracy using DBSCAN. Evaluation with Silhouette score shows an average accuracy of 0.7916 and using DBI reaches 0.19791 at the lowest, while Z-score and MSE measurements show high cluster density with anomaly detection accuracy reaching 98.41% and 0.555537. The DE approach to parameter selection is effective in improving the performance of DBSCAN in detecting anomalies in bank transaction data. Suggestions for future research are to increase the number of data records and conduct experiments on a wider variety of data attributes.

Keywords: Anomaly, Bank Transaction Data, DBSCAN, Differential Evolution, Optimize

Abstrak

Anomali dalam data transaksi bank sering kali menandakan adanya aktivitas penipuan atau kesalahan. Penelitian ini bertujuan mendeteksi anomali pada data transaksi bank dengan optimasi parameter DBSCAN menggunakan metode *Differential Evolution* (DE) karena terdapat kekurangan yaitu sulitnya menentukan parameter yang tepat untuk membuat *cluster* yang tepat agar dapat mendeteksi anomali pada data transaksi bank dengan baik. Data yang digunakan adalah data transaksi dari Bank XYZ dengan lebih dari 1011 *record* data. Tahapan penelitian meliputi pengumpulan data, *preprocessing* data (pembersihan, normalisasi, dan transformasi data), perancangan sistem, implementasi algoritma, serta analisa dan pengujian menggunakan metode *Silhouette score* dan *Z-score*. Metode DE digunakan untuk menentukan parameter optimal MinPts dan Epsilon secara otomatis. Hasil penelitian menunjukkan bahwa penggunaan DE dapat menghasilkan parameter yang optimal, dengan peningkatan akurasi deteksi anomali menggunakan DBSCAN. Evaluasi dengan *Silhouette score* menunjukkan akurasi rata-rata 0.7916 dan menggunakan DBI mencapai 0.19791 paling rendah, sementara pengukuran *Z-score* dan MSE menunjukkan kepadatan *cluster* yang tinggi dengan akurasi deteksi anomali mencapai 98.41% dan 0.555537. Pendekatan DE untuk pemilihan parameter ini efektif dalam meningkatkan performa DBSCAN dalam mendeteksi anomali pada data transaksi bank. Saran untuk penelitian selanjutnya adalah meningkatkan jumlah *record* data dan lakukan percobaan pada variasi atribut data yang lebih beragam.

Kata kunci: Anomali, Data Transaksi Bank, DBSCAN, *Differential Evolution*, Optimasi

1. PENDAHULUAN

Anomali didefinisikan sebagai pola yang menyimpang dari perilaku yang diharapkan dari data. Anomali pada data bisa diketahui melalui beberapa teknik analisis dalam dunia teknologi. Memungkinkan pengidentifikasian kesalahan,

anomali, ketidakefektifan, ketidaksesuaian, dan bias yang sering dikaitkan dengan penipuan, dengan penekanan pada jumlah dolar tertentu yang melebihi ambang batas kontrol internal. Salah satunya anomali pada transaksi bank merujuk pada aktivitas atau pola transaksi yang tidak biasa

atau mencurigakan dan berpotensi menunjukkan adanya penipuan, kesalahan, atau kegiatan ilegal. Anomali ini dapat berupa berbagai bentuk.

Deteksi anomali dapat diterapkan pada permasalahan seperti kecurangan pada transaksi bank, penipuan untuk kartu kredit, perawatan kesehatan, asuransi, dan deteksi kesalahan dalam sistem yang sangat penting sebelum terjadi kerusakan besar[1].

Salah satu metode yang dapat digunakan dalam mendeteksi anomali data adalah metode *clustering*. Di mana metode *clustering* dapat mengidentifikasi dan Tujuannya adalah untuk membuat data dalam *cluster* yang sama lebih mirip satu sama lain dibandingkan dengan data di cluster lain. *Density-Based Spatial Clustering of Applications with Noise* (DBSCAN) adalah salah satu algoritma pengelompokan yang paling umum dan cukup banyak dikutip dalam literatur ilmiah. Algoritma DBSCAN umumnya menghubungkan kelompok sebagai area padat peristiwa dalam sampel data yang terisolasi oleh daerah dengan kepadatan rendah. DBSCAN secara alami mengidentifikasi dan mengabaikan titik-titik yang terletak di luar *cluster* yang padat, sehingga lebih *robust* terhadap *noise*. Ini membuatnya sangat efektif dalam mendeteksi anomali yang muncul sebagai *outliers* di luar area yang padat. DBSCAN juga sangat tepat digunakan pada pengelompokan data yang mempunyai atribut lokasi. Karena keunggulan algoritma DBSCAN, banyak peneliti mempertimbangkan penggunaannya untuk tujuan penelitian mereka. Sebuah algoritma unik berbasis DBSCAN untuk deteksi anomali disajikan dalam Penelitian terdahulu menggunakan metode DBSCAN untuk mendeteksi kecurangan dan anomali data pada Prediksi finansial[2], *Finger Print*[3], *Credit Card* [4], data kendaraan[5].

Namun, meskipun DBSCAN telah menunjukkan kinerja yang baik dalam banyak aplikasi, terdapat beberapa tantangan dan keterbatasan yang dihadapi oleh peneliti. Pada penelitian [2] DBSCAN digunakan untuk memprediksi *error* pada *financial stements*. Dengan hasil *Accuracy* 51.5%, *Precision* 16.4%, *Recall* 53.1%. Dengan melakukan beberapa kali *experiman* untuk menentukan parameter agar dapat membentuk *cluster* yang sesuai. Penelitian terdahulu [6], Kesulitan dalam menentukan parameter yang sesuai ini sering kali memerlukan beberapa iterasi dan penyesuaian berdasarkan uji coba dan *error* atau analisis yang lebih mendalam mengenai distribusi data yang dianalisis. Karena DBSCAN bekerja dengan baik pada data yang memiliki densitas yang seragam, namun menjadi kurang efektif ketika data memiliki variasi densitas yang tinggi. Hal ini membuat penentuan parameter yang tepat menjadi lebih kompleks.

Penelitian terdahulu [7] hasil eksperimen menunjukkan bahwa pendekatan *Modified* data dengan menambahkan data musiman per bulan dan pemilihan parameter yang baik pada DBSCAN mampu mendeteksi lebih banyak titik

anomali (4.79%) dibandingkan dengan DBSCAN standar (2.16%), menunjukkan peningkatan dalam deteksi anomali lokal dalam data musiman. Penelitian ini [8] menunjukan kemampuan DBSCAN dengan memilih parameter yang baik, dalam mendeteksi kesalahan data atau anomali pada data yang terbatas atau kurangnya data historis yang cukup untuk menganalisis dan mengidentifikasi pola anomali.

Sering kali peneliti terdahulu mengalami kesulitan saat menentukan parameter yang tepat untuk DBSCAN, sehingga banyak titik data yang seharusnya menjadi bagian dari *cluster* dianggap sebagai *noise* atau sebaliknya. Pemilihan parameter *Minimum Points* (MinPts) dan *Epsilon* (Eps) yang baik sangat penting untuk identifikasi anomali yang lebih akurat. Namun, metode konvensional dalam menentukan parameter dengan melakukan eksperimen pemilihan parameter ini masih kurang efektif dan efisien. Oleh karena itu, pemilihan parameter yang optimal menjadi salah satu tantangan utama dalam penerapan DBSCAN. Karena itu diperlukan metode untuk menentukan atau melakukan optimasi pemilihan parameter DBSCAN agar memudahkan penggunaan algoritma DBSCAN tersebut. Salah satu pendekatan yang digunakan untuk menangani sulitnya menentukan parameter DBSCAN adalah metode *Differential Evolution* [9].

Defferential Evolution (DE) adalah salah satu algoritma optimasi yang digunakan untuk menemukan solusi optimal pada masalah optimasi global. DE memungkinkan penentuan parameter yang optimal secara otomatis, meningkatkan kualitas hasil *clustering* yang dihasilkan oleh DBSCAN [10]. Penelitian lain [11] DE digunakan bersama dengan *algoritma clustering* DBSCAN untuk membentuk algoritma NCjDE-2LS yang dapat menemukan dan mengoptimalkan *multiple global optima*. Dengan menghasilkan rata-rata akurasi pada tingkat puncak 100%. Algoritma DE juga dapat digunakan untuk mengoptimalkan parameter *epsilon* dalam DBSCAN, penelitian ini [12] menggunakan DE untuk mengoptimasikan parameter Eps karena DE memiliki mekanisme yang baik untuk mengeksplorasi ruang solusi dan mengeksploitasi solusi terbaik. Ini juga penting untuk pengaturan parameter DBSCAN, yang harus menemukan keseimbangan antara kepadatan dan radius pencarian. Selain itu, DE mampu menangani optimasi pada ruang multidimensi, yang membuatnya cocok untuk menemukan kombinasi parameter terbaik.

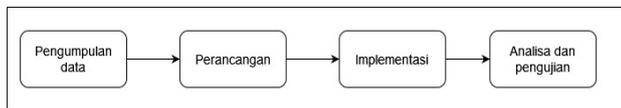
Pada penelitian terdahulu masih melakukan *experimen* menentukan parameter *MinPts* dan *Epsilon* untuk membentuk *cluster* pada DBSCAN [1], [2]. Yang menghasilkan banyak titik data yang seharusnya termasuk dalam kluster dianggap sebagai *noise* (anomali) atau sebaliknya. Dan juga DE digunakan untuk mengoptimasikan parameter Eps [6], [9], [12]. belum banyak peneliti yang menggunakan metode DE untuk membantu DBSCAN menemukan parameter terbaik untuk mendeteksi anomali pada data transaksi bank. DE juga dapat

digunakan untuk mengoptimalkan parameter MinPts dan Epsilon untuk membentuk *cluster* yang baik agar DBSCAN dapat mendeteksi adanya anomali pada data transaksi. Setelah iterasi pengetesan pemilihan parameter untuk proses DBSCAN. Hasil *cluster* akan divalidasi menggunakan metode *Silhouette score* dan *Davies-Bouldin Indeks (DBI)*. untuk mengukur keakuratan anomali yang diidentifikasi menggunakan *Z-score* sebagai pengukuran proporsional data dan menggunakan MSE pengukuran anomali hasil *cluster*.

Berdasarkan uraian tersebut pengenalan algoritma DBSCAN dan metode *Differential evolution*. Diharapkan dapat memberikan hasil yang baik untuk menentukan kedua parameter MinPts dan *Epsilon* agar dapat menghasilkan *cluster* yang baik untuk mendeteksi anomali pada data transaksi bank.

2. METODE PENELITIAN

Pada penelitian ini dilakukan beberapa tahapan penelitian yang bisa dilihat pada gambar 1.



Gambar 1. Tahapan Pengujian

Dapat dilihat dari gambar 1, ada beberapa tahapan yang harus dilewati dimulai dengan Pengumpulan data hingga pada proses Analisa dan pengujian. Berikut adalah uraian dari tahapan pada gambar 1.

2.1 Pengumpulan data

Pada tahap ini melakukan pencarian dan pengumpulan data. Yang di mana data yang diusulkan berupa data transaksi pada bank xyz. Set data yang sesuai untuk penggunaan DBSCAN dengan karakteristik berupa Numerik, Skala yang serupa dan juga relevan untuk *cluster*. Maka dari itu, diperlukan *preprocessing* untuk menyesuaikan data agar bisa diolah pada algoritma DBSCAN.

Jumlah *record* data transaksi lebih dari atau kurang dari 1011 digunakan. Objek deteksi anomali akan digunakan untuk mengidentifikasi transaksi pada data. Dengan DBSCAN, dapat menemukan pengelompokan data dan menemukan adanya data yang tidak biasa pada data transaksi.

Pada tahap *preprocessing* awal akan dilakukan tahap *data clean*, normalisasi data dan transformasi data. Di mana *preprocessing data clean* untuk menghapus *missing value* dan transformasi data untuk mengubah data yang bertipe objek menjadi numerik agar dapat diolah menggunakan algoritma DBSCAN. Berikut adalah data yang diusulkan dapat dilihat pada tabel 1.

Tabel 1. Dataset Transaksi Bank XYZ

Coloum	Tipe	Missing	min	max
TID	Object	0	null	null
CID	Object	0	null	null
CDOB	Object	0	null	null
CG	Object	0	null	null
CL	Object	0	null	null
CAB	float64	0	0	3,053,688
TD	date	0	null	null
TT	int64	0	null	null
TA (INR)	float64	0	188,846	230,002
Latitude	numeric	0	null	null
Longitude	numeric	0	null	null

Tahap pengumpulan data dilakukan dan diambil dari *dataset* yang telah digunakan oleh para peneliti sebelumnya. Data dan *Type* data yang dapat dilihat pada tabel 1. Menunjukkan beberapa kolom dengan akronim sebagai berikut pada tabel 2.

Tabel 2. Dataset nama akronim

Coloum	Name
TID	TransactionID
CID	CustomerID
CDOB	CustomerDOB
CG	CustGender
CL	CustLocation
CAB	CustAccountBalance
TD	TransactionDate
TT	TransactionTime
TA (INR)	TransactionAmount (INR)
Latitude	Latitude
Longitude	Longitude

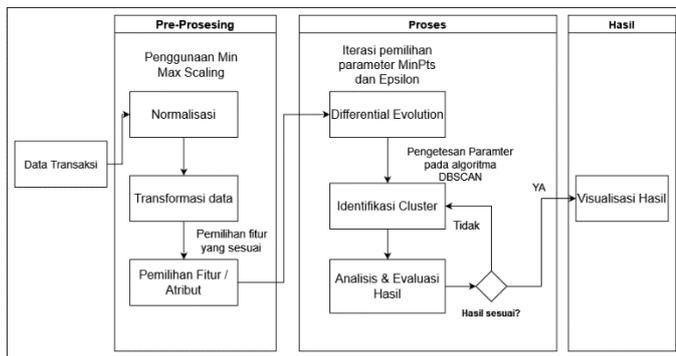
Dari data yang digunakan terdapat ± 1011 *record* data yang terdapat pada *dataset* transaksi bank. Data yang didapatkan akan melalui *preprocessing* agar dapat disesuaikan untuk penggunaan algoritma DBSCAN.

2.2 Perancangan

Pada perancangan ini melakukan tahapan-tahapan mengenai hal yang akan dilakukan pada proses *clustering* data transaksi bank. Tahapan awal yang akan dilakukan adalah *preprocessing* data yaitu normalisasi untuk

menyesuaikan data agar dapat digunakan pada algoritma DBSCAN, Selanjutnya pemilihan fitur yang di mana penyesuaian fitur yang relevan untuk *cluster*, Selanjutnya adalah pemilihan parameter menggunakan metode *Differential Evolution* yang di mana dilakukan beberapa iterasi untuk menemukan parameter yang paling baik agar menghasilkan *cluster* yang baik.

Pada tahap proses dilakukan pengetesan parameter yang telah terdefinisi oleh metode *differential evolution*, untuk menghasilkan *cluster*. Tahap ini juga menguji dan menganalisa hasil apakah akurasi dari *cluster* yang dibentuk untuk mendeteksi anomali akurat. Pada tahap terakhir menampilkan visualisasi hasil dari *cluster anomaly* data. Dapat dilihat rincian dari rancangan pada gambar 2 berikut.



Gambar 2. Tahap Perancangan

2.2.1 *Differential Evolution*

Metode *Differential Evolution* (DE) digunakan dalam bentuk *Binary Differential Evolution* (BDE), di mana solusi-solusi direpresentasikan dalam bentuk biner[11]. DE membantu dalam menentukan parameter-parameter DBSCAN dengan cara mencari kombinasi parameter yang optimal untuk mencapai hasil *clustering* yang baik. DE membantu dalam menyesuaikan nilai Eps dan MinPts secara otomatis berdasarkan evaluasi kualitas *clustering* yang dilakukan selama proses optimasi. Adapun beberapa tahapan penggunaan DE pada proses tahapan optimasi di antaranya sebagai berikut ;

- a) Inisialisasi populasi Langkah ini menciptakan populasi awal secara acak dalam ruang n dimensi dengan mempertimbangkan batasan atas dan bawah untuk setiap variabel. bisa dilihat pada persamaan 2.1

$$X_iG = L + r * (U - L) \tag{2.1}$$

di mana:

- L dan U adalah batas bawah dan atas dari ruang pencarian.
- r adalah vektor acak yang nilainya antara 0 dan 1.
- i adalah indeks individu.
- G adalah indeks generasi.

- b) Mutasi DE secara acak memilih dua vektor populasi yang berbeda dan menggunakan perbedaan antara individu-individu tersebut untuk memutasi individu target. Bisa dilihat pada persamaan 2.2

$$V_iG = X_{r1}G + F * (X_{r2}G - X_{r3}G) \tag{2.2}$$

di mana:

- r1,r2,r3 adalah indeks acak yang berbeda dari i.
- F adalah faktor skala (biasanya antara 0 dan 1).

- c) *Crossover* individu baru dihasilkan dengan menggabungkan individu target dan *mutant*, memastikan bahwa vektor uji tidak menduplikasi vektor individu target. Bisa dilihat pada persamaan 2.3

$$U_{j,i}G + 1 = \{ V_{j,i}G \text{ if } Rand\ j \leq CR \text{ or } J = J_{rand} \tag{2.3}$$

di mana:

- CR adalah rasio *crossover*.
- randj adalah nilai acak antara 0 dan 1.
- J rand adalah indeks acak yang memastikan minimal satu elemen dari v_{i,G+1} dipilih.

- d) Seleksi Algoritma menggunakan seleksi *greedy* untuk memilih individu yang lebih baik antara vektor uji dan vektor individu target untuk generasi berikutnya berdasarkan fungsi kebugaran (*fitness*). Bisa dilihat pada persamaan 2.4

$$X_iG + 1 = \{ U_{j,i}G \text{ jika } f(U_{j,i}G) \leq f(X_iG) \tag{2.4}$$

di mana f(·) adalah fungsi *fitness* yang dievaluasi.

Dengan pendekatan ini, secara efektif menggunakan DE untuk mengeksplorasi ruang parameter dari DBSCAN dan secara otomatis menemukan kombinasi nilai *Epsilon* dan MinPts yang menghasilkan pengklusteran terbaik menurut metrik yang kita pilih. Ini membantu mengatasi tantangan pemilihan parameter manual yang sering kali membutuhkan eksperimen berulang dan pengetahuan mendalam tentang data.

2.2.2 DBSCAN

Algoritma DBSCAN (*Density-Based Spatial Clustering of Applications with Noise*), sebuah algoritma pengelompokan berbasis kepadatan yang populer. Algoritma ini dapat menemukan kelompok data dengan bentuk dan ukuran yang berbeda dari sejumlah besar data, yang mengandung *noise* dan *outliers*. DBSCAN mampu mendeteksi anomali atau *outlier* lebih baik berdasarkan adanya atribut lokasi pada data. Kemampuan DBSCAN memungkinkan untuk mengekstrak pola-pola yang tidak biasa atau anomali pada kerapatan data[13].

Beberapa istilah dalam metode DBSCAN antara lain:

- Core** Pusat titik dalam sebuah *cluster* ditentukan oleh densitas, di mana sejumlah titik harus berada dalam jangkauan Eps (*radius* atau nilai ambang batas). Pengguna menentukan nilai MinPts (jumlah *minimum* titik dalam sebuah *cluster*).
- Border** Titik yang menjadi batasan dalam kawasan titik pusat (*core*).
- Noise** Titik yang tidak dapat dijangkau oleh *core* dan bukan merupakan border seperti persamaan 2.5

$$NOISE : \{x \in X \mid \forall i : x \in C_i\} \quad (2.5)$$

Dimana X merupakan gugus data, dan Ci merupakan *cluster* ke-1.

- Densitas terjangkau secara langsung: Sebuah titik dianggap sebagai titik terjangkau secara langsung jika titik tersebut terhubung secara langsung dengan titik pusat (*core*) seperti persamaan 2.6

$$x \in NEps(y) \wedge |NEps(y)| \geq MinPts \quad (2.6)$$

NEps(y) = titik sekitar y dalam radius Eps
MinPts = minimal titik dalam *cluster*

- Densitas terjangkau: Sebuah titik dianggap sebagai titik terjangkau jika titik tersebut terhubung secara tidak langsung dengan titik pusat (*core*).
- Densitas terhubung: Sebuah titik dianggap terhubung dengan titik lainnya. Dalam DBSCAN, diperlukan dua parameter *input*, yaitu *Epsilon* (Eps) dan titik minimum (MinPts). Eps-titik sekitar didefinisikan seperti persamaan 2.7:

$$NEps(x) = \{y \in D \mid dist(x,y) \leq Eps\} \quad (2.7)$$

N : Eps(x) menunjukkan titik-titik dalam jarak Eps dari titik x.

D : adalah kumpulan data.

Dist : (x, y) merupakan jarak *Euclidean* antara objek x dan y.

Eps : adalah nilai radius atau ambang batas.

Untuk melakukan pengelompokan data, langkah awalnya adalah menentukan nilai MinPts dan Eps. Secara umum, DBSCAN dimulai dengan memilih titik awal secara acak. Berikut adalah langkah-langkah dari penggunaan DBSCAN dengan menggunakan persamaan 2.8, yaitu;

- Tentukan titik awal atau p secara acak.
- Hitung semua jarak titik menggunakan persamaan 2.8

$$E(x,y) = \sqrt{\sum_{i=0}^n (x_i - y_i)^2} = 0 \quad (2.8)$$

- Tentukan nilai epsilon dan jumlah titik minimum. Penentuan nilai optimal untuk epsilon dan MinPts melibatkan serangkaian percobaan yang harus diulangi beberapa kali. Dalam proses tersebut, nilai *epsilon* dan MinPts yang terpilih adalah yang menghasilkan indeks siluet tinggi dan jumlah *cluster* yang signifikan.

2.3 Implementasi

Pada tahap implementasi dilakukan percobaan pembuatan sistem dengan bahasa pemrograman *Python*. Sistem yang dibuat sesuai dengan alur perancangan yang di mana *dataset* yang dilakukan melalui beberapa tahap *preprocessing*. Setelah itu data akan langsung dilakukan proses iterasi pencarian parameter terbaik menggunakan DE sehingga parameter yang di usulkan oleh metode DE bisa digunakan untuk pengolahan data *cluster* DBSCAN untuk mendeteksi anomali pada data.

2.4 Analisa dan Pengujian

Pada tahap pengujian dilakukan *testing* saat praproses melakukan iterasi menggunakan metode *Differential Evolution*. Dan juga ada beberapa pengujian yaitu ;

- Pengukuran akurasi *cluster* menggunakan metode *Silhouette score* [14] dan *Davies-Bouldin Index* (DBI) [15] pada optimasi parameter MinPts dan *Epsilon*.
- Pengukuran akurasi anomali pada *cluster* dengan menggunakan metode *Z-score* [16] dan *Reconstruction Error Mean Squared Error* (RE-MSE) [17].

Evaluasi yang digunakan untuk mengukur performa dari DBSCAN. *Silhouette score* dapat digunakan untuk mengukur performa dari *cluster* yang dihasilkan oleh DBSCAN.

3. HASIL DAN PEMBAHASAN

Penggunaan *Differential Evolution* (DE) dilakukan pada awal percobaan dengan menerapkan persamaan yang telah diuraikan sebelumnya. Tahap pertama penggunaan DE yaitu membangkitkan populasi individu. Individu di sini adalah nilai dari MinPts dan Eps. menggunakan persamaan (2.1) setelah itu dilakukan pengukuran fungsi *fitness* menggunakan metode DBSCAN untuk menghitung *Euclidean distance* pada data transaksi menggunakan persamaan (2.5 – 2.8). Untuk menghitung *score* awal pada percobaan setiap individu menggunakan metode *Silhouette score*. Dapat dilihat pada gambar 3 sebagai berikut.

```
MaxIt = 10
nPop = 10
nVar = 2
bounds = [(2, 20), (0.1, 2)]

# Inisialisasi populasi dengan nilai awal yang telah ditentukan
Population = np.zeros((nPop, nVar))
for i in range(nPop):
    Population[i, :] = [np.random.randint(bounds[0][0], bounds[0][1]), # MinPts
                      np.random.uniform(bounds[1][0], bounds[1][1])] # Eps

def fitness_function(individual, scaled_data):
    dbscan = DBSCAN(eps=individual[1], min_samples=int(individual[0])).fit(scaled_data)
    labels = dbscan.labels_
    n_clusters = len(set(labels)) - (1 if -1 in labels else 0)
    if n_clusters > 1:
        silhouette_avg = silhouette_score(scaled_data, labels)
    else:
        silhouette_avg = -1 # Skor siluet tidak valid jika hanya ada 1 cluster
    return silhouette_avg, {'MinPts': individual[0], 'Eps': individual[1], 'n_clusters': n_clusters}
```

Gambar 3. Code membangkitkan Individu dan Fitness

Pada gambar 3 menampilkan kode untuk menginisialisasi populasi berukuran 10 dengan dua parameter DBSCAN (MinPts dan Eps) dalam batas yang telah ditentukan, lalu

mendefinisikan fungsi *fitness* yang mengevaluasi setiap individu dalam populasi berdasarkan skor siluet yang dihasilkan dari *clustering* DBSCAN pada data yang telah diskalakan, mengembalikan nilai siluet dan informasi parameter kluster.

Setelah membangkitkan individu awal. Selanjutnya DE akan memproses dengan beberapa tahapan yaitu mutasi, perhitungan *crossover* dan seleksi individu baru sesuai dengan ketentuan persamaan yang ada. Pada percobaan ini menggunakan persamaan (2.2 – 2.4). dapat dilihat pada gambar 4 berikut.

```
def mutate(individuals, target_idx, F=0.8):
    idxs = [idx for idx in range(len(individuals)) if idx != target_idx]
    a, b, c = np.random.choice(idxs, 3, replace=False)
    mutant = individuals[a] + F * (individuals[b] - individuals[c])
    return np.clip(mutant, [bounds[0][0], bounds[1][0]], [bounds[0][1], bounds[1][1]])

def crossover(target, mutant, CR=0.7):
    cross_points = np.random.rand(nVar) < CR
    trial = np.where(cross_points, mutant, target)
    return trial

def select(target, trial, scaled_data):
    fitness_ind, _ = fitness_function(target, scaled_data)
    fitness_trial, _ = fitness_function(trial, scaled_data)
    return trial if fitness_trial > fitness_ind else target
```

Gambar 4. Code implementasi Mutate, Crossover dan Seleksi

Kode pada gambar 4 mengimplementasikan tiga komponen utama dari algoritma DE yaitu mutasi, *crossover*, dan seleksi. Fungsi *mutate* menghasilkan vektor mutan dengan memilih tiga individu acak dari populasi dan menggabungkannya dengan faktor skala F yang ditetapkan adalah 0.8, lalu memastikan hasilnya tetap dalam batas yang ditentukan. Fungsi *Crossover* membuat vektor percobaan dengan menggabungkan elemen-elemen dari individu target dan vektor mutan berdasarkan probabilitas *crossover* CR yang ditetapkan adalah 0.7. Fungsi *select* menentukan apakah individu target atau vektor percobaan yang akan masuk ke generasi berikutnya dengan membandingkan nilai *fitness* mereka dan memilih yang memiliki *fitness* lebih tinggi. Setelah proses *select* akan ditetapkan pengulangan percobaan iterasi hingga proses mencapai batas maksimal atau sudah tidak ada perubahan pada pengujian *Silhouette score*. Data yang akan ditampilkan berupa iterasi percobaan setiap individu terbaik pada setiap iterasi.

Differential Evolution (DE) melakukan satu kali percobaan pada *dataset* dengan ±1011 *record* data transaksi. Menghasilkan lima iterasi pemilihan parameter dengan menampilkan hasil *cluster* dari penggunaan *MinPts* dan *Epsilon* pada DBSCAN. Iterasi dapat dilihat pada gambar 5 sebagai berikut.

```
Iterasi Ke - 1 Parameter : {'MinPts': 12.600000000000001, 'Eps': 1.4287135653962773, 'nClusters': 3}
Iterasi Ke - 2 Parameter : {'MinPts': 16.0, 'Eps': 0.15232448270225807, 'nClusters': 8}
Iterasi Ke - 3 Parameter : {'MinPts': 7.0, 'Eps': 1.674463695487391, 'nClusters': 2}
Iterasi Ke - 4 Parameter : {'MinPts': 11.000000000000002, 'Eps': 2.0, 'nClusters': 2}
Iterasi Ke - 5 Parameter : {'MinPts': 7.960000000000001, 'Eps': 2.0, 'nClusters': 2}
```

Gambar 5. Hasil iterasi penggunaan DE

Dapat dilihat pada gambar 5 hasil iterasi yang diberikan pada setiap iterasi memiliki *nCluster* yang berbeda pada iterasi ke-1 dengan *Minpts* 12,6 dan *Epsilon* 1.428..3 mendapatkan *nCluster* sebanyak 3. Iterasi ke-2 dengan

MinPts 16 dan *Epsilon* 0,152..7 mendapatkan *nCluster* 8. Dan iterasi tiga terakhir dengan menghasilkan *nCluster* 2 cenderung memberikan optimasi dari *Epsilon* ≤ 2. Pada percobaan penggunaan DE ke-2 mendapatkan hasil parameter yang lebih stabil bisa dilihat pada gambar 4.

```
Iterasi Ke - 1 Parameter : {'MinPts': 4.0, 'Eps': 1.8296508691942042, 'nClusters': 3}
Iterasi Ke - 2 Parameter : {'MinPts': 10.0, 'Eps': 1.6429907465044589, 'nClusters': 2}
Iterasi Ke - 3 Parameter : {'MinPts': 8.0, 'Eps': 1.8725944223109998, 'nClusters': 2}
Iterasi Ke - 4 Parameter : {'MinPts': 8.0, 'Eps': 2.0, 'nClusters': 2}
```

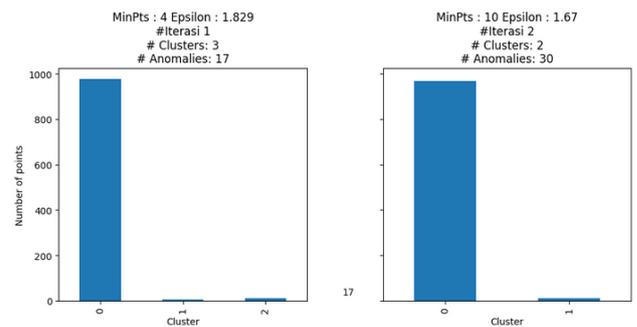
Gambar 6. Hasil Iterasi penggunaan DE ke-2

Pada gambar 6 memberikan informasi percobaan ke-2 penggunaan DE untuk optimasi parameter mendapatkan 4 iterasi yang di mana parameter dan *nCluster* yang diberikan cenderung sama atau lebih stabil. Selanjutnya dilakukan evaluasi *nCluster* menggunakan metode *Silhouette score* untuk mengukur seberapa akurat parameter yang diberikan pada percobaan DE yang ke-2 dapat dilihat pada tabel 3.

Tabel 3. Akurasi dari pengukuran *cluster* yang diuji

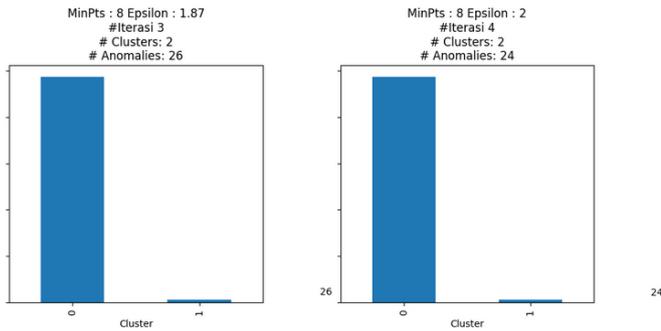
Iterasi	MinPts	Epsilon	nCluster	Silhouette score	DBI
1	4	1.829	3	0.7836	0.19955
2	10	1.67	2	0.7926	0.19791
3	8	1.87	2	0.79463	0.19995
4	8	2	2	0.79563	0.20149

Tabel 3 memberikan informasi mengenai seberapa akurat *nCluster* dari percobaan ke-2 penggunaan DE untuk optimasi parameter *MinPts* dan *Epsilon* pada DBSCAN. Dapat dilihat pada iterasi pertama akurasi yang diperoleh mencapai 0.7836 menggunakan *Silhouette score* dan akurasi DBI 0.19955, iterasi kedua diperoleh akurasi mencapai 0.7926 dan 0.19791, iterasi ketiga memperoleh akurasi mencapai 0.79463 dan 0.19995, iterasi terakhir memperoleh akurasi hingga mencapai 0.79563 dan 0.20149. Dapat dilihat peningkatan akurasi yang tidak terlalu signifikan, tetapi pada setiap iterasi yang dilakukan selalu ada peningkatan untuk akurasi *nCluster* yang diberikan dari optimasi yang dilakukan metode DE tersebut. Kemudian parameter akan diuji untuk mendeteksi anomali oleh DBSCAN dapat dilihat pada gambar 5 untuk iterasi 1 dan 2 dan gambar 6 untuk iterasi 3 dan 4 sebagai berikut.



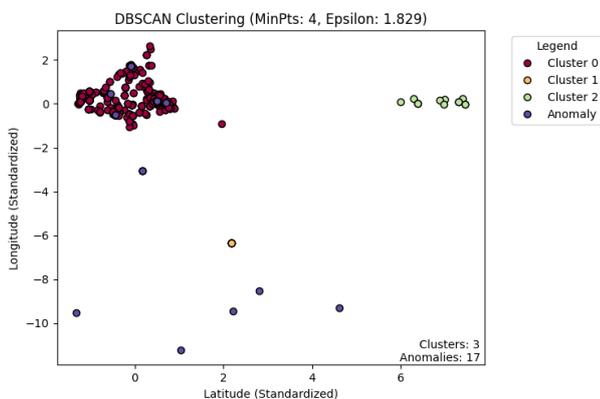
Gambar 7. Hasil deteksi anomali iterasi (a) dan (b)

Bisa dilihat pada gambar 7 menampilkan hasil dari deteksi *anomaly* menggunakan DBSCAN pada iterasi 1 dan 2, di mana iterasi pertama mampu mendeteksi 17 *anomaly* pada 3 *cluster*. Selanjutnya iterasi kedua mampu mendeteksi 30 *anomaly* data dari 2 *cluster*.



Gambar 8. Hasil deteksi *anomaly* iterasi (c) dan (d)

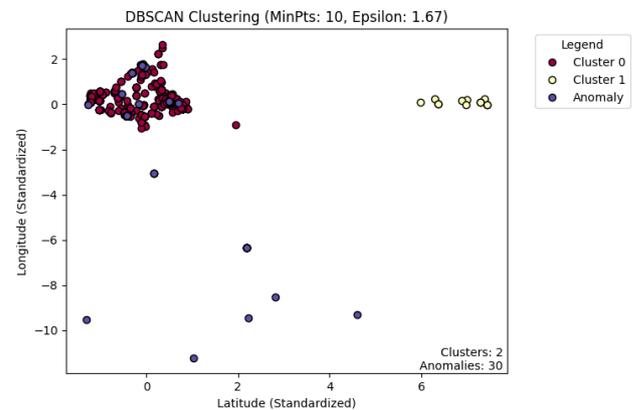
Selanjutnya gambar 8 menunjukkan iterasi ke-3 dan ke-4 pada penggunaan parameter iterasi ke-3 dengan minPts 8 dan *Epsilon* 1.87 dapat mendeteksi *anomaly* sebanyak 26. Iterasi ke-4 dengan minPts 8 dan *Epsilon* 2 dapat mendeteksi sebanyak 24 *anomaly* data. Nilai MinPts yang lebih tinggi seperti pada iterasi ke-2 menghasilkan jumlah *cluster* yang lebih sedikit dan jumlah *anomaly* yang lebih banyak. Ini menunjukkan bahwa dengan meningkatkan nilai MinPts, algoritma menjadi lebih ketat dalam menentukan kepadatan yang diperlukan untuk membentuk kluster, sehingga lebih banyak *point* yang dianggap sebagai *noise* atau *anomaly*. Nilai *Epsilon* yang lebih tinggi seperti pada iterasi ke-4 tampaknya tidak terlalu mempengaruhi jumlah kluster, yang tetap berada di angka 2. Namun, *Epsilon* yang lebih besar biasanya memungkinkan *point* untuk menjadi bagian dari kluster meskipun jaraknya lebih jauh. Ini mungkin alasan mengapa ada sedikit penurunan dalam jumlah *anomaly* dari iterasi ke-2 ke iterasi ke-4, meski perbedaannya tidak signifikan. Selanjutnya pengujian MinPts dan *Epsilon* menggunakan *scatter plot* dapat dilihat pada gambar berikut.



Gambar 9. Hasil *cluster* iterasi ke-1

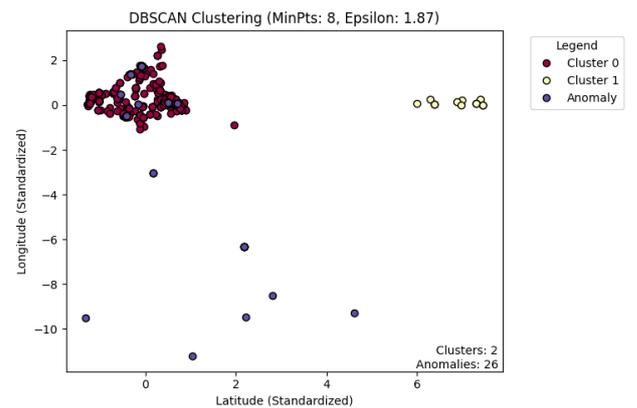
Dapat dilihat pada gambar 9 parameter DBSCAN pada iterasi ke-1 dengan MinPts = 4 dan *Epsilon* = 1,829. Algoritma DBSCAN ini telah mengidentifikasi sebuah

cluster besar (merah), sebuah *cluster* yang lebih kecil (ungu), dan beberapa penciran (hijau) yang dianggap sebagai *anomaly*, pada data yang ada jumlah total *anomaly* adalah 17.



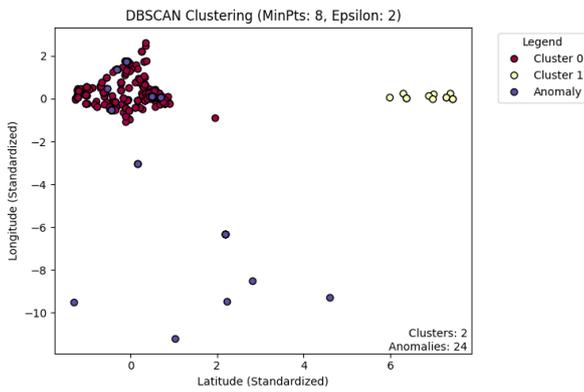
Gambar 10. Hasil *cluster* iterasi ke-2

dalam iterasi ke-2 pada gambar 10 dengan memasukkan parameter MinPts telah meningkat menjadi 10 dan *Epsilon* adalah 1,67. Tampaknya peningkatan MinPts menghasilkan kluster yang lebih tepat dan lebih banyak titik data yang diklasifikasikan sebagai *anomaly* menjadi 30 data.



Gambar 11. Hasil *cluster* iterasi ke-3

Gambar 11 menunjukan iterasi ke-3 MinPts berkurang menjadi 8, dan *Epsilon* meningkat menjadi 1,87. Jumlah *anomaly* telah berkurang menjadi 26 dibandingkan dengan iterasi 2, yang menunjukkan bahwa sedikit peningkatan *Epsilon* memungkinkan lebih banyak titik untuk dimasukkan ke dalam *cluster*.



Gambar 12. Hasil *cluster* iterasi ke-4

Terakhir dapat dilihat pada gambar 12 menunjukan iterasi ke-4 dengan *MinPts* = 8 dan *Epsilon* = 2. Jumlah *anomaly* tetap 24, sama seperti iterasi sebelumnya, namun klaster-klaster tersebut tampaknya bergabung karena peningkatan *Epsilon*, yang memungkinkan adanya lingkungan yang lebih besar di sekitar setiap titik.

Dari iterasi yang dilakukan, dapat dilihat bagaimana pilihan parameter mempengaruhi hasil pengelompokan. Meningkatkan *Epsilon* cenderung menggabungkan *cluster* dan mengurangi jumlah *anomaly*, sementara meningkatkan *MinPts* umumnya menghasilkan lebih banyak *anomaly* dan *cluster* yang lebih kecil dan lebih berbeda. Ini adalah keseimbangan untuk menemukan parameter yang tepat yang paling sesuai dengan pengelompokan alami dalam data. Selanjutnya pengujian pengukuran *anomaly* menggunakan metode *Z-score* yang akan ditunjukkan pada tabel 4 sebagai berikut:

Tabel 4. Hasil pengukuran *Z-score* pada *anomaly*

<i>Its</i>	<i>MPts</i>	<i>Eps</i>	<i>Upper</i>	<i>Lower</i>	<i>Accuracy</i>
1	4	1.829	0.02474076	0.00888931	98.41%
2	10	1.67	0.04013322	0.01921396	97.91%
3	8	1.87	0.03547433	0.01595989	98.05%
4	8	2	0.03312282	0.01435493	98.12%

Rata-rata *Z-score* menunjukkan jarak rata-rata yang dihitung dari pusat *cluster* ke titik data, di mana semakin kecil nilai ini, semakin padat *cluster* tersebut. Dari data, skor tertinggi adalah 2.00 dan terendah adalah 1.67. Skor yang lebih rendah pada *MPts* yang lebih tinggi (10) menunjukkan bahwa *cluster* dengan lebih banyak titik cenderung lebih padat atau homogen. Akurasi mencerminkan seberapa baik model DBSCAN mengidentifikasi titik data sebagai bagian dari *cluster* atau sebagai *anomaly*. Nilai akurasi berkisar dari 97.91% hingga 98.41%. Meskipun perbedaan tidak besar, pengaturan dengan *MPts* 4 mencapai akurasi tertinggi (98.41%).

Tabel 5. Hasil pengukuran *anomaly* menggunakan MSE

<i>Its</i>	<i>MPts</i>	<i>Eps</i>	<i>RE-MSE</i>	<i>MSE-A</i>	<i>MSE-non A</i>
1	4	1.829	0.671778	21.37948	0.33879
2	10	1.67	0.935866	19.06420	0.38148
3	8	1.87	0.555537	13.64477	0.21003
4	8	2	0.695493	14.84811	0.35135

Dapat dilihat pada tabel 5 yang disajikan, hasil eksperimen parameter algoritma DBSCAN untuk mengidentifikasi anomali dalam data transaksi bank. dengan fokus pada *Reconstruction Mean Squared Error* (RE-MSE), data yang terdeteksi anomali (MSE-A) serta data yang terdeteksi non-anomali (MSE-non A). Dari keempat percobaan yang dicatat, Iterasi 3 dengan 8 *MPts* dan *Eps* 1.87 menunjukkan kinerja terbaik dengan RE-MSE terendah sebesar 0.555537, serta nilai MSE-A yang di indikasikan sangat jauh dari data non anomali mencapai 13.64477 menandakan kemampuan yang efisien dalam mendeteksi anomali dengan tingkat kesalahan yang lebih rendah. Sementara itu, nilai MSE untuk non-anomali juga paling rendah pada iterasi ini, yaitu 0.21003, mengindikasikan bahwa model ini secara akurat merekonstruksi mayoritas data non-anomali.

Dalam penelitian ini, mengusulkan metode kombinasi antara algoritma DBSCAN dan *Differential Evolution* (DE) untuk mendeteksi anomali pada data transaksi bank. Metode ini bertujuan untuk mengatasi tantangan utama dalam penggunaan DBSCAN, yaitu penentuan parameter *MinPts* dan *Epsilon* yang optimal. Berdasarkan hasil eksperimen, metode yang diusulkan menunjukkan peningkatan yang signifikan dalam akurasi deteksi anomali dibandingkan dengan pendekatan DBSCAN konvensional. Tahap pertama dalam penelitian ini adalah pengumpulan data transaksi dari Bank XYZ yang melibatkan lebih dari 1011 rekaman. Data tersebut kemudian melalui tahapan praproses seperti pembersihan data, normalisasi, dan transformasi. Praproses ini dilakukan untuk menghapus *missing values* dan memastikan bahwa data dalam format yang sesuai untuk pengelompokan dengan DBSCAN.

Selanjutnya, DE digunakan untuk mengoptimalkan parameter DBSCAN. DE melakukan inisialisasi populasi, mutasi, *crossover*, dan seleksi untuk menemukan kombinasi parameter yang menghasilkan *clustering* terbaik. Proses ini melibatkan beberapa iterasi untuk mengeksplorasi ruang parameter dan mengeksploitasi solusi terbaik yang ditemukan. Hasil iterasi menunjukkan bahwa penggunaan DE mampu meningkatkan akurasi deteksi anomali dengan nilai *Silhouette Score*, *DBI*, *MSE* dan *Z-score* yang lebih baik. Hasil pengelompokan menunjukkan bahwa peningkatan nilai *MinPts* menghasilkan jumlah *cluster* yang lebih sedikit namun lebih padat, sementara peningkatan nilai *Epsilon* cenderung menggabungkan *cluster* dan

mengurangi jumlah anomali. Contohnya, pada iterasi ke-2 dengan parameter MinPts=10 dan Epsilon=1.67, jumlah anomali yang terdeteksi meningkat menjadi 30 data, yang menunjukkan bahwa nilai MinPts yang lebih tinggi membuat algoritma lebih ketat dalam menentukan kepadatan *cluster*.

Penggunaan DE juga terbukti efektif dalam menangani data dengan variasi densitas yang tinggi, yang merupakan salah satu keterbatasan utama dari DBSCAN konvensional. DE membantu menemukan kombinasi parameter yang lebih optimal dengan cara yang lebih efisien daripada metode *trial and error* manual. Secara keseluruhan, penelitian ini menunjukkan bahwa kombinasi DBSCAN dan DE adalah pendekatan yang efektif untuk deteksi anomali pada data transaksi bank. Metode ini tidak hanya meningkatkan akurasi deteksi tetapi juga efisiensi dalam penentuan parameter DBSCAN. Penelitian lebih lanjut dapat mengaplikasikan metode ini pada jenis data lain dan menguji ketahanannya terhadap berbagai skenario anomali.

4. KESIMPULAN

Berdasarkan hasil penelitian ini berhasil mengoptimalkan parameter DBSCAN, yaitu MinPts dan Epsilon, menggunakan metode *Differential Evolution* (DE). Hasilnya menunjukkan bahwa optimasi parameter ini meningkatkan akurasi deteksi anomali dalam data transaksi bank. Iterasi yang dilakukan menunjukkan peningkatan akurasi yang stabil menggunakan dengan DBI mencapai 0.19955, dengan hasil terbaik mencapai akurasi Z-Score 98.41% dan MSE mencapai 0.671778 pada penggunaan MinPts 4 dan Eps 1,829 dapat mendeteksi 17 anomali. Penggunaan DE juga membantu mengurangi kesulitan dalam menentukan parameter secara manual. pendekatan yang diusulkan dapat digunakan untuk meningkatkan deteksi anomali dalam berbagai aplikasi yang membutuhkan analisis data transaksi. Dengan menggunakan optimasi parameter yang lebih efisien. Penelitian ini memiliki beberapa batasan, di antaranya adalah penggunaan *dataset* yang terbatas pada transaksi bank tertentu. Selain itu, meskipun metode DE membantu dalam optimasi parameter, hasilnya masih bergantung pada kualitas dan karakteristik data yang digunakan. Variasi data yang berbeda mungkin memerlukan penyesuaian lebih lanjut. Untuk penelitian selanjutnya, direkomendasikan untuk mengaplikasikan metode optimasi ini pada *dataset* transaksi yang lebih besar dan beragam. Penelitian mendatang juga dapat mengeksplorasi kombinasi algoritma *clustering* lainnya dengan metode optimasi berbeda untuk membandingkan hasilnya. Selain itu, pengembangan sistem deteksi anomali yang *real-time* menggunakan pendekatan ini dapat menjadi kontribusi signifikan bagi masyarakat umum.

DAFTAR PUSTAKA

- [1] P. Jain, M. S. Bajpai, and R. Pamula, "A Modified DBSCAN Algorithm for Anomaly Detection in Time-series Data with Seasonality," *International Arab Journal of Information Technology*, vol. 19, no. 1, pp. 23–28, Jan. 2022, doi: 10.34028/iajit/19/1/3.
- [2] M. Tatusch, G. Klassen, M. Bravidor, and S. Conrad, "Predicting Erroneous Financial Statements Using a Density-Based Clustering Approach," in *ACM International Conference Proceeding Series*, Association for Computing Machinery, Aug. 2020, pp. 89–94. doi: 10.1145/3418653.3418673.
- [3] I. Kamil and B. Pharmasetiawan, "Fingerprint Presence Fraud Detection Using Tight Clustering on Employee's Presence and Activity Data," 2019. doi: DOI: 10.1109/ICITISEE48480.2019.9003914.
- [4] Mohamad Zamini and Gholamali Montazer, *Credit Card Fraud Detection using autoencoder based clustering*. 2018 9th International Symposium on Telecommunications (IST'2018), 2018. doi: DOI: 10.1109/ISTEL.2018.8661129.
- [5] M. Yang and D. Ergu, "Anomaly Detection of Vehicle Data Based on LOF Algorithm," *Frontiers in Signal Processing*, vol. 4, no. 1, Jan. 2020, doi: 10.22606/fsp.2020.41007.
- [6] D. Deng, "Research on Anomaly Detection Method Based on DBSCAN Clustering Algorithm," in *Proceedings - 2020 5th International Conference on Information Science, Computer Technology and Transportation, ISCTT 2020*, Institute of Electrical and Electronics Engineers Inc., Nov. 2020, pp. 439–442. doi: 10.1109/ISCTT51595.2020.00083.
- [7] P. Jain, M. S. Bajpai, and R. Pamula, "A Modified DBSCAN Algorithm for Anomaly Detection in Time-series Data with Seasonality," *International Arab Journal of Information Technology*, vol. 19, no. 1, pp. 23–28, Jan. 2022, doi: 10.34028/iajit/19/1/3.
- [8] Z. Zhang, L. Chen, Q. Liu, and P. Wang, "A Fraud Detection Method for Low-Frequency Transaction," *IEEE Access*, vol. 8, pp. 25210–25220, 2020, doi: 10.1109/ACCESS.2020.2970614.
- [9] M. Z. Hossain, M. J. Islam, M. W. R. Miah, J. H. Rony, and M. Begum, "Develop a dynamic DBSCAN algorithm for solving initial parameter selection problem of the DBSCAN algorithm," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 23, no. 3, pp. 1602–1610, Sep. 2021, doi: 10.11591/ijeecs.v23.i3.pp1602-1610.
- [10] K. I. Ghathwan and A. J. Mohammed, "Intelligent Bat Algorithm for Finding Eps Parameter of DbScan Clustering Algorithm," *Iraqi Journal of Science*, vol. 63, no. 12, pp. 5572–5580, 2022, doi: 10.24996/ijs.2022.63.12.41.

- [11] G. Dominico and R. S. Parpinelli, "Multiple global optima location using differential evolution, clustering, and local search," *Appl Soft Comput*, vol. 108, Sep. 2021, doi: 10.1016/j.asoc.2021.107448.
- [12] M. T. Guerreiro *et al.*, "Anomaly detection in automotive industry using clustering methods—a case study," *Applied Sciences (Switzerland)*, vol. 11, no. 21, Nov. 2021, doi: 10.3390/app11219868.
- [13] Manoj Kumar Reddy Mallidi and Yeshwanth Zagabathuni, "Analysis of Credit Card Fraud detection using Machine Learning models on balanced and imbalanced datasets," *International Journal of Emerging Trends in Engineering Research*, vol. 9, no. 7, pp. 846–852, Jul. 2021, doi: 10.30534/ijeter/2021/02972021.
- [14] H. Sayed Ramadan, H. Amin Maghawry, M. El-Eleamy, and K. El-Bahnasy, "A Heuristic Novel Approach For Determination Of Optimal Epsilon For DBSCAN Clustering Algorithm," *J Theor Appl Inf Technol*, vol. 15, no. 7, 2022, [Online]. Available: www.jatit.org
- [15] N. Gholizadeh, H. Saadatfar, and N. Hanafi, "K-DBSCAN: An improved DBSCAN algorithm for big data," *Journal of Supercomputing*, vol. 77, no. 6, pp. 6214–6235, Jun. 2021, doi: 10.1007/s11227-020-03524-3.
- [16] H. Liu, Y. Wang, and W. G. Chen, "Anomaly detection for condition monitoring data using auxiliary feature vector and density-based clustering," *IET Generation, Transmission and Distribution*, vol. 14, no. 1, pp. 108–118, Jan. 2020, doi: 10.1049/iet-gtd.2019.0682.
- [17] X. Olive and L. Basora, "Identifying Anomalies in past en-route Trajectories with Clustering and Anomaly Detection Methods," 2019. [Online]. Available: <https://hal.science/hal-02345597>



PENERAPAN METODE *COMBINED COMPROMISE SOLUTION* (COCOSO) DALAM PEMILIHAN *FRANCHISE* MINUMAN

Julita Marito¹, Wahyuni Betris Nainggolan², Gede Surya Mahendra³

^{1,2,3}Program Studi Sistem Informasi, Universitas Pendidikan Ganesha
Singaraja, Bali, Indonesia 81116

julita@undiksha.ac.id, wahyuni.betris@undiksha.ac.id, gmahendra@undiksha.ac.id

Abstract

The research in this journal is motivated by the beverage franchise industry, which has become one of the most dynamic global business sectors. This article examines applying the Combined Compromise Solution (CoCoSo) method to the decision-making system (SPK) in selecting beverage franchises. The beverage franchise data that we use consists of 9 brands that are pretty famous among the public, such as Kopi Kenangan, Es Teh Indonesia, Teh Poci, Calais Tea, Puyo Puyo, Gulu Gulu, Kopi Kulo, Xi boba, Kopi Yor. In the process of data collection, we use observation and research. The data analysis process is carried out using the combined compromise solution method, one of the multi-criteria decision-making (MCDM) methods that can be used to select alternatives based on the calculation of criteria weights. This method can facilitate the determination of beverage franchises because it is more effective and efficient in calculating and ranking. Through the decision-making system that has been developed, the value of the beverage franchise can be generated based on predetermined criteria. Calculation of beverage franchises using CoCoSo shows the results of the calculation of the highest preference value obtained by the Puyo Puyo beverage franchise with a final value of 2.3436 and the lowest preference value obtained by the Kopi Kenangan beverage franchise with a final value of 1.3385.

Keywords: Business, CoCoSo, Decision Making System, Franchise, MCDM

Abstrak

Penelitian pada jurnal ini dilatar belakangi oleh industri *franchise* minuman yang kini telah menjadi salah satu sektor yang paling dinamis dalam bisnis global. Artikel ini bertujuan untuk meneliti penerapan metode *Combined Compromise Solution* (CoCoSo) pada Sistem Pengambilan Keputusan (SPK) dalam pemilihan *franchise* minuman. Data *franchise* minuman yang kami gunakan terdiri dari 9 *brand* yang cukup terkenal di kalangan masyarakat seperti, Kopi Kenangan, Es Teh Indonesia, Teh Poci, Calais Tea, Puyo Puyo, Gulu Gulu, Kopi Kulo, Xi boba, Kopi Yor. Dalam proses pengumpulan data yang kami lakukan dengan cara observasi dan *Research*. Proses analisis data dilakukan menggunakan metode *Combined Compromise Solution* yang merupakan salah satu metode *Multi Criteria Decision Making* (MCDM) yang dapat digunakan dalam pemilihan alternatif berdasarkan perhitungan bobot kriteria. Dengan metode ini dapat memudahkan dalam penentuan *Franchise* minuman karena lebih efektif dan efisien dalam proses perhitungan dan perbandingan. Sehingga melalui Sistem Pengambilan Keputusan yang telah dikembangkan dapat dihasilkannya nilai *franchise* minuman berdasarkan dengan kriteria yang telah ditentukan. Perhitungan *franchise* minuman dengan menggunakan CoCoSo menunjukkan bahwa hasil perhitungan nilai preferensi tertinggi didapatkan oleh *franchise* minuman Puyo Puyo dengan nilai akhir 2,3436 serta nilai preferensi terendah didapatkan oleh *franchise* minuman Kopi Kenangan dengan nilai akhir 1.3385.

Kata kunci: Bisnis, CoCoSo, MCDM, Sistem Pengambilan Keputusan, Waralaba

1. PENDAHULUAN

Dalam era globalisasi ini, industri waralaba atau yang juga sering disebut dengan *franchise* telah menjadi salah satu pilar utama dalam perekonomian global. Konsep waralaba memungkinkan pengusaha untuk memperluas bisnis dengan menggunakan merek, sistem, dan dukungan yang telah

teruji dari pemilik merek dagang. Dalam konteks ini, pemilihan *franchise* minuman merupakan keputusan strategi yang membutuhkan analisis yang cermat dan pendekatan yang terstruktur. Penelitian ini menggali lebih dalam penerapan Metode *Combined Compromise Solution* (CoCoSo) dalam konteks pemilihan *franchise* minuman,

dengan mengintegrasikan berbagai faktor kritis yang mempengaruhi keputusan tersebut.

Jenis bisnis waralaba (*franchise*) pertama kali dicetuskan di Amerika Serikat, sekitar satu abad lalu, pada saat perusahaan mesin jahit Singer mulai memperkenalkan konsep waralaba (*franchise*) menjadi suatu cara dalam mengembangkan distribusi produknya [1]. Dalam waralaba (*franchise*) ini dapat dikatakan bahwa sebagai bagian dari kepatuhan mitra usaha terhadap aturan main yang diberikan oleh pengusaha *Franchise* diberikan hak untuk memanfaatkan Hak Atas Kekayaan Intelektual dari pengusaha *Franchisor*, baik dalam penggunaan merek dagang, merek jasa, hak cipta atas logo, desain industri, paten berupa teknologi maupun rahasia dagang dan sebaliknya, pengusaha *Franchisor* memperoleh royalti atas penggunaan Hak Atas Kekayaan Intelektual mereka [2].

Franchise dikelompokkan menjadi tiga berdasarkan jenisnya yaitu: *Product Franchises* yang merupakan model bisnis di mana penerima waralaba membayar kepada *franchisor* atas hak untuk mendistribusikan produk dengan hak eksklusif untuk memasarkan produk di wilayah tertentu. *franchisor* bertindak sebagai produsen produk. *Business Format Franchises* merupakan model bisnis waralaba dimana *franchisor* memberikan izin kepada individu atau perusahaan untuk membuka gerai yang menjual berbagai produk *franchisor*. Pada model ini *franchisor* menyediakan metode bisnis yang telah terbukti atau teruji dan merek dagang serta membantu menjalankan bisnis, Penerima waralaba harus mematuhi standar operasional, membayar biaya lisensi dan diawasi oleh *franchisor*. *Manufacturing Plant Franchises* merupakan model bisnis dimana *franchisor* yang mengungkapkan *know-how* atau formula rahasia yang digunakan dalam produksi produk, Penerima waralaba kemudian memproduksi dan memasarkan produk sesuai standar dengan menggunakan merek yang sama [3].

Franchise merupakan suatu perjanjian dalam metode pendistribusian barang dan jasa kepada konsumen. Di sini *Franchisor* memberi lisensi pada *franchise* agar melakukan kegiatan pendistribusian barang dan jasa di bawah nama dan identitas *franchisor* di wilayah tertentu [4]. Yang mana usaha tersebut dijalankan sesuai dengan prosedur dan cara yang telah ditetapkan *franchisor*, sebagai imbalan *franchise* membayar sejumlah uang sesuai perjanjian yang merupakan *initial fee* dan royalti.

Industri waralaba minuman telah menjadi salah satu sektor yang paling dinamis dalam bisnis global. Minuman Seperti kopi, teh, susu, dan minuman lainnya tidak hanya memenuhi kebutuhan konsumen akan hidrasi, tetapi juga telah menjadi bagian integral dari gaya hidup *modern*. Dengan pertumbuhan pesat ritel *modern* dan perubahan preferensi konsumen, permintaan akan minuman dalam format waralaba semakin meningkat. Di negara kita sendiri (Indonesia) bentuk usaha seperti bisnis ini juga berkembang dengan sangat pesat, dimana bentuk usaha *franchise* ini

banyak digunakan dalam usaha minuman seperti Kopi Kenangan, Es Teh Indonesia, Teh Poci, Calais Tea, Puyo Puyo, Gulu Gulu, Kopi Kulo, Xi boba, Kopi Yor, dan lain sebagainya.

Tabel 1. Logo *Franchise*

<i>Franchise</i>	Logo
Kopi Kenangan	
Es Teh Indonesia	
Teh Poci	
Calais Tea	
Puyo Puyo	
Gulu Gulu	
Kopi Kulo	
Xi Boba	
Kopi Yor	

Artikel ini meneliti penerapan metode *Combined Compromise Solution* (CoCoSo) pada Sistem Pengambilan Keputusan (SPK) dalam pemilihan *franchise* minuman. Sistem Pendukung Keputusan (SPK) adalah alat yang

dimaksudkan untuk membantu individu atau organisasi untuk membuat pilihan yang lebih baik dan lebih terinformasi [5]. SPK menggunakan teknologi data dan komputer untuk mengumpulkan, mengolah, dan menganalisis informasi yang relevan, setelah itu SPK membuat saran atau pilihan alternatif [6]. SPK meningkatkan pengambilan keputusan, efisiensi dan wawasan pengguna yang pada gilirannya dapat meningkatkan kualitas keputusan strategi dan operasional. SPK dapat meramalkan berbagai tren atau skenario dengan menggabungkan berbagai jenis data, termasuk data historis, saat ini dan masa depan. SPK dapat menggunakan metode seperti pemodelan matematis, analisis statistik, teknik kecerdasan buatan, dan pengambilan keputusan multi-kriteria [7]. SPK juga memiliki kemampuan untuk memungkinkan pengguna bekerja sama dalam proses pengambilan keputusan yang memungkinkan pertukaran ide dan penilaian yang lebih baik. Secara keseluruhan, SPK adalah alat yang kuat untuk meningkatkan proses pengambilan keputusan di berbagai bidang dan membantu pengguna membuat keputusan yang lebih tepat dan informasi [8].

Sistem pendukung keputusan memiliki beberapa keuntungan utama; memiliki kemampuan untuk mendukung solusi masalah yang rumit, respons yang cepat, dan kemampuan untuk secara cepat mencoba berbagai strategi dalam konfigurasi yang berbeda, meningkatkan kontrol dan kinerja manajemen, menghemat uang untuk pilihan yang salah, membuat pilihan yang tepat dan konsisten, meningkatkan efektivitas manajemen, dan meningkatkan produktivitas analisis pengambilan keputusan [9]. SPK dirancang untuk beradaptasi dengan perubahan organisasi. Meskipun SPK tidak menggantikan pemakai sebagai pengambil keputusan, mereka menawarkan alternatif dalam proses pengambilan keputusan. SPK adalah sebuah sistem interaktif berbasis komputer yang membantu dalam pembuatan keputusan dengan menggunakan model dan data untuk menyelesaikan masalah yang tidak terstruktur dan semi-terstruktur. Keterbatasan SPK adalah mereka tidak dapat menyelesaikan masalah hal-hal yang tidak terduga seperti manusia, tetapi hanya dengan program yang ditanamkan [10].

Metode *Multi-Criteria Decision Making* (MCDM) digunakan untuk menghasilkan rekomendasi keputusan berdasarkan berbagai kriteria serta alternatif pengambilan keputusan [11]. MCDM menghitung seluruh alternatif berdasarkan kriteria yang bertentangan untuk mendapatkan alternatif terbaik, dan MCDM sangat cocok digunakan ketika terdapat beberapa kriteria dan prioritas [12]. *Combined Compromise Solution CoCoSo* adalah salah satu metode *Multi Criteria Decision Making* (MCDM). MCDM memiliki banyak metode yang dapat digunakan untuk melakukan penentuan alternatif terbaik, seperti: *Weighted Sum Model* (WSM), *Analytical Hierarchy Process* (AHP), *Evaluasi Rasio Aditif* (ARAS), *Technique of Referral by*

Kemiripan Solusi Ideal (TOPSIS), *Distance of Average Solution* (EDAS), *Combined Compromise Solution* (CoCoSo), *Complex Proportional Assesment* (COPRAS), *Multi-Objective Optimization by Ratio Analysis* (MOORA) dan masih banyak metode lain yang dapat digunakan dalam pengambilan keputusan.

Metode CoCoSo ialah model pendekatan berdasarkan bobot aditif yang sederhana dan diintegrasikan dalam model eksponensial. Metode CoCoSo diketahui fleksibel terhadap bobot kriteria dan tidak mudah terpengaruh pada perubahan jumlah alternatif [13]. Metode ini digunakan sebagai solusi untuk memecahkan masalah pengambilan keputusan CoCoSo yang mengacu pada pilihan dan standar yang ada. Penelitian mengenai metode yang digunakan, yaitu pengambilan keputusan dalam pemilihan *franchise* minuman.

Dengan pertimbangan kompleksitas dalam pengambilan keputusan bisnis, terutama pada *franchise* minuman, misalnya. SPK menggunakan metode CoCoSo dapat membantu dalam pemilihan *franchise* minuman. Mempertimbangkan saat membuat keputusan dengan menggunakan CoCoSo dapat menggabungkan dan menyeimbangkan berbagai standar untuk membantu membuat keputusan terbaik. Oleh karena itu penelitian ini bertujuan untuk memberikan panduan yang bermanfaat bagi calon pengusaha dalam menghadapi kesulitan dalam memilih *franchise* minuman, dengan mengidentifikasi komponen penting yang mempengaruhi keputusan tersebut dan mengevaluasi seberapa efektif CoCoSo dalam mengatasi masalah ini. Penelitian ini dibatasi hanya pada pemilihan *franchise* minuman dan penerapan Metode CoCoSo dalam konteks tersebut.

2. METODE PENELITIAN

Penelitian ini menggunakan metode kuantitatif, dimana metode kuantitatif adalah seperti yang dikemukakan oleh Sugiyono (2009:14) diartikan pada metode penelitian yang dilandasi pada filsafat positivisme, metode ini digunakan untuk penelitian pada populasi/sampel tertentu, umumnya teknik pengambilan sampel dilakukan secara *random*, instrumen penelitian digunakan untuk pengumpulan datanya, dan analisis data bersifat kuantitatif/statistik bertujuan untuk pengujian hipotesis yang sudah ditetapkan [14].

2.1 Metode CoCoSo

Metode *Combined Compromise Solution* (CoCoSo) merupakan metode yang dipublikasi dan diperkenalkan pada tahun 2019 oleh Morteza Yazdani dan rekan-rekannya [15]. CoCoSo merupakan salah satu metode MCDM yang dapat digunakan dalam pemilihan alternatif berdasarkan perhitungan bobot kriteria. Metode CoCoSo digunakan dalam penentuan *franchise* minuman karena lebih efektif dan efisien dalam proses perhitungan dan perangkaan.

Tahapan metode CoCoSo dalam pemilihan *franchise* minuman adalah sebagai berikut [16].

1. Identifikasi Kriteria Pemilihan

Identifikasi kriteria dilakukan dengan mengumpulkan faktor-faktor yang dipertimbangkan dalam pemilihan alternatif *franchise* minuman, pada penelitian ini menggunakan 5 kriteria sebagai berikut: *franchise fee*, *brand* lisensi, lama balik modal, reputasi *brand*, dan *royalty fee*.

2. Penentuan Bobot Kriteria

Berdasarkan kriteria-kriteria yang ditetapkan sebelumnya maka ditentukan bobot dari setiap kriteria seperti pada tabel 2.

Tabel 2. Data Kriteria

KODE KRITERIA	C1	C2	C3	C4	C5
NAMA KRITERIA	<i>Franchise Fee</i>	Brand Lisensi	Lama Balik Modal	Reputasi	<i>Royalty Fee</i>
ATRIBUT KRITERIA	<i>Cost</i>	<i>Benefit</i>	<i>Benefit</i>	<i>Benefit</i>	<i>Cost</i>
BOBOT KRITERIA	4	2.4	2.7	1.8	2.1

Tabel 2 di atas menunjukkan kode kriteria yang dibagi mulai dari C1-C5, dengan nama kriteria yang memuat *cost* dan *benefit* yang didapat, serta dibagi menjadi beberapa bobot kriteria.

3. Menghitung Matrik Ternormalisasi

Tahap normalisasi kriteria digunakan persamaan (1) untuk kriteria yang berjenis *benefit*, sedangkan persamaan (2) digunakan untuk kriteria yang berjenis *cost*.

$$r_{ij} = \frac{x_{ij} - \min_i x_{ij}}{\max_i x_{ij} - \min_i x_{ij}} \text{ kriteria } \textit{Benefit} \tag{1}$$

$$r_{ij} = \frac{\max_i x_{ij} - x_{ij}}{\max_i x_{ij} - \min_i x_{ij}} \text{ kriteria } \textit{Cost} \tag{2}$$

4. Menghitung Nilai Si dan Pi

Tahap perhitungan total nilai dari matrik perbandingan untuk seluruh bobot setiap alternatif digunakan persamaan (3) dan (4) berikut.

$$S_i = \sum_{j=1}^n (r_{ij} \times w_j) \tag{3}$$

$$P_i = \sum_{j=1}^n (r_{ij})^{w_j} \tag{4}$$

5. Menghitung Nilai Kia, Kib, dan Kic

Tahap perhitungan bobot relatif dari semua alternatif digunakan teknik agregasi, dimana digunakan tiga persamaan yaitu persamaan (5), (6), dan (7).

$$k_{ia} = \frac{S_i + P_i}{\sum_{i=1}^m (S_i + P_i)} \tag{5}$$

$$k_{ib} = \frac{S_i}{\min_i S_i} + \frac{P_i}{\min_i P_i} \tag{6}$$

$$k_{ic} = \frac{(\lambda)(S_i) + (1-\lambda)(P_i)}{((\lambda)\max_i S_i + (1-\lambda)\max_i P_i)}; 0 \leq \lambda \leq 1 \tag{7}$$

6. Menghitung Total Nilai Ki

Tahap perhitungan total dari nilai Ki dari setiap alternatif dilakukan menggunakan persamaan (8).

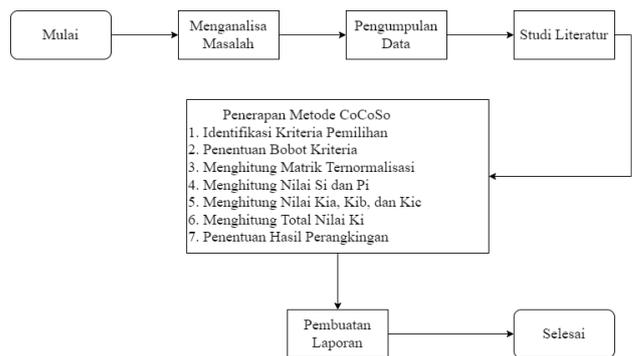
$$k_i = (k_{ia} \times k_{ib} \times k_{ic})^{1/3} + \frac{1}{3} (k_{ia} + k_{ib} + k_{ic}) \tag{8}$$

7. Penentuan Hasil Perangkingan

Tahapan metode CoCoSo diakhiri dengan perangkingan seluruh alternatif.

2.2 Tahapan Penelitian

Tahapan penelitian yang akan dilalui dalam penelitian ini, dapat dilihat pada kerangka penelitian pada gambar 1.



Gambar 1. Kerangka Penelitian

Kerangka penelitian pada gambar 1 di atas memuat alur tahapan penelitian yang dilakukan dengan mulai dari menganalisis masalah sampai tahap pembuatan laporan dan selesai.

a. Menganalisis Masalah

Analisis dilakukan sebagai awal dari penelitian, dengan analisis permasalahan yang ada, maka dapat ditentukan tujuan dari penelitian dan hasil yang diharapkan.

b. Pengumpulan Data

Tahap ini dilakukan proses pengumpulan data yang dibutuhkan dan sesuai dengan tujuan dan hasil akhir yang diharapkan dari berbagai sumber terpercaya.

c. Studi Literatur

Tahap ini dilakukan dengan mengkaji penelitian sejenis dan melakukan pengumpulan data melalui berbagai tindakan,

seperti membaca, mencatat, memahami dan mengelola data yang relevan untuk tujuan penelitian [17].

d. Penerapan Metode

Tahap penerapan metode akan dilakukan dengan menerapkan tahap perhitungan CoCoSo terhadap data yang telah dikumpulkan sebelumnya.

e. Laporan Penelitian

Setelah seluruh tahapan dilakukan, maka hasil dari penelitian dipaparkan dalam sebuah laporan penelitian.

3. HASIL DAN PEMBAHASAN

Hasil yang didapat berdasarkan langkah-langkah perhitungan menggunakan rumus Metode CoCoSo untuk pemilihan *franchise* minuman.

Tabel 3. Data Alternatif dan Kriteria

Alternatif	Kriteria				
	Franchise Fee	Brand Lisensi	Lama Balik Modal	Reputasi	Royalti Fee
	Cost	Benefit	Benefit	Benefit	Cost
	Juta	Tahun	Bulan	Linkert	Persen
Kopi Kenangan	350	2	24	5	5%
Es Teh Indonesia	120	4	6	5	5%
The poci	10	5	3	3	0%
Calais Tea	205	5	12	1	5%
Puyo puyo	100	5	24	2	5%
Gulu gulu	120	3	36	1	3%
Kopi kulo	70	3	12	3	0%
Xi boba	150	3	12	3	7%
Kopi Yor	75	3	13	2	0%

Tabel 3 di atas terbagi menjadi Alternatif dan Kriteria, yang mana pada Alternatif memuat semua *brand franchise* minuman yakni 9 alternatif, dan pada Kriteria memiliki 5 kriteria yang memuat 2 *cost* dan 3 *benefit* dan untuk hasilnya dapat dilihat pada tabel di atas.

Tabel 4. Skala Reputasi

Skala	Keterangan
1	Sangat Rendah
2	Rendah
3	Sedang
4	Tinggi
5	Sangat Tinggi

Berdasarkan tabel 4 di atas menunjukkan ada 5 skala dengan keterangan dari sangat rendah sampai sangat tinggi.

Matriks Keputusan awal dari proses pembobotan kriteria dan data alternatif, disajikan pada matriks di bawah ini:

$$X_{ij} = \begin{bmatrix} 350 & 2 & 24 & 5 & 5 \\ 120 & 4 & 6 & 5 & 5 \\ 10 & 5 & 3 & 3 & 0 \\ 205 & 5 & 12 & 1 & 5 \\ 100 & 5 & 24 & 2 & 5 \\ 120 & 3 & 36 & 1 & 3 \\ 70 & 3 & 12 & 3 & 0 \\ 150 & 3 & 12 & 3 & 7 \\ 75 & 3 & 13 & 2 & 0 \end{bmatrix}$$

Selanjutnya dilakukan normalisasi nilai kriteria, normalisasi ini dilakukan menggunakan persamaan (1) pada kriteria dengan jenis *benefit*.

$$r_{ij} = \frac{x_{ij} - \min_i x_{ij}}{\max_i x_{ij} - \min_i x_{ij}}$$

$$r_{A1,2} = \frac{2 - 2}{5 - 2} = 0,000$$

$$r_{A1,3} = \frac{24 - 3}{36 - 3} = 0,636$$

$$r_{A1,4} = \frac{5 - 1}{5 - 1} = 1,000$$

Sedangkan kriteria dengan jenis *cost* dihitung menggunakan persamaan(2).

$$r_{ij} = \frac{\max_i x_{ij} - x_{ij}}{\max_i x_{ij} - \min_i x_{ij}}$$

$$r_{A1,1} = \frac{350 - 350}{350 - 10} = 0,000$$

$$r_{A1,5} = \frac{7 - 5}{7 - 0} = 0,285$$

Hasil normalisasi kriteria disajikan pada tabel 5 data normalisasi alternatif atau *Normalization of criteria values dataset*.

Tabel 5. Data normalisasi alternatif (*Normalization of criteria values dataset*)

Kriteria	C1	C2	C3	C4	C5
Kode Alternatif	Franchise Fee (Juta)	Brand Lisensi (Tahun)	Lama Balik Modal (Bulan)	Reputasi	Royalti Fee
A1	0.0000	0.0000	0.6364	1.0000	0.2857
A2	0.6765	0.6667	0.0909	1.0000	0.2857
A3	1.0000	1.0000	0.0000	0.5000	1.0000
A4	0.4265	1.0000	0.2727	0.0000	0.2857
A5	0.7353	1.0000	0.6364	0.2500	0.2857
A6	0.6765	0.3333	1.0000	0.0000	0.5714
A7	0.8235	0.3333	0.2727	0.5000	1.0000
A8	0.5882	0.3333	0.2727	0.5000	0.0000
A9	0.8088	0.3333	0.3030	0.2500	1.0000

Tabel data normalisasi alternatif di atas menunjukkan bahwa kriteria C1 *franchise fee* dalam satuan juta, C2 *brand lisensi* dalam satuan tahun, C3 lama balik modal dengan

satuan bulan, sedangkan C4 reputasi dibuat dalam skala likert dan terakhir C5 royalti fee dengan satuan persen, dan untuk hasilnya dapat dilihat pada tabel di atas.

Dilanjutkan melakukan normalisasi alternatif terbobot (S_i) dan (P_i) menggunakan persamaan (3) dan (4).

$$S_i = \sum_{j=1}^n (r_{ij} \times w_j)$$

$$S_1 = \sum \left[\begin{matrix} (0,000 \times 30,77\%); & (0,000 \times 18,46\%); \\ (0,636 \times 20,77\%); & (1,000 \times 13,85\%); \\ & (0,285 \times 16,15\%) \end{matrix} \right]$$

$$S_1 = \sum \left[\begin{matrix} (0,000); & (0,000); \\ (0,132); & (0,138); \\ & (0,046) \end{matrix} \right] = 0,316$$

$$P_i = \sum_{j=1}^n (r_{ij})^{w_j}$$

$$P_1 = \sum \left[\begin{matrix} (0,000)^{30,77\%} & (0,000)^{18,46\%} \\ (0,636)^{20,77\%} & (1,000)^{13,85\%} \\ (0,285)^{16,15\%} & \dots \end{matrix} \right]$$

$$P_1 = \sum \left[\begin{matrix} (0,000)^{30,77\%} & (0,000)^{18,46\%} \\ (0,636)^{20,77\%} & (1,000)^{13,85\%} \\ (0,285)^{16,15\%} & \dots \end{matrix} \right]$$

Data hasil normalisasi alternatif terbobot disajikan pada tabel 6 data normalisasi alternatif terbobot atau *weighted comparability sequence* (S_i).

Tabel 6. Data normalisasi alternatif terbobot (*Weighted comparability Sequence* (S_i))

Kriteria	C1	C2	C3	C4	C5	
Kode Alternatif	Franchise Fee (Juta)	Brand Lisensi (Tahun)	Lama Balik Modal (Bulan)	Reputasi	Royalti Fee	S (i)
A1	0.0000	0.0000	0.1322	0.1385	0.0462	0.3168
A2	0.2081	0.1231	0.0189	0.1385	0.0462	0.5347
A3	0.3077	0.1846	0.0000	0.0692	0.1615	0.7231
A4	0.1312	0.1846	0.0566	0.0000	0.0462	0.4186
A5	0.2262	0.1846	0.1322	0.0346	0.0462	0.6238
A6	0.2081	0.0615	0.2077	0.0000	0.0923	0.5697
A7	0.2534	0.0615	0.0566	0.0692	0.1615	0.6023
A8	0.1810	0.0615	0.0566	0.0692	0.0000	0.3684
A9	0.2489	0.0615	0.0629	0.0346	0.1615	0.5695
					SUM	4.7269
					MAX	0.7231
					MIN	0.3168

Tabel di atas menunjukkan hasil perhitungan yang dilakukan di Ms. Excel, yang memuat nilai *max* 0.7231 dan *min* 0.3168.

Data normalisasi alternatif terbobot atau *Exponentially weighted comparability sequence* (P_i) yang dapat dilihat dari tabel 7 di bawah ini.

Tabel 7. Data normalisasi alternatif terbobot (*Exponentially weighted comparability sequence* (P_i))

Kriteria	C1	C2	C3	C4	C5	
Kode Alternatif	Franchise Fee (Juta)	Brand Lisensi (Tahun)	Lama Balik Modal (Bulan)	Reputasi	Royalti Fee	P (i)
A1	0.0000	0.0000	0.9104	1.0000	0.8168	2.7272
A2	0.8867	0.9279	0.6077	1.0000	0.8168	4.2391
A3	1.0000	1.0000	0.0000	0.9085	1.0000	3.9085
A4	0.7693	1.0000	0.7635	0.0000	0.8168	3.3496
A5	0.9097	1.0000	0.9104	0.8253	0.8168	4.4623
A6	0.8867	0.8164	1.0000	0.0000	0.9136	3.6167
A7	0.9420	0.8164	0.7635	0.9085	1.0000	4.4304
A8	0.8494	0.8164	0.7635	0.9085	0.0000	3.3378
A9	0.9368	0.8164	0.7804	0.8253	1.0000	4.3590
					SUM	34.4305
					MAX	4.4623
					MIN	2.7272

Tabel di atas menunjukkan hasil perhitungan yang dilakukan di Ms. Excel, yang memuat nilai *max* 4.4623 dan *min* 2.7272.

Selanjutnya dilakukan perhitungan bobot relatif seluruh alternatif dengan persamaan (5), (6) dan (7).

$$k_{ia} = \frac{S_i + P_i}{\sum_{i=1}^m (S_i + P_i)}$$

$$k_{1a} = \frac{0.316 + 2.727}{\sum_{i=1}^m (4.726 + 34.430)} = 0,077$$

$$k_{ib} = \frac{S_i}{\min_i S_i} + \frac{P_i}{\min_i P_i}$$

$$k_{ib} = \frac{0.316}{0.316} + \frac{2.727}{2.727} = 2,000$$

$$k_{ic} = \frac{(\lambda)(S_i) + (1 - \lambda)(P_i)}{((\lambda)^{\max_i S_i} + (1 - \lambda)^{\max_i P_i})}; 0 \leq \lambda \leq 1$$

$$k_{ic} = \frac{(0,5)(0,316) + (1 - 0,5)(2,717)}{((0,5)0,723 + (1 - 0,5)4,462)} = 0,587$$

Bobot relatif dari data alternatif yang dilakukan seperti pada tabel 8.

Tabel 8. Bobot relatif dari *dataset* alternatif (*Relative weights of the alternatives dataset*)

Kode Alternatif	Nama Alternatif	Ka	Kb	Kc
A1	Kopi	0.0777	2.0000	0.5870
A2	Kenangan Es The Indonesia	0.1219	3.2423	0.9206
A3	The Poci	0.1183	3.7157	0.8932
A4	Calais Tea	0.0962	2.5498	0.7267
A5	Puyo Puyo	0.1299	3.6054	0.9809

Kode Alternatif	Nama Alternatif	Ka	Kb	Kc
A6	Gulu Gulu	0.1069	3.1245	0.8073
A7	Kopi Kulo	0.1285	3.5260	0.9706
A8	Xi Boba	0.0946	2.3869	0.7147
A9	Kopi Yor	0.1259	3.3961	0.9505

Pada tabel 8 di atas ini menunjukkan setiap kode alternatif mulai dari A1-A9, dengan nama alternatif setiap *brand franchise* minuman yang di-list, dengan kriteria Ka, Kb dan Kc, dengan semua nilai yang dihasilkan.

Tahap perhitungan diakhiri dengan penentuan rangking setiap alternatif yang dibuat dengan menggunakan persamaan (8). Hasil perangkingan dapat dilihat pada tabel 9 *rangking* final dari alternatif atau *final rangking of the alternative Dataset*.

Tabel 9. *Ranking* final dari data alternatif (*Final ranking of the alternatif Dataset*)

Kode Alternatif	Nama Alternatif	Nilai Preferensi	Rank
A1	Kopi Kenangan	1.3385	9
A2	Es The Indonesia	2.1422	5
A3	The Poci	2.3079	2
A4	Calais Tea	1.6871	7
A5	Puyo Puyo	2.3436	1
A6	Gulu Gulu	1.9923	6
A7	Kopi Kulo	2.3022	3
A8	Xi Boba	1.6100	8
A9	Kopi Yor	2.2314	4

Tabel di atas menunjukkan nilai preferensi dan *rank* dari setiap alternatif yang menghasilkan perangkingan 1-9, yang mana pada posisi pertama diraih oleh Puyo-Puyo dengan nilai preferensi sebesar 2.3436 dan posisi terakhir jatuh pada Kopi Kenangan dengan nilai preferensi sebesar 1.3385.

Hasil dari Nilai Preferensi Menggunakan metode CoCoSo dapat juga dilihat pada gambar 2 di bawah ini.



Gambar 2. *Chart* nilai preferensi menggunakan Metode CoCoSo

Setelah dilakukannya langkah-langkah perhitungan sesuai dengan metode CoCoSo maka diketahui dari beberapa alternatif yang dimiliki, *franchise brand* PUYO PUYO merupakan *franchise* minuman yang lebih menguntungkan untuk dilakukan Kerjasama.

Berdasarkan langkah-langkah dari perhitungan yang sudah dilakukan maka dapat diketahui dari beberapa alternatif yang dimiliki, waralaba PUYO-PUYO menunjukkan hasil lebih banyak potensi daripada alternatif lainnya. Hal ini dapat dilihat dari kriteria C3 Normalisasi Alternatif Terbobot dimana alternatif 5 yaitu PUYO-PUYO lebih unggul dari Alternatif 3 yaitu Teh Poci yang menempati posisi kedua yang terjadi karena jangka waktu yang dibutuhkan dalam pengembalian modal sejumlah alternatif 5 lebih lama. Pada proses penelitian pemilihan *brand franchise* minuman menggunakan metode CoCoSo ini, penulis menghadapi beberapa kendala pada saat pengumpulan data-data dari setiap *brand* minuman yang digunakan, dimana tidak semua *brand* menyajikan data yang dibutuhkan secara publik.

4. KESIMPULAN

Dengan menerapkan metode *Combined Compromise Solution* (CoCoSo) pada penentuan pemilihan *franchise* minuman ini akan menghasilkan perbandingan dari 9 alternatif *franchise* minuman di Indonesia yang disajikan penulis dalam jurnal ini. Sehingga melalui sistem pendukung pengambilan keputusan yang telah dikembangkan dapat dihasilkan nilai dari *franchise* minuman berdasarkan dengan kriteria yang telah ditentukan. Perhitungan *franchise* minuman dengan menggunakan *Combined Compromise Solution* (CoCoSo) menunjukkan bahwa hasil perhitungan nilai preferensi tertinggi didapatkan oleh *franchise* minuman Puyo Puyo dengan nilai 2,3436 dan nilai preferensi terendah didapatkan oleh *franchise* minuman Kopi Kenangan dengan nilai 1,3385.

Ucapan Terima Kasih

Penulis mengucapkan terima kasih dan puji syukur kepada Tuhan Yang Maha Esa karena kasih karunianya penulis bisa menyelesaikan penelitian ini. Ucapan terima kasih kepada bapak Gede Surya Mahendra, S.Pd., M.Kom. selaku dosen pengampu mata kuliah SPPK dan pembimbing selama penelitian serta ucapan terima kasih pada rekan seperjuangan dalam pembuatan artikel ini.

DAFTAR PUSTAKA

- [1] S. Redjeki *et al.*, "Waralaba (Franchise) Di Indonesia," *Waralaba di Indones. Lex Jurnalica*, vol. 8, no. 2, p. 127, 2011.
- [2] V. Alapján-, "濟無No Title No Title No Title," pp. 1–23, 2016.
- [3] P. S. Akuntansi, "1* , 2 1,2," vol. 20, no. 1, pp. 105–123, 2022.
- [4] Najla, Y. Dewanthi, and E. Permana, "Strategi Mempertahankan Usaha Franchise Pada Minuman Mixue di Indonesia," *J. Business, Financ. Econ.*, vol. 3, no. 2, pp. 189–198, 2023, doi: 10.32585/jbfe.v3i2.4099.
- [5] C. Maulana, A. Hendrawan, and P. A. R. Pinem, "Pemodelan Penentuan Kredit Simpan Pinjam

- Menggunakan Metode Additive Ratio Assessment (ARAS),” *Pengemb. Rekayasa dan Teknol.*, vol. 15, no. 1, pp. 7–11, 2019, doi: <http://dx.doi.org/10.26623/jprt.v15i1.1483>.
- [6] I. N. A. A. Dwijayadi, “Pengembangan Sistem Pendukung Keputusan Penentuan Hotel Di Kecamatan Buleleng Dengan Metode Analytic Hierarchy Process (Ahp) Dan Technique for Others Reference By Similarity To Ideal Solution (Topsis),” *J. Nas. Pendidik. Tek. Inform.*, vol. 7, no. 2, p. 163, 2019, doi: [10.23887/janapati.v7i2.13435](https://doi.org/10.23887/janapati.v7i2.13435).
- [7] A. D. Wahyudi, “Analisis Kepuasan Terhadap Pelayanan Supplier Menggunakan Metode A New Additive Ratio Assessment (ARAS),” *J. Artificial Intell. Tecno. Inf.*, vol. 2, no. 1, pp. 1–13, 2024, doi: <https://doi.org/10.58602/jaiti.v2i1.105>.
- [8] H. Saputra and T. Ardiansah, “Penerapan Combined Compromise Solution (CoCoSo) Method Dalam Sistem Pendukung Keputusan Pemilihan Modem,” *J. Ilm. Comput. Sci.*, vol. 1, no. 1, pp. 7–16, 2022, [Online]. Available: <https://doi.org/10.58602/jics.v1i1.2>.
- [9] S. J. Bulan and S. J. Bulan, “Penerapan Analytical Hierarchy Process (Ahp) Dalam Perangkingan Bengkel Mobil Terbaik Di Kota Kupang,” *J. Teknol. Terpadu*, vol. 5, no. 1, pp. 5–9, 2019, doi: [10.54914/jtt.v5i1.189](https://doi.org/10.54914/jtt.v5i1.189).
- [10] A. R. Andreyany, “Dosen Pengampu : Yananto Mihadi Putra , SE , M . Si,” *Sist. Basis Data*, no. June, 2020.
- [11] D. U. Iswavigra and L. E. Zen, “Systematic Literature Review: Pengaplikasian Metode VIKOR dalam Decision Support System,” *J. Inf. dan Teknol.*, vol. 5, no. 3, pp. 13–19, 2023, doi: [10.60083/jidt.v5i3.379](https://doi.org/10.60083/jidt.v5i3.379).
- [12] A. P. Aulia Manurung, M. Amin, and H. Herdianto, “Rancang Bangun Aplikasi Sistem Pendukung Keputusan Cleaning Service Menggunakan Metode Fuzzy Multiple Attribute Decision Making,” *J. Sistim Inf. dan Teknol.*, vol. 5, no. 3, pp. 52–62, 2023, doi: [10.60083/jsisfotek.v5i3.285](https://doi.org/10.60083/jsisfotek.v5i3.285).
- [13] H. F. Amran, F. I. Komputer, U. M. Riau, U. Islam, N. Sustan, and S. Kasim, “Jurnal Software Engineering and Information Systems (SEIS),” vol. 3, no. 1, pp. 1–6, 2023.
- [14] K. Abdullah *et al.*, “Metodologi Penelitian Kuantitatif Metodologi Penelitian Kuantitatif”, no. May. 2021.
- [15] M. Yazdani, P. Zarate, E. K. Zavadskas, and Z. Turskis, “A combined compromise solution (CoCoSo) method for multi-criteria decision-making problems,” *Manag. Decis.*, vol. 57, no. 9, pp. 2501–2519, 2019, doi: [10.1108/MD-05-2017-0458](https://doi.org/10.1108/MD-05-2017-0458).
- [16] Y. M. Kristania, “Penerapan Combined Compromise Solution Method Dalam Penentuan Penerima Beasiswa,” *J. Comput. Technol. Comput. Eng. Informatics*, vol. 1, no. 2, pp. 44–55, 2023.
- [17] J. Informatika and S. Informasi, “INFORMASI (Jurnal Informatika dan Sistem Informasi) Volume 15 No.1 / Mei / 2023,” *Inf. (Jurnal Inform. dan Sist. Informasi)*, vol. 15, no. 1, pp. 32–45, 2023.



PERBANDINGAN KLASIFIKASI LABEL TUNGGAL UNTUK SOAL UJIAN FISIKA MENGGUNAKAN NAÏVE BAYES DAN *K-FOLD CROSS VALIDATION*

Christopher Kevin Herijanto¹, Yulia Wahyuningsih²

^{1,2} Ilmu Informatika, Universitas Katolik Darma Cendika
Surabaya, Jawa Timur, Indonesia 60117
Christopher.herijanto@student.ukdc.ac.id, yulia@ukdc.ac.id

Abstract

This research evaluates the use of the Naïve Bayes algorithm in classifying Physics questions with single labels. The main objective is to identify the best algorithm for classifying Physics questions to assist high school students with difficulty understanding them. The research method involves using a dataset containing Physics questions that need to be classified to facilitate learning for high school students. The Naïve Bayes algorithm is implemented using Google Colab to train the classification model using features extracted from the text of the Physics questions. Additionally, several other classification algorithms, such as Support Vector Machine (SVM), Logistic Regression, Decision Tree, and Random Forest, are tested, and their performance is compared. Experimental results show that Naïve Bayes provides competitive results in classifying single-label Physics questions. However, there are significant performance differences between Naïve Bayes and other algorithms, depending on the type and complexity of the classified Physics problems. In this study, SVM achieved higher accuracy, but Naïve Bayes excelled in training time. This research provides a deeper understanding of the strengths and weaknesses of Naïve Bayes in solving the task of classifying single-label Physics problems. These findings guide the development of more accurate classification models for application in the context of Physics learning.

Keywords: Machine Learning, Naïve Bayes, Physics, Random Forest, Single Label Classification

Abstrak

Penelitian ini mengevaluasi penggunaan algoritma Naïve Bayes dalam mengklasifikasikan soal Fisika dengan label tunggal. Tujuan utamanya adalah untuk mendapatkan algoritma terbaik untuk mengklasifikasikan soal fisika yang akan digunakan untuk siswa SMA yang kesulitan untuk memahaminya. Metode penelitian ini menggunakan *dataset* yang berisi soal-soal Fisika yang perlu diklasifikasikan untuk memudahkan pembelajaran siswa SMA. Algoritma Naïve Bayes diimplementasikan menggunakan *Google Collab* untuk melatih model klasifikasi menggunakan fitur-fitur yang diekstraksi dari teks soal Fisika. Selain itu, beberapa algoritma klasifikasi lain seperti *Support Vector Machine (SVM)*, *Logistic Regression*, *Decision Tree*, dan *Random Forest* juga diuji dan dibandingkan kinerjanya. Hasil eksperimen menunjukkan bahwa Naïve Bayes memberikan hasil yang bersaing dalam klasifikasi soal Fisika label tunggal. Namun, terdapat perbedaan performa yang signifikan antara Naïve Bayes dan algoritma lainnya, bergantung pada jenis dan kompleksitas masalah soal Fisika yang diklasifikasikan. Pada penelitian ini, SVM memiliki nilai akurasi yang lebih tinggi, tetapi Naïve Bayes unggul dalam *training time*. Penelitian ini memberikan pemahaman lebih dalam tentang kelebihan dan kelemahan Naïve Bayes dalam menyelesaikan tugas klasifikasi masalah Fisika label tunggal. Temuan ini menjadi panduan untuk mengembangkan model klasifikasi yang lebih akurat untuk aplikasi dalam konteks pembelajaran Fisika.

Kata kunci: Fisika, Klasifikasi Label Tunggal, Naïve Bayes, Pembelajaran Mesin, *Random Forest*

1. PENDAHULUAN

Klasifikasi teks adalah proses mengklasifikasikan dokumen ke dalam kategori yang berbeda[1]. Klasifikasi teks memiliki banyak fungsi, seperti klasifikasi topik, analisis sentimen, deteksi *spam*, dan pemfilteran informasi. Dalam *machine learning*, proses klasifikasi teks dapat dibagi menjadi empat bagian umum: ekstraksi fitur, reduksi

dimensi, pemilihan pengklasifikasi, dan penilaian. Proses ekstraksi fitur mengubah data teks menjadi format yang dapat digunakan di komputer. Banyak metode yang telah dikembangkan untuk klasifikasi, seperti Naïve Bayes, *Random Forest*, *SVM*, *Decision Tree*, *Logistic Regression*, dan algoritma lain yang dapat digunakan untuk klasifikasi. Mengumpulkan kumpulan data multi-label adalah hal yang

lebih sulit dan kurang terukur dibandingkan mengumpulkan satu label *dataset*, karena pengumpulannya konsisten dan menyeluruh daftar label untuk setiap gambar memerlukan usaha yang signifikan[2].

Klasifikasi terhadap soal Fisika dibutuhkan karena adanya kesulitan bagi siswa SMA untuk memahami soal-soal fisika yang dikarenakan ada banyaknya rumus dan konsep yang harus dipahami[3]. Maka dari itu, klasifikasi sangat bermanfaat untuk memberikan solusi yang lebih efisien pada tingkat pemahaman siswa terhadap soal fisika.

Naïve Bayes adalah metode klasifikasi yang mengasumsikan variabel tidak bergantung satu sama lain [4][5]. Asumsi ini menjadi dasar proses klasifikasi Naïve Bayes. Naïve Bayes dikembangkan oleh Thomas Bayes sebagai metode pembelajaran terawasi berbasis teori. Naïve Bayes juga menjadi metode pengambilan keputusan dengan perhitungan probabilitas[6]. Naïve Bayes sendiri untuk klasifikasi teks bisa dikatakan sebagai metode dengan hasil yang baik dengan kalkulasi yang tidak rumit. Model pembelajaran ini mengasumsikan bahwa variabel-variabelnya bersifat independen. Misalnya kelas c pada himpunan kelas C mempunyai fitur t , dimana t merupakan subset dari himpunan fitur T . Oleh karena itu, probabilitas kelas c dengan fitur t mengikuti teori *Bayesian* pada Rumus 1[1].

$$\Pr(c|t) = \frac{\Pr(c)\Pr(t|c)}{\Pr(t)} \quad (1)$$

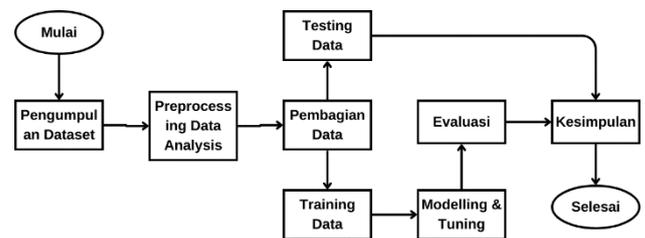
Logistic Regression adalah model linier yang biasa digunakan untuk klasifikasi biner yang menggunakan fungsi logistik untuk memodelkan ketergantungan biner *variable* [5][7]. *Random Forest* adalah algoritma pembelajaran ansambel yang menggunakan dan membangun struktur pohon secara bertahap. Jika digunakan, pohon keputusan dibuat dengan memilih atau mengambil sampel data secara acak. Untuk menentukan kelas data, *Random Forest* menggunakan sistem *voting* sebanyak hasil berdasarkan pohon keputusan[7]. *Support Vector Machine* (SVM) adalah kumpulan teknik pembelajaran terawasi yang membuat *hyperplanes* atau serangkaian *hyperplanes* dalam proses klasifikasi, regresi, dan deteksi *outlier*. Salah satu kegunaannya adalah untuk mengelompokkan teks dan *hypertext*. Kelebihan SVM ini adalah efektif pada ruang berdimensi tinggi, efektif bila jumlah dimensi lebih besar dibandingkan jumlah sampel, dan hemat memori karena menggunakan sub set titik pelatihan[8].

Menurut penelitian sebelumnya, dari hasil analisis seluruh pengujian yang dilakukan pada penelitian klasifikasi multi-label untuk topik berita Indonesia, dapat diambil kesimpulan model klasifikasi multi-label menggunakan metode *multinomial* Naïve Bayes yang memberikan hasil paling rendah. Pada penelitian tersebut, didapatkan hasil dari metode *hamming loss* yaitu 0,18. Selanjutnya model *multinomial* Naïve Bayes dengan menggunakan teknik *stemming* pada bagian *preprocessing* mencapai hasil yang lebih baik dibandingkan tanpa menggunakan teknik *stemming* pada bagian *preprocessing*[9]. Sebagai contoh

penerapan algoritma Naïve Bayes, hasil perbandingan penerapan algoritma klasifikasi *K-Nearest Neighbors* dan Naïve Bayes *Classifier* pada analisis sentimen dari dataset opini pengguna Twitter mengenai infeksi virus corona baru, algoritma *K-Nearest Neighbors* mencapai akurasi sebesar 72,37%, sedangkan algoritma Naïve Bayes mencapai akurasi sebesar 67,84% [10].

Selanjutnya, apakah algoritma Naïve Bayes juga merupakan algoritma yang efisien dalam membangun sistem klasifikasi untuk mengklasifikasikan data masalah fisik ke dalam label, dan apakah algoritma Naïve Bayes merupakan model yang baik untuk masalah klasifikasi multi-label. Timbul pertanyaan apakah algoritma tersebut bisa berhasil beradaptasi dengan Oleh karena itu, fokus penelitian ini adalah mengembangkan sistem klasifikasi teks multi-label pada data permasalahan fisika dengan mengimplementasikan algoritma Naïve Bayes dan menganalisis pengaruh jenis teknik transformasi permasalahan yang digunakan. Hasil klasifikasi yang dihasilkan juga dianalisis untuk memahami sumber kesalahan model saat memprediksi kelas dan label dari data. Metode evaluasi yang digunakan dalam penelitian ini adalah perbandingan hasil pelatihan dan tes serta *K-fold cross validation*.

2. METODE PENELITIAN



Gambar 1. Tahapan Metode Penelitian

Tahapan metode penelitian seperti yang terlihat pada gambar 1 dilakukan dari pengumpulan *dataset*, studi literatur, menentukan *basic statistic*, melakukan *preprocessing & exploratory* data, melakukan SMOTE (*Synthetic Minority Oversampling Technique*) atau *balancing dataset*, dan pengujian model. Terdapat 4 teknik pengolahan data yang digunakan yaitu *case folding*, *stopword removal*, *tokenizing* dan *stemming*. Setelah itu dilakukan pengujian dengan melihat hasil dari akurasi, *training time*, dan *prediction time*.

2.1 Dataset Collection

Studi kasus ini berfokus terutama pada ujian tingkat sekolah dasar di Indonesia, dan memerlukan data primer untuk dipusatkan pada soal-soal ujian fisika yang cocok untuk siswa SMA. *Input*-nya berupa data diekstrak dari sekolah-sekolah di Surabaya. *Dataset* yang digunakan adalah kumpulan soal-soal Fisika sebanyak 583 soal.

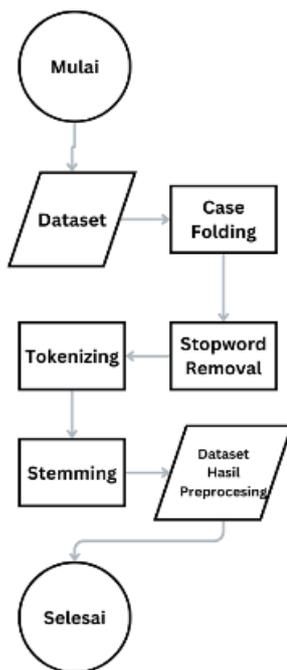
2.2 Klasifikasi

Suatu metode dalam *data mining* yang beroperasi dengan mengidentifikasi karakteristik dari kelompok data yang telah diketahui dan ditentukan sebelumnya disebut sebagai klasifikasi. Klasifikasi memiliki salah satu ciri khas, yaitu adanya kelas atau target yang ditetapkan. Teknik ini berfungsi dengan cara mengelompokkan data ke dalam kelas atau target tertentu untuk membentuk suatu model atau aturan[11]. Model atau aturan yang telah dibentuk ini kemudian dapat digunakan untuk mengklasifikasikan data baru ke dalam kelas atau target yang sesuai pada periode yang akan datang.

2.3 Basic Statistics

Pada proses ini, dilakukan perhitungan *statistic* deskriptif terhadap *dataset* untuk mengetahui nilai pemusatan data (*mean, median, modus*), ukuran letak data (kuartil dan desil), dan *standard deviation*. Statistik deskriptif pada dasarnya digunakan untuk memberikan pemahaman yang lebih mendalam tentang karakteristik dari *dataset* yang digunakan dalam klasifikasi label tunggal. Statistik deskriptif dapat membantu dalam menilai kebersihan data, mendeteksi nilai-nilai yang hilang, atau *outlier* yang mungkin mempengaruhi kualitas model klasifikasi.

2.4 Preprocessing



Gambar 2. Tahapan Preprocessing

Dataset yang digunakan pada tahapan seperti yang terlihat di gambar 2 ini berisi data yang masih belum terstruktur. Pada fase ini data akan dibersihkan terlebih dahulu dan mendapatkan hasil yaitu data yang sudah terstruktur. Penerapan metode *preprocessing* pada kumpulan data tertentu memungkinkan penyempurnaan teks secara signifikan kualitas klasifikasi dengan peningkatan yang dapat dicapai hingga 80%. Ada beberapa proses yang harus dilakukan yaitu *case folding, stopwords removal, tokenizing* dan *stemming*.

Untuk proses *case folding*, *input*-nya berupa teks lengkap artikel, dan *output*-nya berupa teks dengan rata-rata huruf kecil tanpa tanda baca. Hal ini dilakukan untuk menghindari ketidaksamaan nilai yang terjadi ketika menggunakan huruf besar dan kecil dalam kata yang sama[8][9]. Contoh hasil dari proses *case folding* dapat dilihat di tabel 1.

Tabel 1. Hasil dari Proses Case Folding

Soal	Contoh Hasil Proses Case Folding
\tSebuah bola besi digantungkan pada langit...	sebuah bola besi digantungkan pada langit lang...
\tYang bukan merupakan pasangan gaya aksi...	yang bukan merupakan pasangan gaya aksi reaksi...
\tPerhatikan pilihan gambar tentang seorang...	perhatikan pilihan gambar tentang seorang anak...
\tPada gambar di bawah pasangan...	pada gambar di bawah pasangan gaya aksi dan re...
\tSebuah benda digantungkan pada langit-langit...	sebuah benda digantungkan pada langit langit...

Setelah itu dilakukan proses *stopword removal* yang merupakan proses penghapusan kata-kata yang dianggap berdampak kecil, seperti kata yang memiliki tingkat kemunculan yang lebih tinggi dibandingkan dengan kata-kata lainnya[12]. Contoh hasil dari proses *stopword removal* dapat dilihat di tabel 2.

Tabel 2. Hasil dari Proses Stopword Removal

Soal	Contoh Hasil Proses Stopword Removal
\tSebuah bola besi digantungkan pada langit...	bola, besi, digantungkan, langit
\tYang bukan merupakan pasangan gaya aksi...	pasangan, gaya, aksi, reaksi
\tPerhatikan pilihan gambar tentang seorang...	perhatikan, pilihan, gambar, anak, ditarik
\tPada gambar di bawah pasangan...	gambar, pasangan, gaya, aksi, reaksi
\tSebuah benda digantungkan pada langit-langit...	benda, digantungkan, langit, langit, gambar

Tokenizing adalah proses pemecahan kalimat dari setiap dokumen pada *dataset* yang sudah dilakukan pembersihan ke dalam kata per kata[13]. Contoh hasil dari proses *tokenizing* dapat dilihat di tabel 3.

Tabel 3. Hasil dari Proses Tokenizing

Soal	Contoh Hasil Proses Tokenizing
\tSebuah bola besi digantungkan pada langit...	sebuah, bola, besi, digantungkan, pada, langit
\tYang bukan merupakan pasangan gaya aksi...	yang, bukan, merupakan, pasangan, gaya, aksi...

Soal	Contoh Hasil Proses <i>Tokenizing</i>
\tPerhatikan pilihan gambar tentang seorang...	perhatikan, pilihan, gambar, tentang, seorang...
\tPada gambar di bawah pasangan...	pada, gambar, di, bawah, pasangan, gaya, aksi...
\tSebuah benda digantungkan pada langit-langit...	sebuah, benda, digantungkan, pada, langit...

Stemming adalah suatu proses untuk mengubah kata menjadi bentuk dasar. Contoh hasil dari proses *stemming* dapat dilihat di tabel 4.

Tabel 4. Hasil dari Proses *Stemming*

Soal	Contoh Hasil Proses <i>Stemming</i>
\tSebuah bola besi digantungkan pada langit...	bola, besi, gantung, langit, langit
\tYang bukan merupakan pasangan gaya aksi...	pasang, gaya, aksi, reaksi
\tPerhatikan pilihan gambar tentang seorang...	perhati, pilih, gambar, anak, tarik, mobil
\tPada gambar di bawah pasangan...	gambar, pasang, gaya, aksi, reaksi
\tSebuah benda digantungkan pada langit-langit...	benda, gantung, langit, langit, gambar, gue

2.5 Accuracy

Akurasi atau *accuracy* setiap bagian data adalah persentase label yang diprediksi dengan benar dibagi dengan jumlah total label yang diprediksi dan label sebenarnya dari bagian data tersebut. Presisi keseluruhan adalah nilai presisi rata-rata setiap titik data dibagi dengan jumlah titik data [14].

2.6 K-Fold Cross Validation

K-Fold adalah metode validasi silang umum yang melipat data sebanyak k kali dan mengulanginya sebanyak k kali. Validasi silang *K-fold* membagi data menjadi segmen yang sama. Satu segmen digunakan sebagai data pelatihan dan segmen lainnya digunakan sebagai data pengujian. Bedanya, metode ini mengharuskan seluruh data yang terdapat dalam *dataset* digunakan baik sebagai data latih maupun data uji. Validasi silang membagi semua data menjadi k kelompok. Jumlah tersebut adalah *record* dibagi dengan jumlah k yang digunakan [15].

3. HASIL DAN PEMBAHASAN

Dalam menentukan parameter terbaik dari metode yang digunakan, penulis melakukan tiga skenario pengujian terhadap model klasifikasi yang dibangun, yaitu sebagai berikut:

3.1 Skenario 1

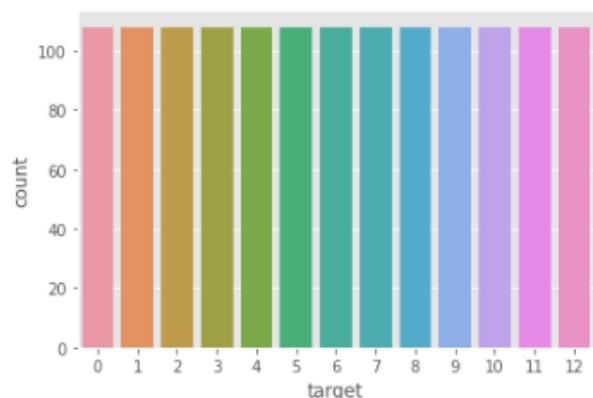
Analisis terhadap algoritma Naïve Bayes, SVM, *Random Forest*, *Decision Tree*, dan *Logistic Regression* dengan *dataset* soal-soal fisika tanpa dilakukan SMOTE atau *data balancing* terhadap *dataset*. Data yang didapatkan dari analisis ini berupa *training time*, *prediction time*, *testing accuracy*, dan *training accuracy*.

Tabel 5. Hasil Pengujian Tanpa SMOTE

	<i>Training Time</i>	<i>Prediction Time</i>	<i>Testing Accuracy</i>	<i>Training Accuracy</i>
Naïve Bayes	0.003	0.0006	0.61	0.71
SVM	0.045	0.0005	0.68	0.88
Random Forest	0.302	0.0162	0.74	0.95
Decision Tree	0.030	0.0008	0.68	0.95
Logistic Regression	0.262	0.0018	0.74	0.89

Berdasarkan tabel 5, Algoritma Naïve Bayes memiliki *training time* tercepat dengan nilai 0.003 detik. Untuk *prediction time* tercepat diperoleh oleh algoritma SVM dengan nilai 0.0005 detik. Untuk *testing accuracy*, ada dua algoritma yang memiliki hasil yang sama yaitu algoritma *Random Forest* dan *Logistic Regression* yang memiliki nilai sebesar 74%. Untuk *training accuracy*, ada dua algoritma yang memiliki hasil yang sama yaitu algoritma *Random Forest* dan *Decision Tree* yang memiliki nilai sebesar 95%.

3.2 Skenario 2



Gambar 3. Proses SMOTE (*Data Balancing*)

Analisis terhadap algoritma Naïve Bayes, SVM, *Random Forest*, *Decision Tree*, dan *Logistic Regression* dengan *dataset* soal-soal fisika dengan dilakukan *smote* atau *data balancing* terhadap *dataset*. Data yang didapatkan dari analisis ini berupa *training time*, *prediction time*, *testing accuracy*, dan *training accuracy* yang diperoleh dari *dataset* yang sudah dilakukan *data balancing* atau SMOTE. Untuk

Hasil Proses SMOTE atau *data balancing* bisa dilihat di gambar 3.

Tabel 6. Hasil Pengujian dengan SMOTE (*Data Balancing*)

	<i>Training Time</i>	<i>Prediction Time</i>	<i>Testing Accuracy</i>	<i>Training Accuracy</i>
Naïve Bayes	0.009	0.0017	0.80	0.91
SVM	0.029	0.0007	0.88	0.990
RandomFo rest	0.290	0.0163	0.85	0.996
Decision Tree	0.127	0.0008	0.77	0.997
Logistic Regression	0.844	0.0009	0.89	0.95

Berdasarkan tabel 6, Algoritma Naïve Bayes memiliki *training time* tercepat dengan nilai 0.009 detik. Untuk *prediction time* tercepat diperoleh oleh algoritma SVM dengan nilai 0.0005 detik. Untuk *testing accuracy*, ada dua algoritma yang memiliki hasil yang sama yaitu algoritma *Random Forest* dan *Logistic Regression* yang memiliki nilai sebesar 74%. Untuk *training accuracy*, ada dua algoritma yang memiliki hasil yang sama yaitu algoritma *Random Forest* dan *Decision Tree* yang memiliki nilai sebesar 95%.

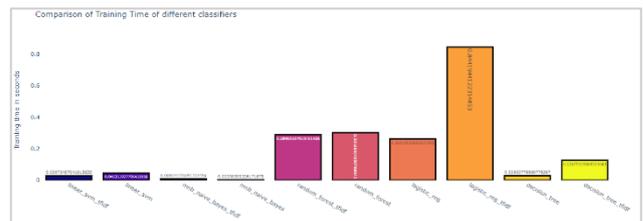
3.3 Skenario 3

Analisis terhadap algoritma Naïve Bayes, SVM, *Random Forest*, *Decision Tree*, dan *Logistic Regression* dengan dataset soal-soal fisika dengan dilakukan SMOTE atau data balancing terhadap dataset. Pada skenario ini, terdapat metode tambahan yaitu metode K-Folds Cross Validation dengan *k* sebanyak 10 untuk menguji *dataset* dengan tingkat akurasi yang lebih tinggi.

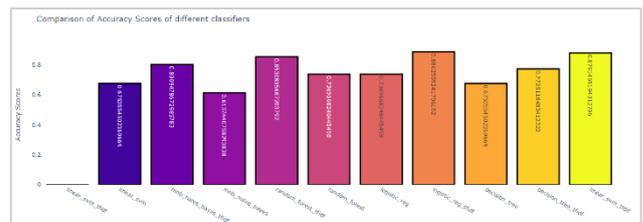
Tabel 7. Hasil Pengujian K-Fold Cross Validation dengan SMOTE

	<i>Maximum Accuracy</i>	<i>Minimum Accuracy</i>	<i>Overall Accuracy</i>
Naïve Bayes	86.52%	79.43%	83.12%
SVM	94.33%	88.57%	91.52%
RandomForest	90.71%	85%	88.68%
Decision Tree	82.86%	72.34%	78.78%
Logistic Regression	-	-	-

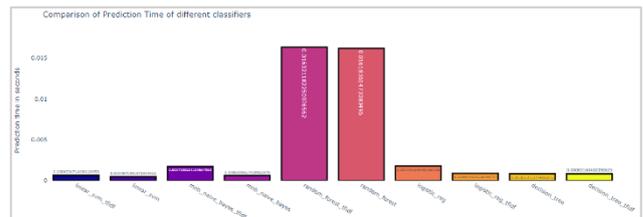
Berdasarkan tabel 7, Algoritma SVM memiliki *maximum accuracy* tertinggi dengan nilai 94.33% dibandingkan dengan Naïve Bayes yang memiliki nilai 86.52%. Untuk *minimum accuracy* tertinggi juga diperoleh oleh algoritma SVM dengan nilai 88.57% dibandingkan dengan Naïve Bayes yang memiliki nilai 79.43%. Untuk *overall accuracy*, algoritma SVM memiliki hasil tertinggi dengan nilai 91.52% dibandingkan dengan Naïve Bayes dengan nilai 83.12%. Berdasarkan hasil dari pengujian K-Fold Cross Validation, Naïve Bayes belum dapat memiliki nilai akurasi yang lebih baik dibandingkan oleh nilai akurasi yang dimiliki oleh algoritma SVM. Tetapi untuk *training time* algoritma SVM dan Naïve Bayes dengan *dataset* yang sudah dilakukan SMOTE, Naïve Bayes masih unggul dengan nilai 0.009 detik.



Gambar 4. Komparasi Training Time Setiap Algoritma



Gambar 5. Komparasi Accuracy Setiap Algoritma



Gambar 6. Komparasi Prediction Time Setiap Algoritma

Berdasarkan gambar 4, dapat disimpulkan bahwa algoritma *Logistic Regression* dengan data yang sudah melakukan data *balance* memiliki *training time* terlama dengan nilai 0.844 detik dibandingkan dengan algoritma lainnya. Berdasarkan gambar 5, dapat disimpulkan bahwa *Logistic Regression* dengan data yang sudah dilakukan SMOTE memiliki hasil akurasi tertinggi dengan nilai 89% dibanding dengan algoritma lainnya. Berdasarkan gambar 6, dapat disimpulkan bahwa SVM menggunakan data yang tidak dilakukan SMOTE memiliki *prediction time* tercepat dengan waktu sebesar 0.000487 detik.

Berdasarkan hasil di atas, dapat disimpulkan bahwa algoritma Naïve Bayes memiliki waktu *training data* tercepat dengan hasil akurasi yang baik dibandingkan dengan algoritma-algoritma lainnya. Maka dari itu, algoritma Naïve Bayes dapat digunakan untuk mengklasifikasikan soal fisika dan hasil dari klasifikasi tersebut dapat digunakan oleh siswa SMA untuk bisa memahami soal-soal tersebut dengan lebih mudah dan cepat.

4. KESIMPULAN

Dari penelitian ini, didapatkan bahwa Naïve Bayes adalah algoritma yang memiliki tingkat kecepatan *training* yang tinggi dibandingkan dengan algoritma lainnya. Maka dari itu, algoritma Naïve Bayes dapat menjadi solusi untuk mengklasifikasikan soal fisika dengan cepat dan memberikan kemudahan untuk siswa SMA untuk dapat memahami soal fisika dengan lebih baik. Tetapi kekurangan dari algoritma ini adalah nilai akurasi yang dihasilkan bukan merupakan nilai tertinggi apabila dibandingkan dengan algoritma SVM. Untuk penelitian yang akan mendatang, dapat disarankan untuk melakukan kombinasi algoritma dan evaluasi yang lebih komprehensif untuk dapat meningkatkan nilai akurasi.

Ucapan Terima Kasih

Saya mengucapkan terima kasih kepada seluruh pihak yang mendukung pembuatan jurnal ini.

DAFTAR PUSTAKA

- [1] M. M. N. Karsana, K. M. L, and W. Astuti, "Single-Label and Multi-Label Text Classification using ANN and Comparison with Naïve Bayes and SVM," vol. 7, no. April, pp. 857–863, 2023, doi: 10.30865/mib.v7i2.6024.
- [2] T. Durand and G. Mori, "Learning a Deep ConvNet for Multi-Label Classification With Partial Labels," pp. 647–657.
- [3] P. Mata, P. Matematika, D. Z. Azhari, I. S. Damanik, and D. Suhendro, "Penerapan Algoritma C4 . 5 Untuk Klasifikasi Tingkat Pemahaman Siswa," vol. 1, no. 1, pp. 11–20, 2022.
- [4] A. Fathiarahma, A. Voutama, T. Ridwan, and N. Heryana, "Analisis Text Mining Klasifikasi Kegiatan Keluarga menggunakan Orange dengan Metode Naive Bayes," *J. Teknol. Terpadu*, vol. 9, no. 1, pp. 35–41, 2023, doi: 10.54914/jtt.v9i1.606.
- [5] H. Hendriyana, I. M. Karo Karo, and S. Dewi, "Analisis perbandingan Algoritma Support Vector Machine, Naive Bayes dan Regresi Logistik untuk Memprediksi Donor Darah," *J. Teknol. Terpadu*, vol. 8, no. 2, pp. 121–126, 2022, doi: 10.54914/jtt.v8i2.581.
- [6] I. M. Karo Karo and H. Hendriyana, "Klasifikasi Penderita Diabetes menggunakan Algoritma Machine Learning dan Z-Score," *J. Teknol. Terpadu*, vol. 8, no. 2, pp. 94–99, 2022, doi: 10.54914/jtt.v8i2.564.
- [7] A. Wiraguna, S. Al Faraby, and Adiwijaya, "Klasifikasi Topik Multi Label pada Hadis Bukhari dalam Terjemahan Bahasa Indonesia Menggunakan Random Forest," *e-Proceeding Eng.*, vol. 6, no. 1, pp. 2144–2153, 2019, [Online]. Available: <https://openlibrarypublications.telkomuniversity.ac.id/index.php/engineering/article/view/8507%0Ahttps://openlibrarypublications.telkomuniversity.ac.id/index.php/engineering/article/download/8507/8377>
- [8] M. R. Adrian, M. P. Putra, M. H. Rafialdy, and N. A. Rakhmawati, "Perbandingan Metode Klasifikasi Random Forest dan SVM Pada Analisis Sentimen PSBB," *J. Inform. Upgris*, vol. 7, no. 1, pp. 36–40, 2021.
- [9] I. Made Riartha Prawira and M. Syahrul Mubarak, "Klasifikasi Multi-Label Pada Topik Berita Berbahasa Indonesia Menggunakan Multinomial Naïve Bayes," *e-Proceeding Eng.*, vol. 5, no. 3, pp. 7774–7781, 2018.
- [10] H. A. N. S. Habibi, A. Nugroho, and R. Firliana, "Perbandingan Algoritma Naïve Bayes Classifier Dan K-Nearest Neighbors Untuk Analisis Sentimen Covid-19 Di Twitter," *J. Ilm. Inform.*, vol. 11, no. 01, pp. 54–62, 2023, doi: 10.33884/jif.v11i01.7069.
- [11] N. Nurainun, E. Haerani, F. Syafria, and L. Oktavia, "Penerapan Algoritma Naïve Bayes Classifier Dalam Klasifikasi Status Gizi Balita dengan Pengujian K-Fold Cross Validation," *J. Comput. Syst. Informatics*, vol. 4, no. 3, pp. 578–586, 2023, doi: 10.47065/josyc.v4i3.3414.
- [12] D. C. Hidayati, S. Al Faraby, and A. Adiwijaya, "Klasifikasi Topik Multi Label pada Hadis Shahih Bukhari Menggunakan K-Nearest Neighbor dan Latent Semantic Analysis," *JURIKOM (Jurnal Ris. Komputer)*, vol. 7, no. 1, p. 140, 2020, doi: 10.30865/jurikom.v7i1.2013.
- [13] A. Hanafi, A. Adiwijaya, and W. Astuti, "Klasifikasi Multi Label pada Hadis Bukhari Terjemahan Bahasa Indonesia Menggunakan Mutual Information dan k-Nearest Neighbor," *J. Sisfokom (Sistem Inf. dan Komputer)*, vol. 9, no. 3, pp. 357–364, 2020, doi: 10.32736/sisfokom.v9i3.980.
- [14] I. P. Y. Indrawan, I. G. Indrawan, and I. M. Candiasa, "Klasifikasi Multilabel Pada Abstrak Tugas Akhir Menggunakan Vector Space Model dan K-Nearest Neighbors Program Studi Magister Ilmu Komputer, Program Pascasarjana," vol. 2, no. 2, pp. 91–97, 2019.
- [15] N. Isnaini, Adiwijaya, M. S. Mubarak, and M. Y. A. Bakar, "A multi-label classification on topics of Indonesian news using K-Nearest Neighbor," *J. Phys. Conf. Ser.*, vol. 1192, no. 1, 2019, doi: 10.1088/1742-6596/1192/1/012027.



IMPLEMENTASI BI-LSTM DENGAN EKSTRAKSI FITUR WORD2VEC UNTUK PENGEMBANGAN ANALISIS SENTIMEN APLIKASI IDENTITAS KEPENDUDUKAN DIGITAL

Romario Onsu¹, Daniel Febrian Sengkey², Feisy Diane Kambey³

^{1, 2, 3}Teknik Elektro, Universitas Sam Ratulangi
Manado, Sulawesi Utara, Indonesia 95115

romarioonsu026@student.unsrat.ac.id, danielsengkey@unsrat.ac.id, feisykambey@unsrat.ac.id

Abstract

The Indonesian government is striving to enhance digital public services, including the Digital Identity Application (IKD) launched in 2022 by the Directorate General of Population and Civil Registration. Since its launch, IKD has received various responses from the public. User reviews on Google Play Store indicate a decline in ratings from June to December 2023. Review analysis is essential to understand user satisfaction, identify issues, and guide application improvements. This study aims to perform sentiment analysis on IKD user reviews using Bidirectional Long Short-Term Memory (Bi-LSTM) and Word2Vec methods. Bi-LSTM and Word2Vec are used to develop sentiment analysis from previous research that still used Machine Learning methods. This research is expected to contribute to the development of sentiment analysis models using Deep Learning for the IKD application. Review data was collected from the Google Play Store using scraping techniques for the period January-December 2023 and categorized into positive and negative. The Bi-LSTM model was trained with Word2Vec CBOW and Skip-Gram variations with dimensions of 100, 200, and 300. The results show that the combination of Bi-LSTM and Word2Vec CBOW with a dimension of 200 and a data split ratio of 80/20 produced the highest accuracy of 96.06%, with a precision of 96.44%, recall of 95.64%, and an f1-score of 96.04%. All combinations of Bi-LSTM and Word2Vec outperformed other Machine Learning algorithms.

Keywords: Bi-LSTM, IKD, Sentiment Analysis, Text Mining, Word2Vec

Abstrak

Pemerintah Indonesia berupaya meningkatkan layanan publik berbasis digital, termasuk aplikasi Identitas Kependudukan Digital (IKD) yang diluncurkan pada 2022 oleh Dirjen Kependudukan dan Pencatatan Sipil. Sejak diluncurkan, IKD mendapat berbagai tanggapan dari masyarakat. Data ulasan di Google Play Store menunjukkan penurunan rating dari Juni hingga Desember 2023. Analisis ulasan penting untuk memahami kepuasan pengguna dan mengidentifikasi masalah serta memandu perbaikan aplikasi. Penelitian ini bertujuan melakukan analisis sentimen ulasan pengguna IKD menggunakan metode Bidirectional Long Short-Term Memory (Bi-LSTM) dan Word2Vec. Bi-LSTM dan Word2Vec digunakan untuk mengembangkan analisis sentimen dari penelitian sebelumnya yang masih menggunakan metode Machine Learning. Penelitian ini diharapkan berkontribusi dalam pengembangan model analisis sentimen menggunakan Deep Learning untuk aplikasi IKD. Data ulasan dikumpulkan dari Google Play Store dengan teknik scraping pada periode Januari-Desember 2023 dan dibagi menjadi kategori positif dan negatif. Model Bi-LSTM dilatih dengan variasi Word2Vec CBOW dan Skip-Gram dengan dimensi 100, 200, dan 300. Hasil penelitian menunjukkan kombinasi Bi-LSTM dan Word2Vec CBOW dengan dimensi 200 dan proporsi data 80/20 menghasilkan akurasi tertinggi sebesar 96,06%, dengan precision 96,44%, recall 95,64%, dan f1 score 96,04%. Semua kombinasi Bi-LSTM dan Word2Vec menunjukkan hasil lebih tinggi dibandingkan algoritma Machine Learning lainnya.

Kata kunci: Analisis Sentimen, Bi-LSTM, IKD, Text Mining, Word2Vec

1. PENDAHULUAN

Dalam beberapa tahun terakhir, pemerintah Indonesia berupaya mengembangkan layanan publik menjadi digital untuk mencapai transformasi digital di era teknologi,

diharapkan bisa memudahkan masyarakat di berbagai sektor, terkhusus dalam layanan publik. Transformasi layanan publik menjadi digital bertujuan meningkatkan transparansi, efisiensi, dan responsivitas terhadap

kebutuhan warga negara[1]. Salah satu upaya mengembangkan layanan publik berbasis digital. Pemerintah Indonesia melalui Direktorat Jenderal Kependudukan dan Pencatatan Sipil meluncurkan aplikasi Identitas Kependudukan Digital (IKD) di tahun 2022. Aplikasi Identitas Kependudukan Digital mengintegrasikan berbagai layanan seperti kesehatan, pelayanan sosial, pendidikan, perpajakan, perbankan, dan lainnya ke dalam satu aplikasi untuk aktivasi[2]. Sejak diluncurkan, IKD telah mendapatkan berbagai tanggapan dari masyarakat, terutama melalui ulasan dari pengguna di *Google Play Store*, yang berisi berbagai macam opini masyarakat berupa pujian, kritik sampai masukan terhadap aplikasi. Aplikasi IKD mengalami penurunan *rating* dalam rentang waktu yang cukup lama dari Juni–Desember 2023. Ulasan pengguna dapat menjadi sumber informasi berharga mengenai tanggapan masyarakat dan umpan balik terhadap aplikasi, serta mengetahui kekurangan dan potensi perbaikan yang dapat diperbaiki. Namun, dengan begitu banyak jumlah ulasan, sulit untuk membedakan secara manual, ulasan yang memiliki sentimen bersifat positif dan bersifat negatif[3].

Dalam menghadapi masalah yang ditemukan, diperlukan analisis sentimen untuk mengklasifikasikan sentimen secara otomatis terhadap ulasan pengguna aplikasi IKD dari *Google Play Store*. Penggunaan analisis sentimen terhadap ulasan pengguna aplikasi layanan digital seperti IKD, Sangat berguna bagi Pemerintah dan pengembang dalam mengukur seberapa baik aplikasi yang di jalankan mampu menyediakan layanan publik digital dengan baik kepada pengguna[4][5]. Dalam upaya untuk meningkatkan kinerja aplikasi yang bisa diterima masyarakat, analisis sentimen terhadap aplikasi Identitas Kependudukan Digital sangat perlu dilakukan sebagai pertimbangan untuk evaluasi perkembangan aplikasi dan menilai potensi perbaikan. Penelitian terdahulu terkait analisis sentimen pada aplikasi Identitas Kependudukan Digital masih menggunakan pengolahan dengan metode *Machine Learning Konvensional*[2][6][7]. Dikarenakan penelitian terkait aplikasi IKD sebelumnya hanya menggunakan metode *Machine Learning* konvensional, terdapat kebutuhan yang belum terpenuhi untuk mengeksplorasi potensi metode *Deep Learning* terhadap analisis sentimen ulasan aplikasi IKD. Model *Deep Learning* seperti *Artificial Neural Networks*, sangat baik dalam memproses volume data yang besar dengan efisien, dengan mengekstraksi pola-pola kompleks dari data teks[8]. Model *Deep Learning* ini, mempelajari banyak lapisan representasi atau fitur dari data, yang menghasilkan hasil prediksi paling akurat[9]. Di beberapa penelitian analisis sentimen yang membandingkan antara penerapan metode *Machine Learning* dan *Deep Learning*. *Deep Learning* memiliki performa lebih baik dibandingkan dengan *Machine Learning*[10][11][12]. Algoritma *Deep Learning* seperti *Recurrent Neural Network* (RNN) dengan turunannya seperti *Long Short-Term Memory* (LSTM) yang banyak digunakan untuk analisis sentimen.

Sebelum pembentukan model *Deep Learning*, data teks yang akan dianalisis sentimen harus diekstraksi fiturnya terlebih dahulu. Dalam proses ini, kata-kata dalam teks akan diubah menjadi vektor. Algoritma *Deep Learning* seperti LSTM cukup efektif jika disandingkan dengan Word2Vec untuk ekstraksi fitur pada teks[13][14]. Namun LSTM hanya memproses data *input* melalui satu arah, sehingga model LSTM dikembangkan dengan *input*-nya menjadi dua arah. Memungkinkannya untuk menangkap informasi kontekstual dengan lebih efektif[15]. Metode *Bidirectional LSTM* yang dikombinasikan dengan Word2Vec menunjukkan hasil yang sangat baik, dan melebihi LSTM biasa. Penelitian oleh [16], mengenai analisis ulasan destinasi wisata di Pulau Bali, menggunakan *Bidirectional LSTM* mendapatkan hasil yang sangat baik dengan akurasi sebesar 96,86%. Beberapa penelitian yang menggunakan Bi-LSTM dengan Word2Vec juga menunjukkan performa yang sangat baik dibandingkan dengan algoritma lain[17][18].

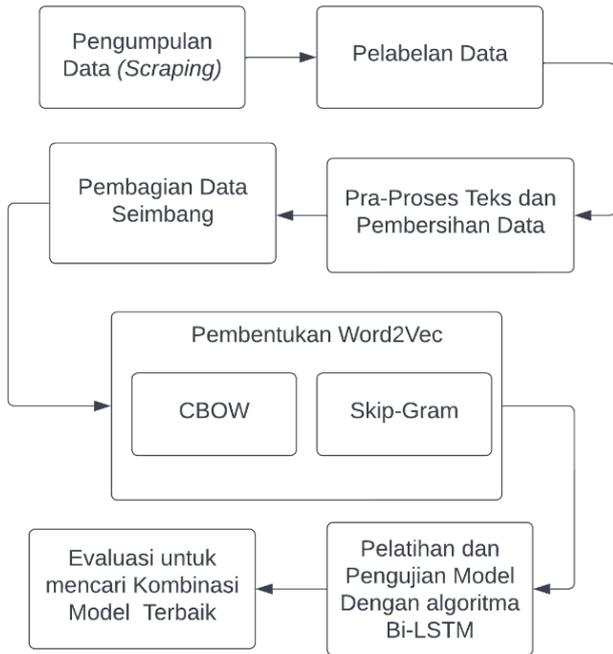
Sebagaimana telah dibahas pada penelitian-penelitian sebelumnya, Bi-LSTM memiliki keunggulan pada pengolahan analisis sentimen terutama dengan *dataset* yang besar. Oleh karena itu, penelitian ini akan menggunakan Bi-LSTM sebagai algoritma *Deep Learning* dengan Word2Vec sebagai fitur ekstraksi yang akan di terapkan untuk mengembangkan metode analisis sentimen pada ulasan pengguna aplikasi IKD, yang pada penelitian-penelitian sebelumnya belum menggunakan penerapan metode *Deep Learning*.

Penelitian ini bertujuan untuk melakukan analisis sentimen terhadap ulasan pengguna aplikasi Identitas Kependudukan Digital menggunakan metode *Bidirectional Long Short-Term Memory* dan Word2Vec. Kombinasi metode ini dipilih berdasarkan hasil dari uraian di atas menunjukkan bahwa, Bi-LSTM dan Word2Vec merupakan mode yang sangat baik untuk digunakan dalam analisis sentimen. Model analisis sentimen sangat berguna bagi pengembang aplikasi dan Pemerintah untuk mengukur keberhasilan manajemen aplikasi IKD dan merupakan bahan evaluasi untuk mengembangkan IKD. Hasil dari penelitian ini diharapkan dapat memberikan kontribusi sebagai pengembangan dari metode analisis sentimen untuk menganalisis sentimen ulasan masyarakat terhadap aplikasi Identitas Kependudukan Digital menggunakan penerapan metode *Deep Learning*, agar bisa mempermudah dalam menganalisis sentimen publik.

2. METODE PENELITIAN

Penelitian ini melibatkan serangkaian tahapan dan pengujian dalam pelaksanaan proses penelitian. Tahapan yang dilakukan meliputi pengumpulan data, pelabelan data, *text preprocessing*, pembentukan model Word2Vec, pelatihan model, pengujian menggunakan Bi-LSTM dengan variasi Word2Vec, dan evaluasi model. Evaluasi model dilakukan menggunakan *Confusion Matrix* untuk mencari model terbaik. Metode penelitian ini mencakup langkah-

langkah yang sistematis untuk memastikan setiap tahap dilakukan dengan baik dan menghasilkan model yang optimal. Pada gambar 1 di bawah ini merupakan diagram alir penelitian yang akan menjadi alur dari penelitian.

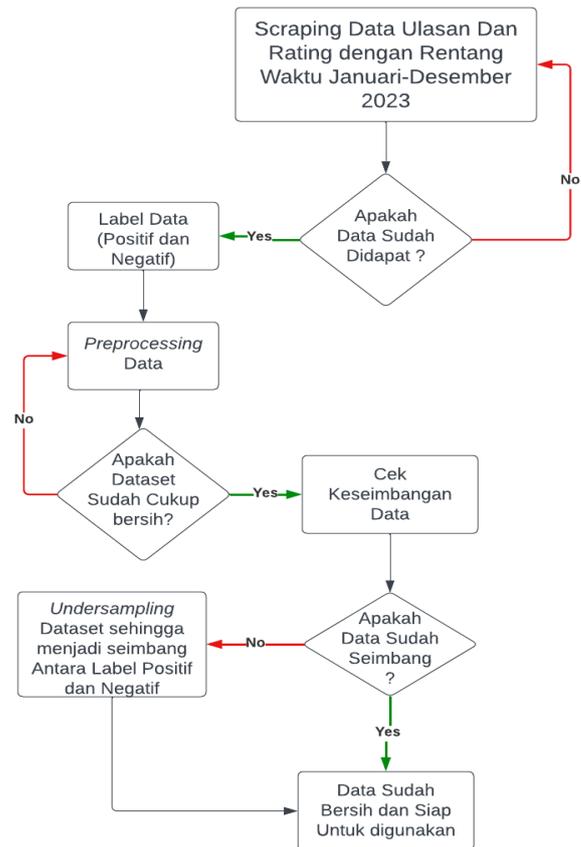


Gambar 1. Diagram Alir Penelitian

Penelitian ini menggunakan data ulasan dan *rating* yang dikumpulkan menggunakan pustaka Python *Google Play Scraper*. Setelah data terkumpul, dilakukan pelabelan dan pembersihan data (*preprocessing*). Setelah data dinilai sudah cukup bersih maka selanjutnya adalah dengan membagi data dengan nilai yang seimbang antara data yang berlabel sentimen positif dan sentimen negatif agar *dataset* yang akan digunakan berkualitas baik untuk digunakan dalam model. Berikutnya, dibuat model Word2Vec untuk memetakan kata-kata dalam ulasan ke dalam vektor numerik yang merepresentasikan makna kata. Model klasifikasi sentimen kemudian dilatih menggunakan algoritma Bi-LSTM dalam menangani data teks berurutan. Kedua variasi Word2Vec diuji untuk menentukan performa model terbaik dengan kombinasi Bi-LSTM. Terakhir, model dievaluasi menggunakan *Confusion Matrix* untuk mengukur akurasi, presisi, *recall*, dan F1-score.

2.1 Pemrosesan Data

Penelitian ini melakukan serangkaian pemrosesan data secara komprehensif seperti pada gambar 2 di bawah ini yang merupakan Diagram Alir Pemrosesan Data.



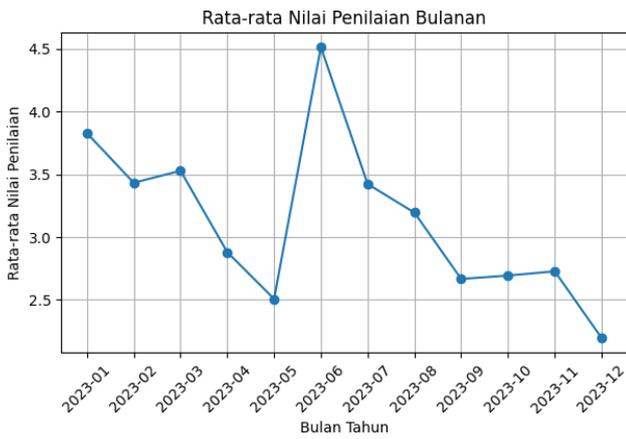
Gambar 2. Diagram Alir Pemrosesan Data

Pada pemrosesan data, pertama dilakukan pengumpulan data lewat *Google Play Store* dengan menggunakan *scraping*. Selanjutnya dilakukan Pelabelan data dengan positif dan negatif. Berikutnya masuk pada tahap *preprocessing*. Setelah data terlihat cukup bersih untuk digunakan, dilakukan penyeimbang data dengan melakukan *undersampling* sehingga data seimbang dan siap untuk digunakan.

2.1.1 Pengumpulan Data

Pada pengumpulan data, penelitian ini melakukan *scraping* data ulasan dan *rating* pada *Google Play Store* terhadap aplikasi Identitas Kependudukan Digital menggunakan pustaka Python *Google-Play-Scraper*. Pengambilan data menggunakan *Google Collab* dengan Bahasa pemrograman Python. Data yang diambil pada rentang waktu 1 tahun yaitu dari bulan Januari 2023–Desember 2023 dengan jumlah 14.000 data ulasan beserta *rating* yang diberikan pada ulasan aplikasi IKD.

Berikut pada gambar 3 di bawah ini merupakan grafik dari rata-rata penilaian terhadap aplikasi IKD dengan rentang waktu 1 tahun antara Januari 2023–Desember 2023.



Gambar 3. Grafik Rata-Rata Penilaian Aplikasi IKD

Pada grafik penilaian aplikasi IKD di atas terlihat pada bulan Juni sampai dengan bulan Desember 2023, mengalami penurunan rating terhadap aplikasi IKD, ini menunjukkan adanya penurunan tingkat kepuasan pengguna terhadap aplikasi IKD pada periode rentang waktu tersebut. Berikut contoh data ulasan dan rating terhadap aplikasi IKD yang bisa dilihat pada tabel 1.

Tabel 1. Contoh Ulasan dan Rating pada Dataset

Ulasan	Rating
Pas di upgrade ke android 12 jadi gak bisa kebuka, malah menyambungkan terus	2
Tak perlu repot jg tak ribet	5
Mantap Dan Sukses..	5
scan barcode harus dtg ke dukcapil...wasting time..katanya digital, hadeuuh..	1

2.1.2 Pelabelan Data

Pada tahap pelabelan data dilakukan *labeling* secara manual dengan mengategorikan ulasan yang bernilai 4 dan 5 sebagai Positif sedangkan 1 dan 2 sebagai Negatif. Kemudian Ulasan yang bernilai 3 akan dihapus dari *dataset*. Sehingga data yang digunakan hanya Positif dan Negatif. Pada tabel 2 merupakan contoh data ulasan yang ada pada *dataset*.

Tabel 2. Contoh Data Ulasan dan Pelabelan pada Dataset

Ulasan	Sentiment
Pas di upgrade ke android 12 jadi gak bisa kebuka, malah menyambungkan terus	Negatif
Tak perlu repot jg tak ribet	Positif
Mantap Dan Sukses..	Positif
scan barcode harus dtg ke dukcapil...wasting time..katanya digital, hadeuuh..	Negatif

2.1.3 Text Preprocessing

Pra-proses teks atau *text preprocessing* adalah serangkaian proses yang digunakan untuk menyiapkan teks mentah agar lebih siap untuk dianalisis. *Text preprocessing* sangat penting dan diperlukan dalam pengolahan *Natural Language Processing (NLP)*[19]. Tujuan utama *text preprocessing* adalah mengurangi kebisingan dan menyederhanakan teks sehingga analisis lanjutan atau algoritma pembelajaran mesin dapat bekerja lebih efisien dan efektif. Berikut serangkaian tahapan pada *text preprocessing* yang akan digunakan. Pada tahapan *text preprocessing* semua dilakukan dengan menggunakan *library* dari *Natural Language Toolkit (NLTK)*, untuk memudahkan proses *text preprocessing*.

a. Case Folding

Case folding adalah salah satu bentuk *teks preprocessing*, untuk mengubah semua huruf dalam dokumen menjadi huruf kecil. Pada tabel 3 di bawah ini merupakan contoh penggunaan *case folding* pada *dataset*.

Tabel 3. Contoh Case Folding pada Dataset

Sebelum Case Folding	Setelah Case Folding
Sangat memudahkan masyarakat untuk akses dokumen kependudukan. Terima kasih Dirjen Dukcapil Kementerian Dalam Negeri	sangat memudahkan masyarakat untuk akses dokumen kependudukan terima kasih dirjen dukcapil kementerian dalam negeri
Aktivasinya sulit malah suruh ke dukcapil segala....	aktivasinya sulit malah suruh ke dukcapil segala

b. Tokenizing

Tokenizing adalah proses memecah teks menjadi unit-unit kecil yang atau token. Token bisa berupa kata, tanda baca, atau entitas penting lainnya. Pada tabel 4 berikut adalah contoh penggunaan *tokenizing* pada penelitian ini.

Tabel 4. Contoh Tokenizing pada Dataset

Sebelum Tokenizing	Setelah Tokenizing
sangat memudahkan masyarakat untuk akses dokumen kependudukan terima kasih dirjen dukcapil kementerian dalam negeri	['sangat','memudahkan','masyarakat','untuk','akses','kependudukan','terima kasih','dokumen','kependudukan','terima','kasih','dirjen','dukcapil','kementerian',',','dalam','negeri']
aktivasinya sulit malah suruh ke dukcapil segala	['aktivasinya','sulit','malah','suruh','ke','dukcapil','segala']

c. Filtering (Stopword Removal)

Filtering adalah suatu proses untuk menghilangkan kata-kata umum dan tidak penting pada *dataset*. Kata-kata seperti "juga," atau," dan kata-kata serupa lainnya yang sering muncul dianggap sebagai *stopword*. Berikut pada tabel 5 merupakan contoh penerapan *Filtering* yang dilakukan pada penelitian ini.

Tabel 5. Contoh *Filtering* pada *Dataset*

Sebelum <i>Filtering</i>	Setelah <i>Filtering</i>
['sangat', 'memudahkan', 'masyarakat', 'untuk', 'akses', 'dokumen', 'kependudukan', 'terima', 'kasih', 'dirjen', 'dukcapil', 'kementrian', 'dalam', 'negeri']	['memudahkan', 'masyarakat', 'akses', 'dokumen', 'kependudukan', 'terima', 'kasih', 'dirjen', 'dukcapil', 'kementrian', 'negeri']
['aktivasinya', 'sulit', 'malah', 'suruh', 'ke', 'dukcapil', 'segala']	['aktivasinya', 'sulit', 'suruh', 'dukcapil']

d. *Stemming*

Stemming merupakan proses mengurangi kata-kata ke bentuk dasarnya atau akarnya. Tujuannya adalah untuk menyederhanakan berbagai bentuk kata menjadi satu bentuk dasar, misalnya mengubah "berlari," "berlari-lari," dan "pelari" menjadi "lari". Berikut pada tabel 6 merupakan contoh penerapan *Stemming* terhadap *dataset* yang dilakukan terhadap penelitian ini.

Tabel 6. Contoh *Stemming* pada *Dataset*

Sebelum <i>Stemming</i>	Setelah <i>Stemming</i>
['memudahkan', 'masyarakat', 'akses', 'dokumen', 'kependudukan', 'terima', 'kasih', 'dirjen', 'dukcapil', 'kementrian', 'negeri']	['mudah', 'rakyat', 'akses', 'dokumen', 'penduduk', 'terima', 'kasih', 'dirjen', 'dukcapil', 'mentri', 'negeri']
['aktivasinya', 'sulit', 'suruh', 'dukcapil']	['aktiv', 'sulit', 'suruh', 'dukcapil']

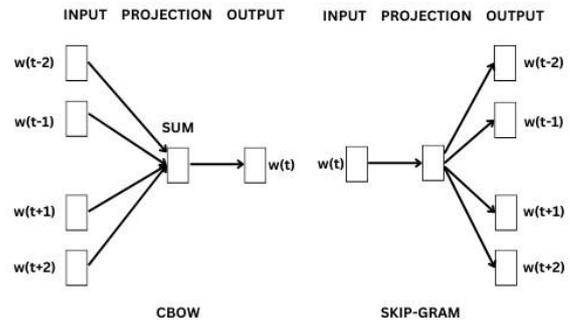
2.1.4 Pembagian Data

Data ulasan yang telah melewati tahap pra-proses data dan menjadi data bersih. kemudian dilakukan pembagian terhadap *sentiment* positif dan negatif dengan jumlah data yang seimbang. Ini perlu dilakukan agar tidak mengalami *imbalanced* pada data yang akan di gunakan. Metode yang digunakan dalam menyeimbangkan data adalah dengan menggunakan *Undersampling* terhadap data sentimen yang lebih besar, dalam hal ini data salah satu sentimen yang lebih besar akan dikurangkan menjadi seimbang keduanya. *Dataset* yang telah diproses tersisa 10800 data dengan pembagian data dengan *sentiment* positif 5400 dan dengan *sentiment* negatif 5400.

2.2 Pembentukan Model Word2Vec

Model Word2Vec adalah sebuah *model word embedding* yang diperkenalkan oleh Mikolov pada tahun 2013. Model ini mampu memahami makna semantik dari kata-kata dalam bahasa alami, sehingga dapat merepresentasikan setiap kata sebagai sebuah vektor. Pada penelitian ini, Word2Vec di bangun menggunakan *library* dari *Gensim* dengan menggunakan *corpus* yang di ambil dari *Wikipedia* sebanyak 200.000 artikel berbahasa Indonesia. Model Word2Vec dibangun dengan variasi CBOW dan Skip-Gram

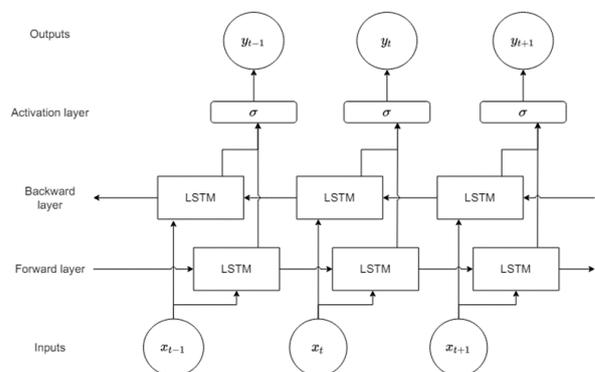
dengan jumlah dimensi vektor masing-masing 100, 200 dan 300. CBOW bekerja dengan memanfaatkan konteks untuk mengetahui target kata, memiliki kelebihan waktu *training* yang lebih cepat dan memiliki keuntungan untuk memprediksi kata-kata yang sering muncul. Sementara itu, Skip-Gram menggunakan satu kata untuk memprediksi konteks target dan memiliki kelebihan saat melakukan pemrosesan pada *dataset* yang relatif lebih sedikit, Skip-Gram mampu mengolah kata-kata yang jarang muncul dengan lebih baik. Pada gambar 4 di bawah ini merupakan arsitektur dari Word2Vec dengan 2 variasi CBOW dan Skip-Gram.



Gambar 4. Arsitektur Pembentukan Word2Vec CBOW dan Skip-Gram

2.3 Pembentukan *Bidirectional Long Short-Term Memory (Bi-LSTM)*

Bidirectional Long Short Term Memory (Bi-LSTM) adalah salah satu varian dari *Long Short Term Memory (LSTM)* yang telah dikembangkan lebih lanjut. Bi-LSTM menerima dua jenis masukan: *input forward* dan *input backward*. Hasil keluaran dari kedua *input* tersebut kemudian digabungkan. Dengan demikian, model mampu mempelajari informasi dari masa lalu dan masa depan untuk setiap urutan *input* yang diberikan. Kemampuan pemrosesan secara *bidirectional* memungkinkan model untuk memproses konteks tidak hanya konteks masa lalu tetapi juga masa depan, yang mengarah pada prediksi yang lebih terinformasi dan peningkatan akurasi dalam analisis sentimen[20]. Berikut pada gambar 5 di bawah ini merupakan Arsitektur dari *Bidirectional Long Short-Term Memory (Bi-LSTM)*.



Gambar 5. Arsitektur Bi-LSTM[21]

Bi-LSTM merupakan arsitektur LSTM yang terdiri dari dua lapisan LSTM yang berjalan paralel. Satu lapisan memproses sekuens dari awal ke akhir (*forward LSTM*), dan lapisan lainnya memproses sekuens dari akhir ke awal (*backward LSTM*). Hasil dari kedua lapisan kemudian digabungkan untuk menghasilkan representasi akhir.

Forward LSTM:

$$ht^{\rightarrow} = LSTM(xt, ht^{\rightarrow}1) \quad (1)$$

Backward LSTM :

$$ht^{\leftarrow} = LSTM(xt, ht^{\leftarrow} + 1) \quad (2)$$

Keluaran dari Bi-LSTM adalah kombinasi dari kedua arah:

$$ht = [ht^{\rightarrow}, ht^{\leftarrow}] \quad (3)$$

Di mana ht^{\rightarrow} adalah *hidden state* dari *forward LSTM* dan ht^{\leftarrow} adalah *hidden state* dari *backward LSTM*.

2.4 Skenario Penelitian

Penelitian ini mengevaluasi kinerja model analisis sentimen menggunakan algoritma Bi-LSTM dengan kombinasi metode *embedding* kata Word2Vec (CBOW dan Skip-Gram) dan proporsi pembagian data pelatihan dan pengujian. Penelitian ini akan melakukan sebanyak 18 kali pengujian terhadap kombinasi dari variabel berikut.

1. Algoritma Bi-LSTM
2. Word2Vec: CBOW dan Skip-Gram
3. Ukuran dimensi *word vector* CBOW : 100, 200, dan 300
4. Ukuran dimensi *word vector* Skip-Gram : 100, 200, dan 300
5. Proporsi pembagian data: 70/30, 80/20, dan 90/10

Selanjutnya, hasil dari penelitian ini akan dibandingkan dengan pengujian dengan beberapa metode dari *machine learning* klasik, seperti *Naïve Bayes*, *LGB Machine*, dan *Gradient Boosting*.

2.5 Evaluasi Model

Pada evaluasi model, Kinerja model akan diukur melalui serangkaian metrik, seperti *accuracy*, *recall*, *precision*, dan *F1-score*. Penggunaan evaluasi model ini sangat penting dalam menentukan seberapa baik model tersebut dalam melakukan prediksi berdasarkan data yang diberikan[22]. Data latih yang digunakan dalam proses pembelajaran model ini merupakan kumpulan data yang memungkinkan model mengenali pola dan menghasilkan prediksi. Data uji yang berbeda dari data latih digunakan untuk menguji kemampuan model yang telah dilatih. Data uji ini membantu dalam mengevaluasi seberapa baik model dapat menggeneralisasi dan memberikan prediksi yang akurat pada data yang belum pernah dilihat sebelumnya. *Confusion Matrix* (CM) akan digunakan dalam tahap evaluasi model untuk memberikan analisis yang lebih mendalam tentang performa model. *Confusion Matrix* (CM) ini berbentuk tabel

yang akan digunakan untuk menghitung dan mengevaluasi performa model yang diuji. Pada tabel 7 di bawah ini merupakan contoh dari penghitungan *Confusion Matrix* yang digunakan untuk melakukan *Evaluasi Model*.

Tabel 7. *Confusion Matrix* Untuk Evaluasi Model

		Actual Values	
		Positif (1)	Negatif (0)
Predicted Values	Positif (1)	TP (True Positif)	FP (False Positif)
	Negatif (0)	FN (False Negative)	TN (True Negative)

- a. *True Positif* (TP) adalah jumlah data yang memang positif dan diprediksi positif oleh model.
- b. *False Positif* (FP) adalah jumlah data yang sebenarnya negatif tetapi diprediksi positif oleh model.
- c. *False Negative* (FN) adalah Jumlah data yang sebenarnya positif tetapi diprediksi negatif oleh model.
- d. *True Negative* (TN) adalah Jumlah data yang memang negatif dan diprediksi negatif oleh model.

Berikut persamaan dari *confusion matrix* yang digunakan sebagai perhitungan performa.

$$Accuracy (\%) = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% \quad (4)$$

$$Recall (\%) = \frac{TP}{TP + FN} \times 100\% \quad (5)$$

$$Precision (\%) = \frac{TP}{TP + FP} \times 100\% \quad (6)$$

$$F1-Score (\%) = \frac{2 \times precision \times recall}{precision + recall} \times 100\% \quad (7)$$

3. HASIL DAN PEMBAHASAN

Penelitian ini melakukan pengujian terhadap keseluruhan variabel dengan berbagai kombinasi menghasilkan 18 kali pengujian.

Hasil Pengujian Kombinasi Bi-LSTM dan Skip-Gram

Pada tabel 8, merupakan hasil dari pengujian menggunakan kombinasi antara Bi-LSTM dan Word2Vec dengan variasi Skip-Gram.

Tabel 8. Hasil Pengujian Dengan kombinasi Bi-LSTM dan Word2Vec Skip-Gram

Proporsi Pembagian Data	Dimensi Vektor	Akurasi
70/30	100	94,72 %
70/30	200	95,03 %
70/30	300	95,03 %
80/20	100	95,51 %
80/20	200	94,60 %

Proporsi Pembagian Data	Dimensi Vektor	Akurasi
80/20	300	95,65 %
90/10	100	95,00 %
90/10	200	94,81 %
90/10	300	94,63 %

Hasil Pengujian Kombinasi CBOW

Tabel 9 berikut merupakan hasil dari pengujian menggunakan kombinasi antara Bi-LSTM dan Word2Vec dengan variasi Word2Vec.

Tabel 9. Hasil Pengujian Dengan kombinasi Bi-LSTM dan Word2Vec Skip-Gram

Proporsi Pembagian Data	Dimensi Vektor	Akurasi
70/30	100	95,06 %
70/30	200	95,46 %
70/30	300	95,15 %
80/20	100	95,74 %
80/20	200	96,06 %
80/20	300	95,56 %
90/10	100	94,81 %
90/10	200	94,72 %
90/10	300	94,91 %

Word Embedding

Setelah penelitian dilaksanakan, kedua metode *embedding* kata, yaitu Skip-Gram dan CBOW menunjukkan performa yang baik dengan akurasi di atas 94% pada semua kombinasi. CBOW cenderung memberikan hasil yang sedikit lebih baik dalam beberapa kombinasi dibandingkan dengan Skip-Gram. Pada proporsi pembagian data 80/20 dengan dimensi vektor 200, CBOW mencapai akurasi

tertinggi sebesar 96,06%, sedangkan Skip-Gram mencapai akurasi tertinggi sebesar 95,65% pada proporsi dan dimensi yang sama.

Pengaruh Ukuran Dimensi Vektor

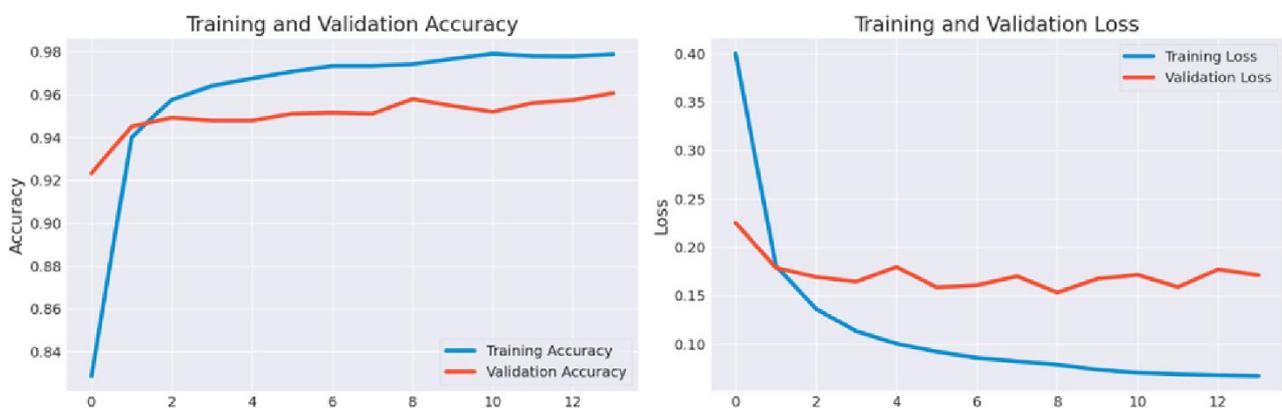
Pada metode Skip-Gram, akurasi sedikit bervariasi dengan perubahan dimensi vektor. Contohnya, pada proporsi pembagian data 70/30, akurasi meningkat dari 94,72% (dimensi 100) menjadi 95,03% (dimensi 200), namun tidak ada peningkatan lebih lanjut ketika dimensi dinaikkan menjadi 300. Pada metode CBOW juga tidak mengalami peningkatan akurasi yang konsisten dengan peningkatan dimensi vektor.

Pengaruh Proporsi Pembagian Data

Pada metode Skip-Gram, proporsi pembagian data 80/20 dengan dimensi vektor 300 memberikan akurasi tertinggi sebesar 95,65%. Kemudian pada metode CBOW, proporsi pembagian data 80/20 dengan dimensi vektor 200 juga memberikan akurasi tertinggi sebesar 96,06%. Untuk kedua metode, proporsi pembagian data 80/20 cenderung memberikan akurasi yang lebih tinggi dibandingkan dengan proporsi 70/30 dan 90/10, menunjukkan bahwa proporsi data pelatihan yang optimal berkontribusi terhadap kinerja model yang lebih baik.

Setelah dilaksanakan penelitian, didapati model terbaik dari *Bidirectional Long Short Term Memory* dan Word2Vec untuk pengembangan analisis sentimen aplikasi Identitas Kependudukan Digital. Kombinasi Word2Vec CBOW dengan pembagian data 80/20 dan ukuran vektor sebesar 200 dimensi menjadi model terbaik pada penelitian ini.

Berikut pada gambar 6 merupakan grafik dari hasil *Training Accuracy* dan *Validation Accuracy* model CBOW dengan ukuran *Word Vector* 200 dan pembagian data 80/20.



Gambar 6. Grafik *Training and Validation Accuracy* Model CBOW 200 dengan pembagian data 80/20

Pada tabel 10 merupakan hasil dari nilai evaluasi *Confusion Matrix* pada Model terbaik dengan kombinasi Variasi CBOV dengan ukuran dimensi Word Vector 200 dan Pembagian Data 80/20.

Tabel 10. Hasil *Confusion Matrix*

Predicted Values	Actual Values	Actual Values	
		Positif (1)	Negatif (0)
Positif (1)	Positif (1)	1032	47
	Negatif (0)	38	1043

Pada tabel 11 di bawah ini merupakan hasil dari *Confusion Matrix* yang telah di hitung, sehingga memperoleh hasil *Accuracy*, *Precision*, *Recall*, dan *F1-Score* terhadap Model CBOV 200 dan Pembagian Data 80/20.

Tabel 11. Hasil Evaluasi dari *Confusion Matrix*

Confusion Matrix Evaluation			
Accuracy	Precision	Recall	F1-Score
96,06	96,44	95,64	96,04

Hasil dari Pengujian terhadap Metode *Machine Learning* Klasik

Penelitian ini juga membandingkan hasil yang diperoleh dari penerapan *Deep Learning* melalui Bi-LSTM dan Word2Vec dengan penerapan *Machine learning* klasik. Dalam penelitian ini menggunakan beberapa algoritma seperti Naive Bayes, *LGB Machine*, dan *Gradient Boosting* sebagai pembanding dari hasil penelitian. Berikut pada tabel 12 merupakan hasil dari penggunaan *Machine Learning* klasik terhadap ulasan pengguna aplikasi Identitas Kependudukan Digital.

Tabel 12. Hasil Pengujian dengan Metode *Machine Learning* Klasik

Model	Accuracy	Roc Scores
Naive Bayes	93,07	93,05
LGB Machine	90,70	91,71
Gradient Boosting	86,33	86,25

Hasil dari pengujian terhadap metode *machine learning*, model Naive Bayes mendapatkan nilai akurasi tertinggi dengan 93,07%, melebihi kedua algoritma *machine learning* yang lain. Namun hasil tertinggi dari metode *machine learning* klasik ini belum bisa melebihi hasil dari kombinasi Bi-LSTM dan Word2Vec keseluruhan.

Secara keseluruhan, penelitian ini mengevaluasi kinerja model analisis sentimen menggunakan algoritma Bi-LSTM dan Word2Vec untuk mengklasifikasikan opini publik terhadap aplikasi Identitas Kependudukan Digital (IKD).

Evaluasi dilakukan berdasarkan nilai-nilai dalam *confusion matrix*, yang menunjukkan performa model dalam membedakan antara sentimen positif dan negatif. Hasil menunjukkan bahwa model terbaik mencapai *accuracy* sebesar 96,06%, *precision* 96,44%, *recall* 95,64%, dan *f1-score* 96,04%. Dengan akurasi model yang tinggi menunjukkan kinerja yang baik dalam mengklasifikasikan sentimen dalam teks terkait aplikasi IKD. Pengembangan analisis sentimen ini dapat membantu berbagai pihak seperti Pemerintah dan pengembang dalam memahami opini publik. Dengan memahami sentimen publik, pemerintah dan pengembang aplikasi IKD dapat membuat keputusan yang lebih tepat sasaran dan bermanfaat bagi masyarakat luas dengan memberikan pengalaman pengguna yang lebih baik. Hasil penelitian ini cukup baik jika dibandingkan dengan penelitian sebelumnya. Misalnya, penelitian oleh [2] yang menggunakan algoritma *Naive Bayes* untuk analisis sentimen terhadap aplikasi IKD, dan penelitian oleh [13] yang menggunakan algoritma LSTM dan Word2Vec untuk analisis sentimen terhadap ulasan film. Perbandingan ini menunjukkan keunggulan penggunaan Bi-LSTM dan Word2Vec dalam analisis sentimen, khususnya untuk aplikasi IKD. Secara keseluruhan, penelitian ini memberikan kontribusi dalam bidang analisis sentimen dengan menggunakan pendekatan yang berbeda dari penelitian sebelumnya.

4. KESIMPULAN

Berdasarkan penelitian yang telah dilakukan, penerapan metode *Bidirectional LSTM* dengan Word2Vec dapat melakukan analisis sentimen terhadap data ulasan aplikasi Identitas Kependudukan Digital dengan sangat baik dan melebihi akurasi dari penelitian terkait aplikasi Identitas Kependudukan Digital sebelumnya yang menggunakan metode *Machine Learning* klasik. Hasil dari penelitian ini memenuhi kebutuhan akan penerapan metode *Deep Learning* terhadap ulasan aplikasi Identitas Kependudukan Digital. Pada pengujian terhadap ulasan aplikasi Identitas Kependudukan Digital model kombinasi dari Bi-LSTM dan Word2Vec variasi CBOV mendapatkan akurasi rata-rata lebih tinggi dibandingkan dengan kombinasi dari Bi-LSTM dan Word2Vec variasi Skip-Gram. Kemudian pembagian data dengan proporsi 80/20 mendapatkan hasil rata-rata lebih tinggi dibandingkan dengan pembagian data 70/30 dan 90/10. Implikasi dari penelitian ini dijadikan sebagai pengembangan analisis sentimen dari penelitian sebelumnya yang menggunakan metode *machine learning* terhadap aplikasi Identitas Kependudukan Digital dan sebagai perbandingan dengan penelitian-penelitian sebelumnya. Untuk penelitian selanjutnya, disarankan untuk membandingkan Word2Vec dengan metode *word embedding* lainnya seperti *FastText*, serta menggunakan data yang lebih besar dan relevan dengan topik analisis sentimen yang akan dikerjakan, guna meningkatkan akurasi dan relevansi hasil penelitian.

DAFTAR PUSTAKA

- [1] A. Yulanda and M. Fachri Adnan, "Transformasi Digital: Meningkatkan Efisiensi Pelayanan Publik Ditinjau dari Perspektif Administrasi Publik," 2023. [Online]. Available: <https://isora.tpublising.org/index.php/isora>
- [2] R. Hidayat, R. Nur Rahman, M. Reifin Perdana, P. Teknik Informatika, F. Sains dan Teknologi, and U. Muhammadiyah Kalimantan Timur, "Analisis Sentimen Aplikasi Identitas Kependudukan Digital (IKD) Menggunakan Metode Naïve Bayes," *Jurnal Sistem Informasi dan Ilmu Komputer*, vol. 2, no. 1, pp. 129–140, 2024, doi: 10.59581/jusiik-widyakarya.v2i1.2320.
- [3] L. M. Azizah, D. B. Ajipratama, N. A. R. Putri, and C. Damarjati, "Analisa Sentimen Masyarakat Terhadap Kebijakan Vaksinasi Covid-19 Di Indonesia Pada Twitter Menggunakan Algoritma LSTM La," *JURNAL IPTEKKOM Jurnal Ilmu Pengetahuan & Teknologi Informasi*, vol. 24, no. 2, pp. 161–172, Dec. 2022, doi: 10.17933/iptekkom.24.2.2022.161-172.
- [4] R. Maulana, A. Voutama, and T. Ridwan, "Analisis Sentimen Ulasan Aplikasi MyPertamina pada Google Play Store menggunakan Algoritma NBC," *Jurnal Teknologi Terpadu*, vol. 9, no. 1, 2023, doi: 10.54914/jtt.v9i1.609.
- [5] P. SV *et al.*, "Twitter-Based Sentiment Analysis and Topic Modeling of Social Media Posts Using Natural Language Processing, to Understand People's Perspectives Regarding COVID-19 Booster Vaccine Shots in India: Crucial to Expanding Vaccination Coverage," *Vaccines (Basel)*, vol. 10, no. 11, Nov. 2022, doi: 10.3390/vaccines10111929.
- [6] A. Komarudin and A. M. Hilda, "Analisis Sentimen Ulasan Aplikasi Identitas Kependudukan Digital Pada Play Store Menggunakan Metode Naïve Bayes," *Computer Science (CO-SCIENCE)*, vol. 4, no. 1, 2024, doi: 10.31294/coscience.v4i1.2955.
- [7] R. A. Lestari, A. Erfina, and W. Jatmiko, "Penerapan Algoritma Support Vector Machine pada Analisis Sentimen Terhadap Identitas Kependudukan Digital," *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 10, no. 5, 2023, doi: 10.25126/jtiik.20231057264.
- [8] V. Umarani, A. Julian, and J. Deepa, "Sentiment Analysis using various Machine Learning and Deep Learning Techniques," *Journal of the Nigerian Society of Physical Sciences*, vol. 3, no. 4, 2021, doi: 10.46481/jnsps.2021.308.
- [9] L. Zhang, S. Wang, and B. Liu, "Deep learning for sentiment analysis: A survey," *Wiley Interdiscip Rev Data Min Knowl Discov*, vol. 8, no. 4, 2018, doi: 10.1002/widm.1253.
- [10] A. Rahman *et al.*, "JEPIN (Jurnal Edukasi dan Penelitian Informatika) Analisis Perbandingan Algoritma LSTM dan Naive Bayes untuk Analisis Sentimen".
- [11] M. Z. Rahman, Y. A. Sari, and N. Yudistira, "Analisis Sentimen Tweet COVID-19 menggunakan Word Embedding dan Metode Long Short-Term Memory (LSTM)," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 5, no. 11, pp. 5120–5127, 2021, [Online]. Available: <http://j-ptiik.ub.ac.id>
- [12] A. F. Hidayatullah, S. Cahyaningtyas, and A. M. Hakim, "Sentiment Analysis on Twitter using Neural Network: Indonesian Presidential Election 2019 Dataset," *IOP Conf Ser Mater Sci Eng*, vol. 1077, no. 1, 2021, doi: 10.1088/1757-899x/1077/1/012001.
- [13] W. Widayat, "Analisis Sentimen Movie Review menggunakan Word2Vec dan metode LSTM Deep Learning," *JURNAL MEDIA INFORMATIKA BUDIDARMA*, vol. 5, no. 3, p. 1018, Jul. 2021, doi: 10.30865/mib.v5i3.3111.
- [14] P. F. Muhammad, R. Kusumaningrum, and A. Wibowo, "Sentiment Analysis Using Word2vec and Long Short-Term Memory (LSTM) for Indonesian Hotel Reviews," in *Procedia Computer Science*, Elsevier B.V., 2021, pp. 728–735. doi: 10.1016/j.procs.2021.01.061.
- [15] G. Xu, Y. Meng, X. Qiu, Z. Yu, and X. Wu, "Sentiment analysis of comment texts based on BiLSTM," *IEEE Access*, vol. 7, 2019, doi: 10.1109/ACCESS.2019.2909919.
- [16] D. I. Af'idah, D. Dairoh, S. F. Handayani, R. W. Pratiwi, and S. I. Sari, "Sentimen Ulasan Destinasi Wisata Pulau Bali Menggunakan Bidirectional Long Short Term Memory," *MATRIK: Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer*, vol. 21, no. 3, pp. 607–618, Jul. 2022, doi: 10.30812/matrik.v21i3.1402.
- [17] J. Zhou, Y. Lu, H. N. Dai, H. Wang, and H. Xiao, "Sentiment analysis of Chinese microblog based on stacked bidirectional LSTM," *IEEE Access*, vol. 7, 2019, doi: 10.1109/ACCESS.2019.2905048.
- [18] F. A. Nugraha, N. H. Harani, R. Habibi, and Rd. N. S. Fatonah, "Sentiment Analysis on Social Distancing and Physical Distancing on Twitter Social Media using Recurrent Neural Network

- (RNN) Algorithm,” *Jurnal Online Informatika*, vol. 5, no. 2, p. 195, Dec. 2020, doi: 10.15575/join.v5i2.632.
- [19] D. F. Sengkey, A. Jacobus, and F. J. Manoppo, “Effects of kernels and the proportion of training data on the accuracy of svm sentiment analysis in lecturer evaluation,” *IAES International Journal of Artificial Intelligence*, vol. 9, no. 4, 2020, doi: 10.11591/ijai.v9.i4.pp734-743.
- [20] D. Shrimal, L. Suganya, and P. Deshmukh, “*Kanak Pandit 2 Harshali Patil Regular paper Comparative Analysis of Deep Learning Models for Sentiment Analysis on IMDB Reviews,” 2024.
- [21] S. Mikhailov and A. Kashevnik, “Car tourist trajectory prediction based on bidirectional lstm neural network,” *Electronics (Switzerland)*, vol. 10, no. 12, 2021, doi: 10.3390/electronics10121390.
- [22] O. Majdoubi, A. Benba, and A. Hammouch, “Comprehensive Machine Learning And Deep Learning Approaches For Parkinson’s Disease Classification And Severity Assessment,” *Informatyka, Automatyka, Pomiar w Gospodarce i Ochronie Srodowiska*, vol. 13, no. 4, 2023, doi: 10.35784/iapgos.5309.



DETEKSI CITRA DAUN UNTUK KLASIFIKASI PENYAKIT PADI MENGUNAKAN PENDEKATAN *DEEP LEARNING* DENGAN MODEL CNN

Muhammad Rijal¹, Andi Muhammad Yani², Abdul Rahman³

^{1,3}Nautika, Akademi Maritim Indonesia AIPI

²Teknika, Akademi Maritim Indonesia AIPI

Makassar, Sulawesi Selatan, Indonesia 90211

muhammadrijal3598@gmail.com, amyani.aipi@gmail.com, abdulrahmandgmakuling@gmail.com

Abstract

Agriculture is a vital sector related to food security. Rice is one of the productions that currently ranks third behind wheat and corn. However, in 2023, rice production in Indonesia will decrease 2022 by 1.12 million tons of GKG, and Diseases in plants are one of the causes of the reduced quantity of agricultural products. This research aims to detect disease in rice plants using leaf images with three classification classes and a test matrix to measure the model built. This research uses the Convolutional Neural Network (CNN) method to classify rice plants based on leaf images with 3 test scenarios using the Jupyter Notebook text editor tool for system coding. Research results with training show that the CNN model can classify diseases in rice based on leaf images. Of the 3 test scenarios carried out, scenario 2 shows the best results with Epoch 50 with training values from the last Epoch, namely training accuracy 0.9905 and training loss 0.0280 while validation accuracy 0.8000 and The validation loss is 0.9222 with the confusion matrix showing the suitability of predictions based on class with the classification report good recall, precision and f1-score values, namely 1.00.

Keywords: Classification, CNN, Deep Learning, Diseases, Rice

Abstrak

Pertanian merupakan sektor yang sangat penting yang berhubungan dengan ketahanan pangan, Padi merupakan salah satu produksi yang saat ini menempati urutan yang ketiga dari gandum dan jagung, tapi di tahun 2023 produksi padi di Indonesia mengalami penurunan dari tahun 2022 sebanyak 1,12 juta ton GKG dan penyakit pada tanaman merupakan salah satu penyebab berkurangnya kuantitas hasil pertanian. Penelitian ini bertujuan untuk mendeteksi penyakit pada tanaman padi berdasarkan citra daun dengan 3 kelas klasifikasi dan *matrix* pengujian untuk mengukur model yang dibangun. Pada penelitian ini menggunakan metode *Convolutional Neural Network* (CNN) untuk klasifikasi pada tanaman padi berdasarkan citra daun dengan 3 skenario pengujian menggunakan *tools* teks editor *Jupyter Notebook* untuk pengkodean sistem. Hasil Penelitian dengan *training* menunjukkan model CNN dapat melakukan klasifikasi penyakit pada padi berdasarkan citra daun, dari 3 skenario pengujian yang dilakukan skenario 2 menunjukkan hasil terbaik dengan *Epoch 50* dengan nilai *training* dari *Epoch* terakhir yaitu *training accuracy* 0.9905 dan *training loss* 0.0280 sedangkan validasi akurasi 0.8000 dan validasi *lossnya* 0.9222 dengan *confusion matrix* menunjukkan kesesuaian prediksi berdasarkan kelas dengan *classification report* nilai *recall*, *precision* dan *f1-score* yang bagus yaitu 1.00.

Kata kunci: CNN, Deep Learning, Klasifikasi, Padi, Penyakit

1. PENDAHULUAN

Pertanian merupakan sektor yang sangat penting yang berhubungan dengan ketahanan pangan dan teknologi kecerdasan banyak dimanfaatkan pada sektor tersebut[1]. *Oryza sativa* atau padi adalah salah satu produksi yang saat ini menempati urutan yang ketiga dari semua jenis biji-bijian setelah gandum dan jagung[2].

Produksi padi di negara Indonesia dari Januari-September 2023 diperkirakan sebesar 45,33 Juta ton GKG atau dapat disebut mengalami penurunan sekitar 105,09 ribu ton GKG (0,23 persen) dibandingkan Januari-September pada tahun 2022 yang sebesar 45,43 juta ton GKG. Dan diperkirakan potensi sepanjang Oktober-Desember 2023 ialah 8,30 juta ton GKG, oleh karena itu total produksi padi pada 2023 diperkirakan sebesar 53,63 juta ton GKG atau mengalami

penurunan sebanyak 1,12 juta ton GKG (2,05 persen) dibandingkan 2022 yang sebesar 54,75 juta ton GKG[3].

Penyakit pada tanaman adalah salah satu dari penyebab penurunan kuantitas dan kualitas pada hasil pertanian[4]. Penyakit yang menyerang daun tanaman padi dapat mengakibatkan berkurangnya jumlah produksi padi. Jenis penyakit pada daun tanaman padi sangat beragam. Namun jenis penyakit pada daun diberikan solusi untuk mengatasinya. Namun jenis penyakit pada daun tanaman padi ini terkadang tidak teridentifikasi oleh orang yang awam, sehingga akan berakibat kesalahan mengidentifikasi jenis penyakit dan penanganannya. Maka dari itu untuk mengatasi hal tersebut dapat dilakukan proses identifikasi dengan *Computer Vision*[5].

Perkembangan komputer dan teknologi informasi memungkinkan dapat mengidentifikasi suatu penyakit dengan memanfaatkan *Artificial Intelligence* yang salah satunya dengan menggunakan model pengenalan pola citra. Salah satu cabang *Artificial Intelligence* yaitu *Machine Learning* telah mengalami perkembangan yang pesat yang kemudian menuju kepada munculnya pendalaman ilmu *Machine Learning* yaitu *Deep Learning*[6]. Kemajuan teknologi di bidang kecerdasan buatan seperti *Deep Learning*, sudah menjadi pendorong transformasi pada berbagai sektor[7]. Metode *Deep Learning* dianggap lebih unggul kalau dibandingkan dengan metode *Machine Learning* yang lain[8].

Metode *Convolutional Neural Network* (CNN) merupakan salah satu algoritma yang populer dalam *Deep Learning* dan dapat digunakan untuk *Computer Vision*[9] atau memiliki kemampuan mengolah informasi citra[10]. CNN mempunyai keunggulan yang tak tertandingi dalam mengekstraksi fitur gambar[11]. Model CNN banyak digunakan untuk tugas *image processing* karena memiliki tingkat akurasi yang tinggi dan lebih baik dalam pengenalan citra. Terdapat sejumlah penelitian yang melakukan klasifikasi gambar citra menggunakan model CNN. Dalam penelitian yang berjudul “Implementasi *Convolutional Neural Network* (CNN) untuk Klasifikasi Ekspresi Citra Wajah pada Fer-2013 *Dataset*” menggunakan beberapa *optimizer* dan tanpa *dropout*. Akurasi pelatihan untuk validasi tertinggi diraih saat menggunakan beberapa *optimizer Adam* dengan nilai yang didapat 66% dan akurasi pelatihan sebesar 84% tapi mengalami *overfitting* itu berdasarkan dari nilai akurasi validasi dan akurasi pelatihan yang cukup jauh[12]. Di penelitian selanjutnya tentang deteksi beberapa penyakit pada apel dengan Model CNN mendapatkan akurasi 94,9%[13].

Maka dari itu pada penelitian ini yang berjudul “Deteksi Citra Daun untuk Klasifikasi Penyakit Padi Menggunakan Pendekatan *Deep Learning* Dengan Model CNN” guna melakukan penelitian untuk mendeteksi penyakit padi dan untuk mengetahui tingkat akurasi model CNN dalam melakukan klasifikasi penyakit pada padi berdasarkan citra

daun untuk mengedukasi para petani untuk mengetahui dini penyakit padi yang dalam penelitian ini 3 jenis penyakit padi yang akan dilakukan klasifikasi adalah *bacterial leaf blight*, *leaf smut* dan *brown spot*, dan untuk menguji model CNN dengan 3 Skenario Pengujian yaitu dengan 10 *Epoch* 50 *Epoch* dan 100 *Epoch* yang bertujuan mengetahui skenario pengujian yang terbaik.

2. METODE PENELITIAN

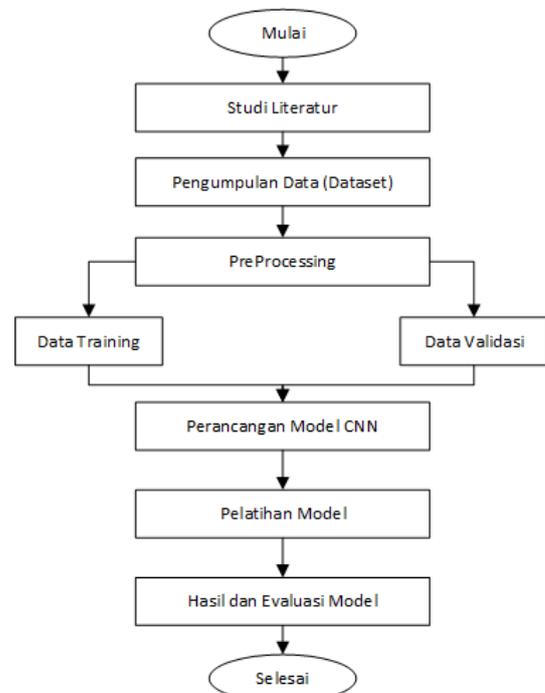
2.1 Metode Pengumpulan Data dan Metode Pengujian

Penelitian ini menggunakan data berupa *dataset* yang digunakan dalam pelatihan model diambil dari *website* penyedia *dataset* yaitu *Kaggle.com Dataset* berisi citra daun padi yang terdiri dari 3 kelas penyakit padi yaitu *bacterial leaf blight*, *leaf smut* dan *brown spot*.

Model yang digunakan dalam penelitian yaitu menggunakan pendekatan *Deep Learning* yaitu metode *Convolutional Neural Network* (CNN). Model ini digunakan untuk mengolah citra pada daun untuk melakukan klasifikasi jenis penyakit pada padi. Klasifikasi merupakan teknik dalam membedakan suatu objek[14]. Kemudian hasil pelatihan metode yang dilakukan kemudian dilakukan pengujian berdasarkan 3 skenario pengujian. *Tools* teks editor yang digunakan dalam penelitian untuk menguji model adalah *Jupyter Notebook* dengan bahasa pemrograman *Python* sebagai pengkodean sistem.

2.2 Tahapan Penelitian

Pada penelitian ini terdapat beberapa proses tahapan cara untuk melakukan pengklasifikasian penyakit pada daun tanaman padi adalah sebagai berikut:



Gambar 1. Tahapan Penelitian

Gambar 1 menunjukkan gambaran dari alur penelitian mulai sampai selesai pada penelitian ini. Proses-proses alur penelitian ini dimulai dengan studi *literatur*, pengumpulan *dataset*, *preprocessing* data. Kemudian model dirancang dengan metode atau Model CNN, pelatihan model, serta hasil dan evaluasi model yang dibangun.

2.2.1 Studi Literatur

Studi literatur yang dilakukan didapat dari buku, jurnal dan *website (Online Research)* yang berkaitan dengan *Deep Learning*, Model CNN, dan klasifikasi penyakit tanaman.

2.2.2 Pengumpulan Dataset

Pengumpulan data berupa *dataset* yang digunakan dalam pelatihan model diambil dari *website* penyedia *dataset* yaitu *Kaggle.com Dataset* berisi citra daun padi yang terdiri dari 3 kelas penyakit padi yaitu *bacterial leaf blight*, *leaf smut* dan *brown spot* dapat dilihat pada tabel 1 yaitu *Dataset*:

Tabel 1. Dataset

No	Kelas	Leaf Smut	Brown Spot	Bacterial leaf blight
1	Jumlah data	40	40	40
2	Data Latih	35	35	35
3	Data Validasi	5	5	5

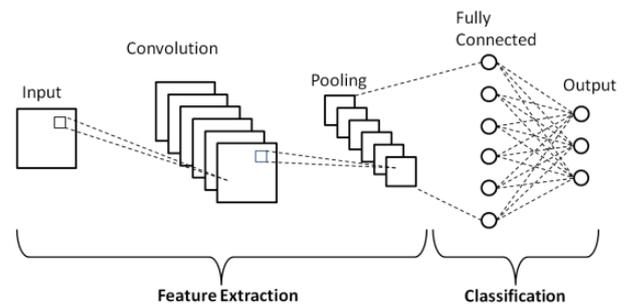
Gambar 

2.2.3 Preprocessing Data

Proses *Preprocessing* data adalah suatu proses persiapan data sebelum diolah dan digunakan untuk klasifikasi[15] yang dilakukan dimulai dari *crop image* atau memotong gambar untuk menjadi rasio *square*, kemudian *augmentasi image* selanjutnya yaitu *split dataset* atau membagi *dataset* yang mau dilatih menjadi data *training*, data *validation*.

2.2.4 Perancangan Model CNN

Perancangan Model menggunakan model *Convolutional Neural Network (CNN)* yaitu salah satu jenis model *supervised learning* yang bekerja dengan menerima input berupa citra atau gambar. Perancangan Model CNN terdiri berbagai tahapan pengulangan antara lapisan *Convolutional* dan lapisan *pooling* yang diakhiri dengan lapisan *fully connected*. Lebih jelasnya dimulai dengan *input dataset* selanjutnya ada *Convolutional Layer*, *ReLU* dan *Maxpooling* dengan beberapa *layer*. Selanjutnya proses klasifikasi pada tahapan *fully connected layer* yaitu *flatten* dan *dense* selanjutnya *output* dari Model CNN adalah hasil pelatihan yang terdiri dari data Akurasi, *loss* dan *Confusion Matrix*. Berikut Arsitektur CNN:



Gambar 2. Arsitektur CNN[16]

2.2.5 Pelatihan Model

Pada tahapan pelatihan model CNN yang dilakukan akan dilakukan pelatihan atau *train model* dengan menggunakan data *training* dan data *validasi*. Pada tahapan ini model melakukan prediksi dan menghitung dengan menggunakan *Confusion Matrix*. Model dalam *dataset* juga diolah kemudian dilatih dengan 3 skenario pengujian yaitu menggunakan 10 *Epoch*, 50 *Epoch* dan 100 *Epoch* dengan masing-masing pembagian 35 data sebagai data *training* dan 5 data sebagai data *validation* untuk mengetahui skenario yang paling bagus dalam proses pelatihan yang dilakukan.

2.2.6 Hasil dan Evaluasi Model

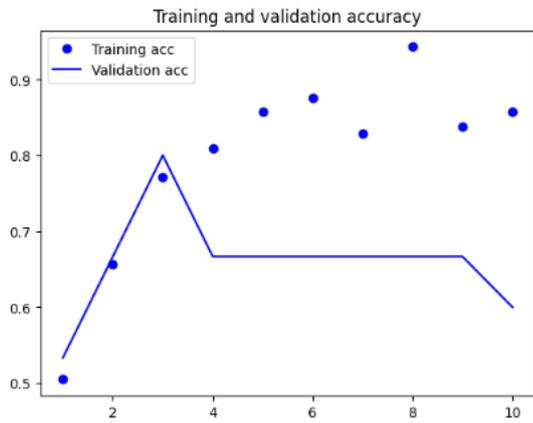
Hasil dan evaluasi model merupakan tahap terakhir dari pemeriksaan ini. Keakuratan dan presentasi *loss* dari hasil model CNN yang dibuat akan ditentukan pada tahap ini. Evaluasi dan hasil tersebut kemudian menjadi landasan untuk mengambil kesimpulan dari penelitian yang dilakukan.

3. HASIL DAN PEMBAHASAN

Pada tahapan ini merupakan hasil dari arsitektur model CNN yang telah dibangun tentang penerapan Deteksi Citra Daun untuk klasifikasi penyakit pada padi menggunakan pendekatan *Deep Learning* dengan Mode CNN. Pada penelitian ini metode CNN yang dibangun dapat memprediksi padi dan mengklasifikasikan berdasarkan kelas penyakit dibuktikan dengan data *training* dan data *validasi*. Adapun *layer convolution*, *layer pooling*, *layer flatten* serta *layer dense* untuk klasifikasi penyakit pada tanaman padi menggunakan CNN. Dalam proses *training* yang dilakukan ada 3 skenario pengujian yang dilakukan yaitu skenario 1 menggunakan *Epoch* 10, skenario 2 menggunakan 50 *Epoch* dan skenario 3 menggunakan 10 *Epoch* dan masing proses pengujian akan berhenti ketika memenuhi kondisi dari setiap *Epoch* yang ditentukan.

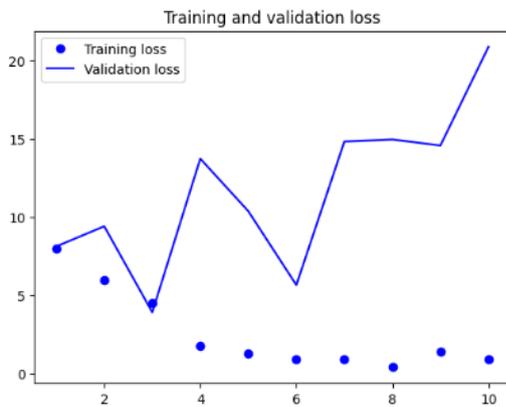
3.1 Skenario 1 Epoch 10

Dari pengujian model yang dilakukan pada skenario 1 dengan menggunakan 10 *Epoch*, dapat diketahui hasil yang didapatkan terhadap data *training* dan *data validation* sebagai berikut:



Gambar 3. Epoch 10 Train dan Valid Acc

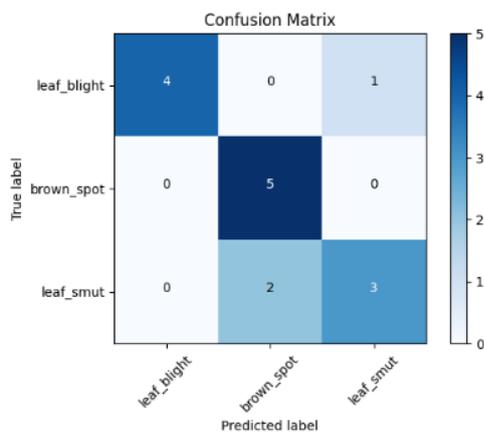
Gambar 3 Epoch 10 Training dan validasi accuracy pada skenario 1 dengan Epoch 10 memperlihatkan grafik akurasi model yang dilatih. Dari Epoch terakhir diperoleh nilai pada training accuracy 0.8571 dan nilai pada validasi accuracy 0.6000.



Gambar 4. Epoch 10 Train dan Valid Loss

Gambar 4 Epoch 10 Training dan Validasi Loss menunjukkan grafik yang dihasilkan. Pada nilai Epoch yang terakhir memperlihatkan Training loss 0.9306 dan Validasi Loss 20.8793.

Selanjutnya dapat dilihat hasil dari pengujian menggunakan Confusion Matrix di gambar berikut:



Gambar 5. Epoch 10 Confusion Matrix

Di gambar 5 hasil Confusion Matrix Epoch 10 Confusion Matrix terdapat ketidakcocokan pada leaf blight 1 dan di leaf smut 2, kemudian menghasilkan sebuah classification report dengan hasil yang diperoleh dari matrix pengujian sebagai berikut:

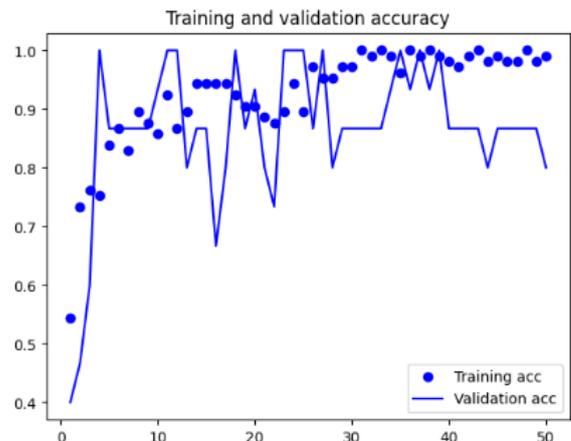
	precision	recall	f1-score	support
bacterial_leaf_blight	1.00	0.80	0.89	5
brown_spot	0.71	1.00	0.83	5
leaf_smut	0.75	0.60	0.67	5
accuracy			0.80	15
macro avg	0.82	0.80	0.80	15
weighted avg	0.82	0.80	0.80	15

Gambar 6. Epoch 10 Classification Report

Dari gambar 6 Epoch 10 Classification Report, didapatkan hasil pada pengujian model di skenario 1 dengan Epoch 10 menggunakan confusion matrix dalam mendapatkan klasifikasi pada tanaman padi itu menghasilkan nilai dengan akurasi 80% karena terdapat ketidaksesuaian 1 di leaf blight dan 2 leaf smut.

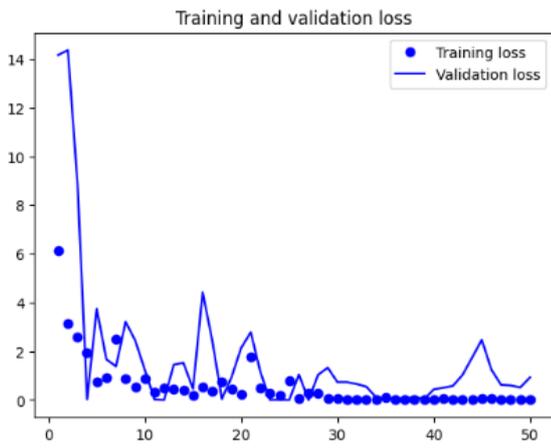
3.2 Skenario 2 Epoch 50

Dari pengujian model yang dilakukan pada skenario 2 dengan menggunakan 50 Epoch, dapat diketahui hasil yang didapatkan terhadap data training dan data validation sebagai berikut:



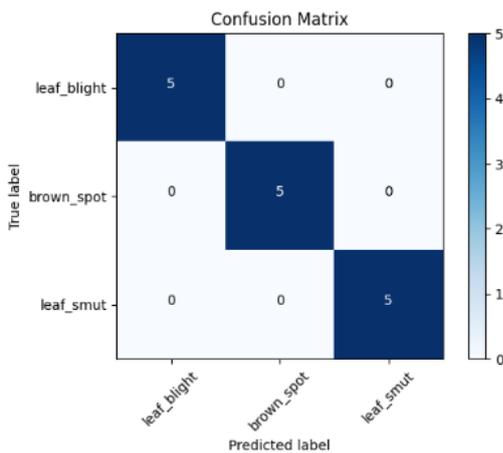
Gambar 7. Epoch 50 Train dan Valid Acc

Gambar 7 Epoch 50 Training dan validasi accuracy pada skenario 2 dengan Epoch 50 memperlihatkan grafik akurasi model yang dilatih. Dari Epoch terakhir diperoleh nilai pada training accuracy 0.9905 dan nilai pada validasi accuracy 0.8000.



Gambar 8. Epoch 50 Train dan Valid Loss

Sedangkan untuk gambar 8 Epoch 50 Training dan validasi loss pada skenario 2 dengan Epoch 50 memperlihatkan grafik akurasi model yang dilatih. Dari Epoch terakhir diperoleh nilai pada training loss 0.0280 dan nilai pada validasi 0.9222.



Gambar 9. Epoch 50 Confusion Matrix

Dari hasil Confusion Matrix di gambar 9 Epoch 50 Confusion Matrix didapatkan bahwa dari masing-masing 5 data validasi dengan masing-masing kelas semua memberikan hasil prediksi klasifikasi dengan benar sesuai dengan kelas asli dari data yang diuji, kemudian menghasilkan sebuah Classification report dengan hasil yang diperoleh dari matrix pengujian sebagai berikut:

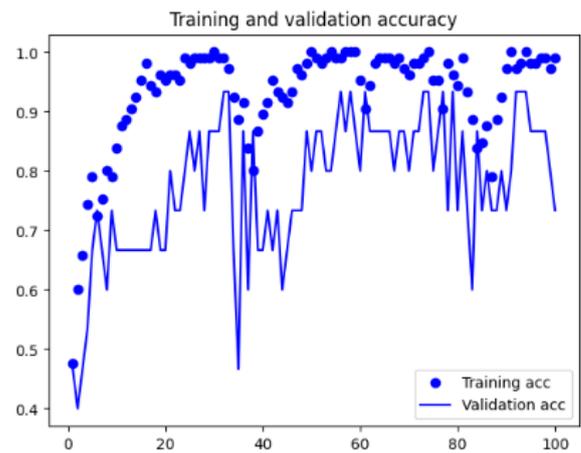
	precision	recall	f1-score	support
bacterial_leaf_blight	1.00	1.00	1.00	5
brown_spot	1.00	1.00	1.00	5
leaf_smut	1.00	1.00	1.00	5
accuracy			1.00	15
macro avg	1.00	1.00	1.00	15
weighted avg	1.00	1.00	1.00	15

Gambar 10. Epoch 50 Classification Report

Dalam Gambar 10 Epoch 50 Classification Report dapat dilihat pada kelas Brown Spot, Bacterial Leaf Blight dan Leaf smut didapatkan nilai recall, precision dan f1-score yang bagus yaitu 1.00.

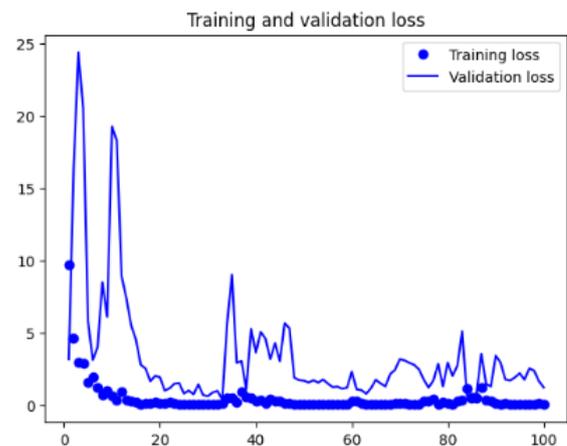
3.3 Skenario 3 Epoch 100

Dari pengujian model yang dilakukan pada skenario 3 dengan menggunakan 100 Epoch, dapat diketahui hasil yang didapatkan terhadap data training dan data validation sebagai berikut:



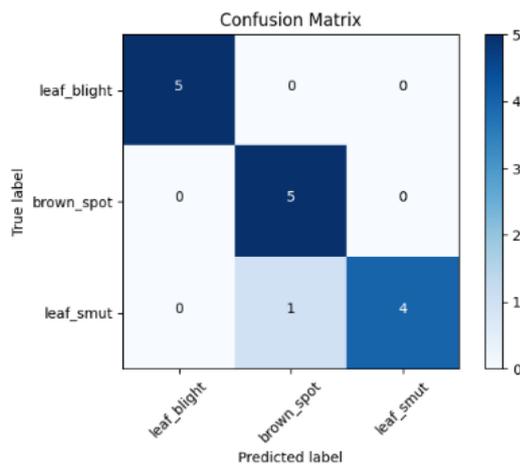
Gambar 11. Epoch 100 Train dan Valid Acc

Gambar 11 Epoch 100 Training dan validasi accuracy pada skenario 3 dengan Epoch 100 memperlihatkan grafik akurasi model yang dilatih. Dari Epoch terakhir diperoleh nilai pada training accuracy 0.9905 dan nilai pada validasi accuracy 0.7333.



Gambar 12. Epoch 100 Train dan Valid Loss

Sedangkan untuk gambar 12 Epoch 100 Training dan validasi loss pada skenario 3 dengan Epoch 100 memperlihatkan grafik akurasi model yang dilatih. Dari Epoch terakhir diperoleh nilai pada training loss 0.0245 dan nilai pada validasi 0.7333.



Gambar 13. Epoch 100 Confusion Matrix

Dari hasil *Confusion Matrix* di gambar 13 *Epoch 100 Confusion Matrix* terdapat ketidakcocokan pada *leaf smut* 1 dan untuk 2 kelas lainnya sudah terdapat kecocokan pada masing-masing kelas, kemudian menghasilkan sebuah *Classification report* dengan hasil yang diperoleh dari *matrix* pengujian sebagai berikut:

	precision	recall	f1-score	support
bacterial_leaf_blight	1.00	1.00	1.00	5
brown_spot	0.83	1.00	0.91	5
leaf_smut	1.00	0.80	0.89	5
accuracy			0.93	15
macro avg	0.94	0.93	0.93	15
weighted avg	0.94	0.93	0.93	15

Gambar 14. Epoch 100 Classification Report

Dalam Gambar 14 *Epoch 100 Classification Report* dapat dilihat bahwa kelas *brown spot* dengan *precision* 0.83 dan *f1-score* 0.91 dan *leaf smut* dengan nilai *recall* 0.8 dan *f1-score* 0.89 ini menunjukkan terdapat ketidaksesuaian kelas yaitu antara *leaf smut* dan *brown spot*. Dari proses pelatihan yang dilakukan dapat dievaluasi bahwasanya jumlah *Epoch* berpengaruh terhadap hasil dari *training*, dan dari evaluasi *matrix* yang dilakukan skenario 2 menunjukkan hasil terbaik dari pada skenario yang lainnya. Serta diharapkan dengan penerapan model CNN dalam mendeteksi penyakit padi petani bisa mengetahui dini penyakit padi untuk mengurangi risiko penurunan produksi padi akibat penyakit.

4. KESIMPULAN

Berdasarkan hasil pengujian yang dilakukan pada penelitian dengan menggunakan Model CNN, dapat ditarik kesimpulan yaitu Pengujian berdasarkan beberapa skenario dengan menggunakan *Epoch* yang berbeda dapat mempengaruhi nilai dari akurasi dan *confusion matrix* pengujian, model CNN untuk klasifikasi penyakit pada tanaman padi berdasarkan citra daun terdapat perbedaan hasil dari 3 skenario pengujian yang dilakukan.

Hasil dari pelatihan model dari pengujian 3 skenario didapatkan pada skenario 2 dengan menggunakan *Epoch 50* dengan nilai dari *Epoch* terakhir yaitu *training* akurasi

0.9905 dan nilai *training loss* 0.0280 sedangkan untuk validasi akurasi 0.8000 dan validasi *lossnya* 0.9222. Kemudian dilakukan evaluasi *matrix* menggunakan *confusion matrix* yang dapat memberikan hasil prediksi klasifikasi dengan benar sesuai dengan kelas asli dari data yang diuji. Selanjutnya *Classification Report* menunjukkan kelas *Brown Spot*, *Bacterial Leaf Blight* dan *leaf smut* didapatkan nilai *recall*, *precision* dan *f1-score* yang bagus yaitu 1.00 ini menunjukkan jumlah *Epoch* berpengaruh pada hasil dari proses *training* yang dilakukan sehingga di dapatkan bahwasanya jumlah *Epoch* berpengaruh terhadap hasil *training* berdasarkan hasil *training* yang dilakukan skenario 2 menunjukkan hasil yang lebih bagus dari skenario 1 dan 3. Untuk penelitian yang akan datang diharapkan dapat melakukan pengujian dengan jumlah *dataset* dan kelas penyakit padi yang lebih banyak lagi.

DAFTAR PUSTAKA

- [1] F. F. Maulana and N. Rochmawati, "Klasifikasi Citra Buah Menggunakan Convolutional Neural Network," *J. Informatics Comput. Sci.*, vol. 1, no. 02, pp. 104–108, 2020, doi: 10.26740/jinacs.v1n02.p104-108.
- [2] G. Zhou, W. Zhang, A. Chen, M. He, and X. Ma, "Rapid Detection of Rice Disease Based on FCM-KM and Faster R-CNN Fusion," *IEEE Access*, vol. 7, pp. 143190–143206, 2019, doi: 10.1109/ACCESS.2019.2943454.
- [3] Badan Pusat Statistik, "Luas Panen dan Produksi Padi di Indonesia 2023 (Angka Sementara)," *Badan Pus. Stat.*, vol. 2023, no. 68, pp. 1–8, 2023, [Online]. Available: <https://www.bps.go.id/pressrelease/2023/10/16/2037/luas-panen-dan-produksi-padi-di-indonesia-2023--angka-sementara-.html#:~:text=Produksi beras pada 2023 untuk,sebesar 31%2C54 juta ton.>
- [4] H. B. Prajapati, J. P. Shah, and V. K. Dabhi, "Detection and classification of rice plant diseases," *Intell. Decis. Technol.*, vol. 11, no. 3, pp. 357–373, 2017, doi: 10.3233/IDT-170301.
- [5] A. Jinan, B. H. Hayadi, and U. P. Utama, "Journal of Computer and Engineering Science Volume 1, Nomor 2, April 2022 Klasifikasi Penyakit Tanaman Padi Menggunakan Metode Convolutional Neural Network Melalui Citra Daun (Multilayer Perceptron)," vol. 1, no. April, pp. 37–44, 2022.
- [6] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015, doi: 10.1038/nature14539.
- [7] K. Marlin, E. Tantrisna, B. Mardikawati, R. Anggraini, and E. Susilawati, "Manfaat dan Tantangan Penggunaan Artificial Intelligences (AI) Chat GPT Terhadap Proses Pendidikan Etika dan Kompetensi Mahasiswa Di Perguruan Tinggi," *Innov. J. Soc. Sci. Res.*, vol. 3, no. 6, pp. 5192–5201, 2023, [Online]. Available: <https://j->

- innovative.org/index.php/Innovative/article/view/7119.
- [8] M. Z. Alom *et al.*, “A state-of-the-art survey on deep learning theory and architectures,” *Electron.*, vol. 8, no. 3, 2019, doi: 10.3390/electronics8030292.
- [9] W. jie Liang, H. Zhang, G. feng Zhang, and H. xin Cao, “Rice Blast Disease Recognition Using a Deep Convolutional Neural Network,” *Sci. Rep.*, vol. 9, no. 1, pp. 1–10, 2019, doi: 10.1038/s41598-019-38966-0.
- [10] P. A. Nugroho, I. Fenriana, and R. Arijanto, “Implementasi Deep Learning Menggunakan Convolutional Neural Network (CNN) Pada Ekspresi Manusia,” *Algor*, vol. 2, no. 1, pp. 12–21, 2020.
- [11] I. Hermawan, D. Arnaldy, M. Agustin, M. F. Widiono, D. Nathanael, and M. T. Mulyani, “Sistem Pengenalan Benih Padi menggunakan Metode Light Convolutional Neural Network pada Raspberry PI 4 B,” *J. Teknol. Terpadu*, vol. 7, no. 2, pp. 120–126, 2021, doi: 10.54914/jtt.v7i2.443.
- [12] D. Alamsyah and D. Pratama, “Implementasi Convolutional Neural Networks (CNN) untuk Klasifikasi Ekspresi Citra Wajah pada FER-2013 Dataset,” *J. Teknol. Inf.*, vol. 4, no. 2, pp. 350–355, 2020, doi: 10.36294/jurti.v4i2.1714.
- [13] G. Wicaksono, S. Andryana, and B. -, “Aplikasi Pendeteksi Penyakit Pada Daun Tanaman Apel Dengan Metode Convolutional Neural Network,” *JOINTECS (Journal Inf. Technol. Comput. Sci.)*, vol. 5, no. 1, p. 9, 2020, doi: 10.31328/jointecs.v5i1.1221.
- [14] A. Pratiwi and A. Fauzi, “Implementation of Deep Learning on Flower Classification Using Implementasi Deep Learning Pada Pengklasifikasian Bunga,” vol. 5, no. 2, pp. 487–495, 2024.
- [15] T. Arifin and A. Herliana, “Optimizing decision tree using particle swarm optimization to identify eye diseases based on texture analysis,” *J. Teknol. dan Sist. Komput.*, vol. 8, no. 1, pp. 59–63, 2020, doi: 10.14710/jtsiskom.8.1.2020.59-63.
- [16] V. H. Phung and E. J. Rhee, “A High-accuracy model average ensemble of convolutional neural networks for classification of cloud image patches on small datasets,” *Appl. Sci.*, vol. 9, no. 21, 2019, doi: 10.3390/app9214500.



PENGARUH JARAK OBJEK CITRA PADA MODEL DETEKSI DAN KLASIFIKASI BOTOL PLASTIK MENGGUNAKAN YOLO

Nurvelly Rosanti¹, Retnani Latifah², Sirojul Munir³, Izzuddin Al Qossam Maududi⁴

^{1,2,4}Teknik Informatika, Universitas Muhammadiyah Jakarta

³Teknik Informatika, Sekolah Tinggi Teknologi Terpadu Nurul Fikri
Jakarta Selatan, DKI Jakarta, Indonesia 12640

nurvelly.rosanti@umj.ac.id, retnani.latifah@umj.ac.id, rojulman@nurulfikri.ac.id, izzuddinalqossammaududi@gmail.com

Abstract

Plastic bottle waste must be separated based on shape and size to facilitate recycling. Sorting plastic bottles can use object detection technology to facilitate classification using images. Image distance capture affects the classification of bottle waste because large bottles will look small when seen from a distance and vice versa. This study aims to create a plastic bottle detection and classification model using the YOLOv8 algorithm with the same bottle shape but different sizes and measure the effect of image distance on the model. Bottles consist of three sizes: large bottles measuring 1500 ml, medium bottles measuring 600 ml, and small sizes 330 ml. Pictures for the bottle image dataset were shot between 80 and 100 centimeters away. Roboflow was used to produce the dataset. Model performance evaluation used Mean Average Precision, and model testing used a confusion matrix. The test results for the same model with an image capture distance had an accuracy value of 100%. Testing of 80 cm distance images applied to the 100 cm model had an accuracy of 67%. Testing for 100 cm distance images applied to the 80 cm model was still quite good, with an accuracy of 91.6%. The results obtained show that the image distance affects the results of the model that has been built, so use an image that matches the distance applied to the model.

Keywords: Confusion Matrix, Mean Average Precision, Object Detection, Roboflow, YOLOv8

Abstrak

Sampah botol plastik perlu dipisahkan baik berdasarkan bentuk dan ukuran untuk mempermudah proses daur ulang. Pemilahan botol plastik dapat menggunakan teknologi deteksi objek untuk mempermudah klasifikasi menggunakan citra. Pengambilan jarak citra mempengaruhi klasifikasi sampah botol, hal ini disebabkan botol yang ukuran besar jika dilihat dari jauh akan terlihat kecil begitu juga sebaliknya. Penelitian ini bertujuan membuat model deteksi dan klasifikasi botol plastik menggunakan algoritma YOLOv8 dengan bentuk botol yang sama tapi ukuran berbeda serta mengukur pengaruh jarak citra terhadap model. Botol terdiri dari tiga ukuran yaitu botol besar ukuran 1500 ml, botol sedang ukuran 600 ml dan ukuran kecil 330 ml. Pengambilan foto untuk *dataset* citra botol dilakukan pada jarak 80 cm dan 100 cm. Pembuatan *dataset* menggunakan Roboflow. Evaluasi kinerja model menggunakan Mean Average Precision dan pengujian model menggunakan *confusion matrix*. Hasil pengujian untuk model yang sama dengan jarak pengambilan citra memiliki nilai akurasi 100%. Pengujian citra jarak 80 cm diterapkan ke model 100 cm memiliki akurasi 67%. Pengujian untuk citra jarak 100 cm diterapkan pada model 80 cm masih cukup baik dengan akurasi 91,6%. Hasil yang diperoleh bahwa jarak citra mempengaruhi hasil pada model yang sudah dibangun sehingga gunakan citra yang sesuai dengan jarak yang diterapkan pada model.

Kata kunci: Confusion Matrix, Deteksi Objek, Mean Average Precision, Roboflow, YOLOv8

1. PENDAHULUAN

Pemerintah dalam pengelolaan sampah sudah membuat aturan yang tertuang dalam PP No 81 Tahun 2012. Peraturan ini juga memuat tentang pemilahan sampah berdasarkan jenisnya. Jenis sampah ada tiga yaitu sampah organik, anorganik serta sampah bahan beracun dan berbahaya (B3). Dinas kebersihan Kota sudah melengkapi sarana umum dengan sarana tempat sampah yang berwarna-warni minimal tiga tempat sampah berdekatan dengan

keterangan organik, non organik dan B3. Dalam hal ini masyarakat dalam membuang sampah harus memasukkan jenis sampah sesuai dengan tempat yang tertulis pada tempat sampah tersebut. Penelitian terkait sampah juga sudah dilakukan dengan membuat prototipe aplikasi bank sampah dengan tujuan dapat mengelola sampah terpadu [1]. Pemisahan sampah menggunakan tempat sampah seperti ini masih pada tahap awal jika sampah ingin didaur ulang. Sampah yang sering ditemukan di area umum adalah

sampah botol plastik minuman sedangkan plastik sulit terurai di tanah. Sampah botol plastik yang ditemukan ada yang berbeda bentuk dan ukuran sesuai dengan merek produk.

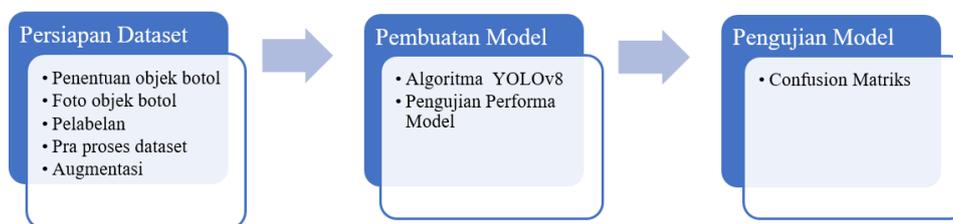
Untuk memudahkan dalam proses daur ulang dari berbagai jenis botol plastik tersebut, sampah botol plastik ini perlu dipisahkan baik berdasarkan bentuk, ukuran maupun merek sesuai dengan kebutuhan. Proses pemilahan sampah botol plastik tersebut dapat menggunakan teknologi deteksi objek. Deteksi objek menggunakan *machine learning* dengan metode *deep learning* sudah mampu memproses data gambar [2]. Penelitian terkait pemilahan sampah menggunakan deteksi objek sudah banyak dilakukan. Penelitian yang dilakukan oleh Wahyutama & Hwang [3] terkait deteksi objek tempat sampah telah dilakukan menggunakan algoritma YOLOv4 (*You Only Look Once version 4*). Penelitian tersebut fokus pada deteksi objek sampah kemudian mengelompokkannya sesuai dengan dengan empat jenis sampah secara otomatis yaitu kertas, kaleng, plastik dan selain benda ini. Di samping itu Wahyutama & Hwang [3] juga memanfaatkan teknologi *internet of things (iot)* untuk mendeteksi tempat sampah sudah penuh di wilayah tertentu. Hasil dari penelitian ini memiliki tingkat akurasi sampai 91% dalam deteksi objek sampah. Penelitian lain terkait deteksi sampah juga dilakukan oleh Parasian & Utamingrum [4] metode yang digunakan dalam deteksi objek juga menggunakan algoritma YOLO. Objek yang dideteksi dalam penelitian Parasian & Utamingrum adalah deteksi sampah organik dan anorganik. Untuk sampah organik terdiri dari empat

kelas dan sampah anorganik terdiri dari tiga kelas. Tingkat akurasi yang diperoleh dari hasil pengujian untuk deteksi sampah organik paling tinggi adalah 98% sedangkan akurasi untuk deteksi sampah anorganik paling tinggi adalah 94,6%.

Berdasarkan penelitian terdahulu bahwa tingkat akurasi deteksi objek menggunakan YOLO di atas 90% sehingga YOLO merupakan metode yang baik untuk diterapkan dalam deteksi objek digital [5][6]. Penelitian sebelumnya juga melakukan pemisahan sampah berdasarkan organik dan anorganik tentu secara objek jelas perbedaan bentuknya secara visual [7]. Pemilahan sampah botol plastik untuk daur ulang juga melakukan pemisahan sampah berdasarkan ukuran besar dan kecilnya botol agar memudahkan dalam pengelompokan sampah botol plastik tersebut. Penelitian ini membuat model deteksi objek sampah botol plastik menggunakan algoritma YOLOv8 dengan khusus objek sampah botol dengan bentuk botol yang sama hanya ukuran berbeda yaitu botol besar, sedang dan kecil. Penelitian ini bertujuan untuk membuat model klasifikasi objek botol yaitu kelas botol besar, botol sedang dan botol kecil, serta untuk mengetahui pengaruh jarak objek gambar terhadap klasifikasi. Pengklasifikasian ukuran botol ini dapat mempermudah proses pemilahan atau pemisahan sampah botol plastik berdasarkan ukuran botol plastik.

2. METODE PENELITIAN

Pembuatan model deteksi objek sampah botol plastik untuk mengelompokkan sampah botol berdasarkan ukuran menggunakan algoritma YOLOv8 dapat melalui tahap-tahapan seperti pada Gambar 1 berikut.



Gambar 1. Tahapan Pembuatan Model

Pada Gambar 1 secara garis besar terdapat empat fase dalam tahap-tahapan pembuatan model deteksi objek botol yaitu dimulai dari *input* berupa *dataset*, kemudian proses pembuatan model menggunakan algoritma YOLOv8. Setelah model terbentuk maka masuk tahapan berikutnya yaitu pengujian model dengan mengukur tingkat akurasi deteksi objek. Bagian terakhir adalah *output* berupa hasil klasifikasi botol yang terdeteksi yaitu pengelompokan botol besar, botol sedang dan botol kecil. Secara detail dari masing-masing fase sebagai berikut.

2.1. Persiapan Dataset

Pengumpulan *dataset* merupakan bagian penentu dari keberhasilan model. Proses pembuatan *dataset* dilakukan dengan membuat foto botol minimal 100 foto. *Dataset* yang dibutuhkan terdiri dari *dataset* latih dan *dataset* uji dengan

perbandingan 80% data latih dan 20% data uji [8]. Langkah-langkah yang dilakukan adalah sebagai berikut

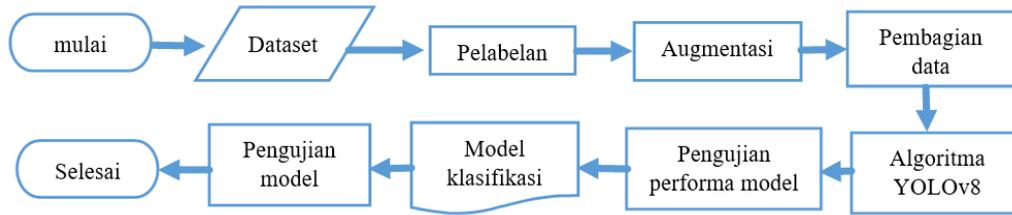
- Mempersiapkan botol plastik minuman mineral ukuran 1500 ml, 600 ml dan 300 ml.
- Mempersiapkan kamera *webcam* untuk pengambilan gambar dengan posisi tetap.
- Botol plastik dengan masing-masing ukuran difoto dengan posisi tegak, tidur dan miring.
- Hasil foto disimpan kemudian dilakukan pelabelan menggunakan *Roboflow* [9].

2.2 Pembuatan Model

Pada tahap ini *dataset* yang sudah disiapkan dan dilabeli maka masuk proses membuat model dengan melatih *dataset*

untuk mengenali objek botol. Algoritma YOLOv8 digunakan untuk pembuatan model ini disebabkan YOLOv8 baik pada objek detektor *real time*, bekerja baik pada ukuran objek yang kecil dan memiliki kemampuan deteksi objek

dengan akurasi yang tinggi [10][11]. Perancangan model yang diterapkan untuk klasifikasi botol berdasarkan jarak gambar dapat dilihat pada Gambar 2.



Gambar 2. Flowchart pembuatan model dengan YOLOv8

Pada Gambar 2 dapat diuraikan bahwa setelah persiapan *dataset* seperti proses pelabelan dan augmentasi maka tahap berikutnya masuk pada penggunaan algoritma YOLOv8 dengan pengaturan pembagian data.

2.3 Evaluasi Kinerja Model

Pada fase ini menguji kinerja model yang sudah dibuat dengan tujuan untuk mengevaluasi kinerja model deteksi objek yang telah dilatih dengan mengukur tingkat akurasi dan performa model. Pengujian model dilakukan menggunakan metode :

- a) *Mean Average Precision* (mAP) yaitu Matriks untuk mengukur akurasi deteksi objek dengan menghitung rata-rata presisi di setiap nilai recall. Semakin tinggi nilai mAP, semakin baik performa model deteksi objek [12][13]. Formula dari mAP sebagai berikut :

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i$$

N = jumlah kelas

AP_i = rata-rata presisi kelas ke i

- b) *Intersection over Union (IoU)* yaitu mengukur rasio antara luas area yang tumpang tindih antara beberapa bounding box dengan luas area gabungan dari bounding box tersebut [14].

2.4 Pengujian Model

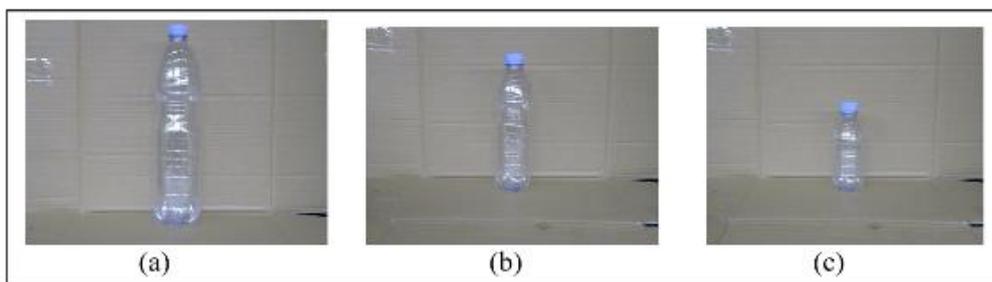
Pada tahap ini penerapan model yang sudah dihasilkan kemudian diuji menggunakan metode *confusion matrix* [15]. Pengujian model ini menggunakan *dataset* uji dengan hasil klasifikasi objek botol plastik sesuai dengan kelas botol plastik yang dideteksi yaitu kelas botol besar, kelas botol sedang dan kelas botol kecil berdasarkan jarak citra botol 80 cm dan 100 cm.

3. HASIL DAN PEMBAHASAN

3.1 Pembuatan Dataset

- a) Penentuan Objek Botol

Objek botol yang ditentukan adalah botol plastik berukuran 1500 ml, 600 ml dan 330 ml dengan bentuk botol yang sama hanya ukuran botol saja yang berbeda seperti pada Gambar 3 berikut.

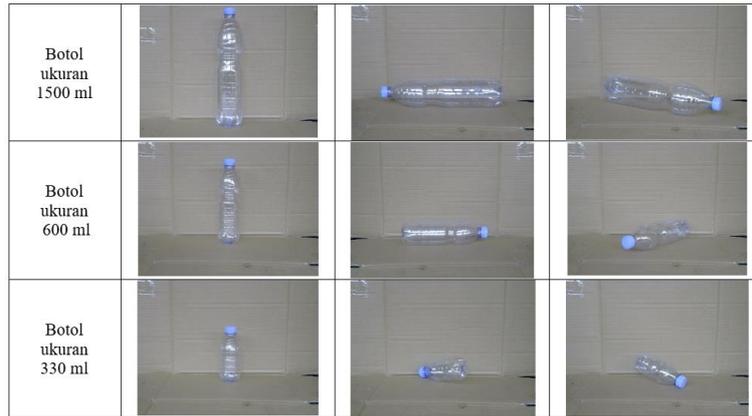


Gambar 3. (a) botol ukuran 1500 ml (b) botol ukuran 600 ml
(c) botol ukuran 330 ml

- b) Foto Objek Botol

Botol yang sudah ada kemudian difoto dengan skenario untuk jarak 80 cm dan 100 cm dari kamera. Untuk jarak 80 cm diambil foto botol ukuran 1500 ml dengan posisi tegak, tidur dan miring kiri dan miring kanan sebanyak 21 foto. Kemudian jarak 80 cm diambil foto botol ukuran 600 ml

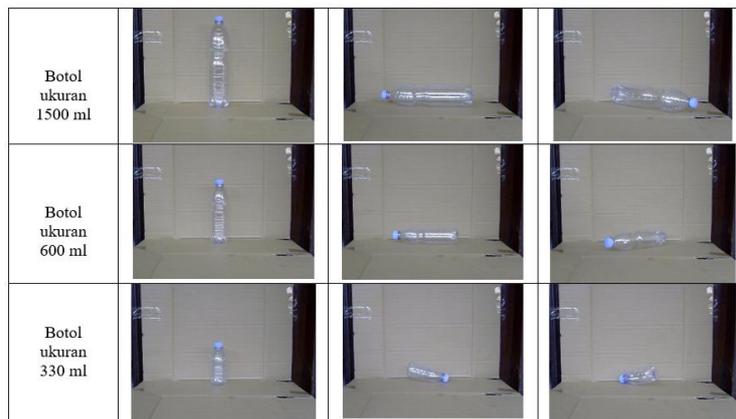
dengan posisi tegak, tidur dan miring kiri dan miring kanan sebanyak 21 foto. Berikutnya masih jarak 80 cm diambil foto botol ukuran 330 ml dengan posisi tegak, tidur dan miring kiri dan miring kanan sebanyak 21 foto. Total citra botol yang diperoleh dari jarak 80 cm adalah 63 citra. Hasil foto *dataset* jarak 80 cm seperti pada Gambar 4.



Gambar 4. Dataset foto jarak 80 cm

Untuk jarak 100 cm diambil foto botol ukuran 1500 ml dengan posisi tegak, tidur dan miring kiri dan miring kanan sebanyak 21 foto. Kemudian jarak 100 cm diambil foto botol ukuran 600 ml dengan posisi tegak, tidur dan miring kiri dan miring kanan sebanyak 21 foto. Berikutnya masih

jarak 100 cm diambil foto botol ukuran 330 ml dengan posisi tegak, tidur dan miring kiri dan miring kanan sebanyak 21 foto. Total citra botol yang diperoleh dari jarak 100 cm adalah 63 citra. Hasil foto *dataset* jarak 100 cm seperti pada Gambar 5.



Gambar 5. Dataset foto jarak 100 cm

Hasil foto botol dikelompokkan menjadi dua *dataset* yaitu *dataset* jarak 100 m dan *dataset* jarak 80 m terlihat bahwa botol ukuran botol yang sama tetapi diambil dari jarak yang berbeda terlihat berbeda.

Setelah foto objek botol selesai diambil, proses berikutnya adalah pelabelan. Pelabelan adalah menentukan kelas dari masing-masing botol yaitu kelas 1,5 L untuk botol ukuran besar, kelas 330 ml untuk botol ukuran sedang dan kelas 330 ml untuk botol ukuran kecil. Pelabelan menggunakan aplikasi *Roboflow* untuk *dataset* jarak 80 cm dan 100 cm secara terpisah, hasil pelabelan seperti pada Gambar 6.

c) Pelabelan



Gambar 6. Proses Pelabelan Dataset

Pada Gambar 6 *dataset* sudah berlabel dan terklasifikasi 1,5L, 600 ml dan 330 ml menjadi data *train* sebanyak 44 citra, data validasi 10 citra dan data testing 9 citra dengan

persentase *split dataset* sebesar 70%, 16% dan 14% seperti pada Gambar 7.

Tabel 1. Pengujian jarak 80 cm untuk model 80 dan 100 cm

Jarak	Uji	prediksi			
		model 80 cm		model 100 cm	
		confidence	class	confidence	class
aktual 80 cm	1,5-1	0,921	1,5L	0,72	1,5L
	1,5-2	0,957	1,5L	0,902	1,5L
	1,5-3	0,963	1,5L	0,878	1,5L
	1,5-4	0,96	1,5L	0,884	1,5L
	600-1	0,929	600ML	0,779	600ML
	600-2	0,938	600ML	0,925	600ML
	600-3	0,947	600ML	0,93	600ML
	600-4	0,957	600ML	0,945	600ML
	330-1	0,931	330ML	0,906	600ML
	330-2	0,957	330ML	0,69	600ML
	330-3	0,951	330ML	0,888	600ML
	330-4	0,945	330ML	0,921	600ML

Tabel 2. Pengujian jarak 100 cm untuk model 80 dan 100 cm

Jarak	Uji	prediksi			
		model 80 cm		model 100 cm	
		confidence	class	confidence	class
aktual 100 cm	1,5-1	0,921	600ML	0,879	1,5L
	1,5-2	0,957	1,5L	0,923	1,5L
	1,5-3	0,956	1,5L	0,918	1,5L
	1,5-4	0,955	1,5L	0,931	1,5L
	600-1	0,917	600ML	0,914	600ML
	600-2	0,939	600ML	0,921	600ML
	600-3	0,959	600ML	0,935	600ML
	600-4	0,951	600ML	0,932	600ML
	330-1	0,913	330ML	0,904	330ML
	330-2	0,929	330ML	0,939	330ML
	330-3	0,947	330ML	0,948	330ML
	330-4	0,817	330ML	0,939	330ML

Berdasarkan Tabel 1 dan 2 tahap berikutnya dilakukan pengujian menggunakan *confusion matrix* dengan hasil pada Tabel 3 dan 4.

Tabel 3. Evaluasi Model 80 cm berdasarkan jarak citra

Jarak Objek Citra	Precision	Recall	F1-Score
Jarak 80 cm	1	1	1
Jarak 100 cm	0,916	0,916	0,916

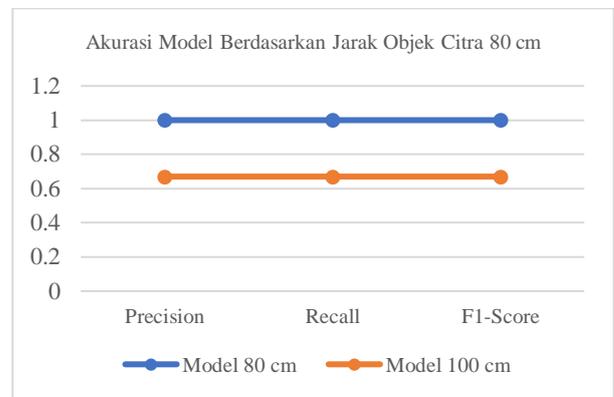
Pada Tabel 3 untuk data uji jarak 80 cm diterapkan pada model 80 cm diperoleh nilai *Precision*, *Recall*, *F1-Score*, adalah 1 atau dimaknai 100%. Untuk data uji jarak 100 cm diterapkan pada model 80 cm diperoleh nilai *Precision*, *Recall*, *F1-Score* adalah 0,916 atau 91,6%.

Tabel 4. Evaluasi Model 100 cm berdasarkan jarak citra

Jarak Objek Citra	Precision	Recall	F1-Score
Jarak 80 cm	0,67	0,67	0,67
Jarak 100 cm	1	1	1

Pada Tabel 4 untuk data uji jarak 80 cm diterapkan pada model 100 cm diperoleh nilai *Precision*, *Recall*, *F1-Score* adalah 0,67 atau 67%. Untuk data uji jarak 100 cm diterapkan pada model 100 cm diperoleh nilai *Precision*, *Recall*, *F1-Score* adalah 1 atau 100%.

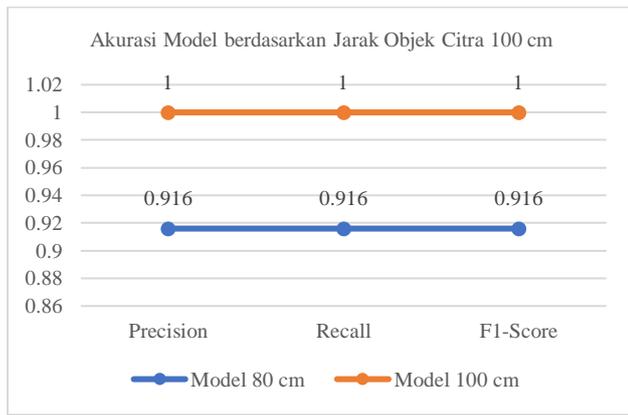
Berdasarkan evaluasi model pada Tabel 3 dan 4, maka dapat disimpulkan performa model berdasarkan jarak pengambilan citra objek botol seperti pada Gambar 9.



Gambar 12. Perbandingan akurasi berdasarkan jarak 80 cm

Pada Gambar 12, jika jarak pengambilan citra berjarak 80 cm diterapkan pada model 80 cm maka memperoleh akurasi yang baik yaitu 100%, tetapi jika diterapkan pada model 100 cm maka menghasilkan akurasi performa 67%.

Pada Gambar 13, untuk pengambilan citra botol pada jarak 100 cm diterapkan pada model 80 cm memiliki performa akurasi sebesar 91,6%, namun jika diterapkan pada model 100 cm memperoleh hasil akurasi 100%.



Gambar 13. Perbandingan akurasi berdasarkan jarak 100 cm

4. KESIMPULAN

Model 80 cm dan model 100 cm sangat baik diterapkan untuk data citra dengan jarak yang sesuai dengan modelnya, seperti untuk data uji jarak 80 cm diterapkan pada model 80 cm maka menghasilkan tingkat akurasi 100%, ini berarti citra dapat mengenali kelasnya dengan tepat yaitu kelas 1,5L, 600 ml dan 330 ml begitu juga untuk data uji dengan jarak 100 cm diterapkan pada model 100 cm. Hasil yang tidak baik terlihat pada data uji dengan jarak 80 cm diterapkan pada model 100 cm yaitu dengan hasil tingkat akurasi 67%, ini bermakna bahwa terdapat beberapa kesalahan prediksi objek citra, sedangkan untuk citra jarak 100 cm diterapkan pada model 80 cm masih cukup baik dengan akurasi 91,6%. Berdasarkan hasil penelitian bahwa jarak pengambilan citra mempengaruhi klasifikasi deteksi objek citra.

Ucapan Terima Kasih

Terima kasih kepada Universitas Muhammadiyah Jakarta melalui LPPM UMJ atas bantuan hibah penelitian ini.

DAFTAR PUSTAKA

- [1] Sirojul Munir, Ahmad Adriansyah, Radon Dhelika, "Implementasi Arsitektur Aplikasi MVC pada Perancangan Aplikasi Bank Sampah Berbasis Web," *J. Teknol. Terpadu*, vol. 4, no. 2, hal. 76–81, 2018, doi: <https://doi.org/10.54914/jtt.v4i2.131>.
- [2] Imamul Arifin, Reydiko Fakhra Haidi, "Penerapan Computer Vision Menggunakan Metode Deep Learning Pada Perspektif Generasi Ulul Albab," *J. Teknol. Terpadu*, vol. 7, no. 2, hal. 98–107, 2021, doi: <https://doi.org/10.54914/jtt.v7i2.436>.
- [3] A. B. Wahyutama dan M. Hwang, "YOLO-Based Object Detection for Separate Collection of Recyclables and Capacity Monitoring of Trash Bins," *Electron.*, vol. 11, no. 9, 2022, doi: 10.3390/electronics11091323.
- [4] P. O. Parasian dan F. Utamingrum, "Rancang Bangun Sistem Pengklasifikasi Jenis Sampah Organik dan Sampah Daur Ulang menggunakan Resnet50," vol. 6, no. 4, hal. 2548–964, 2022, [Daring]. Tersedia pada: <http://j-ptiik.ub.ac.id>.

- [5] W. L. Mao, W. C. Chen, H. I. K. Fathurrahman, dan Y. H. Lin, "Deep learning networks for real-time regional domestic waste detection," *J. Clean. Prod.*, vol. 344, no. October 2021, hal. 131096, 2022, doi: 10.1016/j.jclepro.2022.131096.
- [6] X. Long *et al.*, "PP-YOLO: An Effective and Efficient Implementation of Object Detector," 2020, [Daring]. Tersedia pada: <http://arxiv.org/abs/2007.12099>.
- [7] Y. Resnet, K. R. Tanjung, H. Juwiantho, dan J. S. Surabaya, "Klasifikasi Benda Organik dan Anorganik Dengan Metode."
- [8] T. L. C. Tran, Z. C. Huang, K. H. Tseng, dan P. H. Chou, "Detection of Bottle Marine Debris Using Unmanned Aerial Vehicles and Machine Learning Techniques," *Drones*, vol. 6, no. 12, hal. 14–16, 2022, doi: 10.3390/drones6120401.
- [9] R. A. M. AlShehri dan A. K. J. Saudagar, "Detecting Threats from Live Videos using Deep Learning Algorithms," *Int. J. Adv. Comput. Sci. Appl.*, vol. 14, no. 11, hal. 643–658, 2023, doi: 10.14569/IJACSA.2023.0141166.
- [10] H. Lou *et al.*, "DC-YOLOv8: Small-Size Object Detection Algorithm Based on Camera Sensor," *Electron.*, vol. 12, no. 10, hal. 1–14, 2023, doi: 10.3390/electronics12102323.
- [11] M. Safaldin, N. Zaghden, dan M. Mejdoub, "An Improved YOLOv8 to Detect Moving Objects," *IEEE Access*, vol. 12, no. May, hal. 59782–59806, 2024, doi: 10.1109/ACCESS.2024.3393835.
- [12] P. Jiang, D. Ergu, F. Liu, Y. Cai, dan B. Ma, "A Review of Yolo Algorithm Developments," *Procedia Comput. Sci.*, vol. 199, hal. 1066–1073, 2021, doi: 10.1016/j.procs.2022.01.135.
- [13] R. Hesananda, D. N. Simatupang, N. Wiliani, I. Technology, dan S. Program, "Cloth Bag Object Detection Using The Yolo Algorithm V5," vol. 18, no. 2, 2022, doi: 10.33480/pilar.v18i2.3019.
- [14] Jupiyandi Saniputra, F. R. Pratama, dan Yoga Dharmawan, "Pengembangan Deteksi Citra Mobil Untuk Mengetahui Jumlah Tempat Parkir Menggunakan Cuda Dan Modified Yolo Development of Car Image Detection To Find Out the Number of Parking Space Using Cuda and Modified Yolo," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 6, no. 4, hal. 413–419, 2019, doi: 10.25126/jtiik.201961275.
- [15] L. Yu, M. Qian, Q. Chen, F. Sun, dan J. Pan, "An Improved YOLOv5 Model: Application to Mixed Impurities Detection for Walnut Kernels," *Foods*, vol. 12, no. 3, 2023, doi: 10.3390/foods12030624.



PEMANFAATAN DATA ULASAN PENGGUNA UNTUK MEMBANGUN SISTEM KLASTERISASI BERDASARKAN *PAIN POINTS* MENGGUNAKAN ALGORITMA *K-MEANS*

Ikhya Ulummuddin¹, Angraini Puspita Sari², Made Hanindia Prami Swari³

^{1,2,3} Informatika, Universitas Pembangunan Nasional “Veteran” Jawa Timur
Surabaya, Jawa Timur, Indonesia 60294

20081010120@student.upnjatim.ac.id, angraini.puspita.if@upnjatim.ac.id, madehanindia.fik@upnjatim.ac.id

Abstract

In design thinking, empathizing and defining stages are part of UX research. The goal is to analyze pain points or complaints experienced by users using qualitative data. However, this process is always done manually, which can be time-consuming and resource-intensive. The objective of this research is to develop a system for clustering qualitative data based on problem topics using K-Means clustering and several evaluation methods, namely silhouette score, Davies-Bouldin Index, and Calinski-Harabasz Index, implemented in Python programming language and run on Google Colaboratory. User review data for the Gojek app version 4.9.3 from November 2021 to January 2024, obtained from Kaggle and preprocessed, will be used as the object for system development. Based on testing for each cluster number, the results obtained are 14 clusters or problem topics with a silhouette score of 0.65, Davies-Bouldin Index of 0.35, and Calinski-Harabasz Index of 40.7, where each evaluation method has good accuracy. The system requires a computation time of 127.4 seconds. The K-Means algorithm is effective when clustered user review data based on complaint topics. UX researchers can utilize the system from this research to assist them in analyzing pain points more quickly and efficiently.

Keywords: Clustering, K-Means, Natural Language Processing, Pain Points, User Experience

Abstrak

Dalam *design thinking*, tahap *empathize* dan *define* termasuk bagian *UX research*. Tujuannya untuk menganalisis *pain points* atau keluhan yang dialami oleh para pengguna dari data kualitatif yang didapatkan. Namun, proses ini selalu dilakukan secara manual, yang mana dapat memakan waktu dan sumber daya yang lebih banyak. Tujuan dari penelitian ini adalah mengembangkan sistem pengelompokan data kualitatif berdasarkan topik masalah menggunakan *K-Means clustering* dan beberapa metode evaluasi, yaitu *silhouette score*, *Davies-Bouldin Index*, dan *Calinski-Harabasz Index* dengan bahasa pemrograman *Python* yang dijalankan pada *Google Colaboratory*. Data ulasan pengguna aplikasi Gojek versi 4.9.3 dengan rentang waktu November 2021 sampai dengan Januari 2024 dari Kaggle yang telah dilakukan beberapa tahap *preprocessing*, akan dimanfaatkan sebagai objek pengembangan sistem. Berdasarkan pengujian setiap jumlah kluster, hasil yang didapatkan adalah 14 kluster atau topik masalah dengan *silhouette score* 0.65, *Davies-Bouldin Index* 0.35 dan *Calinski-Harabasz Index* 40.7, yang mana setiap metode evaluasi memiliki kualitas akurasi yang baik. Sistem membutuhkan waktu komputasi selama 127.4 detik. Algoritma *K-Means* menunjukkan keefektifannya ketika digunakan untuk mengelompokkan data ulasan pengguna berdasarkan topik keluhannya. Sistem dari hasil penelitian ini dapat dimanfaatkan oleh para *UX researcher* untuk membantu mereka dalam menganalisis *pain points* lebih mudah dan efisien.

Kata kunci: Klusterisasi, *K-Means*, Pemrosesan Bahasa Alami, Titik Nyeri, Pengalaman Pengguna

1. PENDAHULUAN

Seiring dengan berjalannya waktu, banyak hal yang mulai berkembang di kehidupan manusia. Awalnya, pemahaman seseorang tentang desain hanya sebatas tentang keestetikaan saja. Namun, sekarang desain dapat digunakan untuk memberikan solusi terhadap masalah-masalah yang ada, contohnya dalam mengembangkan

produk digital. Mendesain produk digital dengan baik dapat memberikan banyak keuntungan, salah satunya adalah bisnis. Untuk mendapatkan desain produk digital yang baik, diperlukan penerapan *user interface (UI)* dan *user experience (UX)*. *UI* merupakan visualisasi setiap komponen produk yang dapat diinteraksi oleh penggunanya, sedangkan *UX* merupakan semua

pengalaman pengguna ketika berinteraksi dengan produk. Konsep tersebut dapat menuntun suatu produk digital agar sesuai dengan kebutuhan para penggunanya, mulai dari desain antarmuka sampai dengan pengalaman penggunanya. Suatu produk digital yang berhasil pasti memiliki desain antarmuka dan pengalaman pengguna yang berdasarkan analisis kebutuhan pengguna yang dikaji terus menerus [1]. Untuk menerapkan konsep *UI* dan *UX* secara terstruktur dan sesuai dengan kebutuhan pengguna produk, perlu digunakan sebuah kerangka kerja, salah satunya adalah *design thinking*. *Design thinking* merupakan kerangka kerja desain untuk mencari solusi inovatif dari masalah yang dialami pengguna secara langsung. Pendekatan *design thinking* digunakan untuk mengetahui kebutuhan, sifat-sifat, dan preferensi yang memprioritaskan pengguna akhir secara langsung [2]. Setiap tahapan *design thinking* berasal dan memang ditujukan kepada manusia secara langsung [3]. Ada beberapa tahapan dari *design thinking*, yaitu *empathize*, *define*, *ideate*, *prototype*, dan *test*. Setiap tahapan dapat dilakukan tidak berurutan sesuai dengan kebutuhan. Metode *design thinking* dilakukan secara beriterasi secara terus menerus. Tahap *empathize* dan *define* dalam *design thinking* termasuk dalam tahap proses *UX research* juga. *UX research* merupakan tahap mengumpulkan *insight* tentang kebutuhan dan keluhan dari pengguna produk secara langsung dengan tujuan mengidentifikasi perilaku pengguna ketika berinteraksi dengan produk. *UX research* sudah banyak diterapkan ketika produk masih dalam proses pengembangan dengan menggunakan persepsi subjektif pengguna terhadap produk [4]. Tahap *empathize* menjadi kunci awal dari mendesain suatu produk karena tahap tersebut yang menentukan bagaimana desain dapat memenuhi kebutuhan pengguna dan bisnis dengan baik. Untuk mendapatkan informasi tentang pengguna, biasanya *UX researcher* mengumpulkan data ulasan pengguna terhadap aplikasi tersebut di suatu layanan distribusi digital. Tahap *UX research* selanjutnya adalah *define*, yaitu berisi aktivitas menganalisis dan merangkum data yang dikumpulkan di tahap *empathize* untuk diidentifikasi topik masalah yang sering dirasakan oleh pengguna. Selama ini, proses penentuan topik masalah selalu dilakukan secara manual oleh para *UX researcher*. Cara manual tersebut membuat tahap *define* dapat berlangsung selama beberapa hari hingga minggu [5]. Durasi dan sumber daya untuk menentukan topik masalah akan dibutuhkan lebih banyak lagi apabila data kualitatif yang didapatkan ketika tahap *empathize* lebih banyak dan kompleks. Berdasarkan adanya masalah tersebut, perlu di desain sistem untuk melakukan klasterisasi data kualitatif yang telah didapatkan oleh *UX researcher* ketika tahap *empathize* ke dalam beberapa topik *pain points* dengan menggunakan salah satu *artificial intelligence (AI)*, yaitu metode algoritma *K-Means clustering*.

Algoritma *K-Means clustering* merupakan *unsupervised machine learning*. Tujuan dari algoritma klasterisasi

tersebut adalah untuk mengelompokkan kumpulan data ke dalam suatu *cluster* yang memiliki kesamaan tertentu. Untuk pengujian sistem klasterisasi tersebut, dimanfaatkan *dataset* ulasan pengguna aplikasi Gojek yang didapatkan dari Kaggle sebagai objek penelitian. Tujuannya untuk mengetahui apakah sistem yang dirancang sudah cukup memenuhi untuk mengelompokkan data kualitatif yang didapatkan pada tahap *empathize* kedepannya. Sebelum *dataset* digunakan, perlu dilakukan tahap *text preprocessing* agar dapat diklasterisasi menggunakan algoritma *K-Means*. Tahapan-tahapannya di mulai dari *tokenizing*, *stopwords removal*, *stemming*, *text cleaning*, dan vektorisasi numerik menggunakan *TF-IDF*. Setelah itu, setiap jumlah *cluster* akan dievaluasi menggunakan metrik pengujian *silhouette score*, *Davies-Bouldin Index (DBI)*, dan *Calinski-Harabasz Index (CHI)* untuk menentukan jumlah kluster optimal. Nantinya setiap kluster akan mempresentasikan topik masalah atau *pain points* apa saja yang dialami oleh pengguna aplikasi Gojek berdasarkan *dataset* yang digunakan sebagai objek penelitian.

Ada beberapa penelitian yang membahas tentang pengelompokan data kualitatif menggunakan algoritma *K-Means*, yaitu penelitian yang dilakukan oleh Sembiring, T. A. B. dan Hasibuan, M. S. tentang mengelompokkan bahasa Karo berdasarkan dialeknya menggunakan algoritma *K-Means* dengan cara mengambil nilai pembobotan *Term Frequency-Inverse Document Frequency (TF-IDF)* [6]. Penelitian lainnya yaitu tentang mencari kesamaan pola tema dari terjemahan ayat Al-Qur'an menggunakan algoritma *K-Means clustering* dan skema pembobotan dengan *TF-IDF* yang dilakukan oleh Wahyudi, M. D. R. [7]. Penelitian terdahulu selanjutnya yang dilakukan oleh Marisa, F., dkk., yaitu menganalisis nilai setiap kluster terhadap hasil pengelompokan data pelanggan pada pendekatan *K-Means* dengan metode *elbow* [8]. Berdasarkan beberapa penelitian terdahulu yang berkaitan dengan pengelompokan data teks menggunakan algoritma *K-Means*, ada peluang yang cukup untuk membuat sistem klasterisasi data keluhan pengguna produk berdasarkan topik *pain points*-nya untuk membantu *UX researcher* dalam melakukan tahap *define*. Jika waktu yang dibutuhkan pada tahap tersebut lebih cepat, maka para *UX researcher* bisa memiliki waktu lebih banyak untuk mencari solusi kreatif dalam tahap *ideate* terhadap masalah yang telah didefinisikan.

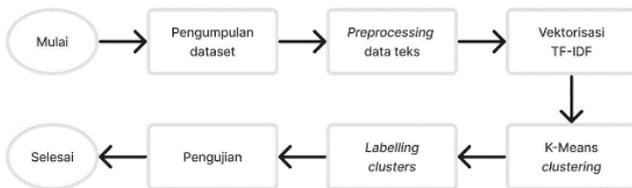
2. METODE PENELITIAN

2.1 Metode pengumpulan data

Data yang digunakan untuk pengujian sistem dari penelitian ini adalah *dataset* ulasan pengguna aplikasi Gojek yang didapatkan dari Kaggle yang diunggah oleh Ari Zidane [9]. Diambil data ulasan pengguna dengan versi aplikasi 4.9.3 dari semua *rating*.

2.2 Tahapan penelitian

Di dalam penelitian yang dilakukan, ada beberapa tahapan yang perlu dilakukan untuk membuat sistem pengelompokan data ulasan yang sesuai dengan topik masalahnya, yaitu mengambil *dataset* ulasan pengguna, *preprocessing* teks, melakukan vektorisasi numerik, *clustering* dengan algoritma *K-Means*, memberikan label ke setiap *cluster*, dan mengevaluasi setiap jumlah kluster menggunakan metrik pengujian. Tahapan-tahapan dari penelitian ini dapat dilihat pada Gambar 1.



Gambar 1. Tahapan penelitian

2.2.1 Pengumpulan *dataset*

Dataset berisi data ulasan pengguna aplikasi Gojek yang diambil dari Kaggle berisi data *username* pengguna, data ulasan, *rating*, waktu penulisan ulasan, dan versi aplikasinya. *Dataset* tersebut berisi 225.002 data ulasan pengguna aplikasi Gojek.

2.2.2 *Preprocessing* data teks

Sebelum menggunakan data teks sebagai objek analisis, diperlukan *preprocessing* agar data teks memiliki kualitas yang baik dan lebih mudah untuk diolah. Kualitas data teks memiliki peran penting untuk meningkatkan kinerja algoritma karena jika data tidak dilakukan *preprocessing*, maka algoritma yang digunakan akan memiliki performa yang berkurang karena data yang tidak konsisten [10]. Semua tahap *text preprocessing* menggunakan *library* dari *Natural Language Toolkit* (NLTK). Di dalam tahap *preprocessing*, kumpulan data teks yang berisi masalah pengguna akan dilakukan *tokenizing*, *stopwords removal*, *stemming*, serta *text cleaning* dan dijelaskan sebagai berikut.

a. *Tokenizing*

Tokenizing adalah tahap *preprocessing* data teks yang bertujuan untuk memisahkan atau memecah kumpulan teks menjadi kumpulan unit yang lebih kecil atau kata individual.

b. *Stopwords removal*

Stopword removal adalah tahap *preprocessing* data teks yang bertujuan untuk menghapus kumpulan kata yang kurang memiliki makna signifikan dan tidak memiliki informasi yang tinggi dalam data teks. Tidak hanya menghapus atau membuat *stoplist*, tetapi juga dapat membuat *wordlist* atau menyimpan kata penting yang tidak ingin dihapus dari proses *stopword removal*.

c. *Stemming*

Stemming adalah tahap *preprocessing* data teks yang bertujuan untuk mengubah kumpulan kata dalam teks menjadi bentuk dasarnya dengan mempertimbangkan konteksnya. Proses ini dilakukan dengan menghapus imbuhan, seperti awalan, akhiran, dan sisipan dalam setiap kata.

d. *Text cleaning*

Text cleaning adalah tahap untuk menghilangkan simbol-simbol yang ada di dalam data teks. Tujuannya adalah agar teks lebih mudah untuk diproses dan terhindar dari kumpulan simbol yang tidak relevan atau diinginkan, seperti simbol, tanda baca, dan karakter khusus lainnya.

2.2.3 Vektorisasi TF-IDF

TF-IDF adalah teknik untuk mengukur nilai setiap kata dalam teks berdasarkan frekuensi kemunculan dan umumnya kata tersebut secara keseluruhan. TF-IDF merupakan gabungan dari dua konsep yaitu nilai TF dan nilai IDF. TF bertujuan untuk menghitung banyaknya frekuensi kata keluar di dalam teks, sedangkan IDF bertujuan untuk menghitung banyaknya dokumen dalam *corpus* yang mengandung kata tersebut. Persamaan (1) merupakan rumus menghitung nilai TF.

$$TF(t, d) = \frac{n(t, d)}{N(d)} \quad (1)$$

Keterangan:

t = Kata yang ingin dihitung TF-IDF-nya.

d = Dokumen yang ingin dihitung TF-IDF-nya

$n(t, d)$ = Jumlah kemunculan kata t dalam dokumen d

$N(d)$ = Jumlah kata dalam dokumen d

Untuk menghitung IDF, dapat menggunakan persamaan (2).

$$IDF(t, d) = \log \left(\frac{N}{df(t)} \right) \quad (2)$$

Keterangan:

N = Jumlah dokumen dalam *corpus*

$df(t)$ = Jumlah dokumen yang mengandung kata t

Untuk menghitung TF-IDF-nya, dapat menggunakan persamaan (3).

$$TF - IDF(t, d) = TF(t, d) * IDF(t) \quad (3)$$

Proses pembobotan di setiap kata di dalam *dataset* perlu dilakukan untuk mendapatkan performa algoritma yang digunakan bisa lebih baik karena ada beberapa algoritma yang membutuhkan proses tersebut [11]. Vektorisasi teks adalah tahap *preprocessing* data teks yang bertujuan untuk mengubah data teks menjadi representasi vektor numerik

yang dapat diidentifikasi oleh komputer. Data teks yang belum divektorisasi tidak dapat langsung digunakan untuk model *training* atau algoritma tertentu karena operasi ekstraksi fitur belum dilakukan [12]. Vektorisasi numerik TF-IDF yang dilakukan pada sistem ini menggunakan *library Scikit-learn*.

2.2.4 K-Means clustering

K-Means clustering termasuk algoritma sistem pembatas yang digunakan untuk menentukan total permulaan kelompok yang mendefinisikan nilai permulaan dari titik pusat atau *centroid* [13]. Kata *K* artinya konstanta terhadap jumlah *cluster* yang diperlukan dan kata *means* artinya nilai rata-rata dari kelompok-kelompok data yang telah didefinisikan menjadi *cluster* [14]. Algoritma *K-Means* bertujuan untuk membagi kumpulan data menjadi beberapa *k* kelompok atau *clusters* yang berbeda dengan meminimalkan variansi antar data dalam kelompok dan memaksimalkan variansi antar kelompok. Ada beberapa tahap proses algoritma *K-Means clustering*, mulai dari menentukan nilai *k* atau *cluster* yang ingin dibuat, menginisialisasi nilai *centroid* atau *center* dari *cluster* secara *random*, menetapkan jarak setiap data *point* ke *centroid* terdekat, melakukan perhitungan ulang nilai *centroid* dari *cluster* yang telah terbuat, melakukan optimasi agar kriteria terpenuhi dengan melakukan iterasi tahap ke tiga dan empat. Setiap *cluster* mempunyai *centroid* atau titik pusat yang menggambarkan *cluster* tersebut [15]. Untuk menghitung jarak antara dua data *point*, dapat menggunakan persamaan (4).

$$d(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (4)$$

Keterangan:

- D = Selisih jarak data *point* *x* dan *y*
x = Nilai data ke-*i*
y = Nilai *centroid* data ke-*i*
n = Jumlah atribut data

2.2.5 Labelling clusters

Cluster yang ditentukan oleh algoritma *K-Means* masih berupa angka belum menjadi nama topik dari *cluster* tersebut. Pengubahan angka klaster menjadi topik inti masalah klaster dapat dilakukan dengan cara mengambil semua data teks ulasan berdasarkan setiap klaster. Data setiap kalimat di dalam *cluster* akan dihitung berdasarkan keberadaan kata-kata yang paling umum dan memiliki nilai tertinggi menggunakan TF-IDF. Proses *labelling clusters* dengan TF-IDF menggunakan bantuan *library Scikit-learn*. Data kalimat yang sesuai dengan ketentuan akan dipilih sebagai ringkasan dari *cluster* tersebut.

2.2.6 Pengujian

Pengujian dilakukan untuk mengetahui kualitas dan efisiensi sistem dalam mengelompokkan data ulasan pengguna berdasarkan topik masalahnya. Untuk menguji kualitas pengelompokannya, dilakukan evaluasi pada setiap jumlah klasternya menggunakan beberapa metrik pengujian, yaitu *silhouette score*, *Davies-Bouldin Index*, dan *Calinski-Harabasz Index*. Hasil dari beberapa metrik pengujian akan dibandingkan untuk diketahui jumlah klaster optimal dari 2 sampai dengan jumlah data sampelnya. Setiap metrik tersebut dapat diimplementasikan menggunakan *library Scikit-learn*. Untuk menguji efisiensi sistem, dilakukan penghitungan waktu eksekusi sistem dalam mengelompokkan data ulasan pengguna dari awal hingga akhir proses.

3. HASIL DAN PEMBAHASAN

3.1 Pengumpulan data

Dataset yang digunakan untuk penelitian ini adalah data ulasan pengguna aplikasi Gojek dengan versi aplikasi 4.9.3 dan rentang waktu dari bulan November 2021 sampai dengan Januari 2024. Didapatkan 112 data ulasan pengguna dengan *rating* bintang 1 sampai dengan 5. *Dataset* tersebut dilakukan *filtering* dan *fixing* data karena ada beberapa data ulasan yang struktur kalimatnya kurang baik, tidak relevan, dan tidak menggambarkan suatu masalah yang dialami oleh para pengguna aplikasi. Setelah dilakukan *filtering* dan *fixing* data, didapatkan 33 data ulasan pengguna. Beberapa data ulasan pengguna yang telah dilakukan *filtering* dan *fixing* ditampilkan pada Tabel 1.

Tabel 1. Hasil pengumpulan data ulasan pengguna

Data ulasan
Susah dan lama untuk menemukan <i>driver</i>
Ada beberapa lokasi yang belum diperbarui
Ongkirnya mahal banget dan tidak ada promo atau voucher diskon
Aplikasi dibuka malah keluar sendiri terus
Lambat sekali aplikasi nya tidak seperti dulu

3.2 Preprocessing data teks

Dilakukan beberapa tahapan *preprocessing* data teks, yaitu *tokenizing*, *stopwords removal*, *stemming*, dan *text cleaning*. Namun, ada beberapa kata yang dimasukkan ke dalam daftar *wordlist* agar intisari dari suatu kalimat tidak hilang. Hasil dari *preprocessing* data teks ditampilkan pada Tabel 2.

Tabel 2. Hasil *preprocessing* data

Data ulasan
susah lama menemukan <i>driver</i>
lokasi belum diperbarui

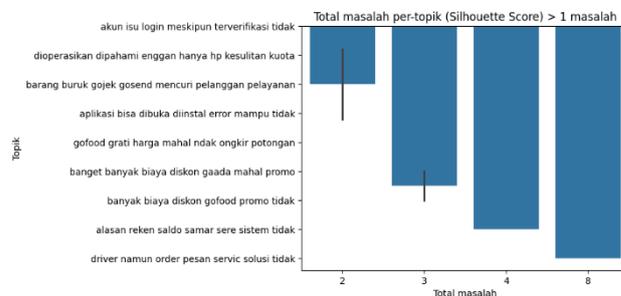
Jumlah klaster	Silhouette score	Davies-Bouldin Index	Calinski-Harabasz Index
7	0.185	1.040	2.992
8	0.240	1.135	3.899
9	0.564	0.811	8.681
10	0.579	0.890	11.435
11	0.304	1.055	4.243
12	0.451	1.037	7.423
13	0.311	0.821	5.343
14	0.655	0.351	40.704
15	0.463	0.720	11.276
16	0.614	0.397	48.660
17	0.471	0.544	34.614
18	0.474	0.762	6.705
19	0.347	0.480	24.310
20	0.413	0.431	44.792
21	0.438	0.362	66.842
22	0.338	0.309	32.726
23	0.314	0.442	39.220
24	0.405	0.214	57.140
25	0.321	0.141	226.708
26	0.281	0.268	19.364
27	0.249	0.215	28.366
28	0.223	0.184	531.237
29	0.149	0.289	35.520
30	0.085	0.194	182.865
31	0.019	0.253	38.569

Hasil evaluasi tersebut menunjukkan bahwa pada setiap metrik pengujian memberikan jumlah klaster optimal yang berbeda-beda. Untuk *silhouette score* menunjukkan jumlah klaster optimal 14 dengan nilai *index* 0.65, *Davies-Bouldin Index* menunjukkan jumlah klaster optimal 25 dengan nilai *index* 0.14 dan *Calinski-Harabasz Index* menunjukkan jumlah klaster optimal 28 dengan nilai *index* 531.23. Untuk metrik pengujian *silhouette score* apabila nilai *index*-nya mendekati atau sama dengan 1, itu menunjukkan bahwa klaster tersebut optimal. Pada metrik pengujian *Davies-Bouldin Index* apabila nilai *index*-nya mendekati atau sama dengan 0, itu menunjukkan bahwa klaster tersebut optimal. Di metrik pengujian *Calinski-Harabasz Index* apabila nilai *index*-nya melebihi 0, itu menunjukkan bahwa klaster tersebut telah optimal. Perbedaan jumlah klaster optimal yang didapatkan oleh setiap metrik pengujian diharuskan untuk mempertimbangkan hasil pengelompokan terhadap setiap jumlah klaster, yaitu 14, 25, dan 28. Tujuannya untuk mendapatkan hasil yang

terbaik dengan menyampingkan nilai *index* yang dianggap oleh metrik pengujian bahwa itu termasuk jumlah klaster yang optimal.

Berdasarkan evaluasi hasil pengelompokan data ulasan pengguna untuk klaster 14, 25, dan 28 pada setiap metrik pengujian, yaitu *silhouette score*, *Davies-Bouldin Index*, dan *Calinski-Harabasz Index*, memiliki beberapa kekurangan dan kelebihan. Untuk *silhouette score*, kekurangannya adalah ada beberapa data ulasan pengguna yang masuk ke dalam klaster yang tidak seharusnya dan kelebihan adalah ada banyak data ulasan pengguna yang masuk ke dalam klaster yang sesuai dengan topik masalahnya, sebanyak 6 klaster. Untuk hasil pengelompokan dari *Davies-Bouldin Index* dan *Calinski-Harabasz Index* memiliki kekurangan, yaitu ada beberapa data ulasan pengguna yang membuat klaster sendiri meskipun ada klaster lain yang memiliki topik yang sama. Untuk data yang masuk ke dalam klaster yang sesuai dengan topiknya, *Davies-Bouldin Index* memiliki 5 klaster dan *Calinski-Harabasz Index* memiliki 4 klaster.

Oleh karena itu, jumlah klaster optimal sebanyak 14 yang didapatkan dari metrik pengujian *silhouette score* akan digunakan sebagai hasil pengelompokan data ulasan pengguna aplikasi Gojek. Hasil pengelompokan data ulasan pengguna tersebut divisualisasikan menggunakan diagram batang agar diketahui topik masalah mana yang paling banyak dialami oleh pengguna. Visualisasi hasil pengelompokan dapat ditampilkan pada Gambar 5.



Gambar 5. Frekuensi data setiap topik masalah

Dari hasil grafik tersebut menunjukkan bahwa para pengguna merasa aplikasi yang digunakan sering mengalami masalah ketika memesan layanan dan sudah tidak banyak promo serta diskon yang diterima akhir-akhir ini, yang mana membuat biaya layanan menjadi lebih mahal.

4. KESIMPULAN

Berdasarkan penelitian yang telah dilakukan, didapatkan kesimpulan bahwa algoritma *K-Means* cocok diimplementasikan untuk mengelompokkan data kualitatif seperti data ulasan pengguna dari suatu aplikasi. Dari hasil evaluasi pengelompokan pada setiap metrik pengujian, hasil dari metrik pengujian *silhouette score*, dengan jumlah klaster optimal 14 dengan nilai *index* 0.65, dipilih sebagai jumlah klaster optimal karena mendapatkan hasil

pengelompokan yang cukup baik dibandingkan dengan hasil metrik pengujian lainnya. Metrik dari *Davies-Bouldin Index* menunjukkan jumlah kluster optimalnya 25 dengan nilai *index* 0.14 dan *Calinski-Harabasz Index* menunjukkan jumlah kluster optimalnya 28 dengan nilai *index* 531.23, tetapi hasil pengelompokan dari dua metrik tersebut masih terlihat kurang baik daripada hasil dari jumlah kluster yang ditentukan oleh *silhouette score*. Perbedaan jumlah kluster optimal yang ditentukan oleh setiap metrik bisa terjadi karena adanya perbedaan cara pengukuran evaluasi hasil pengelompokan. Untuk saran penelitian selanjutnya dapat dilakukan lagi pengembangan sistem menjadi platform yang bisa digunakan oleh siapa pun dengan lebih mudah, menambahkan beberapa fitur baru yang mungkin diperlukan kedepannya, dan memperbaiki apa yang kurang dari penelitian ini.

DAFTAR PUSTAKA

- [1] S. Tazkiyah, A. Arifin, "Perancangan UI/UX Pada Website Laboratorium Energy Menggunakan Aplikasi Figma," *Jurnal Teknologi Terpadu*, vol. 8, no. 2, pp. 72–78, 2022, doi: 10.54914/jtt.v8i2.513.
- [2] Y. S. Purbo, F. S. Utomo, Y. Purwati, "Analisis dan Perancangan Antarmuka Aplikasi Wisata Menggunakan Metode User-Centered Design (UCD)," *Jurnal Teknologi Terpadu*, vol. 9, no. 2, pp. 123–132, 2023, doi: 10.54914/jtt.v9i2.977.
- [3] L. Setiyani, E. Tjandra, "UI / UX Design Model for Student Complaint Handling Application Using Design Thinking Method (Case Study: STMIK Rosma Karawang)," *International Journal Of Science Technology & Management*, vol. 3, no. 3, pp. 690–702, 2022, doi: 10.46729/ijstm.v3i3.505.
- [4] L. Luther, V. Tiberius, A. Brem, "User Experience (UX) in Business, Management, and Psychology: A Bibliometric Mapping of the Current State of Research," *Multimodal Technologies and Interaction*, vol. 4, no. 2, pp. 1-19, 2020, doi: 10.3390/mti4020018.
- [5] M. Bloemendaal, "A Breakdown of The Time and Resources Needed to Implement Design Thinking," 2023. <https://studiowhy.com/a-breakdown-of-the-time-and-resources-needed-to-implement-design-thinking>.
- [6] T. A. B. Sembiring, M. S. Hasibuan, "Text Clustering in Karo Language Using TF-IDF Weighting and K-Means Clustering," *Jurnal Teknik Informatika*, vol. 4, no. 5, 2023, doi: 10.52436/1.jutif.2023.4.5.1462.
- [7] M. D. R. Wahyudi, "Evaluation of TF-IDF Algorithm Weighting Scheme in The Qur'an Translation Clustering with K-Means Algorithm," *Journal of Information Technology and Computer Science*, vol. 6, no. 2, pp. 117–129, 2021, doi: 10.25126/jitecs.202162295.
- [8] F. Marisa, A. R. Wardhani, W. Purnomowati, A. V. Vitianingsih, A. L. Maukar, E. W. Puspitarini, "Potential Customer Analysis Using K-Means With Elbow Method," *Jurnal Informatika dan Komputer*, vol. 7, no. 2, pp. 307–312, 2023, doi: 10.26798/jiko.v7i2.911.
- [9] A. Zidane, "Gojek App Reviews Bahasa Indonesia," 2024. <https://www.kaggle.com/datasets/ucupsedaya/gojek-app-reviews-bahasa-indonesia>.
- [10] H. Woo, J. Kim, W. Lee, "Validation of Text Data Preprocessing Using a Neural Network Model," *Mathematical Problems in Engineering*, pp. 1–9, 2020, doi: 10.1155/2020/1958149.
- [11] O. I. Gifari, M. Adha, I. R. Hendrawan, F. F. S. Durrand, "Film Review Sentiment Analysis Using TF-IDF and Support Vector Machine," *Journal of Information Technology*, vol. 2, no. 1, pp. 36–40, 2022, doi: 10.46229/jifotech.v2i1.330.
- [12] X. Yang, K. Yang, T. Cui, M. Chen, L. He, "A Study of Text Vectorization Method Combining Topic Model and Transfer Learning," *Processes*, vol. 10, no. 2, pp. 1–16, 2023, doi: 10.3390/pr10020350.
- [13] A. P. Sari, A. N. Sihananto, D. A. Prasetya, M. M. Al Haromainy, "Pengelompokan Tingkat Penyebaran Covid 19 Pada Kabupaten Jombang Dengan Menggunakan Algoritma K-Means," *Jurnal Teknologi Informasi dan Komunikasi*, vol. 17, no. 3, pp. 7–12. 2022.
- [14] D. Nurfitriana, A. Voutama, "Penerapan K-Means dan Rank Order Centroid Pada Proporsi Individu Dengan Keterampilan Teknologi Informasi dan Komputer," *Jurnal Teknologi Terpadu*, vol. 9, no. 2, pp. 70–78, 2023, doi: 10.54914/jtt.v9i2.608.
- [15] Carudin, "Pemanfaatan Data Transaksi Untuk Dasar Membangun Strategi Berdasarkan Karakteristik Pelanggan Dengan Algoritma K-Means Clustering dan Model RFM," *Jurnal Teknologi Terpadu*, vol. 7, no. 1, pp. 7–14, 2021, doi: 10.54914/jtt.v7i1.318.

Jurnal Teknologi Terpadu
Volume. 10 No. 1, Juli 2024

Daftar Isi

Klasifikasi Penyakit Daun Pisang menggunakan <i>Convolutional Neural Network</i> (CNN)	1
M. Duta Pratama, Rendra Gustriansyah, Evi Purnamasari	
Implementasi Metode <i>Hybrid Filtering Technique</i> pada Penentuan <i>Rating</i> Pestisida	7
Ardimansyah, Husain, Herlinda, Kasmawaru, Nurdiansah, Marsa	
Analisis Prediksi Kata Kunci Situs Web MonsterMAC dengan Metode <i>Long Short-Term Memory</i> (LSTM)	14
Fityan Hanif Assalmi, Wahyu Syaifullah Jauharis Saputra, Amri Muhaimin	
Optimasi Parameter DBSCAN menggunakan Metode <i>Differential Evolution</i> untuk Deteksi Anomali pada Data Transaksi Bank	22
Rifqi Karunia Ibadirachman, Yulison Herry Chrisnanto, Puspita Nurul Sabrina	
Penerapan Metode <i>Combined Compromise Solution</i> (CoCoSo) dalam Pemilihan <i>Franchise</i> Minuman	32
Julita Marito, Wahyuni Betris Nainggolan, Gede Surya Mahendra	
Perbandingan Klasifikasi Label Tunggal untuk Soal Ujian Fisika menggunakan <i>Naïve Bayes</i> dan <i>K-Fold Cross Validation</i>	40
Christopher Kevin Herijanto, Yulia Wahyuningsih	
Implementasi Bi-LSTM dengan Ekstraksi Fitur Word2Vec untuk Pengembangan Analisis Sentimen Aplikasi Identitas Kependudukan Digital	46
Romario Onsu, Daniel Febrian Sengkey, Feisy Diane Kambey	
Deteksi Citra Daun untuk Klasifikasi Penyakit Padi menggunakan Pendekatan <i>Deep Learning</i> dengan Model CNN	56
Muhammad Rijal, Andi Muhammad Yani, Abdul Rahman	
Pengaruh Jarak Objek Citra pada Model Deteksi dan Klasifikasi Botol Plastik menggunakan YOLO	63
Nurvelly Rosanti, Retnani Latifah, Sirojul Munir, Izzuddin Al Qossam Maududi	
Pemanfaatan Data Ulasan Pengguna untuk Membangun Sistem Klasterisasi berdasarkan <i>Pain Points</i> menggunakan Algoritma <i>K-Means</i>	70
Ikhya Ulummuddin, Anggraini Puspita Sari, Made Hanindia Prami Swari	

Published by :

LPPM STT Terpadu Nurul Fikri
Jln. Raya Lenteng Agung, no. 20, Srengseng Sawah, Jagakarsa,
Jakarta Selatan, DKI Jakarta 12640

Telp. 021 - 786 3191 WhatsApp. 0851 7444 3360

Email : journal@nurulfikri.ac.id

