ISSN : 2527-5836
e-ISSN : 2528-0074

Vol. 9 No. 3, September 2024

# JISKa

## Jurnal Informatika Sunan Kalijaga

Jurusan Teknik Informatika
Fakultas Sains dan Teknologi
UIN Sunan Kalijaga Yogyakarta

# Tim Pengelola JISKa (Jurnal Informatika Sunan Kalijaga)
# Edisi September 2024

# Mitra Bestari (*Reviewer*)

**Mitra Bestari Internasional (*International Reviewers*)**

1. Ardiansyah Musa Efendi, Ph.D. (Singapore Chipset Algorithm Design Lab, Huawei, Singapore)
2. Dr.Eng. M. Muhammad Syafrudin (Sejong University, Korea Selatan)
3. Mohd. Fikri Azli bin Abdullah, M.Sc. (Multimedia University, Malaysia)
4. Dr.Eng. M. Alex Syaekhoni (Who's Good, Korea Selatan)
5. Norma Latif Fitriyani, M.Sc. (Sejong University, Korea Selatan)

**Mitra Bestari Nasional (*National Reviewers*)**

1. Dr. Ir. Agung Fatwanto (UIN Sunan Kalijaga Yogyakarta, Indonesia)
2. Agus Mulyanto, S.Si., M.Kom., ASEAN Eng. (UIN Sunan Kalijaga Yogyakarta, Indonesia)
3. Ahmad Fathan Hidayatullah, M.Cs. (Universitas Islam Indonesia Yogyakarta, Indonesia)
4. Alam Rahmatulloh, M.T. (Universitas Siliwangi Tasikmalaya, Indonesia)
5. Anggi Rizky Windra Putri, M.Kom. (Universitas Aisyiyah Yogyakarta, Indonesia)
6. Dr. Ir. Bambang Sugiantoro (UIN Sunan Kalijaga Yogyakarta, Indonesia)
7. Dr. Enny Itje Sela (Universitas Teknologi Yogyakarta, Indonesia)
8. Dr.Eng. Ganjar Alfian (Universitas Gadjah Mada, Indonesia)
9. Mandahadi Kusuma, M.Eng. (UIN Sunan Kalijaga Yogyakarta, Indonesia)
10. Maria Ulfa Siregar, Ph.D. (UIN Sunan Kalijaga Yogyakarta, Indonesia)
11. Muhammad Dzulfikar Fauzi, M.Cs. (Telkom University Surabaya, Indonesia)
12. Muhammad Habibi, M.Cs. (Universitas Jenderal Achmad Yani Yogyakarta, Indonesia)
13. Muhammad Rifqi Maarif, M.Eng. (Universitas Jenderal Achmad Yani Yogyakarta, Indonesia)
14. Niki Min Hidayati Robbi, M.Eng. (Universitas Gadjah Mada, Indonesia)
15. Prof. Dr. Hj. Okfalisa, S.T., M.Sc. (UIN Sultan Syarif Kasim Riau, Indonesia)
16. Oman Somantri, M.Kom. (Politeknik Negeri Cilacap, Indonesia)
17. Puguh Jayadi, M.Kom. (Universitas PGRI Madiun, Indonesia)
18. Puji Winar Cahyo, M.Cs. (Universitas Jenderal Achmad Yani Yogyakarta, Indonesia)
19. Qorry Aina Fitroh, M.Kom. (UIN Sunan Kalijaga Yogyakarta, Indonesia)
20. Ridho Surya Kusuma, M.Kom. (Universitas Siber Muhammadiyah, Yogyakarta, Indonesia)
21. Dr. Shofwatul Uyun (UIN Sunan Kalijaga Yogyakarta, Indonesia)
22. Dr. Ir. Sumarsono, M.Kom. (UIN Sunan Kalijaga Yogyakarta, Indonesia)
23. Dr.Eng. Sunu Wibirama, M.Eng. (Universitas Gadjah Mada, Indonesia)
24. Tundo, M.Kom. (Sekolah Tinggi Ilmu Komputer Cipta Karya Informatika (STIKOM CKI), Indonesia)
25. Yudistira Dwi Wardhana Asnar, Ph.D. (Institut Teknologi Bandung, Indonesia)

**ISSN : 2527-5836**

**e-ISSN: 2528-0074**

**JISKa (Jurnal Informatika Sunan Kalijaga)**

Vol. 9, No. 2, SEPTEMBER 2024

# DAFTAR ISI

**JISKA (Jurnal Informatika Sunan Kalijaga)**
**Vol. 9, No. 3, SEPTEMBER, 2024, Pp. 159 – 169**
**ISSN: 2527 – 5836 (print) | 2528 – 0074 (online)**

# Identifikasi Kematangan Buah Pisang Berdasarkan Variasi Jarak Menggunakan Metode K-Nearest Neighbor

**Rizky Putu Ananda [(1)], Febri Liantoni [(2)*], Nurcahya Pradana Taufik Prakisya [(3)]**
Pendidikan Teknik Informatika dan Komputer, Fakultas Keguruan dan Ilmu Pendidikan, Universitas Sebelas Maret, Surakarta, Indonesia
e-mail : {rizky.putu,febri.liantoni,nurcahya.pradana}@gmail.com.
* Penulis korespondensi.
Artikel ini diajukan 2 November 2023, direvisi 14 Juni 2024, diterima 20 Juni 2024, dan dipublikasikan 25 September 2024.

***Abstract***

*This research aims to identify the level of ripeness of kepok bananas based on the color of their skin using the K-Nearest Neighbor (K-NN) method. Bananas are an important commodity in Indonesia, and various ripeness levels need to be identified. The current process of identifying banana ripeness is still done manually, which requires a lot of labor and tends to be subjective. The K-NN method is used to classify bananas based on their skin color. This research involves the collection of banana images with three ripeness levels (raw, ripe, and overripe) and the extraction of RGB color features from these images. Three distance methods, namely Euclidean, Minkowski, and Manhattan, are also employed to compare accuracy results. The evaluation results of this research show that the accuracy value for the Euclidean distance method is 84%, the Minkowski distance method is 82%, and the Manhattan distance method is 80%. Thus, the findings indicate that the K-NN method and the Euclidean distance method provide good results in identifying the ripeness level of bananas. By implementing the K-NN algorithm, this research attempts to address the weaknesses of the time-consuming and subjective manual identification process, with the hope of providing a more accurate and efficient solution for the banana industry. The results of this research can be used to automate the identification process of banana ripeness levels and improve efficiency in banana sorting. It is expected that this research can provide practical benefits to the community and serve as a basis for further research in this field.*

***Keywords: K-NN, Classification, RGB Feature Extraction, Banana, Digital Imaging***

**Abstrak**

Penelitian ini bertujuan untuk mengidentifikasi tingkat kematangan buah pisang kepok berdasarkan warna kulit buah pisang menggunakan metode K-Nearest Neighbor (K-NN). Pisang adalah komoditas penting di Indonesia, dengan berbagai tingkat kematangan yang perlu diidentifikasi. Proses identifikasi kematangan buah pisang saat ini masih dilakukan secara manual, yang memerlukan banyak tenaga kerja dan cenderung subjektif. Metode K-NN digunakan untuk mengklasifikasikan buah pisang berdasarkan warna kulitnya. Penelitian ini melibatkan pengumpulan data citra pisang dengan tiga tingkat kematangan (mentah, matang, dan busuk) dan ekstraksi fitur warna RGB dari citra tersebut. Pada penelititan ini juga menggunakan 3 metode jarak yaitu Euclidean, minkwoski, dan Manhattan yang berfungsi untuk membandingkan hasil *accuracy*. Hasil evaluasi dari penelitian ini yaitu nilai akurasi dari metode jarak Euclidean sebesar 84%, metode jarak Minkowski sebesar 82%, dan Manhattan sebesar 80%. Maka hasilnya menunjukkan bahwa metode K-NN dengan metode jarak Euclidean memberikan hasil yang baik dalam mengidentifikasi tingkat kematangan buah pisang. Dengan menerapkan algoritma K-NN, penelitian ini mencoba mengatasi kelemahan proses identifikasi manual yang memakan waktu dan subjektifitas manusia, dengan harapan memberikan solusi yang lebih akurat dan efisien untuk industri pisang. Hasil penelitian ini dapat digunakan untuk mengotomatisasi proses identifikasi tingkat kematangan buah pisang dan meningkatkan efisiensi dalam pemilahan buah pisang. Diharapkan penelitian ini dapat memberikan manfaat praktis bagi masyarakat dan menjadi dasar untuk penelitian lebih lanjut di bidang ini.

**Kata Kunci: K-NN, Klasifikasi, Ekstraksi Fitur RGB, Pisang, Citra Digital**

## 1. PENDAHULUAN

Buah pisang (*Musa paradisiaca*) merupakan sumber yang kaya akan vitamin, mineral, dan karbohidrat (Limin et al., 2019). Di Indonesia, pisang sering digunakan dalam konsumsi sehari-hari dan dalam hidangan khusus yang populer. Terdapat lebih dari 200 jenis pisang di Indonesia, masing-masing dengan ciri khasnya sendiri (Arifki & Barliana, 2018). Pisang juga memiliki sifat penyembuhan luka yang bermanfaat, terutama pada varietas pisang kepok yang kaya akan senyawa flavonoid dengan potensi antioksidan.

Kendari adalah salah satu kota penghasil buah tropis, termasuk pisang, pepaya, dan nangka. Produksi buah di Kendari telah meningkat signifikan dalam beberapa tahun terakhir, dengan produksi pisang mencapai 7.576 kw pada tahun 2005, diikuti oleh pepaya dan nangka (Limin et al., 2019). Identifikasi jenis dan tingkat kematangan buah masih sering dilakukan secara manual oleh petani, terutama dalam proses pasca panen buah pisang yang diproduksi dalam skala besar atau industri.

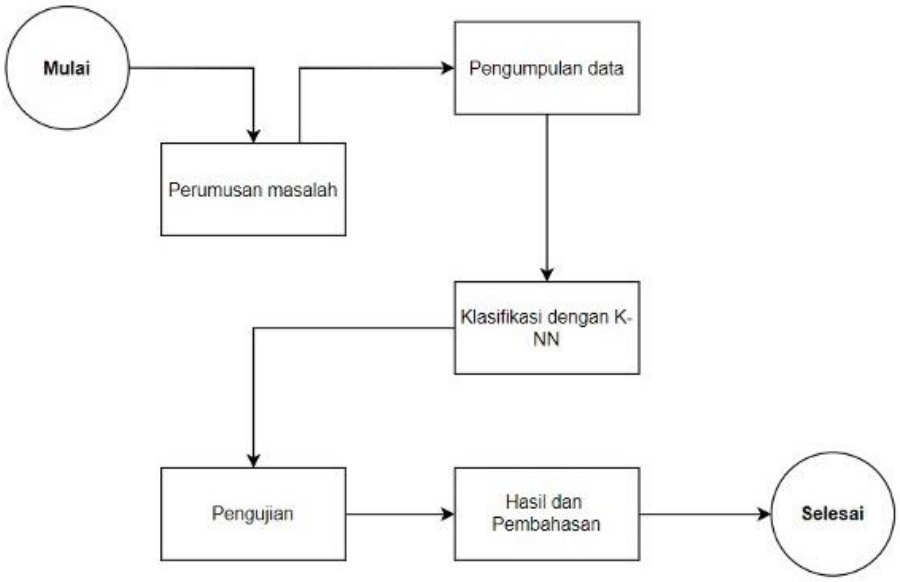
Beberapa penelitian telah mencoba menggunakan metode K-Nearest Neighbor (K-NN) dalam berbagai konteks, seperti prediksi tinggi gelombang, identifikasi gejala penyakit kulit, dan klasifikasi daun (Liantoni & Annisa, 2018). Penelitian terdahulu telah mencoba menggunakan berbagai metode, termasuk algoritma K-Nearest Neighbor (K-NN), dalam berbagai aplikasi pengenalan pola, seperti identifikasi gejala penyakit, klasifikasi daun, dan identifikasi motif kain tradisional. Algoritma K-NN adalah alat yang kuat untuk pengenalan pola dan pengklasifikasian, di mana objek baru diklasifikasikan berdasarkan mayoritas kategori yang dimiliki oleh K-NN terdekat (Wijaya & Ridwan, 2019).

Dalam konteks ini, penelitian ini mencoba mengatasi tantangan dalam identifikasi tingkat kematangan buah pisang dengan memanfaatkan teknologi citra digital (Liantoni & Annisa, 2018). Dengan menggunakan metode K-NN, penelitian ini bertujuan untuk mengembangkan sistem identifikasi yang lebih efisien dan akurat daripada identifikasi manual yang ada. Oleh karena itu, pemahaman dan implementasi metode ini dalam industri pisang menjadi esensial untuk meningkatkan efisiensi dan konsistensi dalam proses penyortiran buah pisang.

Melalui pengolahan citra digital, penelitian ini berusaha menyajikan solusi yang mampu mengatasi kekurangan proses identifikasi manual, memperpendek waktu yang dibutuhkan, dan memberikan hasil yang lebih konsisten dalam menentukan tingkat kematangan buah pisang. Hal ini dapat memberikan manfaat signifikan bagi industri pisang di Indonesia, serta menjadi kontribusi dalam pengembangan teknologi pengenalan pola melalui citra digital. Oleh karena itu diperlukan metode atau pendekatan yang tepat agar dapat dengan mudah dan akurat menentukan tingkat kematangan buah pisang (Bere et al., 2016).

## 2. METODE PENELITIAN

Dalam tahap awal penelitian, perlu dirancang alur penelitian yang terstruktur dan efisien untuk mencapai hasil optimal (Sidiq et al., 2019). Tahapan awal penelitian dilakukan dengan melakukan perumusan masalah, dilanjutkan dengan pengumpulan data, data yang didapatkan dilakukan klasifikasi menggunakan algoritma K-NN, selanjutnya dilakukan pengujian, dan terakhir dilakukan analisis terhadap hasil pemodelan. Langkah-langkah penelitian dapat dilihat pada Gambar 1.




**Gambar 1 Metodologi Penelitian**

### 2.1 Pengumpulan Data

Tahap ini melakukan pengambilan data yang akan diolah pada penelitian yang akan dilaksanakan, yaitu dengan mengambil citra pisang mentah, matang, dan busuk. Citra yang akan diproses dengan pengolahan citra melibatkan citra awal yang diambil melalui kamera sebagai data masukan (Angreni et al., 2019). Citra pisang yang diambil berupa jenis pisang kepok dengan warna latar belakang yang sama seperti yang ditunjukkan Gambar 2.

**Gambar 2 Pisang Kepok Mentah, Matang, dan Busuk**

### 2.2 Klasifikasi dengan K-Nearest Neighbor

K-Nearest Neighbor (K-NN) adalah metode klasifikasi yang memanfaatkan jarak antara data yang akan diklasifikasikan dan data dalam *dataset* pelatihan. Data pelatihan direpresentasikan dalam ruang berdimensi tinggi dengan fitur-fitur yang mewakili data. K-NN mengidentifikasi kategori objek yang diuji berdasarkan mayoritas dari $k$ tetangga terdekatnya dalam ruang dimensi ini (Liantoni, 2016). Fitur warna direkam dan disimpan dalam *database*, lalu diolah menggunakan metode K-NN untuk mengklasifikasikan citra. Hasilnya adalah kelas citra, yang menentukan label berdasarkan karakteristik warna yang dianalisis menggunakan K-NN (Lestari et al., 2019). Proses klasifikasi kematangan pisang menggunakan algoritma K-NN dibagi menjadi tiga tahapan berikut.

### 2.2.1 Tahap Pemrosesan Citra

Pada pemrosesan citra ini menggunakan citra dari data yang telah dikumpulkan dan diekstrasi ke RGB. Sebelum dilakukan ekstrasi ke RGB citra dipotong pada bagian tengah citra dengan rasio 1:1. Kemudian dari citra yang telah dipotong dilakukan ekstrasi nilai RGB-nya dengan menghitung rata-rata nilai RGB tiap *pixel*. Citra berwarna RGB (Red, Green, Blue) terdiri dari tiga komponen khusus: merah, hijau, dan biru (Siswanto et al., 2020). Ketika komponen warna ini digabungkan, mereka membentuk warna akhir (Raysyah et al., 2021). Setiap komponen warna direpresentasikan dalam rentang nilai intensitas 0 hingga 255 dalam format bilangan bulat, di mana 0 menunjukkan ketiadaan warna dan 255 merupakan intensitas maksimum (Stamou et al., 2005).

### 2.2.2 Tahap Pelatihan Data

Tahap pelatihan data dimulai dengan menggunakan data citra yang telah diberi label sesuai klasifikasinya. Selanjutnya, proses ekstraksi nilai RGB dilakukan untuk setiap citra guna memperoleh informasi warna yang menjadi ciri khas dari citra tersebut. Hasil ekstraksi ini kemudian disimpan ke dalam *database* sebagai data latih yang akan digunakan untuk melatih model dalam proses selanjutnya. Data latih ini berperan penting dalam membantu model mengenali pola atau fitur dari citra yang dilatih sehingga mampu melakukan prediksi atau klasifikasi dengan akurat pada tahap pengujian.

### 2.2.3 Tahap Klasifikasi

Pada tahap klasifikasi ini menggunakan algoritma K-Nearest Neighbor dengan data latih yang sudah ada sebelumnya. Pada klasifikasi ini menggunakan nilai $k$ 10,11,12,13, dan 14 untuk menghindari dilema jarak sama yang sama atau *tie breaking* yang dikarenakan data yang diteliti memiliki 3 kelas. K tersebut diambil dari akar jumlah data *training* (Fasnuari et al., 2022). Klasifikasi ini juga menggunakan 3 metode jarak yaitu Euclidean, Manhattan, dan Minkowski (Wahyono et al., 2020). Setelah dilakukan klasifikasi dengan metode-metode tersebut, kemudian didapatkan data yang berupa label mentah, matang, dan busuk. Adapun beberapa rumus dari ketiga metode jarak yaitu:

1) Metode jarak Euclidean

Metode jarak ini adalah metode perhitungan jarak antara dua titik dalam ruang Euclidean (2D, 3D, atau dimensi lebih tinggi), yang digunakan untuk mengukur tingkat kemiripan antara data. Jarak tersebut dihitung menggunakan rumus Euclidean sebagaimana ditunjukkan pada Pers. (1). Dalam rumus ini, $d$ merepresentasikan jarak antara dua titik $x$ dan $y$, di mana $x$ adalah pusat klaster, dan $y$ adalah data atribut yang akan diukur. Indeks $i$ menunjukkan setiap data individu, sedangkan $n$ menunjukkan jumlah keseluruhan data. Nilai $x_i$ merujuk pada koordinat data di pusat klaster ke-$i$, sementara $y_i$ adalah koordinat data pada setiap data ke-$i$. Perhitungan jarak Euclidean ini membantu mengidentifikasi seberapa dekat data tersebut dengan pusat klaster, yang menjadi dasar penting dalam proses klasterisasi atau analisis lainnya.

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

2) Metode jarak Manhattan

Metode jarak Manhattan digunakan untuk menghitung perbedaan absolut (mutlak) antara koordinat dua objek. Perhitungan dilakukan dengan menjumlahkan perbedaan mutlak dari setiap dimensi koordinat antara dua titik, yang dirumuskan pada Pers. (2). Dalam rumus ini, $d$ merepresentasikan jarak antara dua titik $x$ dan $y$, di mana $x$ adalah pusat klaster, dan $y$ adalah data atribut yang akan diukur. Indeks $i$ menunjukkan setiap data individu, sedangkan $n$

menunjukkan jumlah keseluruhan data. Nilai $x_i$ merujuk pada koordinat data di pusat klaster ke-$i$, sementara $y_i$ adalah koordinat data pada setiap data ke-$i$. Metode ini berguna dalam analisis klasterisasi dan pengukuran jarak ketika pergerakan di sepanjang sumbu atau dimensi diutamakan, seperti dalam jaringan jalan kota yang terstruktur sebagai *grid*.

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

3) Metode jarak Minkowski

Metode jarak Minkowski adalah sebuah metrik dalam ruang vektor yang menggeneralisasi metode jarak Euclidean dan metode jarak Manhattan dengan menggunakan norma yang dapat diatur. Metode ini bergantung pada parameter $p$, yang mengontrol tingkat generalisasi dari kedua metode tersebut. Dalam pengukuran menggunakan metode ini, nilai $p$ umumnya diambil 1 (untuk jarak Manhattan) atau 2 (untuk jarak Euclidean), seperti yang dinyatakan oleh Nishom (2019). Rumus untuk menghitung jarak Minkowski ditunjukkan dalam Pers. (3), di mana $d$ adalah jarak antara dua titik $x$ dan $y$, dengan $x$ sebagai pusat klaster dan $y$ sebagai data atribut. Indeks $i$ menunjukkan setiap data individu, sedangkan $n$ menunjukkan jumlah keseluruhan data. Nilai $x_i$ merujuk pada koordinat data di pusat klaster ke-$i$, sementara $y_i$ adalah koordinat data pada setiap data ke-$i$. Variabel $p$ adalah parameter eksponen (*power*) yang menentukan jenis norma yang digunakan dalam perhitungan jarak. Metode ini sangat fleksibel karena dapat mencakup berbagai jenis metrik tergantung pada nilai $p$ yang dipilih.

$$d(x,y) = \left( \sum_{i=1}^{n} |x_i - y_i| p \right)^{\frac{1}{p}} \quad (3)$$

### 2.3 Analisis

Dalam analisis ini, ekstraksi fitur RGB digunakan untuk menggambarkan penerapan metode K-Nearest Neighbor dalam mengidentifikasi tingkat kematangan buah pisang. Evaluasi dilakukan melalui *confusion matrix*. True Positive (TP) mencerminkan data yang benar-benar positif dan berhasil diklasifikasikan sebagai positif oleh model, sementara False Positive (FP) mengacu pada data yang seharusnya negatif tetapi salah diklasifikasikan sebagai positif. False Negative (FN) menggambarkan data yang seharusnya positif tetapi salah diklasifikasikan sebagai negatif, dan True Negative (TN) adalah jumlah data yang benar-benar negatif dan berhasil diklasifikasikan sebagai negatif oleh metode yang digunakan Xu et al. (2020).

Melalui *confusion matrix*, akurasi dapat dihitung menggunakan rumus pada Pers. (4) (Rahayu et al., 2021). Pada analisis ini digunakan *confusion matrix* karena hanya mencari angka. Apabila ingin menampilkan informasi kinerja algoritma klasifikasi dalam bentuk grafik dapat digunakan Receiver Operating Characteristic (ROC). Kurva ROC dibuat berdasarkan nilai yang telah didapatkan dari perhitungan dengan *confusion matrix*, yaitu antara False Positive (FP) *rate* dan True Positive (TP) (Kristiawan & Widjaja, 2021).

$$Accuracy = \frac{TP}{Total\ Data} \times 100\% \quad (4)$$

### 2.4 Pengambilan Kesimpulan

Pada tahap ini, hasil analisis menggunakan ekstraksi fitur RGB dari citra buah pisang untuk menggambarkan implementasi metode K-Nearest Neighbor dalam identifikasi kematangan pisang akan digunakan untuk menarik kesimpulan. Kesimpulan akan didasarkan pada hasil

pengujian yang menggunakan *confusion matrix*. Selain itu, disarankan untuk mengevaluasi lebih lanjut algoritma klasterisasi dokumen sebagai rekomendasi penelitian lanjutan.

## 3. HASIL DAN PEMBAHASAN

Penelitian ini melibatkan tiga tingkat kematangan pisang, yaitu mentah, matang, dan busuk, menggunakan latar belakang yang sama. Data citra terdiri dari 195 citra, dengan masing-masing kelas memiliki 50 citra untuk data latih dan 15 untuk data uji. Citra-citra ini diambil secara langsung untuk merepresentasikan berbagai tingkat kematangan.

Dalam proses pengolahan citra ini menggunakan ekstrasi fitur RGB, kemudian citra-citra dari data yang telah dikumpulkan diekstraksi ke format RGB menggunakan *library* Python OpenCV. Sebelumnya, citra-citra tersebut dipotong pada bagian tengah dengan rasio 1:1. Selanjutnya, ekstraksi nilai RGB dilakukan dengan menghitung rata-rata nilai RGB untuk setiap piksel pada citra yang telah dipotong dengan kode pemrograman yang terdapat pada Gambar 3. Kemudian dari nilai RGB yang diekstraksi, dihasilkan fitur RGB dari data latih dan data uji yang dapat dilihat pada Tabel 1 sampai 3.

```python
def proses_gambar(img_path):
    # Baca gambar
    image = cv2.imread(img_path)

    # Dapatkan dimensi gambar (lebar dan tinggi)
    height, width, _ = image.shape

    # Tentukan ukuran gambar yang dipotong
    new_width = 200  # Misalnya, 200 piksel
    new_height = 200  # Misalnya, 200 piksel

    # Hitung koordinat x dan y untuk memotong gambar di tengah
    x = (width - new_width) // 2
    y = (height - new_height) // 2

    # Potong gambar
    cropped_image = image[y:y+new_height, x:x+new_width]

    # Ekstraksi komponen warna RGB
    blue_channel = cropped_image[:, :, 0]
    green_channel = cropped_image[:, :, 1]
    red_channel = cropped_image[:, :, 2]

    red = np.mean(red_channel)
    green = np.mean(green_channel)
    blue = np.mean(blue_channel)

    # print(f'red={red},  green={green},  blue={blue}')

    # Menghitung rata-rata komponen warna
    average_blue = blue / 255
    average_green = green / 255
    average_red = red / 255

    return [average_red, average_green, average_blue]
```

**Gambar 3 Source Code Ekstraksi Fitur RGB**

Selanjutnya pada tahap klasifikasi, digunakan algoritma K-Nearest Neighbor dengan berbagai nilai $k$, yaitu 10, 11, 12, 13, dan 14. Pada pemilihan nilai $k$ ini diambil dari akar jumlah data uji untuk mendapatkan nilai $k$ yang optimal. Penentuan nilai $k$ ini juga mempengaruhi terhadap *overfitting* dan *underfitting* hasil klasifikasi. Nilai $k$ yang kecil bisa menyebabkan *overfitting* dan nilai $k$ yang besar bisa menyebabkan *underfitting*. Pengujian dilakukan dengan memproses data uji, melakukan ekstraksi fitur RGB, dan kemudian melakukan klasifikasi menggunakan K-Nearest Neighbor. Terdapat tiga metode pengukuran jarak yang digunakan, yaitu Euclidean, Minkowski, dan Manhattan. Hasil klasifikasi ini menjadi dasar untuk pengujian menggunakan *confusion matrix*. Evaluasi dilakukan dengan memeriksa nilai akurasi pada setiap metode jarak dan nilai $k$ yang digunakan. Perhitungan akurasi digunakan untuk menilai tingkat kebenaran prediksi dan dihitung menggunakan persamaan yang relevan. Hasil akurasi dari klasifikasi menggunakan $k$ dari 10 hingga 14 dan dengan tiga metode jarak yang berbeda ditunjukkan pada Tabel 4.

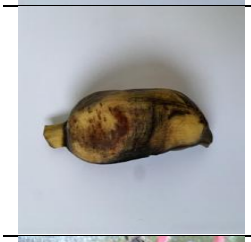
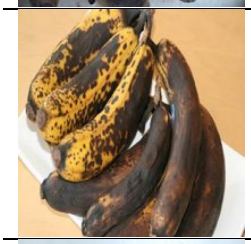
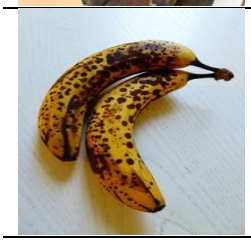
**Tabel 1 Ekstraksi RGB Citra**

| Gambar Asli | R | G | B |
|---|---|---|---|
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Tabel 2 Fitur RGB Data Latih**

| Citra | R | G | B | Kelas Aktual |
|---|---|---|---|---|
| 1 | 0,16 | 0,12 | 0,07 | Busuk |
| 2 | 0,20 | 0,17 | 0,11 | Busuk |
| 3 | 0,19 | 0,16 | 0,10 | Busuk |
| 4 | 0,15 | 0,11 | 0,05 | Busuk |
| 5 | 0,39 | 0,29 | 0,12 | Busuk |
| … | … | … | … | … |
| 150 | 0,28 | 0,33 | 0,15 | Mentah |

**Tabel 3 Fitur RGB Data Uji**

| CITRA | R | G | B |
|---|---|---|---|
|  | 0,16 | 0,13 | 0,10 |
|  | 0,34 | 0,21 | 0,11 |
|  | 0,47 | 0,42 | 0,37 |
|  | 0,42 | 0,33 | 0,22 |
|  | 0,27 | 0,22 | 0,18 |
| … | … | … | … |
|  | 0,32 | 0,36 | 0,19 |

**Tabel 4 Hasil Nilai *Accuracy* dengan 3 Metode Jarak**

| Metode | *k*=10 | *k*=11 | *k*=12 | *k*=13 | *k*=14 |
|---|---|---|---|---|---|
| Euclidean | 0,84 | 0,84 | 00,84 | 0,84 | 0,84 |
| Minkowski | 0,82 | 0,82 | 0,82 | 0,82 | 0,82 |
| Manhattan | 0,80 | 0,80 | 0,80 | 0,80 | 0,80 |

JISKA (Jurnal Informatika Sunan Kalijaga)
Vol. 9, No. 3, SEPTEMBER, 2024: 159 – 169

Perhitungan pada Tabel 4 dihitung menggunakan rumus pada Pers. (4). Pada perhitungan ini diambil dari seluruh jumlah data uji dari setiap kelas yaitu 15 data uji dan terdapat 3 kelas. Pada data uji memiliki seluruh jumlah nilai benar Euclidean 38, Minkowski 37, dan Manhattan 36. Jumlah seluruh data uji terdapat 45 data. Berikut contoh perhitungan nilai akurasi.

1) Euclidean $k$=10

$$accuracy = \frac{38}{45} \times 100\% = 84\%$$

2) Minkowski $k$=10

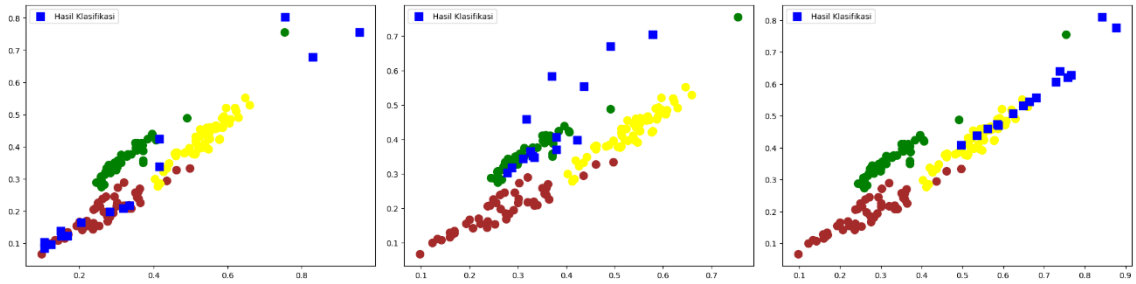
$$accuracy = \frac{37}{45} \times 100\% = 82\%$$

3) Manhattan $k$=10

$$accuracy = \frac{36}{45} \times 100\% = 80\%$$

Rata-rata nilai akurasi yang dihasilkan dari K-NN dengan metode jarak Euclidean, Minkowski, dan Manhattan masing-masing 84%, 82%, dan 80%. Dari nilai akurasi yang dihasilkan, berfungsi untuk mengukur seberapa baik metode K-Nearest Neighbor digunakan untuk mengklasifikasikan data pada identifikasi kematangan buah pisang. Dari nilai akurasi yang dihasilkan didapat K-NN dengan metode jarak Euclidean memiliki hasil akurasi yang lebih baik daripada Minkowski dan Manhattan dengan persentase 84%.

Visualisasi data persebaran digambarkan menggunakan *scatter diagram. Scatter diagram*, atau juga disebut sebagai diagram pencar, digunakan untuk menggambarkan hubungan atau korelasi antara dua faktor atau sebab dan akibat. Ketika kedua variabel tersebut saling berkaitan, titik-titik koordinat akan tersebar sepanjang garis atau kurva. Semakin kuat hubungan atau korelasi, semakin erat titik-titik tersebut berkumpul mendekati garis (Guntara, 2023). Visualisasi data persebaran pengklasifikasian menggunakan metode Euclidean dapat dilihat pada Gambar 4.




**Gambar 4 *Scatter Plot* Persebaran Klasifikasi Busuk, Mentah, dan Matang**

## 4. KESIMPULAN

Hasil penelitian ini menunjukkan bahwa penggunaan algoritma K-Nearest Neighbor (K-NN) dalam mengidentifikasi kematangan buah pisang, dengan tiga tingkat kematangan yang berbeda, telah memberikan hasil yang memuaskan. Tiga metode jarak, yaitu Euclidean, Minkowski, dan Manhattan, digunakan untuk membandingkan tingkat akurasi. Hasil evaluasi menunjukkan bahwa metode Euclidean memiliki akurasi tertinggi, mencapai 84%, diikuti oleh Minkowski dengan 82%, dan Manhattan dengan 80%. Oleh karena itu, disimpulkan bahwa metode Euclidean dengan $k$=10 hingga 14 adalah yang paling akurat untuk penelitian ini. Meskipun kinerja metode K-NN dalam mengidentifikasi kematangan buah pisang terbukti baik, hasilnya sangat tergantung pada parameter yang dipilih dan tujuan identifikasi. Oleh karena itu, diperlukan

eksperimen lebih lanjut untuk memperoleh hasil yang optimal. Berdasarkan penelitian ini, beberapa saran untuk penelitian selanjutnya meliputi pengembangan dengan mempertimbangkan penggunaan metode lain atau metode terbaru, pemilihan jenis pisang yang beragam selain pisang kepok, penambahan fitur lain dalam analisis, serta peningkatan jumlah data latih dan data uji guna meningkatkan akurasi.

## DAFTAR PUSTAKA

Angreni, I. A., Adisasmita, S. A., Ramli, M. I., & Hamid, S. (2019). Pengaruh Nilai K Pada Metode K-Nearest Neighbor (KNN) Terhadap Tingkat Akurasi Identifikasi Kerusakan Jalan. *Rekayasa Sipil*, *7*(2), 63. https://doi.org/10.22441/jrs.2018.v07.i2.01

Arifki, H. H., & Barliana, M. I. (2018). Karakteristik dan Manfaat Tumbuhan Pisang di Indonesia : Review Artikel. *Farmaka*, *16*(3), 196–203. https://doi.org/10.24198/JF.V16I3.17605

Bere, G. A., Tamtjita, E. N., & Kusumaningrum, A. (2016). Klasifikasi Untuk Menentukan Tingkat Kematangan Buah Pisang Sunpride. *Conference SENATIK STT Adisutjipto Yogyakarta*, *2*, 109–113. https://doi.org/10.28989/senatik.v2i0.61

Fasnuari, H. A. D., Yuana, H., & Chulkamdi, M. T. (2022). Penerapan Algoritma K-Nearest Neighbor untuk Klasifikasi Penyakit Diabetes Melitus. *Antivirus : Jurnal Ilmiah Teknik Informatika*, *16*(2), 133–142. https://doi.org/10.35457/antivirus.v16i2.2445

Guntara, R. G. (2023). Visualisasi Data Laporan Penjualan Toko Online Melalui Pendekatan Data Science Menggunakan Google Colab. *ULIL ALBAB : Jurnal Ilmiah Multidisiplin*, *2*(6), 2091–2100. https://doi.org/10.56799/JIM.V2I6.1578

Kristiawan, K., & Widjaja, A. (2021). Perbandingan Algoritma Machine Learning dalam Menilai Sebuah Lokasi Toko Ritel. *Jurnal Teknik Informatika Dan Sistem Informasi*, *7*(1), 35–46. https://doi.org/10.28932/jutisi.v7i1.3182

Lestari, Z. D., Nafi'iyah, N., & Susilo, P. H. (2019). Sistem Klasifikasi Jenis Pisang Berdasarkan Ciri Warna HSV Menggunakan Metode K-NN. *Prosiding Seminar Nasional Teknologi Informasi Dan Komunikasi (SENATIK)*, *2*(1), 11–15. https://prosiding.unipma.ac.id/index.php/SENATIK/article/view/880

Liantoni, F. (2016). Klasifikasi Daun Dengan Perbaikan Fitur Citra Menggunakan Metode K-Nearest Neighbor. *Jurnal ULTIMATICS*, *7*(2), 98–104. https://doi.org/10.31937/ti.v7i2.356

Liantoni, F., & Annisa, F. N. (2018). Fuzzy K-Nearest Neighbor pada Klasifikasi Kematangan Cabai Berdasarkan Fitur HSV Citra. *JIPI (Jurnal Ilmiah Penelitian Dan Pembelajaran Informatika)*, *3*(2), 101–108. https://doi.org/10.29100/jipi.v3i2.851

Limin, N. S., Sari, J. Y., & Purnama, I. P. N. (2019). Identifikasi Tingkat Kematangan Buah Pisang Menggunakan Metode Ektraksi Ciri Statistik Pada Warna Kulit Buah. *ULTIMATICS*, *10*(2), 98–102. https://doi.org/10.31937/ti.v10i2.1004

Nishom, M. (2019). Perbandingan Akurasi Euclidean Distance, Minkowski Distance, dan Manhattan Distance pada Algoritma K-Means Clustering berbasis Chi-Square. *Jurnal Informatika: Jurnal Pengembangan IT*, *4*(1), 20–24. https://doi.org/10.30591/jpit.v4i1.1253

Rahayu, W. I., Prianto, C., & Novia, E. A. (2021). Perbandingan Algoritma K-Means dan Naïve Bayes untuk Memprediksi Prioritas Pembayaran Tagihan Rumah Sakit Berdasarkan Tingkat Kepentingan pada PT. Pertamina (Persero). *Jurnal Teknik Informatika*, *13*(2), 1–8. https://ejurnal.ulbi.ac.id/index.php/informatika/article/view/1383

Raysyah, S. R., Arinal, V., & Mulyana, D. I. (2021). Klasifikasi Tingkat Kematangan Buah Kopi Berdasarkan Deteksi Warna Menggunakan Metode KNN dan PCA. *JSiI (Jurnal Sistem Informasi)*, *8*(2), 88–95. https://doi.org/10.30656/jsii.v8i2.3638

Sidiq, U., Choiri, Moh. M., & Mujahidin, A. (2019). *Metode Penelitian Kualitatif di Bidang Pendidikan* (A. Mujahidin, Ed.). CV. Nata Karya. https://opac.perpusnas.go.id/DetailOpac.aspx?id=1257824

Siswanto, I., Utami, E., & Raharjo, S. (2020). Klasifikasi Tingkat Kematangan Buah Berdasarkan Warna dan Tekstur Menggunakan Metode K-Nearest Neighbor dan Nearest Mena Classifier. *Inspiration: Jurnal Teknologi Informasi Dan Komunikasi*, *10*(1), 93–101. https://doi.org/10.35585/inspir.v10i1.2559

Stamou, G., Krinidis, M., Loutas, E., Nikolaidis, N., & Pitas, I. (2005). 2D and 3D Motion Tracking in Digital Video. In *Handbook of Image and Video Processing* (pp. 491–XVIII). Elsevier. https://doi.org/10.1016/B978-012119792-6/50093-0

Wahyono, W., Trisna, I. N. P., Sariwening, S. L., Fajar, M., & Wijayanto, D. (2020). Comparison of distance measurement on k-nearest neighbour in textual data classification. *Jurnal Teknologi Dan Sistem Komputer*, *8*(1), 54–58. https://doi.org/10.14710/jtsiskom.8.1.2020.54-58

Wijaya, N., & Ridwan, A. (2019). Klasifikasi Jenis Buah Apel dengan Metode K-Nearest Neighbors dengan Ekstraksi Fitur HSV dan LBP. *Jurnal Sisfokom (Sistem Informasi Dan Komputer)*, *8*(1), 74–78. https://doi.org/10.32736/sisfokom.v8i1.610

Xu, J., Zhang, Y., & Miao, D. (2020). Three-way confusion matrix for classification: A measure driven view. *Information Sciences*, *507*, 772–794. https://doi.org/10.1016/j.ins.2019.06.064

**JISKA (Jurnal Informatika Sunan Kalijaga)**
**Vol. 9, No. 3, SEPTEMBER, 2024, Pp. 170 – 177**
**ISSN: 2527 – 5836 (print) | 2528 – 0074 (online)**

# Segmentasi Pelanggan *E-Commerce* Menggunakan Fitur *Recency, Frequency, Monetary* (RFM) dan Algoritma Klasterisasi K-Means

**Reyhan Muhammad Fauzan [(1)], Ganjar Alfian [(2)*]**
Teknik Elektro dan Informatika, Sekolah Vokasi, Universitas Gadjah Mada, Yogyakarta, Indonesia
e-mail : reyhanmuhammad@mail.ugm.ac.id, ganjar.alfian@ugm.ac.id.
* Penulis korespondensi.
Artikel ini diajukan 25 Januari 2024, direvisi 1 Mei 2024, diterima 2 Mei 2024, dan dipublikasikan 25 September 2024.

***Abstract***

*The rapid growth in the e-commerce industry demands the development of smarter and more focused marketing strategies. One approach that can be applied is customer segmentation using various features such as Recency, Frequency, and Monetary (RFM), along with machine learning-based clustering methods. The objective of this study is to design and develop a web-based e-commerce customer segmentation application using a combination of RFM features and clustering methods. The study proposes the K-Means algorithm and compares it with K-Medoids and Fuzzy C Means using publicly available e-commerce datasets. Experimental results showed that the K-Means algorithm outperformed K-Medoids and Fuzzy C Means (FCM) based on the Silhouette Score of 0.67305, Davies Bouldin Index of 0.51435, and Calinski Harabasz Index of 5647.89. Through analysis and testing, the designed application has proven effective in grouping customers into relevant segments. These segments are divided into three categories: Loyal, Need Attention, and Promising, visualized in a web-based application dashboard using Streamlit. The developed application allows e-commerce business owners and users from the business, management, and marketing divisions to categorize customers based on transaction data. The results of this study are expected to provide valuable insights to e-commerce management and marketing professionals who are facing increasingly fierce competition.*

***Keywords: E-Commerce, Customer Segmentation, RFM, K-Means, Web Application***

**Abstrak**

Peningkatan pesat dalam industri *e-commerce* menuntut pengembangan strategi pemasaran yang lebih cerdas dan terfokus. Salah satu pendekatan yang dapat diterapkan adalah segmentasi pelanggan menggunakan berbagai fitur, seperti *Recency, Frequency,* dan *Monetary* (RFM), serta metode klasterisasi berbasis *machine learning*. Tujuan dari penelitian ini adalah untuk merancang dan membangun aplikasi segmentasi pelanggan *e-commerce* berbasis web yang menggunakan kombinasi fitur RFM dan metode klasterisasi. Penelitian ini mengusulkan algoritma K-Means dan membandingkannya dengan K-Medoids, serta Fuzzy C Means pada *dataset e-commerce* yang tersedia secara publik. Hasil penelitian menunjukkan bahwa algoritma K-Means lebih unggul dibanding algoritma K-Medoids dan Fuzzy C Means (FCM) berdasarkan nilai Silhouette Coefficient sebesar 0,67305, Davies Bouldin Index sebesar 0,51435, dan Calinski Harabasz Index sebesar 5647,89. Melalui analisis dan pengujian, aplikasi yang dirancang telah terbukti efektif dalam mengelompokkan pelanggan ke dalam segmen yang relevan. Segmen tersebut dibagi menjadi tiga kategori yaitu *Loyal, Need Attention,* dan *Promising,* kemudian divisualisasikan dalam bentuk dashboard pada aplikasi berbasis web menggunakan *Streamlit.* Aplikasi yang dikembangkan dalam penelitian ini memungkinkan pemilik bisnis *e-commerce* ataupun pengguna dari bidang bisnis, divisi manajemen, dan pemasaran untuk mengelompokkan pelanggan berdasarkan data transaksi. Hasil dari penelitian ini diharapkan dapat memberikan informasi berharga kepada manajemen *e-commerce* maupun bidang pemasaran dalam menghadapi persaingan yang semakin ketat.

**Kata Kunci: *E-Commerce*, Segmentasi Pelanggan, RFM, K-Means, Aplikasi Web**

JISKA (Jurnal Informatika Sunan Kalijaga)
Vol. 9, No. 3, SEPTEMBER, 2024: 170 – 177

## 1. PENDAHULUAN

Di era digital yang dipenuhi oleh industri *e-commerce* yang pesat, persaingan antarperusahaan untuk menarik, mempertahankan, dan memahami pelanggan semakin ketat. Aktivitas perdagangan melalui penerapan *e-commerce* sangat praktis hanya menggunakan perangkat elektronik seperti laptop, komputer, atau *smartphone* dan menggunakan internet sebagai perantara (Molla & Licker, 2005). Namun, dengan pertumbuhan pesat ini, juga muncul tantangan besar dalam mengelola dan memahami pelanggan *e-commerce*.

Mengingat kompleksitas bisnis dan pasar *e-commerce* yang terus berkembang, keberhasilan bisnis dalam sektor ini sangat bergantung pada kemampuan mereka untuk memahami dan merespon perubahan perilaku pelanggan dengan cepat dan efektif. Memahami kebutuhan dan perilaku pelanggan merupakan hal yang krusial dalam merancang strategi pemasaran yang efektif dan meningkatkan kualitas layanan. Salah satu tantangan utama dalam *e-commerce* adalah bagaimana mengelompokkan pelanggan ke dalam segmen yang relevan. Segmentasi pelanggan diperlukan untuk mengelompokkan pelanggan yang memiliki kesamaan karakteristik (Kim et al., 2006). Selanjutnya, hasil kelompok pelanggan akan mendapatkan perlakuan yang berbeda-beda dalam strategi pemasaran (Li et al., 2010; Shirole et al., 2021).

Analisis RFM adalah salah satu cara untuk melakukan segmentasi pelanggan berdasarkan *recency, frequency,* dan *monetary*. Faktor *recency* menilai seberapa baru pelanggan melakukan transaksi. *Frequency* menilai seberapa sering pelanggan bertransaksi, sedangkan *monetary* menilai seberapa besar total pengeluaran yang dilakukan oleh pelanggan saat bertransaksi (Anitha & Patil, 2022). Selanjutnya nilai RFM juga dapat digunakan sebagai fitur untuk segmentasi pelanggan menggunakan metode klasterisasi yang berbasis *unsupervised learning*.

Penelitian sebelumnya menganalisis pola penggunaan Mass Rapid Transit (MRT) dan layanan berbagi sepeda atau *bike sharing* (YouBike) di Taipei, Taiwan, dengan menggunakan data historis. Dengan menggunakan algoritma RFM dan K-Means *clustering*, penelitian ini mengidentifikasi tiga kelompok pengguna MRT-YouBike yang berbeda: *potential, vulnerable*, dan *loyal* (Chen et al., 2022). Penelitian selanjutnya berfokus pada data *e-retailer* sebagai studi kasusnya. Pada penelitian ini, RFM (*recency, frequency, monetary*) digunakan sebagai solusi pembuatan klaster. Penelitian ini menganalisis informasi pembelian pelanggan selama delapan bulan. Kemudian klaster dievaluasi menggunakan *metrics* Silhouette Coefficient untuk algoritma K-Means dengan jumlah klaster yang berbeda. Hasil yang didapatkan menunjukkan bahwa klaster dengan jumlah 3 lebih baik daripada klaster yang berjumlah 5 (Anitha & Patil, 2022). Implementasi model K-Means untuk segmentasi pelanggan menjadi 3 klaster yaitu *loyal, promising,* dan *need attention* sudah dilakukan oleh penelitian sebelumnya (Hilmy et al., 2023). Hasilnya menunjukkan bahwa model K-Means lebih baik dibanding dengan model K-Medoids dalam segmentasi data pelanggan.

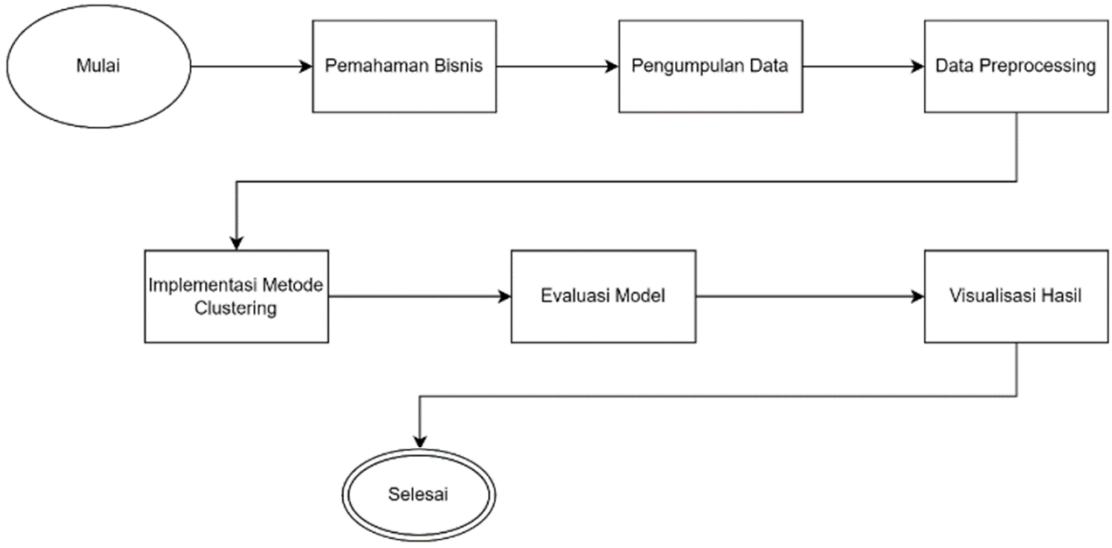
Dengan menggunakan metode klasterisasi K-Means, dapat menentukan kategori dan strategi yang baik untuk pelanggan. Hasil penelitian sebelumnya menunjukkan bahwa metode klasterisasi K-Means dengan skor atribut RFM yang berbeda berhasil mengelompokkan 14.979 data pelanggan kedalam 5 klaster (Sutresno et al., 2018). Pada penelitian selanjutnya menganalisis model RFM dan klasterisasi K-Means pada *online bookstore*. Masalah dalam penelitian ini adalah menurunnya jumlah transaksi dalam rentang bulan Januari 2019 hingga bulan November 2020. Sehingga dilakukan analisis untuk menciptakan klaster yang optimal menggunakan *elbow method* dan dilakukan pengujian performa menggunakan *metrics* Silhouette Coefficient dan Calinski Harabasz Index. Hasil jumlah klaster yang optimal untuk strategi pemasaran dari 23.152 pelanggan adalah berjumlah 3 klaster (Juhari & Juarna, 2022).

Selain berfokus terhadap analisis, penelitian selanjutnya memvisualisasikan hasil dari analisis yang didapat menjadi *dashboard* berbasis web. Pada penelitian kali ini data pelanggan diperoleh dari *platform* Kaggle. Penelitian ini menggunakan model RFM dan algoritma K-Means untuk menganalisis segmentasi pelanggan. Dalam penelitian ini juga digunakan metode Silhouette

untuk mengoptimalkan klaster yang berjumlah 4 klaster. Hasil yang diperoleh kemudian divisualisasikan menjadi bentuk *dashboard* menggunakan *platform* Streamlit (Alzami et al., 2023). Penelitian sebelumnya telah mengeksplorasi penggunaan metode RFM dan klasterisasi K-Means dalam segmentasi pelanggan, tetapi masih terdapat kekurangan dalam membandingkan metode ini dengan algoritma klasterisasi lain seperti K-Medoids dan Fuzzy C Means, serta dalam mengevaluasi efektivitas model klasterisasi tersebut. Dalam penelitian ini, akan dikembangkan sebuah aplikasi web yang mengintegrasikan RFM dan K-Means untuk segmentasi pelanggan. Efisiensi algoritma K-Means akan diukur dan dibandingkan dengan K-medoids dan Fuzzy C Means menggunakan tiga metrik validasi klaster: Skor Silhouette, Indeks Calinski-Harabasz, dan Indeks Davies-Bouldin. Tujuan akhir dari penelitian ini adalah untuk menyediakan sebuah alat yang efektif bagi perusahaan, khususnya di sektor bisnis dan *e-commerce*, untuk meningkatkan strategi pemasaran, mempertahankan pelanggan, dan membina hubungan jangka panjang yang lebih kuat dengan pelanggan.

## 2. METODE PENELITIAN

Langkah-langkah dalam proses analisis mencakup beberapa tahapan yang dimulai dengan memahami secara menyeluruh kebutuhan bisnis. Proses tersebut mencakup pengumpulan data yang relevan, pemrosesan awal untuk memastikan kualitasnya, pembuatan model yang sesuai, evaluasi kinerjanya, dan visualisasi hasil (Anitha & Patil, 2022). Semua langkah ini dapat dilihat detail dalam Gambar 1.




**Gambar 1 Alur Penelitian**

Alur dimulai dari memahami secara menyeluruh tentang bagaimana suatu bisnis atau organisasi beroperasi, tujuan bisnisnya, serta konteks eksternal yang mempengaruhi keberhasilannya. Pemahaman bisnis juga sangat penting dalam pengembangan perangkat lunak atau solusi teknologi, karena membantu para pengembang membangun solusi yang sesuai dengan kebutuhan bisnis dan memberikan nilai tambah. Selanjutnya adalah pengumpulan data yang dibutuhkan untuk penelitian. *Dataset* yang digunakan pada penelitian ini diperoleh melalui *platform online data.world* dengan nama Global Superstore. *Dataset* ini merupakan kumpulan data yang berasal dari komunitas Tableu dengan total 51.290 baris. Kumpulan data dimulai dari tahun 2011 hingga tahun 2014 yang berpusat pada transaksi pelanggan dari berbagai vendor dan pasar yang berbeda. Penelitian dengan *dataset* ini sudah dilakukan sebelumnya oleh (Mahfuza et al., 2022) dengan tujuan untuk melakukan segmentasi pelanggan. Adapun detail *dataset* dapat dilihat pada Gambar 2.

| Order ID | Order Date | Ship Date | Ship Mode | Customer ID | Customer Name | Segment | City | State | ... | Product ID | Category | Sub-Category |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CA-2012-124891 | 7/31/2012 | 7/31/2012 | Same Day | RH-19495 | Rick Hansen | Consumer | New York City | New York | ... | TEC-AC-10003033 | Technology | Accessories |
| IN-2013-77878 | 2/5/2013 | 2/7/2013 | Second Class | JR-16210 | Justin Ritter | Corporate | Wollongong | New South Wales | ... | FUR-CH-10003950 | Furniture | Chairs |
| IN-2013-71249 | 10/17/2013 | 10/18/2013 | First Class | CR-12730 | Craig Reiter | Consumer | Brisbane | Queensland | ... | TEC-PH-10004664 | Technology | Phones |
| ES- | | | | | | | | | | | | |

**Gambar 2 Ilustrasi *Dataset* Penelitian**

Sebelum melakukan analisis diperlukan pembersihan data dengan melakukan *data preprocessing* agar hasil analisis dapat sesuai dengan keinginan. Melalui *data preprocessing dataset* dapat dibersihkan seperti mengisi nilai yang hilang, mengkoreksi data ganda, ataupun mengkoreksi ketidakcocokan antar data (Anitha & Patil, 2022). Pada tahap selanjutnya yaitu menentukan metode analisis dengan menggunakan RFM (*recency, frequency, monetary*) dan K-Means. RFM memberikan kerangka kerja yang kuat dengan mempertimbangkan faktor seberapa baru pelanggan berbelanja (*recency*), seberapa sering mereka berbelanja (*frequency*), dan seberapa besar total belanjaan mereka (*monetary*). Integrasi metode klasterisasi seperti K-Means membawa dimensi tambahan dalam proses segmentasi, memungkinkan identifikasi kelompok pelanggan yang memiliki pola perilaku serupa. Pembuatan model dimulai dengan menghitung skor *recency, frequency*, dan *monetary* yang diperoleh dari metode RFM. Untuk memberikan gambaran perhitungan tersebut, contoh data pelanggan disajikan dalam Tabel 1.

**Tabel 1 Contoh Data Pelanggan**

| Nama Pelanggan | Tanggal Pembelian | Jumlah Pembelian |
|---|---|---|
| A | 12 November 2023 | $150 |
| B | 13 November 2023 | $200 |
| C | 15 November 2023 | $50 |
| B | 17 November 2023 | $300 |
| D | 19 November 2023 | $200 |
| D | 20 November 2023 | $100 |

Dengan mengasumsikan bahwa hari ini adalah tanggal 23 November 2023, skor *Recency* dihitung berdasarkan perbedaan antara tanggal saat ini dan tanggal transaksi terakhir pelanggan. Skor *Frequency* diperoleh dengan menghitung total transaksi yang dilakukan, sementara skor *Monetary* dihitung dengan menjumlahkan total pengeluaran dari setiap transaksi. Hasil dari ketiga perhitungan skor tersebut dapat dilihat pada Tabel 2.

**Tabel 2 Contoh Hasil Perhitungan RFM**

| Nama Pelanggan | *Recency* | *Frequency* | *Monetary* |
|---|---|---|---|
| A | 11 | 1 | $150 |
| B | 6 | 2 | $500 |
| C | 8 | 1 | $50 |
| D | 3 | 2 | $300 |

Skor yang telah dihitung dapat berfungsi sebagai fitur untuk segmentasi pelanggan pada algoritma K-Means. Algoritma K-Means bekerja dengan pertama menentukan jumlah klaster (*K*) yang diinginkan (Adiyanto & Arie Wijaya, 2023). Kemudian, secara acak memilih titik-titik awal sebagai *centroid* atau titik tengah. Setiap titik dalam *dataset* kemudian dikelompokkan ke klaster

yang paling dekat berdasarkan jarak Euclidean ke *centroid* tersebut. Jarak (*d*) antara titik data *y* dan pusat massa *x* dihitung menggunakan rumus jarak Euclidean, sebagaimana tertulis pada Pers. (1). Setelah pengelompokan, posisi *centroid* diperbarui dengan menghitung rata-rata dari semua titik dalam klaster tersebut. Proses ini diulang – pengelompokan data dan pembaruan *centroid* – hingga klaster stabil, yaitu ketika tidak ada perubahan signifikan dalam pengelompokan atau sampai batas iterasi maksimum tercapai. Hasil akhirnya adalah pembentukan klaster dengan titik-titik data yang memiliki kesamaan ciri di dalamnya dan titik-titik *centroid* yang stabil.

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

Langkah berikutnya melibatkan evaluasi model pada fase penilaian hasil klaster. Silhouette Coefficient (SC) adalah metrik evaluasi yang digunakan untuk mengukur sejauh mana titik data dalam suatu klaster cocok dengan klaster tempatnya berada dan seberapa dekat atau jauh dari klaster lainnya. Metrik ini merupakan metode evaluasi klaster gabungan antara separasi dan kohesi (Paembonan & Abduh, 2021). Adapun rumus SC dapat dilihat pada Pers. (2).

$$S(i) = \frac{b(i) - a(i)}{\max\big(a(i), b(i)\big)} \quad (2)$$

Evaluasi selanjutnya dilakukan dengan menggunakan Davies Bouldin Index (DBI), yang mengukur perbandingan antara jarak di dalam suatu klaster dan jarak antar klaster. Pers. (3) menjelaskan DBI, dengan *k* sebagai jumlah total klaster yang digunakan, dan ($R_{i,j}$) sebagai rasio antara klaster *i* dan klaster *j*. Nilai DBI yang lebih rendah menandakan bahwa kualitas klasterisasi yang dihasilkan lebih baik (Agustino & Budaya, 2023).

$$DBI = \frac{1}{K}\sum_{i}^{K} max_{i \neq j}\big(R_{i,j}\big) \quad (3)$$

Metrik evaluasi terakhir adalah Calinski Harabasz Index (CHI) yang digunakan untuk mengukur kualitas suatu klasterisasi pada data. Keuntungan dari CHI adalah bahwa nilai yang lebih tinggi mencerminkan klaster yang lebih jelas (Sikana & Wijayanto, 2021). Perhitungan CHI dapat dilihat pada Pers. 4.

$$CHI = \frac{tr(B_k)}{tr(W_k)} \times \frac{(N-k)}{(k-1)} \quad (4)$$

Dalam melakukan segmentasi pelanggan, digunakan model *machine learning*. Model klasterisasi diterapkan menggunakan bahasa pemrograman Python dengan menggunakan pustaka Scikit-learn, dan parameter bawaan dari Scikit-learn yang digunakan. Setelah melalui tahap pengujian, langkah selanjutnya adalah fase implementasi model, yang melibatkan pemanfaatan pustaka Streamlit. Pustaka ini mempermudah pembuatan dasbor yang menampilkan hasil analisis secara visual. Untuk menjalankan *dashboard* menggunakan Streamlit, diperlukan koneksi ke GitHub untuk mengakses kode sumber yang sebelumnya telah diunggah ke GitHub. Melalui proses ini, program yang sebelumnya berjalan secara lokal dapat diakses secara publik.

## 3. HASIL DAN PEMBAHASAN

### 3.1 Perbandingan Model Klaterisasi

Perbandingan antara metode klasterisasi K-Means, K-Medoids, dan Fuzzy C Means dapat dievaluasi berdasarkan tiga metrik utama, yaitu Silhouette Coefficient (SC), Davies Bouldin Index (DBI), dan Calinski Harabasz Index (CHI). Silhouette Coefficient memberikan indikasi seberapa

baik objek dalam klaster terpisah dan saling berdekatan, dengan nilai mendekati 1 menandakan pembentukan klaster yang baik. Davies Bouldin Index menilai sejauh mana batas klaster terdefinisi dan seberapa homogen klaster tersebut. Nilai rendah pada DBI mengindikasikan pembentukan klaster yang baik. Sementara itu, Calinski Harabasz Index mengevaluasi homogenitas dan pemisahan klaster, dengan nilai tinggi menandakan pembentukan klaster yang baik. Hasil perhitungan yang telah dilakukan menunjukkan bahwa algoritma K-Means lebih unggul daripada algoritma K-Medoids dan Fuzzy C Means. Nilai ketiga indeks tersebut menunjukkan nilai 0,67305 pada Silhouette Coefficient, nilai 0,51435 Davies Bouldin Index, dan nilai 5647,89 pada Calinski Harabasz Index, sebagaimana tergambar pada Tabel 3.

**Tabel 3 Perbandingan Model Klasterisasi**

| Matriks | K-Means | K-Medoids | Fuzzy C Means |
|---|---|---|---|
| Silhouette Coefficient (SC) | 0,67305 | 0,66130 | 0,67303 |
| Davies Bouldin Index (DBI) | 0,51435 | 0,52575 | 0,51467 |
| Calinski Harabasz Index (CHI) | 5647,89 | 5271,67 | 5644,48 |

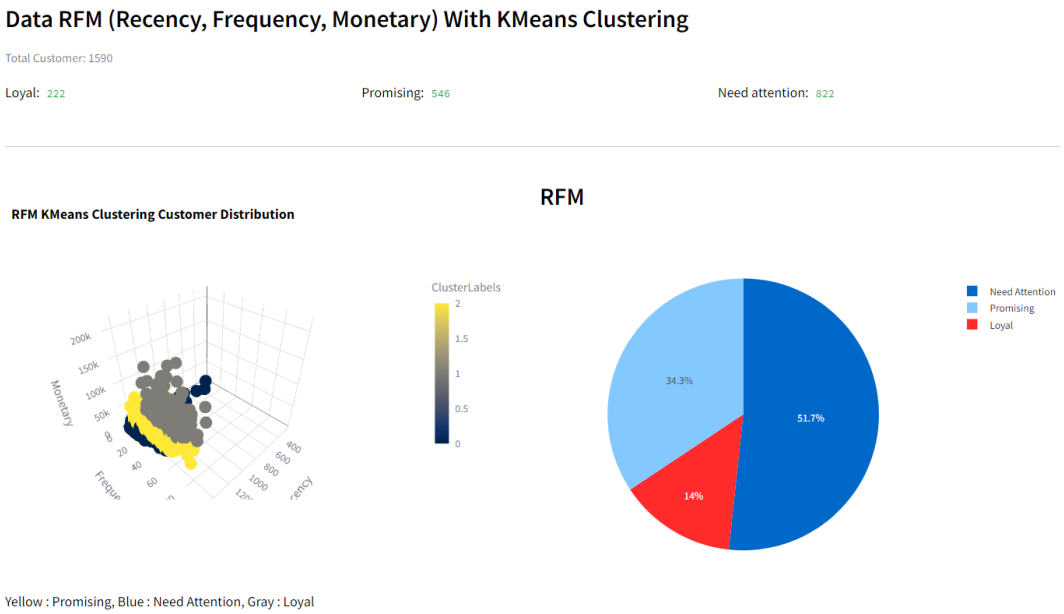
Hasil segmentasi menggunakan fitur RFM melalui penerapan algoritma K-Means menunjukkan bahwa pelanggan dapat diklasifikasikan ke dalam tiga kategori, yaitu *loyal, promising,* dan *need attention* (Hilmy et al., 2023). Jumlah masing-masing pelanggan/konsumen dalam kelompok *loyal, promising*, dan *need attention* adalah 222, 546, dan 822. Dari jumlah tersebut, 222 pelanggan dianggap *loyal* karena memiliki frekuensi tinggi dan nilai *monetary* yang tinggi. Kategori *promising* mencakup 546 pelanggan dengan frekuensi dan *monetary* sedang. Sementara itu, kategori *need attention* terdiri dari 822 pelanggan dengan frekuensi dan *monetary* yang rendah. Hasil pengelompokkan menggunakan K-Medoids menunjukkan bahwa 316 pelanggan dikategorikan sebagai *loyal*, 478 pelanggan dalam kategori *promising*, dan 796 pelanggan diklasifikasikan *need attention*. Algoritma Fuzzy C Means menghasilkan 225 pelanggan pada kategori *loyal*, 546 pelanggan pada *promising*, dan 819 pelanggan pada kategori *need attention*. Rangkuman jumlah pelanggan dari hasil klasterisasi ini dapat dilihat pada Tabel 4.

**Tabel 4 Hasil Pengelompokkan Klasterisasi**

| Kategori | K-Means | K-Medoids | Fuzzy C Means |
|---|---|---|---|
| *Loyal* | 222 | 316 | 225 |
| *Promising* | 546 | 478 | 546 |
| *Need Attention* | 822 | 796 | 819 |

### 3.2 Visualisasi Hasil

Penelitian ini bertujuan untuk merancang dan mengembangkan sistem segmentasi pelanggan *e-commerce* berbasis web untuk membantu memberikan dukungan pada proses pengambilan keputusan manajemen. Sistem ini dikonstruksi dengan menggunakan bahasa pemrograman Python, memanfaatkan perpustakaan Streamlit untuk antarmuka web dan memanfaatkan Google Sheets untuk meletakkan *file dataset* yang telah dimiliki. Proses pengelompokan data pelanggan dilakukan dengan menggunakan perpustakaan Scikit-learn. Setelah berhasil melakukan *login*, pengguna akan diarahkan ke halaman *dashboard*, sebagaimana terlihat pada Gambar 3, yang menampilkan hasil klastersasi.




**Gambar 3 Hasil Visualisasi**

## 4. KESIMPULAN

Segmentasi pelanggan menggunakan model RFM telah berhasil dijalankan, dan dari hasil analisis tersebut, beberapa kesimpulan dapat diambil. Pertama, fitur RFM (*Recency, Frequency, Monetary*) bersama dengan algoritma K-Means, K-Medoids, dan Fuzzy C Means berhasil menampilkan 3 segmen pelanggan utama, yaitu *Loyal, Promising,* dan *Need Attention*. Kedua, performa algoritma K-Means, K-Medoids, dan FCM dievaluasi menggunakan metrik seperti Silhouette Coefficient (SC), Davies Bouldin Index (DBI), dan Calinski Harabasz Index (CHI). Hasil evaluasi menunjukkan bahwa algoritma K-Means mencapai hasil paling optimal dengan nilai 0,67305 pada SC, 0,51435 pada DBI, dan 5647,89 pada CHI. Ketiga, *output* dari analisis dapat disajikan secara visual melalui aplikasi web menggunakan pustaka Streamlit, sehingga dapat memberikan dukungan untuk pengambilan keputusan yang efektif. Visualisasi menggunakan *scatter plots, pie charts,* dan *line charts* memudahkan penyajian hasil analisis dengan jelas, membantu pemahaman pola dan tren pelanggan secara intuitif. Kedepannya, optimalisasi model dan komparasi dengan model klasterisasi lain akan kami sajikan dalam penelitian selanjutnya.

## DAFTAR PUSTAKA

Adiyanto, A., & Arie Wijaya, Y. (2023). PENERAPAN ALGORITMA K-MEANS PADA PENGELOMPOKAN DATA SET BAHAN PANGAN INDONESIA TAHUN 2022-2023. *JATI (Jurnal Mahasiswa Teknik Informatika)*, *7*(2), 1344–1350. https://doi.org/10.36040/jati.v7i2.6849

Agustino, D. P., & Budaya, I. G. B. A. (2023). Evaluasi Performa Segmentasi Pelanggan Tenant Inkubator Bisnis dengan Menggunakan Model Consensus Clustering. *Prosiding CORISINDO 2023*, 236–240. https://www.stmikpontianak.org/ojs/index.php/corisindo/article/view/62

Alzami, F., Sambasri, F. D., Nabila, M., Megantara, R. A., Akrom, A., Pramunendar, R. A., Prabowo, D. P., & Sulistiyawati, P. (2023). Implementation of RFM Method and K-Means Algorithm for Customer Segmentation in E-Commerce with Streamlit. *ILKOM Jurnal Ilmiah*, *15*(1), 32–44. https://doi.org/10.33096/ilkom.v15i1.1524.32-44

Anitha, P., & Patil, M. M. (2022). RFM model for customer purchase behavior using K-Means algorithm. *Journal of King Saud University - Computer and Information Sciences*, *34*(5), 1785–1792. https://doi.org/10.1016/j.jksuci.2019.12.011

Chen, A. H. L., Liang, Y.-C., Chang, W.-J., Siauw, H.-Y., & Minanda, V. (2022). RFM Model and K-Means Clustering Analysis of Transit Traveller Profiles: A Case Study. *Journal of Advanced Transportation*, *2022*(1), 1–14. https://doi.org/10.1155/2022/1108105

Hilmy, F. M., Nurhaliza, R. A., Huzyan Octava, M. Q., & Alfian, G. (2023). Web-based E-Commerce Customer Segmentation System Using RFM and K-Means Model. *2023 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT)*, 83–87. https://doi.org/10.1109/3ICT60104.2023.10391650

Juhari, T., & Juarna, A. (2022). IMPLEMENTATION RFM ANALYSIS MODEL FOR CUSTOMER SEGMENTATION USING THE K-MEANS ALGORITHM CASE STUDY XYZ ONLINE BOOKSTORE. *EXPLORE*, *12*(1), 107–118. https://doi.org/10.35200/explore.v12i1.548

Kim, S.-Y., Jung, T.-S., Suh, E.-H., & Hwang, H.-S. (2006). Customer segmentation and strategy development based on customer lifetime value: A case study. *Expert Systems with Applications*, *31*(1), 101–107. https://doi.org/10.1016/j.eswa.2005.09.004

Li, W., Wu, X., Sun, Y., & Zhang, Q. (2010). Credit Card Customer Segmentation and Target Marketing Based on Data Mining. *2010 International Conference on Computational Intelligence and Security*, 73–76. https://doi.org/10.1109/CIS.2010.23

Mahfuza, R., Islam, N., Toyeb, Md., Emon, M. A. F., Chowdhury, S. A., & Alam, Md. G. R. (2022). LRFMV: An efficient customer segmentation model for superstores. *PLOS ONE*, *17*(12), e0279262. https://doi.org/10.1371/journal.pone.0279262

Molla, A., & Licker, P. S. (2005). eCommerce adoption in developing countries: a model and instrument. *Information & Management*, *42*(6), 877–899. https://doi.org/10.1016/j.im.2004.09.002

Paembonan, S., & Abduh, H. (2021). Penerapan Metode Silhouette Coefficient untuk Evaluasi Clustering Obat. *PENA TEKNIK: Jurnal Ilmiah Ilmu-Ilmu Teknik*, *6*(2), 48–54. https://doi.org/10.51557/pt_jiit.v6i2.659

Shirole, R., Salokhe, L., & Jadhav, S. (2021). Customer Segmentation using RFM Model and K-Means Clustering. *International Journal of Scientific Research in Science and Technology*, 591–597. https://doi.org/10.32628/IJSRST2183118

Sikana, A. M., & Wijayanto, A. W. (2021). Analisis Perbandingan Pengelompokan Indeks Pembangunan Manusia Indonesia Tahun 2019 dengan Metode Partitioning dan Hierarchical Clustering. *Jurnal Ilmu Komputer*, *14*(2), 66–78. https://doi.org/10.24843/JIK.2021.v14.i02.p01

Sutresno, S. A., Iriani, A., & Sediyono, E. (2018). Metode K-Means Clustering dengan Atribut RFM untuk Mempertahankan Pelanggan. *Jurnal Teknik Informatika Dan Sistem Informasi*, *4*(3), 433-440–433–440. https://doi.org/10.28932/jutisi.v4i3.878

**JISKA (Jurnal Informatika Sunan Kalijaga)**
**Vol. 9, No. 3, SEPTEMBER, 2024, Pp. 178 – 191**
**ISSN: 2527 – 5836 (print) | 2528 – 0074 (online)**

# Analisis Performa Normalisasi Data untuk Klasifikasi K-Nearest Neighbor pada *Dataset* Penyakit

**Petronilia Palinggik Allorerung [(1)*], Angdy Erna [(2)], Muhammad Bagussahrir [(3)], Samsu Alam [(3)]**
[1,3,4] Teknik Informatika, Universitas Dipa Makassar, Makassar, Indonesia
[2] Sistem Informasi, Universitas Dipa Makassar, Makassar, Indonesia
e-mail : {petroniliaallorerung,bagussahrir}@gmail.com, {angdy,alam}@undipa.ac.id.
* Penulis korespondensi.
Artikel ini diajukan 28 Februari 2024, direvisi 14 Agustus 2024, diterima 15 Agustus 2024, dan dipublikasikan 25 September 2024.

***Abstract***

*This study investigates four normalization methods (Min-Max, Z-Score, Decimal Scaling, MaxAbs) across prostate, kidney, and heart disease datasets for K-Nearest Neighbor (K-NN) classification. Imbalanced feature scales can hinder K-NN performance, making normalization crucial. Results show that Decimal Scaling achieves 90.00% accuracy in prostate cancer, while Min-Max and Z-Score yield 97.50% in kidney disease. MaxAbs performs well with 96.25% accuracy in kidney disease. In heart disease, Min-Max and MaxAbs attain accuracies of 82.93% and 81.95%, respectively. These findings suggest Decimal Scaling suits datasets with few instances, limited features, and normal distribution. Min-Max and MaxAbs are better for datasets with numerous instances and non-normal distribution. Z-Score fits datasets with a wide range of feature numbers and near-normal distribution. This study aids in selecting the appropriate normalization method based on dataset characteristics to enhance K-NN classification accuracy in disease diagnosis. The experiments involve datasets with different attributes, continuous and categorical data, and binary classification. Data conditions such as the number of instances, the number of features, and data distribution affect the performance of normalization and classification.*

***Keywords: Data Normalization, Disease, Min-Max, Z-Score, Decimal Scaling, MaxAbs, K-Nearest Neighbor***

**Abstrak**

Penelitian ini menginvestigasi empat metode normalisasi (*Min-Max, Z-Score, Decimal Scaling, MaxAbs*) pada *dataset* kanker prostat, ginjal, dan jantung untuk klasifikasi K-Nearest Neighbor (K-NN). Skala fitur yang tidak seimbang dapat menghambat kinerja K-NN, sehingga normalisasi data menjadi penting. Hasil penelitian menunjukkan bahwa *Decimal Scaling* mencapai akurasi tertinggi sebesar 90,00% pada penyakit kanker prostat, sementara *Min-Max* maupun *Z-Score* memberikan akurasi tertinggi sebesar 97,50% pada penyakit ginjal. *MaxAbs* juga tampil baik dengan akurasi 96,25% pada penyakit ginjal. Pada penyakit jantung, *Min-Max* dan *MaxAbs* mencapai akurasi masing-masing sebesar 82,93% dan 81,95%. Temuan ini menyimpulkan bahwa *Decimal Scaling* secara umum cocok untuk *dataset* dengan jumlah *instance* yang sedikit, jumlah fitur terbatas, dan berdistribusi normal. *Min-Max* dan *MaxAbs* cenderung lebih sesuai untuk *dataset* dengan jumlah *instance* yang banyak, fitur yang banyak, dan berdistribusi tidak normal. *Z-Score* cocok untuk *dataset* dengan jumlah fitur yang relatif besar atau kecil dan cocok untuk *dataset* berdistribusi normal atau mendekati normal. Penelitian ini membantu menentukan metode normalisasi yang sesuai dengan karakteristik *dataset* untuk meningkatkan akurasi model klasifikasi K-NN dalam mendiagnosis penyakit. Eksperimen menggunakan tiga *dataset* penyakit dengan atribut yang berbeda-beda, jenis data kontinu dan kategorikal, serta klasifikasi biner. Kondisi data seperti jumlah *instance*, jumlah fitur, dan distribusi data mempengaruhi performa normalisasi dan klasifikasi.

**Kata Kunci: Normalisasi Data, Penyakit, *Min-Max*, *Z-Score*, *Decimal Scaling*, *MaxAbs*, K-Nearest Neighbor**

JISKA (Jurnal Informatika Sunan Kalijaga)
Vol. 9, No. 3, SEPTEMBER, 2024: 178 – 191

## 1. PENDAHULUAN

*Data transformation* adalah proses pengubahan data menjadi bentuk yang cocok untuk proses eksplorasi data (Marlina & Bakri, 2021). Salah satu cara transformasi data adalah normalisasi data. Normalisasi data ialah metode menskalakan ulang data menjadi lebih kecil (Ambarwari et al., 2020). Penggunaan normalisasi data sangat diperlukan jika terdapat perbedaan skala antar fitur dalam *dataset*. Konsekuensi jika tidak melakukan normalisasi data adalah menghasilkan nilai akurasi yang kecil dan menyebabkan fitur berskala rendah tidak berpengaruh saat mengimplementasikan algoritma *data mining* yang melibatkan pengukuran jarak, atau dengan kata lain hasil analisis *data mining* hanya bergantung pada fitur berskala tinggi, padahal fitur berskala rendah juga memiliki peranan yang sama pentingnya (Kusnaidi et al., 2022).

Penggunaan normalisasi data sangat penting dilakukan apabila menggunakan metode *data mining* yang melibatkan pengukuran jarak seperti K-Nearest Neighbor (K-NN). K-NN ialah salah satu *method* yang didasarkan pada mayoritas tetangga terdekatnya yang dihasilkan dari proses perhitungan jarak euclidean. Jika jarak nilai setiap fitur sangat besar, maka perhitungan euclidean *distance* kurang maksimal. Perhitungan *distance* yang kurang maksimal memberikan hasil yang kurang tepat dan menunjukkan bahwa *dataset* yang gunakan tidak berkualitas (Whendasmoro & Joseph, 2022).

Penerapan normalisasi data tidak digunakan pada semua *dataset*, melainkan hanya pada *dataset* yang memiliki perbedaan skala pada fiturnya. Salah satu *dataset* yang memiliki karakteristik tersebut adalah *dataset* penyakit. *Dataset* penyakit adalah salah satu contoh data yang sering dipakai dalam studi *data mining*. *Dataset* penyakit sendiri umumnya memiliki nilai numerik, di mana terdapat perbedaan skala pada fiturnya, seperti fitur tinggi badan dan berat badan. Perbedaan skala antar fitur menyebabkan fitur dengan nilai yang lebih kecil tidak bermanfaat dibandingkan dengan fitur lainnya. Oleh karenanya, perlu dilakukan normalisasi data untuk menyeimbangkan skala setiap fitur pada *dataset* ke rasio yang lebih kecil. Normalisasi data dapat dilakukan dengan metode yang umumnya digunakan oleh para peneliti seperti *Min-Max*, *Z-Score*, *Decimal Scaling* (Pagan et al., 2023) dan *MaxAbs* (Permana & Salisah, 2022).

Penelitian tentang penggunaan normalisasi data sudah pernah dilakukan pada penelitian prediksi penyakit diabetes dengan dua normalisasi yakni *Min-Max* dan *Z-Score* dengan algoritma K-NN dan didapat bahwa normalisasi *Min-Max* memperoleh akurasi tertinggi (Sholeh et al., 2022). Adapula penelitian yang membandingkan tiga normalisasi yakni *Min-Max, Z-Score,* dan *Decimal Scaling* untuk klasifikasi *wine* dengan metode K-NN dan didapat bahwa akurasi tertinggi ada pada *Min-Max* (Chandra et al., 2022). Pada penelitian klasifikasi status gizi balita dilakukan perbandingan dua normalisasi data yaitu *Z-Score* dan *Min-Max* dengan metode K-NN dan didapat bahwa metode *Z-Score* memiliki akurasi tertinggi (HS et al., 2023). Dari penelitian sebelumnya, dapat dilihat bahwa *Min-Max* merupakan metode normalisasi data terbaik. Akan tetapi, penelitian tersebut masih membandingkan dua metode normalisasi data, sedangkan perbandingan dengan metode *Min-Max, Z-Score, Decimal Scaling*, dan *MaxAbs* belum dilakukan, sehingga belum diketahui apakah *MaxAbs* memiliki akurasi lebih tinggi atau tidak dari *Min-Max.* Selain itu*,* terdapat pula penelitian oleh Pagan et al. (2023) yang membandingkan tiga normalisasi (*Min-Max, Z-Score*, dan *Decimal Scaling*) untuk sepuluh *dataset* yang dipilih oleh peneliti berdasarkan keragaman ukuran, kompleksitas, dan dimensi fiturnya. Hasil penelitian ini menunjukkan ada enam *dataset* yang menghasilkan *Min-Max* sebagai metode terbaik bahkan mayoritas menghasilkan akurasi di atas 85%, sedangkan empat *dataset* lainnya menghasilkan *Z-Score* sebagai metode terbaik dan mayoritas menghasilkan akurasi di atas 97%. Hal ini menunjukkan bahwa penemuan akurasi tertinggi melalui metode normalisasi yang sama disebabkan oleh kemiripan karakteristiknya, namun penelitian ini tidak menjelaskan secara pasti jumlah ukuran, kompleksitas, dan dimensi fitur tiap *dataset*. Oleh karena itu, metode normalisasi data terbaik yang dihasilkan belum bisa dijadikan acuan untuk setiap kondisi *dataset*.

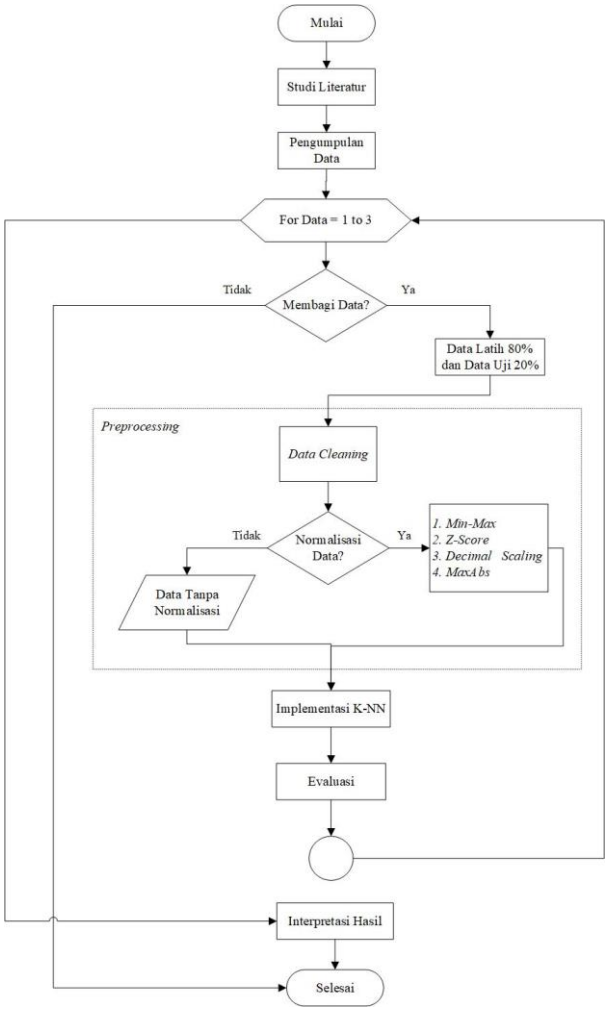
Berdasarkan penjabaran di atas, maka penulis akan mengeksplorasi empat metode normalisasi data yakni *Min-Max Normalization*, *Z-Score Normalization*, *Decimal Scaling Normalization*, dan

*MaxAbs Normalization* dengan algoritma K-NN. Metode tersebut digunakan dalam mengklasifikasi tiga *dataset* penyakit yang diperoleh dari *website* Kaggle. Tujuannya adalah untuk mengidentifikasi metode normalisasi terbaik sesuai dengan karakteristik data tersebut.

## 2. METODE PENELITIAN

### 2.1 Tahap Penelitian

Tahap-tahap yang akan dilaksanakan divisualisasikan oleh Gambar 1. *Flowchart* ini menggambarkan proses penelitian untuk menentukan metode normalisasi terbaik dalam meningkatkan akurasi klasifikasi penyakit menggunakan K-Nearest Neighbor *(*K-NN*)*. Proses dimulai dengan studi literatur dan pengumpulan data, diikuti dengan *looping* untuk tiga *dataset* penyakit (kanker prostat, ginjal, dan jantung). Data kemudian dibagi menjadi 80% data latih dan 20% data uji. Langkah berikutnya adalah *preprocessing*, termasuk *data cleaning* dan normalisasi data menggunakan metode *Min-Max, Z-Score, Decimal Scaling*, dan *MaxAbs*. Setelah itu, algoritma K-NN diterapkan pada data yang sudah diproses, dan kinerja model dievaluasi. Hasil evaluasi kemudian diinterpretasikan untuk mendapatkan kesimpulan akhir sebelum proses penelitian dinyatakan selesai.




**Gambar 1 *Flowchart* Tahap Penelitian**

### 2.2 Pengumpulan Data

Data yang dikumpulkan bersumber dari *website* Kaggle yaitu *dataset* penyakit kanker prostat sebanyak 100 data, penyakit ginjal sebanyak 400 data, dan penyakit jantung sebanyak 1025 data. Adapun struktur ketiga *dataset* tersebut ditunjukkan oleh Tabel 1 sampai 3. Pemilihan ketiga *dataset* ini didasarkan pada beberapa alasan yang komprehensif seperti berikut:

1) **Keberagaman Ukuran *Dataset***: Ketiga *dataset* dipilih karena mewakili variasi dalam ukuran *dataset*. Penggunaan ukuran *dataset* yang bervariasi penting dalam mengevaluasi kinerja algoritma pembelajaran mesin karena menurut penelitian oleh Pagan et al. (2023) yang menggunakan sepuluh *dataset* ditemukan bahwa keberagaman ukuran *dataset* dapat mempengaruhi kemampuan algoritma untuk akurasi prediksi.
2) **Variasi dalam Atribut**: Setiap *dataset* memiliki jumlah dan jenis atribut yang berbeda, yang memungkinkan untuk menguji metode normalisasi dalam konteks yang berbeda. Variasi ini penting karena berdasarkan penggunaan jumlah dan jenis fitur yang beragam pada penelitian Pagan et al. (2023), ditemukan bahwa variasi dalam atribut dapat mempengaruhi kinerja algoritma pembelajaran mesin dan normalisasi.
3) **Kelas Biner**: Ketiga *dataset* memiliki kelas biner (dua kelas) yang konsisten. Ini penting karena klasifikasi biner adalah salah satu jenis klasifikasi yang paling umum dalam analisis kesehatan, dan menurut Riaz et al. (2022), klasifikasi biner lebih sederhana dan hampir 60% penelitian membahas klasifikasi biner. Oleh karena itu, klasifikasi biner menjadi langkah awal yang penting sebelum melakukan klasifikasi dengan lebih banyak kelas.
4) **Skala Atribut yang Beragam**: Setiap *dataset* menggunakan skala yang berbeda untuk atribut mereka, mulai dari atribut kontinu hingga atribut diskrit. Variasi skala ini memungkinkan penelitian untuk menguji efektivitas berbagai metode normalisasi yang berbeda dalam menangani skala atribut yang berbeda, yang menurut Jain et al. (2000), dapat signifikan dalam mempengaruhi kinerja algoritma pembelajaran mesin.
5) **Signifikansi Klinis**: Pemilihan *dataset* penyakit kanker prostat, ginjal, dan jantung didasarkan pada prevalensi dan signifikansi klinis penyakit tersebut. Ketiganya adalah penyakit serius dengan dampak besar pada kesehatan masyarakat, sehingga penelitian yang meningkatkan diagnosis mereka sangat berharga. Menurut data World Health Organization yang dirilis tahun 2020, penyebab kematian utama di dunia pada tahun 2019 adalah penyakit, diantaranya kardiovaskular dan kanker, sehingga membuat penelitian ini sangat relevan.

Dengan mempertimbangkan faktor-faktor di atas, ketiga *dataset* ini memberikan basis yang komprehensif untuk mengevaluasi dan membandingkan performa metode normalisasi dalam klasifikasi K-NN. Selain itu, variasi dalam jumlah data, jumlah atribut, dan skala atribut memastikan bahwa temuan penelitian ini akan lebih umum dan aplikatif dalam konteks yang lebih luas.

**Tabel 1 Struktur Data Penyakit Kanker Prostat**

| Fitur | Nilai |
|---|---|
| *Radius* | 9 – 25 |
| *Texture* | 11 – 27 |
| *Perimeter* | 52 – 172 |
| *Area* | 202 – 1878 |
| *Smoothness* | 0,070 – 0,143 |
| *Compactness* | 0,038 – 0,345 |
| *Symmetry* | 0,135 – 0,304 |
| *Fractal Dimension* | 0,053 – 0,097 |
| *Class* | *Malignant dan Benign* |

**Tabel 2 Struktur Data Penyakit Ginjal**

| Fitur | Nilai |
|---|---|
| *Blood Pressure* | 50 – 180 |
| *Specific Gravity* | 1.005 – 1.025 |
| *Albumin* | 0 – 5 |
| *Sugar* | 0 – 5 |
| *Red Blood Cell* | 0 dan 1 |
| *Blood Urea* | 1.500 – 391 |
| *Serum Creatinine* | 0.400 – 76 |
| *Sodium* | 4.500 – 163 |
| *Potassium* | 2.500 – 47 |
| *Hemoglobin* | 3.100 – 17.800 |
| *White Blood Cell Count* | 2200 – 26400 |
| *Red Blood Cell Count* | 2.100 – 8 |
| *Hypertension* | 0 dan 1 |
| *Class* | 0 dan 1 |

**Tabel 3 Struktur Data Penyakit Jantung**

| Fitur | Nilai |
|---|---|
| *Age* | 29 – 77 |
| *Sex* | 0 dan 1 |
| *Chest Pain Type* | 0 – 3 |
| *Resting Blood Pressure* | 94 – 200 |
| *Serum Cholesterol* | 126 – 564 |
| *Fasting Blood Sugar* | 0 dan 1 |
| *Resting Electrocardiographic Result* | 0 – 2 |
| *Maximum Heart Rate Achieved* | 71 – 202 |
| *Exercise Induced Angina* | 0 dan 1 |
| *Oldpeak* | 0 – 6.200 |
| *Slope* | 0 – 2 |
| *Number of Major Vessels* | 0 – 4 |
| *Thalassemia* | 0 – 3 |
| *Class* | 0 dan 1 |

## 2.3 Teknik Normalisasi

### 2.3.1 *Min-Max Normalization*

*Min-Max* adalah jenis normalisasi yang mentransformasi linier pada *original data* (Chandra et al., 2022). Skala yang dihasilkan berada di antara 0 hingga 1. Rumus normalisasi *Min-Max* ditunjukkan pada Pers. (1). Di mana $x_i$ adalah data yang dinormalisasi, $x'$ menunjukkan hasil normalisasi, $min(x)$ merupakan data terkecil suatu fitur, dan $max(x)$ adalah data terbesar suatu fitur.

$$x' = \frac{x_i - min(x)}{max(x) - min(x)} \quad (1)$$

### 2.3.2 *Z-Score Normalization*

*Z-Score* ialah jenis normalisasi yang didasarkan pada *mean* dan standar deviasi (Chandra et al., 2022). *Z-Score* mengubah data asli menjadi skala yang lebih kecil, tetapi tidak memiliki skala yang tetap. Pes. (2) menunjukkan rumus normalisasi *Z-Score*. Di mana $x_i$ merupakan data yang

dinormalisasi, $x'$ menunjukkan hasil normalisasi, $mean(x)$ adalah nilai rata-rata suatu fitur, dan $std(x)$ merupakan standar deviasi suatu fitur.

$$x' = \frac{x_i - mean(x)}{std(x)} \quad (2)$$

### 2.3.3 *Decimal Scaling Normalization*

*Decimal Scaling* adalah jenis normalisasi yang menggeser nilai desimal data (Chandra et al., 2022). Skala yang dihasilkan *Decimal Scaling* berada di antara 0 hingga 1. Rumus normalisasi *Decimal Scaling* dituliskan pada Pers. (3). Di mana $x'$ adalah hasil normalisasi, $x_i$ merupakan data yang dinormalisasi, dan $j$ menunjukkan jumlah digit dari data terbesar pada tabel.

$$x' = \frac{x_i}{10^j} \quad (3)$$

### 2.3.4 *MaxAbs Normalization*

*MaxAbs* merupakan jenis normalisasi yang melakukan pembagian seluruh data dengan nilai absolut maksimum (Permana & Salisah, 2022). Data asli yang dinormalisasi dengan *MaxAbs* biasanya berada pada skala -1 hingga 1. Rumus normalisasi *MaxAbs* ditunjukkan pada Pers. (4). Di mana $x_i$ merupakan data yang dinormalisasi, $x'$ menunjukkan hasil normalisasi n, dan $max(x)$ adalah data terbesar suatu fitur.

$$x' = \frac{x_i}{|max(x)|} \quad (4)$$

## 2.4 Algoritma K-Nearest Neighbor (K-NN)

K-Nearest Neighbor (K-NN) ialah *supervised learning method* yang digunakan dalam mengklasifikasikan objek baru berdasarkan *class* terbanyak pada tetangga terdekatnya. Berikut langkah-langkah dalam mengimplementasikan K-NN (Cahyanti et al., 2020):

1) Menentukan nilai K yang terdiri dari dua cara, yaitu menggunakan bilangan ganjil yakni 1,3,5,7, dan 9 atau menerapkan rumus pada Pers. (5). Di mana $N$ adalah banyaknya data latih dan $k$ adalah jumlah tetangga terdekat.

$$k = \sqrt{N} \quad (5)$$

2) Hitung *distance* antara *data testing* dan semua *data training*. Perhitungan tersebut menerapkan rumus Euclidean Distance pada Pers. (6). Di mana $p_i$ merupakan *data training*, $q_i$ adalah *data testing*, $i$ menunjukkan data variabel, dan $n$ adalah banyak data.

$$Euclidean = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2} \quad (6)$$

3) Urutkan hasil perhitungan jarak dari terkecil ke terbesar.
4) Menentukan tetangga terdekat berdasarkan jarak terpendek hingga nilai K.
5) Menentukan *class* berdasarkan tetangga terdekat.
6) Menemukan *class* mayoritas dari tetangga terdekat dan menetapkannya sebagai *class data testing*.

### 2.5 Teknik Pengujian

#### 2.5.1 Uji Normalitas

Studi ini menerapkan uji normalitas secara statistik dengan Kolmogorov-Smirnov menggunakan aplikasi SPSS. Pengujian ini memakai nilai statistik uji dan nilai tabel Kolmogorov-Smirnov. Uji Kolmogorov-Smirnov yang tersedia di SPSS memungkinkan peneliti untuk dengan mudah membandingkan distribusi sampel dengan distribusi teoretis (normal) tanpa perlu melakukan pemrograman yang kompleks. Hasil pengujian tiap *dataset* tersebut dijelaskan oleh Tabel 4. Berdasarkan Tabel 4 maka dapat diketahui bahwa *dataset* penyakit kanker prostat berdistribusi normal karena nilai statistik uji < nilai tabel *K-S*, sebaliknya *dataset* penyakit ginjal dan *dataset* penyakit jantung tidak berdistribusi normal.

**Tabel 4 Hasil Uji Normalitas**

| ***Dataset*** | **Nilai Statistik Uji** | **Nilai Tabel *K-S*** |
|---|---|---|
| Penyakit Kanker Prostat | 0,077 | 0,13581 |
| Penyakit Ginjal | 0,079 | 0,06790 |
| Penyakit Jantung | 0,066 | 0,04242 |

Menurut Singh & Singh (2020), normalisasi sangat penting untuk memastikan bahwa setiap fitur memiliki kontribusi yang sama sehingga dapat meningkatkan kualitas data dan kinerja algoritma. Data yang tidak berdistribusi normal dapat menyebabkan model K-NN menjadi lebih bias atau memiliki *variance* yang lebih tinggi. Jika distribusi data tidak terdistribusi secara merata, K-NN dapat cenderung *overfitting* pada daerah tertentu dari data atau *underfitting* pada daerah lain. Ini dapat mengurangi kemampuan generalisasi model. Normalisasi membantu menangani *outlier* yang mana menurut Barus & Sutarman (2023), kehadiran *outlier* juga dapat menyebabkan distribusi data yang tidak normal sehingga menciptakan bias dan menurunkan performa algoritma *machine learning*.

#### 2.5.2 Uji Kinerja Algoritma

Akurasi menunjukkan persentase data yang berhasil diklasifikasikan dengan benar. Nilai akurasi dihasilkan melalui rumus pada Pers. (7). AUC (*Area Under the ROC Curve*) menunjukkan seberapa baik pola yang terbentuk dalam memprediksi kelas dengan tepat. Nilai AUC dibagi menjadi lima kelompok yang ditunjukkan pada Tabel 5.

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% \quad (7)$$

**Tabel 5 Nilai AUC**

| **Nilai** | **Klasifikasi** |
|---|---|
| 0,90 – 1,00 | Sangat Baik |
| 0,80 – 0,90 | Baik |
| 0,70 – 0,80 | Cukup |
| 0,60 – 0,70 | Buruk |
| 0,50 – 0,60 | Salah |

## 3. HASIL DAN PEMBAHASAN

### 3.1 Membagi Data (*Split Data)*

*Spliting data* menjadi data latih dan data uji diterapkan untuk ketiga *dataset* penyakit dengan persentase 80% sebagai data latih dan 20% sebagai data uji. Proses pembagian ini dikerjakan oleh RapidMiner secara default sehingga tidak menyebabkan perbedaan hasil ketika dilakukan analisis secara berulang-ulang. Hasil *split data* ini dijelaskan oleh Tabel 6.

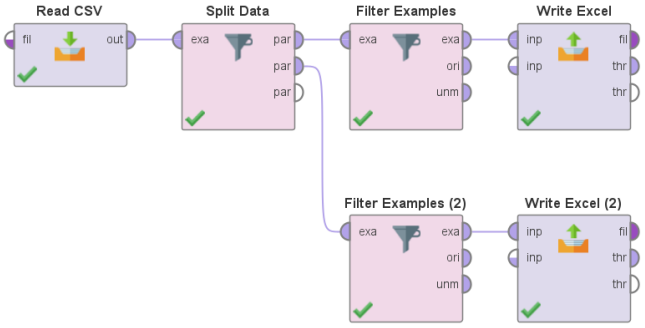
**Tabel 6 Hasil Split Data**

| *Dataset* | Total Data | Jumlah Data Latih | Jumlah Data Uji |
|---|---|---|---|
| Penyakit Kanker Prostat | 100 | 80 | 20 |
| Penyakit Ginjal | 400 | 320 | 80 |
| Penyakit Jantung | 1025 | 820 | 205 |

### 3.2 *Preprocessing*

*Preprocessing* terbagi menjadi dua yakni *cleaning data* dan normalisasi data. *Cleaning data* ialah proses menghapus data tidak lengkap (*missing value)*. Normalisasi data hanya dilakukan pada fitur yang memiliki rentang yang lebih besar dari fitur lainnya yaitu pada fitur di luar rentang 0 – 1. Berdasarkan struktur *dataset* penyakit kanker prostat pada Tabel 1, maka akan dilakukan normalisasi pada empat fitur pertama. Sedangkan berdasarkan struktur *dataset* penyakit ginjal pada Tabel 2 akan dilakukan normalisasi pada semua fitur kecuali fitur *Red Blood Cell* dan *Hypertension.* Sementara untuk *dataset* penyakit jantung, dilihat dari struktur datanya pada Tabel 3 maka akan dilakukan normalisasi pada semua fitur kecuali fitur *Sex, Fasting Blood Sugar,* dan *Exercise Induced Angina*. Berikut adalah rangkaian *preprocessing* dengan menggunakan beberapa operator yang ada pada RapidMiner.

1) Tanpa Normalisasi



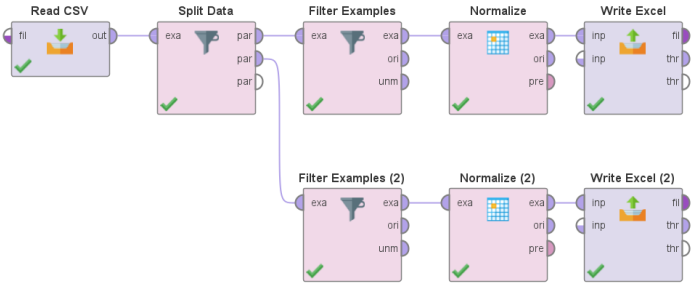

**Gambar 2 *Preprocessing* Tanpa Normalisasi**

Gambar 2 menunjukkan tahap *preprocessing* tanpa menggunakan normalisasi dimulai dengan membaca *dataset* penyakit dalam format *csv* menggunakan operator *Read CSV.* Setelah itu, dilakukan proses pembagian *dataset* dengan memanfaatkan operator *Split Data.* Hasil dari pembagian data tersebut kemudian dibersihkan untuk menghilangkan data yang tidak lengkap (*missing value*) menggunakan operator *Filter Example*. Karena ketiga *dataset* yang digunakan lengkap, maka jumlah data sebelum dan setelah dibersihkan tetap sama. Hasil dari tahap *preprocessing* ini selanjutnya disimpan ke dalam format *excel* menggunakan operator *Write Excel*. Jika *dataset* memiliki skala yang konsisten atau perbedaan rentang nilai antar atribut yang tidak berbeda jauh, tanpa normalisasi dapat menjadi pilihan yang baik (Permana & Salisah, 2022). Data yang digunakan juga tetap dalam bentuk aslinya sehingga tidak ada risiko informasi yang hilang atau terdistorsi.

2) *Min-Max*

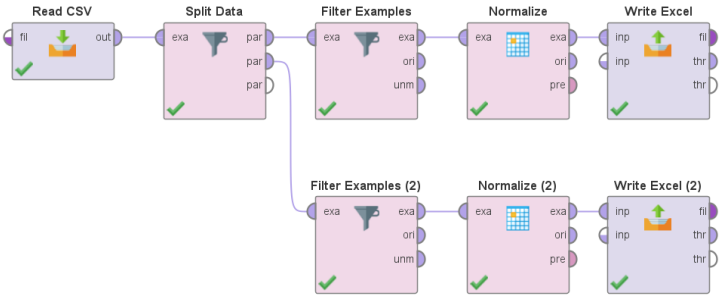
Gambar 3 menunjukkan tahap *preprocessing* menggunakan metode normalisasi *Min-Max* yang mana tahapan awalnya sama dengan *preprocessing* tanpa normalisasi yaitu membaca *dataset*, membagi *dataset*, dan *cleaning dataset*. Setelah menyelesaikan tahap tersebut, dilakukanlah normalisasi data dengan metode *Min-Max* menggunakan operator *Normalize* dan mengatur parameternya yaitu *attribute filter type* menjadi *subset* sehingga dapat memilih fitur yang akan dinormalisasi dengan cara mengisi nama fitur pada bagian *Select Attributes*, dalam hal ini adalah fitur dengan nilai di luar rentang 0 – 1. Selain itu, digunakan pula parameter *method* untuk memilih

metode normalisasi data yaitu *range transformation* untuk *Min-Max*. Hasil dari tahap *preprocessing* ini selanjutnya disimpan ke dalam format *excel* menggunakan operator *Write Excel*. Metode normalisasi *Min-Max* menjadikan semua fitur dalam rentang yang sama, biasanya antara 0 dan 1, sehingga memudahkan komparasi antar fitur (Permana & Salisah, 2022).




**Gambar 3 *Preprocessing* dengan *Min-Max***

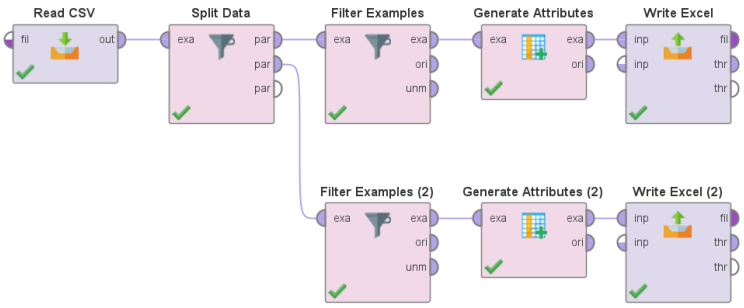
3) *Z-Score*




**Gambar 4 *Preprocessing* dengan *Z-Score***

Gambar 4 menunjukkan tahap *preprocessing* menggunakan metode normalisasi *Z-Score* yang mana tahap awalnya hingga akhir hampir sama dengan metode normalisasi *Min-Max* pada Gambar 3. Perbedaan keduanya terletak pada pengaturan operator *Normalize* pada bagian parameter *method. Min-Max* menggunakan *method range transformation* sedangkan *Z-Score* menggunakan *method Z-transformation*. Metode normalisasi *Z-score* dapat menanggulangi masalah *outlier* dengan cara mengukur berapa banyak standar deviasi suatu data poin dari *mean*.

4) *Decimal Scaling*

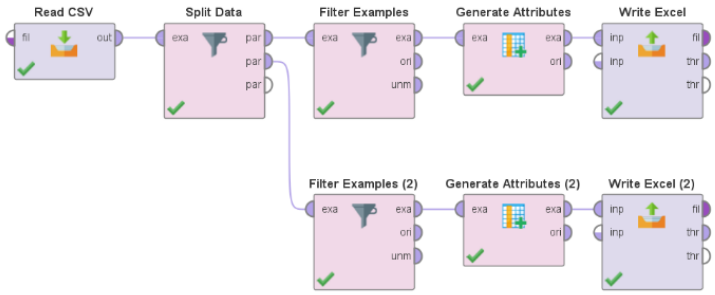



**Gambar 5 *Preprocessing* dengan *Decimal Scaling***

Gambar 5 menunjukkan tahap *preprocessing* menggunakan metode normalisasi *Decimal Scaling* yang mana tahapan awalnya sama dengan *preprocessing* tanpa normalisasi yaitu membaca

*dataset*, membagi *dataset*, dan *cleaning dataset*. Setelah menyelesaikan tahap tersebut, dilakukanlah normalisasi data dengan metode *Decimal Scaling* dengan cara menuliskan rumus dari metode tersebut menggunakan operator *Generate Attributes*. Hasil dari tahap *preprocessing* ini selanjutnya disimpan ke dalam format *excel* menggunakan operator *Write Excel*. *Decimal Scaling* dilakukan dengan membagi nilai data berdasarkan kelipatan 10. Teknik ini cocok untuk data dengan rentang suatu nilai adalah antara 0 dan 1 sedangkan nilai lain pada rentang 0 dan 1000 (Kusnaidi et al., 2022).

5) *MaxAbs*




**Gambar 6 *Preprocessing* dengan *MaxAbs***

Gambar 6 menunjukkan tahap *preprocessing* menggunakan metode normalisasi *MaxAbs* yang mana tahap awalnya hingga akhir hampir sama dengan metode normalisasi *Decimal Scaling* pada Gambar 5. Perbedaan keduanya terletak pada penulisan rumus menggunakan operator *Generate Attributes.* Skala data teknik *MaxAbs* berdasarkan nilai absolut maksimum, mempertahankan *sparsity* pada data *sparse* (Permana & Salisah, 2022). Teknik ini sangat efektif untuk data yang mengandung nilai negatif dan positif dengan skala besar.

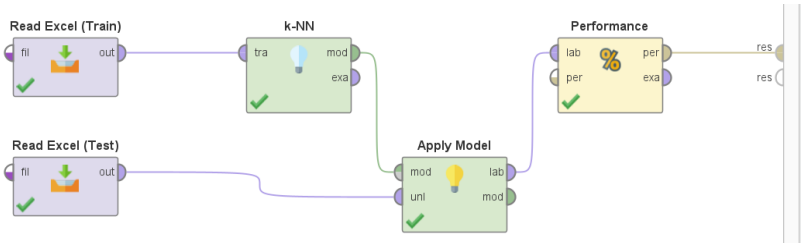
### 3.3 Implementasi K-NN

Pada tahap ini, algoritma K-Nearest Neighbor (K-NN) diimplementasikan berdasarkan hasil *preprocessing* data. Sebelum melanjutkan ke tahap implementasi, dilakukan terlebih dahulu penentuan nilai $k$ yang optimal untuk masing-masing dataset. Penentuan nilai $k$ ini menggunakan rumus yang ditunjukkan pada Pers. (5). Pemilihan nilai $k$ sangat penting karena mempengaruhi performa dan akurasi dari model K-NN dalam proses klasifikasi atau identifikasi yang dilakukan pada dataset yang telah diproses sebelumnya.

$$K_{(Penyakit\ Kanker\ Prostat)} = \sqrt{80} = 8.94 = 9$$

$$K_{(Penyakit\ Ginjal)} = \sqrt{320} = 17.88 = 18$$

$$K_{(Penyakit\ Jantung)} = \sqrt{820} = 28.63 = 29$$



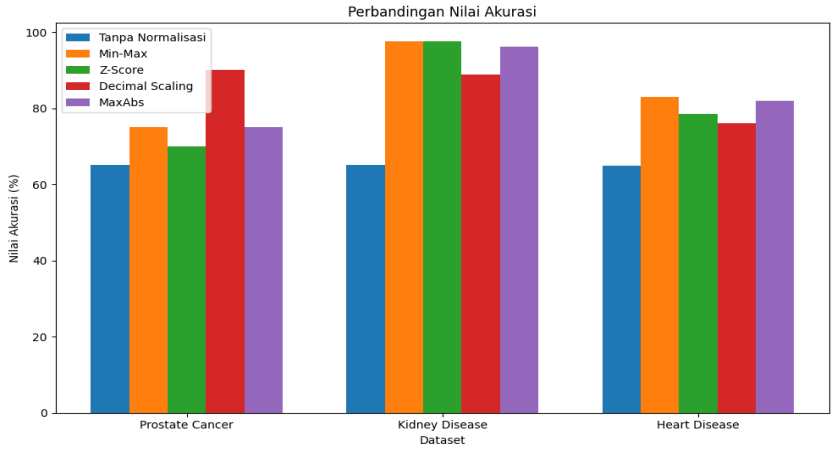

**Gambar 7 Proses Implementasi K-NN**

Gambar 7 menunjukkan proses implementasi metode K-NN yang dimulai dengan membaca data latih dan data uji hasil normalisasi dan tanpa normalisasi dalam format *excel* menggunakan operator *Read Excel.* Setelah itu, diterapkan K-Nearest Neighbor melalui operator K-NN dan mengatur parameternya yaitu *k* untuk menentukan jumlah tetangga terdekat. Penentuan nilai *k* untuk setiap *dataset* tergantung dari jumlah data latihnya, sehingga pada penelitian ini dimanfaatkan suatu rumus yaitu $\sqrt{N}$ . Selain itu, digunakan pula parameter *measure types* untuk memilih metode pengukuran jarak berdasarkan jenis datanya, dalam hal ini digunakan data numerik sehingga dipilih opsi *NumericalMeasures* dan *EuclideanDistance*. Selanjutnya digunakan operator *Apply Model* untuk menerapkan model metode K-NN dari data latih agar melakukan pengklasifikasian pada data uji. Proses klasifikasi dengan metode K-NN ini kemudian di evaluasi untuk mengetahui kinerjanya menggunakan operator *Performance*.

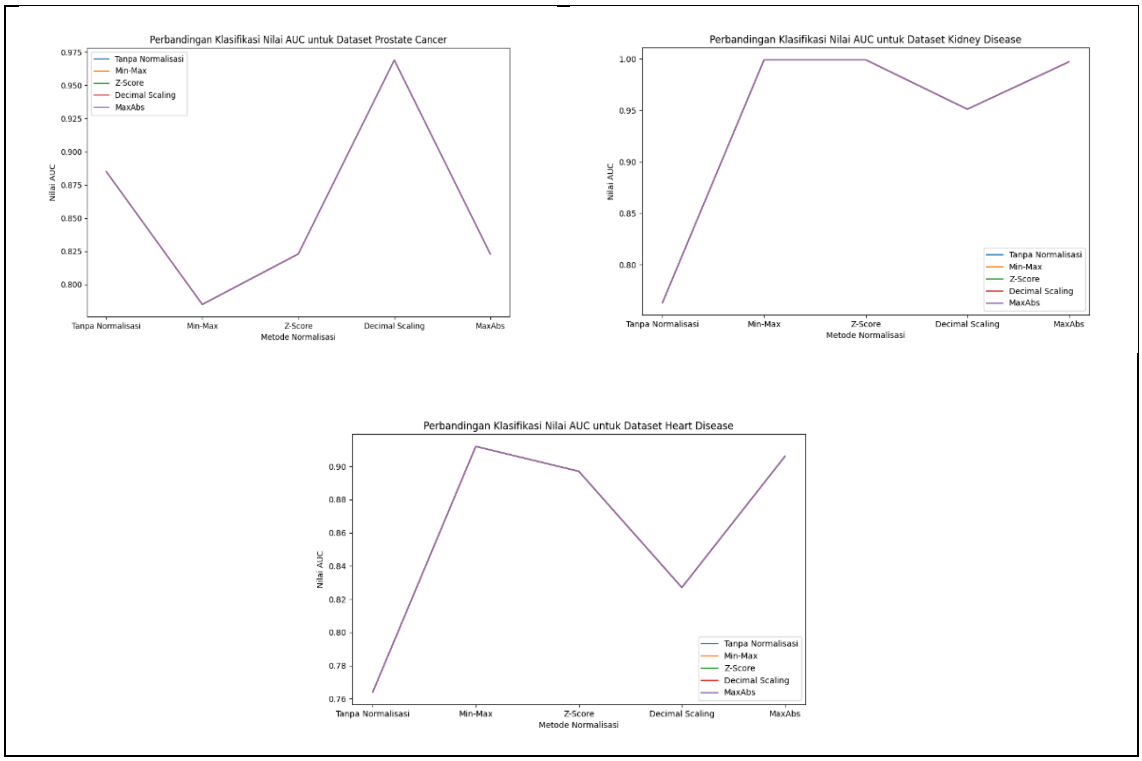
### 3.4 Hasil Analisis

Berdasarkan proses analisis diperoleh hasil perbandingan nilai akurasi dan nilai AUC untuk masing-masing *dataset* yang ditunjukkan melalui grafik Gambar 8 dan Gambar 9. Gambar 8 dan Gambar 9 menunjukkan apabila tidak menggunakan normalisasi data pada ketiga *dataset,* maka akan menghasilkan akurasi yang rendah yaitu ≤ 65%. Sedangkan jika menggunakan normalisasi data, akurasi algoritma K-NN dapat lebih tinggi.

Pada *dataset* penyakit kanker prostat terlihat metode *Decimal Scaling* menghasilkan akurasi tertinggi sebesar 90,00% dan nilai AUC sebesar 0.969. Pada *dataset* penyakit ginjal, metode *Min-Max* dan *Z-Score* menghasilkan akurasi tertinggi sebesar 97,50% dan nilai AUC sebesar 0,999. Metode *MaxAbs* juga menghasilkan akurasi yang tinggi pada *dataset* penyakit ginjal sebesar 96,25% dan nilai AUC sebesar 0,997. Pada *dataset* penyakit jantung, metode *Min-Max* dan *MaxAbs* menghasilkan akurasi yang cukup baik dibanding kedua metode lainnya sebesar 82,93% dan 81,95% serta nilai AUC sebesar 0,912 dan 0,906. Perbedaan penemuan metode normalisasi data terbaik ini disebabkan oleh karakteristik data yaitu, jumlah data, jumlah fitur, dan distribusi data (Pagan et al., 2023). Karakteristik dari ketiga *dataset* tersebut ditunjukkan pada Tabel 7.




**Gambar 8 Perbandingan Nilai Akurasi**




**Gambar 9 Perbandingan Nilai AUC**

**Tabel 7 Karakteristik *Dataset***

| *Dataset* | Total Data | Jumlah Fitur | Distribusi Data |
|---|---|---|---|
| Penyakit Kanker Prostat | 100 | 9 | Normal |
| Penyakit Ginjal | 400 | 14 | Tidak Normal |
| Penyakit Jantung | 1025 | 14 | Tidak Normal |

Berdasarkan nilai akurasi pada Gambar 8, nilai AUC pada Gambar 9, dan karakteristik data pada Tabel 7, maka diperoleh beberapa pengetahuan antara lain: 1) *Decimal Scaling* cenderung cocok dengan karakteristik *dataset* yang memiliki jumlah data yang kecil, jumlah fitur yang sedikit dengan rentang nilai yang kecil, dan data yang berdistribusi normal. 2) *Min-Max* cenderung cocok dengan karakteristik *dataset* yang memiliki jumlah data yang besar, jumlah fitur yang banyak, dan data yang tidak mengikuti distribusi normal. Metode *Min-Max* mereskal nilai-nilai fitur ke dalam rentang yang telah ditentukan (biasanya antara 0 dan 1) dengan mempertahankan bentuk distribusi data. Ini berarti *Min-Max* tidak memerlukan asumsi tentang distribusi data. Kecocokan antara *Min-Max* dengan karakteristik tersebut didukung oleh penelitian sebelumnya yaitu Henderi et al. (2021) dan Sholeh et al. (2022) yang juga menghasilkan *Min-Max* sebagai metode terbaik dengan karakteristik tersebut. 3) Sama seperti *Min-Max*, *MaxAbs* juga cenderung cocok dengan karakteristik data yang memiliki jumlah data yang besar, jumlah fitur yang banyak, dan data yang tidak mengikuti distribusi normal. Namun, jika dibandingkan dengan metode *Min-Max* yang mampu menghasilkan akurasi 97,50% pada *dataset* penyakit ginjal dan 82,93% pada *dataset* penyakit jantung, metode *MaxAbs* justru hanya mampu menghasilkan akurasi 96,25% pada *dataset* penyakit ginjal dan 81,95% pada *dataset* penyakit jantung. Hal ini menunjukkan bahwa performa *MaxAbs* tidak sebaik *Min-Max*. 4) *Z-Score* biasanya cenderung cocok untuk *dataset* yang memiliki jumlah fitur yang relatif besar atau kecil, dan cocok untuk data yang berdistribusi normal atau mendekati distribusi normal (McLeod, 2023). Kecocokan metode *Z-Score* dengan karakteristik tersebut didukung oleh penelitian sebelumnya (Badugu, 2020) yang juga menghasilkan *Z-Score* sebagai metode terbaik. Jika meninjau lebih jauh distribusi datanya, *Z-Score* juga cenderung cocok pada distribusi yang tidak terlalu jauh dari normalitas, di mana pada penelitian ini ditemukan bahwa *dataset* penyakit ginjal memiliki nilai statistik uji yang tidak terlalu

jauh dari nilai tabel Kolmogorov-Smirnov. Hal tersebut membuktikan bahwa *Z-Score* juga dapat diterapkan pada *dataset* yang tidak berdistribusi normal, terutama jika *dataset* tersebut tidak memiliki *outlier* yang signifikan dan distribusinya tidak terlalu jauh dari normalitas (Indeed Editorial Team, 2024).

## 4. KESIMPULAN

Penelitian ini membandingkan empat metode normalisasi data (*Min-Max, Z-Score, Decimal Scaling,* dan *MaxAbs*) untuk meningkatkan akurasi klasifikasi penyakit menggunakan algoritma K-NN. Hasil penelitian mengindikasikan bahwa kinerja setiap metode dipengaruhi oleh karakteristik unik dari setiap *dataset*, seperti jumlah data, jumlah fitur, dan distribusi data. Metode *Decimal Scaling* memberikan akurasi tertinggi sebesar 90,00% pada *dataset* kanker prostat, sementara *Min-Max* dan *Z-Score* masing-masing mencapai akurasi 97,50% pada *dataset* penyakit ginjal. Metode *MaxAbs* juga menunjukkan performa yang baik dengan akurasi 96,25% pada *dataset* yang sama, sedangkan pada *dataset* penyakit jantung, *Min-Max* dan *MaxAbs* mencapai akurasi masing-masing sebesar 82,93% dan 81,95%.

Berdasarkan hasil penelitian ini, disarankan agar pemilihan metode normalisasi dilakukan secara cermat dengan mempertimbangkan karakteristik spesifik dari *dataset* yang akan dianalisis. Untuk *dataset* yang lebih kecil dan sederhana, *Decimal Scaling* dapat menjadi pilihan yang baik. Namun, untuk *dataset* yang lebih besar dan kompleks, *Min-Max* atau *MaxAbs* umumnya lebih disarankan. Metode *Z-Score* dapat menjadi opsi yang fleksibel dan dapat diterapkan pada berbagai jenis *dataset*, terutama jika data tersebut tidak memiliki *outlier* yang signifikan.

Penelitian ini memberikan kontribusi penting dalam pemahaman tentang pengaruh normalisasi data terhadap kinerja algoritma K-NN dalam konteks klasifikasi penyakit. Hasil penelitian ini dapat menjadi acuan bagi peneliti dan praktisi *data mining* untuk meninjau karakteristik *dataset* secara mendalam sebelum memilih metode normalisasi data. Pendekatan yang tepat dapat meningkatkan akurasi prediksi algoritma klasifikasi secara signifikan, yang pada akhirnya dapat menghasilkan keputusan yang lebih baik dalam analisis data kesehatan. Selain itu, penelitian lebih lanjut diperlukan untuk menguji efektivitas metode normalisasi lainnya atau mengombinasikan beberapa metode guna mencapai hasil yang lebih optimal.

## DAFTAR PUSTAKA

Ambarwari, A., Jafar Adrian, Q., & Herdiyeni, Y. (2020). Analysis of the Effect of Data Scaling on the Performance of the Machine Learning Algorithm for Plant Identification. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, *4*(1), 117–122. https://doi.org/10.29207/resti.v4i1.1517

Badugu, S. (2020). Prediction of Heart Problems for Diabetic Patients using Classification Algorithms. *Journal of Advanced Research in Dynamic and Control Systems*, *Volume 12*(02-Special Issue), 904–913. https://doi.org/10.5373/JARDCS/V12SP2/SP20201148

Barus, F. M., & Sutarman, S. (2023). Mendeteksi Outlier pada Data Multivariat dengan Metode Jarak Mahalanobis-Minimum Covariance Determinant (MMCD). *IJM: Indonesian Journal of Multidisciplinary*, *1*(3), 1164–1172. https://journal.csspublishing.com/index.php/ijm/article/view/287

Cahyanti, D., Rahmayani, A., & Husniar, S. A. (2020). Analisis performa metode Knn pada Dataset pasien pengidap Kanker Payudara. *Indonesian Journal of Data and Science*, *1*(2), 39–43. https://doi.org/10.33096/ijodas.v1i2.13

Chandra, R., Chaudhary, K., & Kumar, A. (2022). Comparison of Data Normalization for Wine Classification Using K-NN Algorithm. *IJIIS: International Journal of Informatics and Information Systems*, *5*(4), 175–180. https://doi.org/10.47738/ijiis.v5i4.145

Henderi, H., Wahyuningsih, T., & Rahwanto, E. (2021). Comparison of Min-Max normalization and Z-Score Normalization in the K-nearest neighbor (kNN) Algorithm to Test the Accuracy of Types of Breast Cancer. *IJIIS: International Journal of Informatics and Information Systems*, *4*(1), 13–20. https://doi.org/10.47738/ijiis.v4i1.73

HS, H., Azmi, N., Hazriani, H., & Yuyun, Y. (2023). Klasifikasi Status Gizi Balita Menggunakan Algoritma K-Nearest Neighbor (KNN) | Prosiding SISFOTEK. *Prosiding SISFOTEK*, *7*(1), 313–318. https://seminar.iaii.or.id/index.php/SISFOTEK/article/view/396

Indeed Editorial Team. (2024, August 16). *Normalization Formula: How To Use It on a Data Set | Indeed.com*. Indeed. https://www.indeed.com/career-advice/career-development/normalization-formula

Jain, A. K., Duin, P. W., & Mao, J. (2000). Statistical pattern recognition: a review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *22*(1), 4–37. https://doi.org/10.1109/34.824819

Kusnaidi, M. R., Gulo, T., & Aripin, S. (2022). Penerapan Normalisasi Data Dalam Mengelompokkan Data Mahasiswa Dengan Menggunakan Metode K-Means Untuk Menentukan Prioritas Bantuan Uang Kuliah Tunggal. *Journal of Computer System and Informatics (JoSYC)*, *3*(4), 330–338. https://doi.org/10.47065/josyc.v3i4.2112

Marlina, D., & Bakri, M. (2021). Penerapan Data Mining untuk Memprediksi Transaksi Nasabah dengan Algoritma C4.5. *Jurnal Teknologi Dan Sistem Informasi*, *2*(1), 23–28. https://doi.org/10.33365/JTSI.V2I1.627

McLeod, S. (2023, October 6). *Z-Score: Definition, Formula, Calculation & Interpretation*. Simply Psychology. https://www.simplypsychology.org/z-score.html

Pagan, M., Zarlis, M., & Candra, A. (2023). Investigating the impact of data scaling on the k-nearest neighbor algorithm. *Computer Science and Information Technologies*, *4*(2), 135–142. https://doi.org/10.11591/csit.v4i2.p135-142

Permana, I., & Salisah, F. N. S. (2022). Pengaruh Normalisasi Data Terhadap Performa Hasil Klasifikasi Algoritma Backpropagation. *Indonesian Journal of Informatic Research and Software Engineering (IJIRSE)*, *2*(1), 67–72. https://doi.org/10.57152/ijirse.v2i1.311

Riaz, M., Bashir, M., & Younas, I. (2022). Metaheuristics based COVID-19 detection using medical images: A review. *Computers in Biology and Medicine*, *144*, 105344. https://doi.org/10.1016/j.compbiomed.2022.105344

Sholeh, M., Andayati, D., & Rachmawati, Rr. Y. (2022). Data Mining Model Klasifikasi Menggunakan Algoritma K-Nearest Neighbor dengan Normalisasi untuk Prediksi Penyakit Diabetes. *TeIKa*, *12*(02), 77–87. https://doi.org/10.36342/teika.v12i02.2911

Singh, D., & Singh, B. (2020). Investigating the impact of data normalization on classification performance. *Applied Soft Computing*, *97*, 105524. https://doi.org/10.1016/j.asoc.2019.105524

Whendasmoro, R. G., & Joseph, J. (2022). Analisis Penerapan Normalisasi Data Dengan Menggunakan Z-Score Pada Kinerja Algoritma K-NN. *JURIKOM (Jurnal Riset Komputer)*, *9*(4), 872. https://doi.org/10.30865/jurikom.v9i4.4526

**JISKA (Jurnal Informatika Sunan Kalijaga)**
**Vol. 9, No. 3, SEPTEMBER, 2024, Pp. 192 – 204**
**ISSN: 2527 – 5836 (print) | 2528 – 0074 (online)**

# Implementasi *Data Augmentation* untuk Klasifikasi Sampah Organik dan Non Organik Menggunakan Inception-V3

**Mochamad Rahina Bintang Pambayun [(1)]*, Yufis Azhar [(2)]**
Informatika, Fakultas Teknik, Universitas Muhammadiyah Malang, Malang, Indonesia
e-mail : rahinabintang@gmail.com, yufis@umm.ac.id.
* Penulis korespondensi.
Artikel ini diajukan 25 Maret 2024, direvisi 28 Juni 2024, diterima 22 Juli 2024, dan dipublikasikan 25 September 2024.

***Abstract***

*The surge in global waste, particularly in Indonesia, with a total of 36.218 million tons per year, has become an urgent issue. Challenges in waste management are increasingly complex due to the lack of public understanding and awareness in classifying types of waste. One systemic approach to address waste classification issues involves the use of machine learning technology to categorize waste into two main types: organic and non-organic. The data used in this study comes from a Kaggle website dataset comprising 25,500 entries. This research employs a transfer learning approach with the Inception-V3 architecture and data augmentation implementation. Transfer learning is chosen for its proven performance in image data classification, while data augmentation is implemented to introduce diversity to the dataset. The research stages include business understanding, data preprocessing, data augmentation, data modelling, and evaluation. The results show that the use of transfer learning with the Inception-V3 approach and data augmentation implementation achieves an accuracy rate of 94%, which falls into the excellent category.*

***Keywords: Classification, Transfer Learning, Convolutional Neural Network, Inception-V3, Garbage***

**Abstrak**

Lonjakan jumlah sampah global, terutama di Indonesia dengan total 36,218 juta ton per tahun, menjadi permasalahan mendesak. Tantangan dalam manajemen sampah semakin kompleks akibat minimnya pemahaman dan kesadaran masyarakat dalam pengelompokkan jenis sampah. Salah satu pendekatan sistem yang dapat diimplementasikan untuk mengatasi permasalahan pengelompokkan sampah adalah penggunakan teknologi *machine learning* untuk mengklasifikasikan sampah menjadi 2 kategori utama, yaitu organik dan non-organik. Data yang digunakan dalam penelitian ini adalah *dataset* sampah yang berasal dari *website* Kaggle dengan jumlah data sebanyak 25.500 data. Penelitian ini menggunakan pendekatan *transfer learning* dengan arsitektur Inception-V3 dan implementasi *data augmentation*. Pendekatan *transfer learning* dipilih karena perfomanya yang terbukti sangat baik dalam klasifikasi data citra, sedangkan *data augmentation* diimplementasikan untuk menambah variasi atau keragaman data. Tahapan penelitian ini meliputi pemahaman bisnis, *preprocessing* data, augmentasi data, *modelling* data, dan evaluasi. Hasil dari penelitian menunjukkan bahwa penggunaan pendekatan *transfer learning* Inception-V3 dan implementasi *data augmentation* menghasilkan tingkat akurasi sebesar 94% yang termasuk dalam kategori yang sangat baik.

**Kata Kunci: Klasifikasi, Transfer Pembelajaran, Jaringan Saraf Konvolusional, Inception-V3, Sampah**

## 1. PENDAHULUAN

Peningkatan jumlah sampah dari tahun ke tahun menandai sebuah isu global yang mendesak. Fenomena ini melibatkan pertumbuhan yang cepat dari limbah yang dihasilkan oleh aktivitas manusia yang tentu menciptakan tantangan signifikan dalam manajemen sampah di seluruh dunia (Lebreton & Andrady, 2019). Menurut Sistem Informasi Penanggulangan Sampah Nasional (SIPSN) tahun 2022, total sampah di Indonesia mencapai 36,218 juta ton per tahun, dengan penurunan 14,88% dari tahun sebelumnya. Namun, hanya 64,01% atau sekitar 23,18 juta ton

yang dapat dikelola secara efektif (Kementerian Lingkungan Hidup dan Kehutanan, 2022). Situasi ini mengindikasikan perlu adanya pendekatan yang lebih komprehensif untuk mengatasi permasalahan sampah di Indonesia. Ada dua jenis sampah, sampah non organik dan sampah organik, tergantung dari sifatnya. Sampah organik berasal dari sisa makhluk hidup seperti hewan, manusia, dan tumbuhan yang mengalami pembusukan atau pelapukan. Jenis sampah ini bersifat ramah lingkungan karena dapat diuraikan oleh bakteri dalam waktu yang lama dan berlangsung dengan cepat. Sementara itu, sampah non-organik (anorganik) berasal dari sisa manusia yang sulit diurai oleh bakteri dan membutuhkan waktu yang cukup lama bahkan hingga ratusan tahun untuk mengalami dekomposisi (Fadillah et al., 2019).

Saat ini, di lingkungan sekitar kita sering ditemukan kondisi sampah yang cenderung tercampur dan tidak terpisah (Widodo & Suleman, 2020). Permasalahan ini muncul karena kurangnya pemahaman dan kesadaran masyarakat dalam mengelola pembuangan sampah sesuai dengan jenisnya. Pengelolaan yang optimal, terutama dalam mengelompokkan sampah menjadi kategori organik dan non organik, menjadi hal yang penting untuk mencegah timbulnya bau yang tidak menyenangkan dan potensi penyebaran penyakit (Rosiana & Perdana, 2022). Adanya implementasi suatu sistem deteksi jenis sampah secara otomatis menjadi salah satu solusi untuk mengatasi masalah pengelompokkan sampah di Indonesia. Dengan adanya sistem ini, diharapkan dapat memfasilitasi pengelolaan sampah yang lebih efisien, khususnya untuk proses daur ulang dan pemanfaatan kembali.

Salah satu pendekatan sistem yang dapat diimplementasikan untuk mengatasi permasalahan terkait pengelompokan sampah adalah dengan memanfaatkan teknologi *machine learning*. Dengan menggunakan teknologi ini, kita dapat mengklasifikasikan sampah menjadi dua kategori utama, yaitu organik dan non-organik, dengan tingkat akurasi yang tinggi (Kartiko et al., 2022). Pendekatan ini berpotensi memberikan kontribusi yang signifikan terhadap pengelolaan sampah yang berkelanjutan dan efektif. Sebelumnya, Abdurrahman dan rekan-rekannya telah melakukan penelitian dengan mengaplikasikan Convolutional Neural Network (CNN), salah satu arsitektur *Deep Learning*, untuk mengklasifikasikan sampah organik dan non-organik. Meskipun hasil penelitian tersebut mencapai tingkat akurasi sebesar 89,65%, dari eksperimen yang telah dilakukan mengungkapkan kelemahan model dalam mengeneralisasi sampah non-organik. Hal ini dapat disebabkan oleh ketidakseimbangan kelas dan kurangnya data sampah non-organik pada *dataset* yang digunakan (Ibnul Rasidi et al., 2022). Untuk mengatasi kendala tersebut, penelitian ini akan melibatkan implementasi *data augmentation* dan pendekatan *transfer learning* menggunakan arsitektur model Inception-V3. Pemilihan model Inception-V3 didasarkan pada kemampuannya dalam menangani kompleksitas visual dan kemampuan generalisasi yang lebih baik.

Pendekatan *transfer learning* memungkinkan kita untuk memanfaatkan pengetahuan yang telah diperoleh oleh suatu model dalam menangani tugas tertentu dan mengaplikasikannya pada tugas terkait (Wang et al., 2019). Pemilihan arsitektur Inception-V3 sebagai model *pre-trained* dalam penelitian ini memberikan keuntungan dibanding model CNN karena kemampuannya yang sudah teruji dalam menangani tugas klasifikasi gambar yang kompleks (Minarno, Aripa, et al., 2023). Penelitian sebelumnya tentang *Image Retrieval* yang dilakukan oleh Agus Eko Minarno dan rekan-rekannya juga menunjukkan bahwa Inception-V3 memiliki performa yang sangat baik dalam menangani gambar yang kompleks (Minarno, Hasanuddin, et al., 2023). Model *pre-trained* Inception-V3 tidak hanya dapat mengenali pola dan fitur kompleks dalam data visual, tetapi juga telah dilatih pada dataset yang luas dan beragam, mencakup berbagai kategori gambar (Ahmed et al., 2023). Dengan mentransfer pengetahuan ini ke tugas klasifikasi sampah, diharapkan model dapat dengan cepat dan akurat mengidentifikasi perbedaan antara sampah organik dan non-organik.

Selain *transfer learning*, implementasi *data augmentation* turut diterapkan untuk meningkatkan keberagaman dan jumlah sampel dalam *dataset* pelatihan (Lin et al., 2018). Peningkatan variasi ini diharapkan dapat membantu model untuk mengatasi ketidakseimbangan kelas pada *dataset*

yang digunakan pada penelitian sebelumnya, serta membuat model dapat lebih adaptif terhadap variasi yang mungkin ditemui di lingkungan dunia nyata.

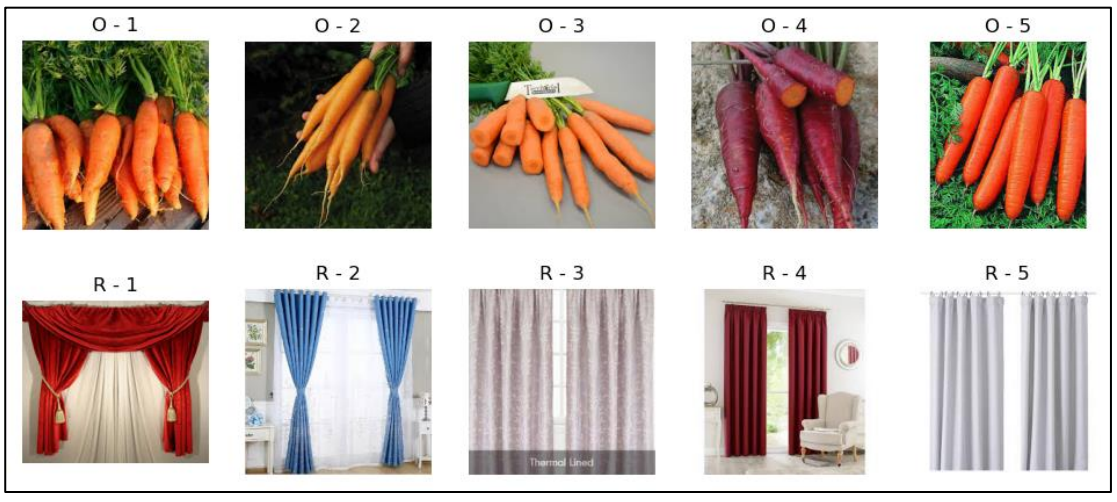
Penelitian ini berupaya untuk mengatasi kelemahan dalam penelitian sebelumnya yang dilakukan oleh Abdurrahman dan rekan-rekannya, yang mungkin mengalami masalah dengan ketidakseimbangan kelas dan generalisasi model (Ibnul Rasidi et al., 2022). Dengan memperkenalkan kombinasi *transfer learning* dan *data augmentation*, diharapkan model yang dihasilkan akan memiliki akurasi dan kemampuan generalisasi yang lebih tinggi, serta mampu mengklasifikasikan sampah organik dan non-organik dengan lebih efektif. Selain itu, penelitian ini juga memberikan kontribusi dalam bidang pengelolaan sampah dengan menyediakan solusi teknologi yang lebih canggih untuk pengelompokan sampah, yang pada gilirannya dapat membantu dalam upaya pengelolaan lingkungan yang lebih baik dan berkelanjutan.

## 2. METODE PENELITIAN

Penelitian ini mengadopsi pendekatan metodologi CRISP-DM (*Cross Industry Standard Process for Data Mining*). CRISP-DM adalah suatu kerangka kerja (*framework*) yang dikenal secara luas dalam dunia *data mining*. *Framework* ini dirancang untuk memberikan struktur yang terorganisir dalam mengatasi tahapan-tahapan utama dalam proses penambangan data. Tahapan penelitian ini mencakup serangkaian langkah yang sistematis, dimulai dari pemahaman bisnis, pemahaman data, persiapan data, pemodelan, evaluasi, hingga penerapan (Hidayati et al., 2021). Dengan menerapkan metodologi CRISP-DM, penelitian ini dapat menjalankan proses penelitian dengan pendekatan yang terstruktur dan terorganisir. Selain itu, penerapan metodologi ini dapat memastikan bahwa setiap tahapan dijalankan secara efisien dan memberikan kontribusi yang signifikan terhadap pemahaman dan solusi terhadap permasalahan klasifikasi sampah organik dan non-organik.

### 2.1 *Dataset*

*Dataset* yang digunakan dalam penelitian ini sama dengan yang digunakan pada penelitian sebelumnya oleh Abdurrahman dan rekan-rekannya (Ibnul Rasidi et al., 2022). *Dataset* ini terdiri dari citra dalam format .jpg yang diperoleh dari *website* Kaggle dengan judul *dataset* “Waste Classification data” (Sekar, 2019). Jumlah total citra yang terkumpul mencapai 25.077 *instance*, yang telah dibagi menjadi dua bagian. Bagian pertama adalah data pelatihan (*train*), yang terdiri dari 12.565 citra untuk kelas organik dan 9.999 citra untuk kelas non-organik (*recycle*), sehingga total citra pada data pelatihan mencapai 22.564 (90%) dari keseluruhan *dataset*. Sementara itu, data pengujian (*test*) terdiri dari 1.401 citra untuk kelas organik dan 1.112 citra untuk kelas non-organik, dengan total 2.513 citra (10%) dari keseluruhan *dataset*.

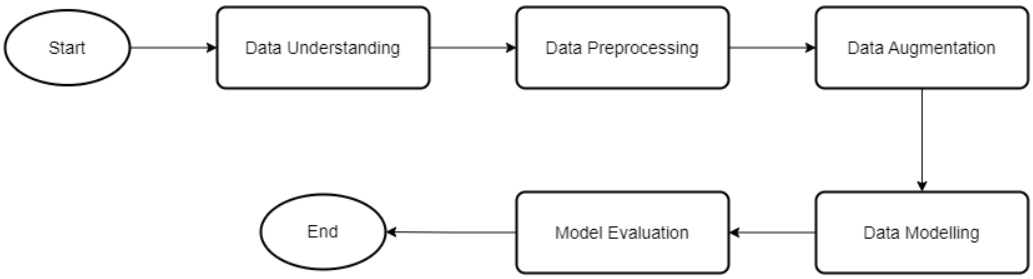



**Gambar 1 Sampel Data**

Untuk mempersingkat proses pelatihan dan mengimbangkan data antara kedua kelas (organik dan non-organik), diputuskan untuk menggunakan sampel dari masing-masing kelas sebanyak 5.000 data citra. Sampel ini diambil dengan tetap mempertahankan representasi proporsional dari data asli. Dengan demikian, total data yang akan digunakan dalam penelitian ini adalah 10.000 citra untuk pelatihan dan 2.513 citra untuk pengujian, yang menghasilkan total citra sebanyak 12.513. Gambar 1 berikut merupakan sampel dari *dataset* yang telah didapatkan. Label 'O' mewakili kelas Organik dan label 'R' mewakili kelas Recycle atau Non Organik.

## 2.2 Tahapan Penelitian

Tahapan penelitian ini berdasar pada metodologi CRISP-DM, di mana Langkah-langkah yang dilakukan di antaranya adalah mempelajari *dataset*, melakukan serangkaian *preprocessing* seperti *data cleaning*, *resizing* dan *normalization*, hingga menerapkan augmentasi untuk memperbanyak variasi data. Gambar 2 berikut merupakan tahapan yang dilakukan dalam pembuatan model klasifikasi sampah.

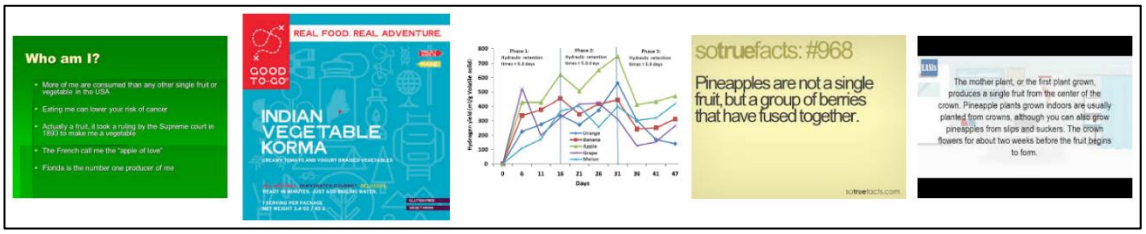



**Gambar 2 Tahapan Penelitian**

### 2.2.1 *Business Understanding*

*Business understanding* adalah tahap awal dalam metodologi CRISP-DM (*Cross Industry Standard Process for Data Mining*) yang bertujuan untuk memahami secara mendalam tantangan bisnis yang dihadapi. Pada tahap ini, fokusnya adalah merumuskan pertanyaan-pertanyaan yang relevan agar dapat diterjemahkan ke dalam tugas analisis data. Dengan pemahaman yang jelas tentang tujuan bisnis dan masalah yang ingin dipecahkan, tim data dapat memastikan bahwa pendekatan analisis data yang dilakukan selaras dengan kebutuhan bisnis (Schröer et al., 2021).

### 2.2.2 *Data Preprocessing*

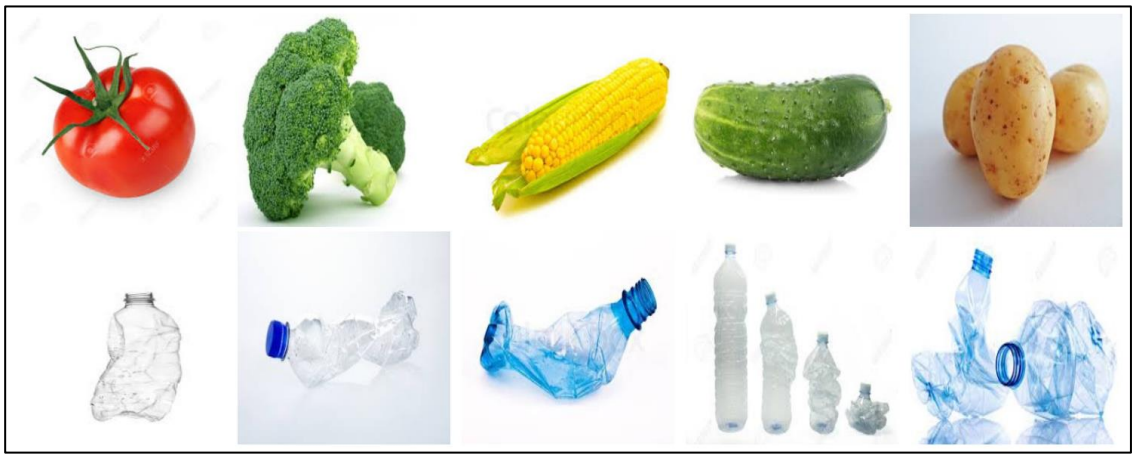



**Gambar 3 Beberapa Sampel Data yang Tidak Relevan**

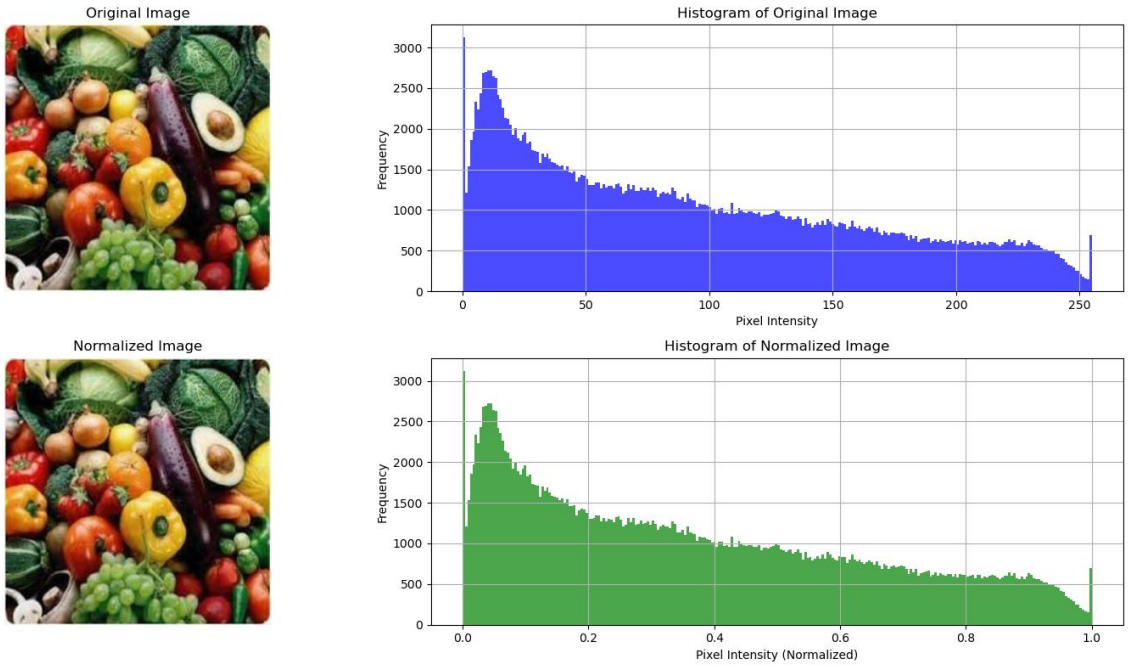
Pra-pemrosesan data memiliki peran yang sangat penting dalam pengembangan model *machine learning* dan membutuhkan perhatian yang seksama. Tujuan utama dari proses ini adalah untuk membersihkan, mengatur, dan menyelaraskan data agar sesuai dengan kebutuhan model yang sedang dikembangkan. Dalam tahap ini, terdapat tiga langkah penting yang umumnya diterapkan: pembersihan data (*data cleaning*), penyesuaian ukuran (*resizing*), dan normalisasi (*normalization*). Pembersihan data dilakukan untuk memastikan bahwa data pelatihan valid, konsisten, dan siap untuk diproses lebih lanjut. Selama proses ini dilakukan, ditemukan berbagai ketidaksesuaian data gambar, seperti ambiguitas antara label organik dan non-organik, kasus

data yang salah label (di mana data organik disalah label sebagai non-organik, dan sebaliknya), hingga adanya data yang tidak relevan seperti terlihat pada Gambar 3. Penanganan masalah-masalah ini sangat penting untuk memastikan integritas dan akurasi hasil pelatihan model.

Langkah selanjutnya adalah *resizing*, di mana dimensi gambar diubah menjadi ukuran yang seragam, yaitu 299 x 299 piksel. Langkah ini tidak hanya meningkatkan efisiensi komputasi dengan memperbolehkan model seperti Inception-V3 untuk menangani input secara lebih terstruktur, tetapi juga mengoptimalkan penggunaan memori sistem (Luke et al., 2019). Ukuran 299 x 299 piksel dipilih berdasarkan rekomendasi dari model Inception-V3 yang digunakan (Fan et al., 2023). Hal ini dilakukan untuk memastikan kesesuaian dengan persyaratan teknis yang diperlukan (*InceptionV3*, 2015). Praktik ini secara signifikan dapat meminimalkan beban komputasi saat memproses *dataset* gambar yang mungkin besar, sekaligus meningkatkan kecepatan dan efisiensi keseluruhan sistem. Gambar 4 menunjukkan sampel citra setelah dilakukan proses *resizing*.



**Gambar 4 Sampel Data Setelah Proses *Resizing***




**Gambar 5 Histogram dari Sampel Data Sebelum dan Sesudah Proses Normalisasi**

Setelah proses r*esizing* dilakukan, langkah selanjutnya adalah melakukan normalisasi pada gambar. Normalisasi pada gambar dilakukan untuk meningkatkan kecepatan konvergensi, akurasi, dan stabilitas model dengan mengubah jangkauan dan distribusi nilai piksel pada gambar (Singh & Singh, 2020). Metode normalisasi yang digunakan adalah metode empiris (*empirical method*) dengan cara membagi setiap citra dengan nilai 255. Pendekatan ini secara efektif menyesuaikan intensitas piksel ke dalam kisaran 0 hingga 1 tanpa memengaruhi distribusi intensitas atau aspek visual (Pei et al., 2023). Gambar 5 menunjukkan perbandingan histogram antara data asli dan data yang telah dinormalisasi.

Dengan mengombinasikan proses *data cleaning, resizing, dan normalization*, *dataset* gambar dapat diproses untuk memastikan keseragaman dan kualitas yang optimal sebelum digunakan untuk melatih model klasifikasi sampah organik dan non-organik. Langkah-langkah ini menjadi landasan penting dalam memastikan bahwa data yang diberikan kepada model bersih, konsisten, dan siap untuk mempelajari pola yang relevan selama proses pelatihan.

### 2.2.3 *Data Augmentation*

*Data augmentation* adalah teknik yang digunakan untuk meningkatkan variasi dalam *dataset* pelatihan dengan membuat modifikasi terkontrol pada data yang ada (Mikolajczyk & Grochowski, 2018). Tujuan dari augmentasi ini adalah untuk memperkaya variasi dalam *dataset* pelatihan, sehingga model dapat menggeneralisasi pola dan fitur dengan lebih baik (Shorten & Khoshgoftaar, 2019).Beberapa metode augmentasi yang umum digunakan untuk mengatur variasi dalam *dataset* pelatihan antara lain adalah *rotation, gamma adjustment, noise injection, dan affine transform* (Zhang et al., 2019). Teknik *rotation* mengubah orientasi citra dalam sudut tertentu seperti 90 derajat atau 180 derajat. Hal ini membantu model memahami variasi posisi objek dalam data. *Gamma adjustment* mengubah kecerahan gambar dengan mengubah nilai *gamma* dalam transformasi *gamma*. Ini memungkinkan model untuk menyesuaikan variasi pencahayaan dalam data. *Noise injection* mengacaukan citra dengan menambahkan *noise* seperti Gaussian *noise*. Pemberian *noise* pada citra dapat meningkatkan toleransi model terhadap variasi *noise* di lingkungan nyata. *Affine transform* mencakup *scaling, translasi, dan shear*, yang dapat memperluas variasi posisi dan skala objek dalam citra untuk meningkatkan kemampuan model dalam mengenali variasi geometris (Vallez et al., 2022). Sampel data citra yang telah di augmentasi dapat dilihat pada Gambar 6.

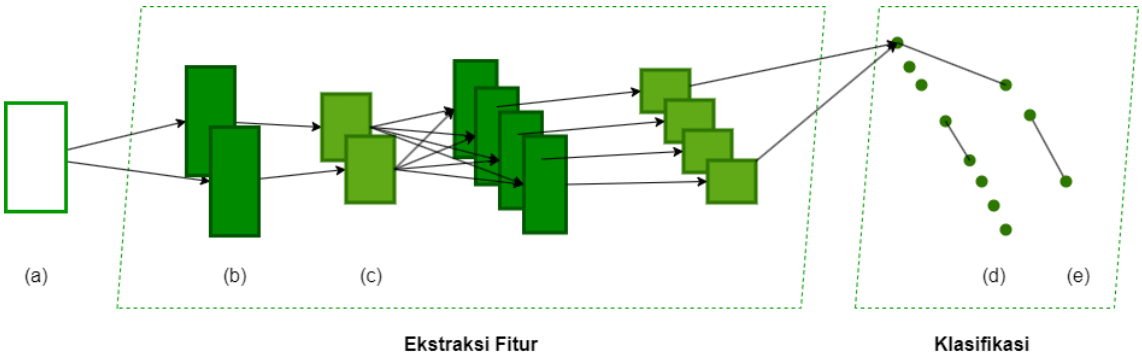
**Gambar 6 Sampel Data Setelah Proses Augmentasi**

Dengan menerapkan teknik-teknik augmentasi data yang telah disebutkan di atas, sebanyak 40.000 citra baru dihasilkan untuk data pelatihan, sehingga total data pelatihan menjadi 50.000 citra. Proses augmentasi ini diharapkan dapat memberikan variasi yang diperlukan untuk melatih model agar lebih efektif dalam mengenali dan menggeneralisasi pola pada sampah organik dan non-organik. Dengan peningkatan jumlah dan keragaman data pelatihan, model diharapkan

mampu menangkap berbagai karakteristik dan variasi sampah yang mungkin ditemui di dunia nyata, sehingga meningkatkan akurasi dan kemampuan generalisasi model secara keseluruhan.

### 2.2.4 *Data Modeling*

Dalam tahap ini, pemodelan dilakukan dengan menerapkan salah satu arsitektur *deep neural network* yang telah terbukti efektif dalam klasifikasi gambar, yaitu Inception-V3. Inception-V3 merupakan model arsitektur Convolutional Neural Network (CNN) yang dikembangkan oleh Google untuk tugas klasifikasi gambar dan telah dilatih pada *dataset* ImageNet yang besar dan beragam (Lin et al., 2018). Kemampuan Inception-V3 untuk mengekstrak fitur dan pola kompleks dari data visual (*image*) menjadi alasan utama dari pemilihan model ini. Representasi diagram cara kerja Inception-V3 ditunjukkan pada Gambar 7. Pada diagram tersebut, (a) menunjukkan *layer* input atau citra, (b) menunjukkan lapisan konvolusi, (c) menunjukkan lapisan *subsampling*, dan (d) serta (e) menunjukkan *layer output* yang terhubung penuh.



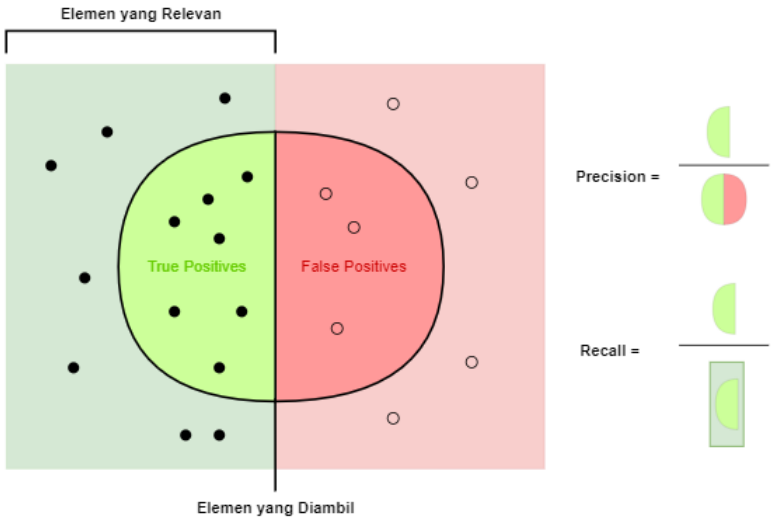

**Gambar 7 Diagram Cara Kerja Inception-V3**

Proses pemodelan dimulai dengan memuat model Inception-V3 yang telah dilatih sebelumnya menggunakan bobot yang telah disesuaikan dengan *dataset* ImageNet. Model ini telah diakui memiliki kemampuan yang handal untuk menangani tugas-tugas klasifikasi gambar yang kompleks, dan pengetahuan yang terkandung dalam model tersebut diadopsi untuk mendukung tugas klasifikasi sampah organik dan non-organik. Setelah model dimuat, dilakukan pengaturan tambahan pada arsitektur, termasuk penggunaan *Global Average Pooling Layer* untuk meratakan *output* dari lapisan sebelumnya. Hal ini dapat menciptakan representasi yang lebih *compact*. Lapisan *Dense* dengan 1024-unit dan fungsi aktivasi ReLU ditambahkan untuk menambah kapasitas model. Lapisan *output* terakhir memiliki satu unit dengan fungsi aktivasi *sigmoid* yang sesuai dengan sifat tugas klasifikasi biner. Model kemudian diatur ulang menggunakan objek *Sequential* dari *library* Keras yang memungkinkan kombinasi lapisan-lapisan secara berurutan (NG, 2019). Proses ini akan menciptakan model yang siap untuk dilatih dengan menggunakan *dataset* sampah yang telah disiapkan sebelumnya.

Langkah terakhir dalam tahap pemodelan adalah pengompilan model. Dalam proses ini, parameter seperti *optimizer* ('adam'), fungsi *loss* ('binary_crossentropy' untuk tugas klasifikasi biner), dan metrik evaluasi (akurasi) ditentukan. Dengan demikian, konfigurasi dan persiapan model Inception-V3 telah berhasil dilakukan untuk tahap pelatihan dengan *dataset* sampah yang telah dipersiapkan sebelumnya.

### 2.2.5 *Model Evaluation*

Tahap evaluasi dalam pengembangan model *machine learning* mirip dengan memberikan penilaian akhir pada kinerja model. Seperti memberikan peringkat akhir pada seorang atlet setelah pertandingan, evaluasi ini menjadi langkah kritis untuk mengukur sejauh mana model memenuhi tujuan dan kebutuhan yang telah ditetapkan sejak awal. Evaluasi bertujuan untuk menilai kemampuan model klasifikasi sampah organik dan non-organik menggunakan teknologi *transfer learning* dan *data augmentation*.




**Gambar 8 *Precision* & *Recall***

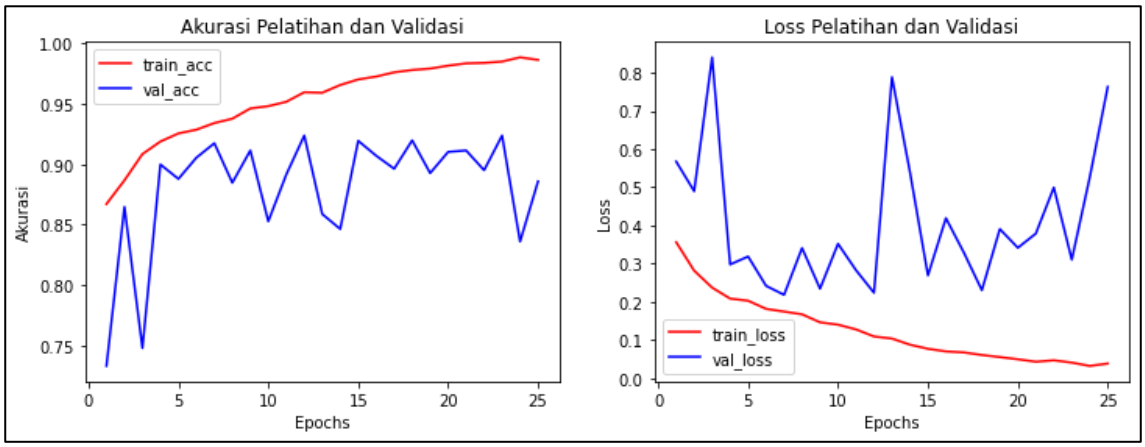
Langkah-langkah evaluasi mencakup berbagai metrik kinerja, seperti akurasi, *precision*, *recall*, dan *F1-score* (Nanmaran et al., 2022). Akurasi mengukur sejauh mana model dapat mengklasifikasikan sampah dengan benar, sedangkan *precision* menilai sejauh mana model memberikan label yang benar untuk kelas tertentu dari semua prediksinya. *Recall* mengukur sejauh mana model dapat mengidentifikasi seluruh sampah yang sebenarnya dalam *dataset*, dan *F1-score* menggabungkan *precision* dan *recall* untuk memberikan gambaran yang lebih holistik tentang kinerja model (Tripathi, 2021). Gambar 8 memberikan gambaran umum tentang bagaimana *precision* dan r*ecall* bekerja. Hasil evaluasi ini menjadi panduan penting dalam menentukan apakah model tersebut layak digunakan dalam situasi tertentu atau apakah perlu dilakukan peningkatan lebih lanjut. Misalnya, jika model memiliki akurasi tinggi tetapi presisi rendah, hal ini dapat mengindikasikan bahwa model cenderung memberikan label yang benar untuk kelas tertentu, tetapi mungkin terlalu sering memberikan label positif. Evaluasi yang cermat membantu dalam mengevaluasi *trade-off* antara berbagai metrik dan memastikan bahwa model dapat memberikan hasil yang dapat diandalkan dalam pengenalan sampah organik dan non-organik. Tahap evaluasi tidak hanya memberikan penilaian akhir pada kualitas model, tetapi juga memberikan wawasan yang berharga untuk perbaikan dan pengembangan model di masa depan.

## 3. HASIL DAN PEMBAHASAN

Bab ini secara sistematis menguraikan proses pelatihan model Inception-V3 untuk klasifikasi sampah organik dan non-organik menggunakan pendekatan *transfer learning*. Proses dimulai dengan *fine-tuning* model selama 25 *epoch* atau iterasi guna mengoptimalkan kinerja. Evaluasi akan dilakukan pada setiap *epoch* menggunakan data pengujian untuk mengukur kemampuan model dalam mengklasifikasikan gambar yang belum pernah dilihat sebelumnya.

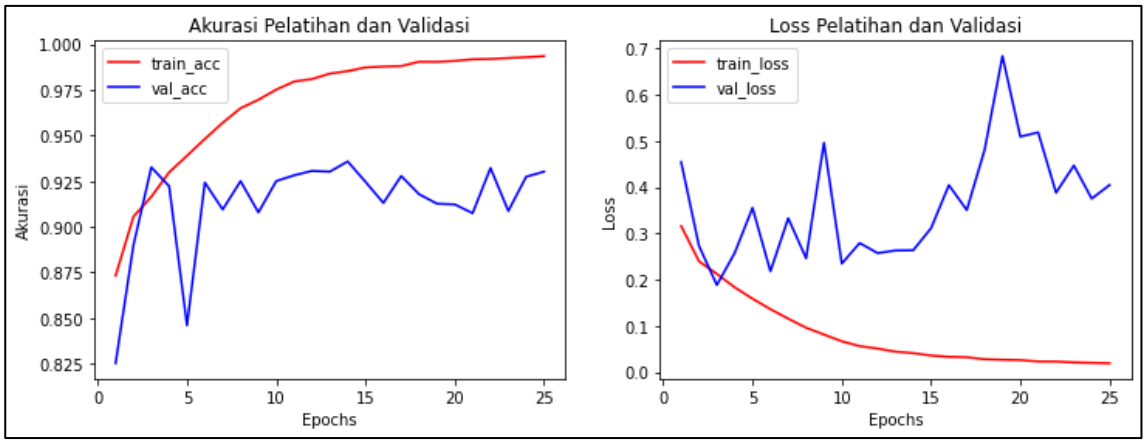
### 3.1 Hasil Pelatihan

Grafik pada Gambar 9 menampilkan perkembangan akurasi pelatihan (*training accuracy*) dan akurasi validasi (*validation accuracy*) serta kerugian pelatihan (*training loss*) dan kerugian validasi (*validation loss*) dari model yang telah dilatih menggunakan *original data* (data yang belum di augmentasi). Terlihat bahwa akurasi pada data pelatihan menunjukkan peningkatan yang konsisten di setiap *epoch*. Meski begitu, akurasi pada data pengujian mengalami fluktuasi dengan nilai akurasi berada pada rentang 0,73 hingga 0,92.




**Gambar 9 Akurasi dan *Loss* dari Model yang menggunakan *Original Data***

Pada grafik *Loss* Pelatihan dan Validasi, terlihat bahwa nilai *loss* pelatihan menurun secara konsisten setiap kali *epoch* berlalu. Hal ini mencerminkan adanya kesinambungan dalam proses pembelajaran model. Namun, terdapat fluktuasi pada nilai loss validasi, yang cenderung meningkat seiring berjalannya waktu. Fenomena ini mengisyaratkan adanya masalah *overfitting* pada model. Selanjutnya**,** model dilatih kembali menggunakan data yang telah di augmentasi. Hasil dari proses pelatihan ini dapat dilihat pada Gambar 10.




**Gambar 10 Akurasi dan *Loss* dari Model yang menggunakan *Augmented Data***

Dapat dilihat bahwa meskipun akurasi dan kerugian pada data uji masih cenderung mengalami fluktuatif, model yang dilatih menggunakan data yang telah di-augmentasi menunjukkan kinerja keseluruhan yang lebih baik dibandingkan dengan model yang menggunakan data asli (data yang belum di-augmentasi).

### 3.2 Evaluasi Model

Evaluasi model dilakukan dengan menggunakan *classification report* dan *confusion matrix* yang dapat dilihat pada Tabel 1 dan Tabel 2. Hasil menunjukkan bahwa model memiliki performa yang memuaskan dalam mengklasifikasikan sampah organik dan non-organik. Untuk kelas ‘Organik’, model dengan data yang telah di augmentasi mencapai nilai *precision* sebesar 0,92, *recall* sebesar 0,97, dan *F1-Score* sebesar 0,94. Hal ini menunjukkan bahwa model memiliki kemampuan tinggi untuk mengidentifikasi sampah organik dengan sangat baik. Di sisi lain, untuk kelas Non Organik, model mencapai nilai *precision* sebesar 0,96, *recall* sebesar 0,90, dan *F1-Score* sebesar 0,93.

JISKA (Jurnal Informatika Sunan Kalijaga)
Vol. 9, No. 3, SEPTEMBER, 2024: 192 – 204

Secara keseluruhan, model berhasil mencapai tingkat akurasi sebesar 0,94. Hal ini mencerminkan kemampuannya dalam melakukan klasifikasi yang tepat pada seluruh *dataset*. Implementasi augmentasi data juga terbukti meningkatkan kinerja model secara keseluruhan hingga 2%. Dalam evaluasi menyeluruh, nilai *precision*, *recall*, dan *F1-Score* menunjukkan keseimbangan yang optimal antara kelas Organik dan Non Organik dengan rata-rata nilai yang tinggi. Hasil evaluasi ini memberikan keyakinan yang kuat bahwa model Inception-V3 berhasil mengklasifikasikan sampah organik dan non-organik dengan tingkat akurasi dan kinerja yang sangat baik.

**Tabel 1 *Model Classification Result***

| | ***Precision*** | | ***Recall*** | | ***F1-Score*** | |
|---|---|---|---|---|---|---|
| | ***Without Augmentation*** | ***With Augmentation*** | ***Without Augmentation*** | ***With Augmentation*** | ***Without Augmentation*** | ***With Augmentation*** |
| **Organik** | 0,91 | 0,92 | 0,95 | 0,97 | 0,93 | 0,94 |
| **Non Organik** | 0,94 | 0,96 | 0,88 | 0,90 | 0,91 | 0,93 |
| ***Accuracy*** | | | | | **0,92** | **0,94** |

**Tabel 2 *Confusion Matrix* Menggunakan *Augmented Data***

| | | **Prediksi** | |
|---|---|---|---|
| | | Organik | Non Organik |
| **Aktual** | Organik | 1348 | 46 |
| | Non Organik | 115 | 997 |

### 3.3 Uji Coba

Percobaan dilakukan dengan menggunakan model untuk melakukan klasifikasi terhadap 100 sampel gambar sampah yang berasal dari berbagai kondisi yang diambil dari di situs Google. Seluruh sampel ini dipilih untuk mencakup keragaman kondisi sampah yang mungkin dihadapi dalam aplikasi praktis, sehingga pengujian dapat mencerminkan situasi yang lebih realistis. Kemampuan model diuji dalam mengenali dan mengklasifikasikan sampah ke dalam dua kategori utama, yaitu organik dan non-organik. Proses ini bertujuan untuk mengevaluasi sejauh mana model dapat memberikan prediksi yang akurat dan konsisten terhadap berbagai jenis sampah. Hasil dari identifikasi sampah menggunakan model Inception-V3 dapat dilihat pada Gambar 11 dan Gambar 12.


**Gambar 11 Sampel Hasil Identifikasi Sampah Organik Menggunakan Inception-V3**


**Gambar 12 Sampel Hasil Identifikasi Sampah Non Organik Menggunakan Inception-V3**

Dari hasil evaluasi percobaan seperti yang terlihat pada Tabel 3, menunjukkan bahwa model mampu mengklasifikasikan sampah organik dengan tingkat akurasi sempurna di 100%, yang menandakan kemampuan model sudah sangat baik dalam mengenali sampah organik. Namun, pada kategori sampah non-organik, meskipun tingkat akurasinya masih tinggi (96%), terdapat beberapa kesalahan klasifikasi. Sebanyak 2 data citra non-organik salah diklasifikasikan sebagai organik. Hal ini memberikan gambaran bahwa model perlu diperbaiki khususnya dalam mengenali sampah non-organik agar dapat meningkatkan akurasi secara keseluruhan.

**Tabel 3 Hasil Pengujian**

| Kondisi Sampah | Data | Benar | Salah | Akurasi | *Error* |
|---|---|---|---|---|---|
| Organik | 50 | 50 | 0 | 1,00 | 0,00 |
| Non Organik | 50 | 48 | 2 | 0,96 | 0,04 |

## 4. KESIMPULAN

Implementasi *transfer learning* dengan arsitektur Inception-V3 dan penerapan *data augmentation* berhasil menciptakan model klasifikasi sampah organik dan non-organik dengan tingkat akurasi data uji mencapai 94%. Model ini menunjukkan keunggulan istimewa dalam mengenali sampah organik dengan akurasi sempurna 100%. Namun, evaluasi yang lebih cermat masih diperlukan terutama dalam klasifikasi sampah non-organik, di mana akurasinya tidak mencapai level yang sama dengan sampah organik.

Penelitian ini berhasil menunjukkan bahwa kombinasi *transfer learning* dan *data augmentation* dapat memberikan peningkatan signifikan dalam akurasi dan kemampuan generalisasi model dibandingkan dengan metode Convolutional Neural Network (CNN) yang digunakan pada penelitian sebelumnya. Keunggulan metode *transfer learning* dengan arsitektur Inception-V3 terletak pada kemampuannya memanfaatkan fitur-fitur yang telah dipelajari dari *dataset* besar, sehingga mempercepat proses pelatihan dan meningkatkan performa pada *dataset* yang lebih kecil dan spesifik.

Untuk penelitian berikutnya, disarankan untuk fokus pada eksplorasi lebih lanjut mengenai teknik *data augmentation* yang lebih canggih. Teknik-teknik seperti Generative Adversarial Networks (GANs) untuk menghasilkan data baru, atau penggunaan augmentasi yang lebih bervariasi, dapat membantu meningkatkan keragaman data dan mengatasi masalah *overfitting* yang terjadi pada model saat ini. Selain itu, penelitian lanjutan juga dapat mengeksplorasi kombinasi arsitektur model lain atau teknik ensembel untuk lebih meningkatkan performa klasifikasi sampah non-organik.

**DAFTAR PUSTAKA**

Ahmed, M., Afreen, N., Ahmed, M., Sameer, M., & Ahamed, J. (2023). An inception V3 approach for malware classification using machine learning and transfer learning. *International Journal of Intelligent Networks*, *4*, 11–18. https://doi.org/10.1016/j.ijin.2022.11.005

Fadillah, I., A, L., Kamil, F. El, Shalahuddin, M., Setiawan, I., N, A., M, H., A, N., S, R., & Fikri, K. (2019). Perubahan Pola Pikir Masyarakat tentang Sampah melalui Sosialisasi Pengolahan Sampah Organik dan Non Organik di Dusun Pondok, Kecamatan Gedangsari, Kab. Gunungkidul. *Prosiding Konferensi Pengabdian Masyarakat* , *1*, 239–242. https://sunankalijaga.org/prosiding/index.php/abdimas/article/view/201

Fan, Y., Li, J., Bhatti, U. A., Shao, C., Gong, C., Cheng, J., & Chen, Y. (2023). A Multi-Watermarking Algorithm for Medical Images Using Inception V3 and DCT. *Computers, Materials & Continua*, *74*(1), 1279–1302. https://doi.org/10.32604/cmc.2023.031445

Hidayati, N., Suntoro, J., & Setiaji, G. G. (2021). Perbandingan Algoritma Klasifikasi untuk Prediksi Cacat Software dengan Pendekatan CRISP-DM. *Jurnal Sains Dan Informatika*, *7*(2), 117–126. https://doi.org/10.34128/jsi.v7i2.313

Ibnul Rasidi, A., Pasaribu, Y. A. H., Ziqri, A., & Adhinata, F. D. (2022). Klasifikasi Sampah Organik dan Non-Organik Menggunakan Convolutional Neural Network. *Jurnal Teknik Informatika Dan Sistem Informasi*, *8*(1), 142-149–142 – 149. https://doi.org/10.28932/jutisi.v8i1.4314

*InceptionV3*. (2015). Google. https://keras.io/api/applications/inceptionv3/

Kartiko, Prima Yudha, A., Dimas Aryanto, N., & Arya Farabi, M. (2022). Klasifikasi Sampah di Saluran Air Menggunakan Algortima CNN. *Indonesian Journal of Data and Science*, *3*(2), 72–81. https://doi.org/10.56705/ijodas.v3i2.33

Kementerian Lingkungan Hidup dan Kehutanan. (2022). *SIPSN - Sistem Informasi Pengelolaan Sampah Nasional*. Kementerian Lingkungan Hidup Dan Kehutanan. https://sipsn.menlhk.go.id/sipsn/

Lebreton, L., & Andrady, A. (2019). Future scenarios of global plastic waste generation and disposal. *Palgrave Communications*, *5*(1), 6. https://doi.org/10.1057/s41599-018-0212-7

Lin, C., Li, L., Luo, W., Wang, K. C. P., & Guo, J. (2018). Transfer Learning Based Traffic Sign Recognition Using Inception-v3 Model. *Periodica Polytechnica Transportation Engineering*, *47*(3), 242–250. https://doi.org/10.3311/PPtr.11480

Luke, J. J., Joseph, R., & Balaji, M. (2019). IMPACT OF IMAGE SIZE ON ACCURACY AND GENERALIZATION OF CONVOLUTIONAL NEURAL NETWORKS. *International Journal of Research and Analytical Reviews*, *6*(1), 70–80. www.ijrar.org

Mikolajczyk, A., & Grochowski, M. (2018). Data augmentation for improving deep learning in image classification problem. *2018 International Interdisciplinary PhD Workshop (IIPhDW)*, 117–122. https://doi.org/10.1109/IIPHDW.2018.8388338

Minarno, A. E., Aripa, L., Azhar, Y., & Munarko, Y. (2023). Classification of Malaria Cell Image using Inception-V3 Architecture. *JOIV : International Journal on Informatics Visualization*, *7*(2), 273. https://doi.org/10.30630/joiv.7.2.1301

Minarno, A. E., Hasanuddin, M. Y., & Azhar, Y. (2023). Batik Images Retrieval Using Pre-trained model and K-Nearest Neighbor. *JOIV : International Journal on Informatics Visualization*, *7*(1), 115. https://doi.org/10.30630/joiv.7.1.1299

Nanmaran, R., Srimathi, S., Yamuna, G., Thanigaivel, S., Vickram, A. S., Priya, A. K., Karthick, A., Karpagam, J., Mohanavel, V., & Muhibbullah, M. (2022). Investigating the Role of Image Fusion in Brain Tumor Classification Models Based on Machine Learning Algorithm for Personalized Medicine. *Computational and Mathematical Methods in Medicine*, *2022*(1), 1–13. https://doi.org/10.1155/2022/7137524

NG, K. (2019). Tuned Inception V3 for Recognizing States of Cooking Ingredients. *State Recognition Symposium*. https://doi.org/10.32555/2019.dl.009

Pei, X., Zhao, Y. hong, Chen, L., Guo, Q., Duan, Z., Pan, Y., & Hou, H. (2023). Robustness of machine learning to color, size change, normalization, and image enhancement on micrograph datasets with large sample differences. *Materials & Design*, *232*, 112086. https://doi.org/10.1016/j.matdes.2023.112086

Rosiana, E., & Perdana, R. (2022). Rancang Bangun Sistem Robot Pemilah Sampah Anorganik dengan Inductive Proximity dan LDR Sebagai Sensor. *Building of Informatics, Technology and Science (BITS)*, *4*(2), 1001–1009. https://doi.org/10.47065/bits.v4i2.2017

Schröer, C., Kruse, F., & Gómez, J. M. (2021). A Systematic Literature Review on Applying CRISP-DM Process Model. *Procedia Computer Science*, *181*, 526–534. https://doi.org/10.1016/j.procs.2021.01.199

Sekar, S. (2019). *Waste Classification data*. Kaggle. https://www.kaggle.com/datasets/techsash/waste-classification-data

Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*, *6*(1), 1–48. https://doi.org/10.1186/S40537-019-0197-0/FIGURES/33

Singh, D., & Singh, B. (2020). Investigating the impact of data normalization on classification performance. *Applied Soft Computing*, *97*, 105524. https://doi.org/10.1016/j.asoc.2019.105524

Tripathi, M. (2021). Analysis of Convolutional Neural Network based Image Classification Techniques. *Journal of Innovative Image Processing*, *3*(2), 100–117. https://doi.org/10.36548/jiip.2021.2.003

Vallez, N., Bueno, G., Deniz, O., & Blanco, S. (2022). Diffeomorphic transforms for data augmentation of highly variable shape and texture objects. *Computer Methods and Programs in Biomedicine*, *219*, 106775. https://doi.org/10.1016/j.cmpb.2022.106775

Wang, Y. H., Ou, Y., Deng, X. D., Zhao, L. R., & Zhang, C. Y. (2019). The Ship Collision Accidents Based on Logistic Regression and Big Data. *Proceedings of the 31st Chinese Control and Decision Conference, CCDC 2019*, 4438–4440. https://doi.org/10.1109/CCDC.2019.8832686

Widodo, A. E., & Suleman, S. (2020). Otomatisasi Pemilah Sampah Berbasis Arduino Uno. *Indonesian Journal on Software Engineering (IJSE)*, *6*(1), 12–18. https://doi.org/10.31294/ijse.v6i1.7781

Zhang, Y. D., Dong, Z., Chen, X., Jia, W., Du, S., Muhammad, K., & Wang, S. H. (2019). Image based fruit category classification by 13-layer deep convolutional neural network and data augmentation. *Multimedia Tools and Applications*, *78*(3), 3613–3632. https://doi.org/10.1007/S11042-017-5243-3/METRICS

**JISKA (Jurnal Informatika Sunan Kalijaga)**
**Vol. 9, No. 3, SEPTEMBER, 2024, Pp. 205 – 216**
**ISSN: 2527 – 5836 (print) | 2528 – 0074 (online)**

# Implementasi K-Means *Clustering* pada Pengelompokan Pasien Penyakit Jantung

**Jihan Wala [(1)*], Herman [(2)], Rusydi Umar [(3)]**
Magister Informatika, Fakultas Teknologi Industri, Universitas Ahmad Dahlan, Yogyakarta, Indonesia
e-mail : 2307048013@webmail.uad.ac.id, {hermankaha,rusydi}@mti.uad.ac.id.
* Penulis korespondensi.
Artikel ini diajukan 15 April 2024, direvisi 25 Juni 2024, diterima 26 Juni 2024, dan dipublikasikan 25 September 2024.

***Abstract***

*Heart disease is a prominent global health concern, necessitating early identification and patient grouping for effective management. This study employs the K-Means clustering algorithm with a medical dataset of 303 patients, encompassing various attributes. These include Age, Gender, Chest Pain Type, Blood Pressure, Serum Cholesterol Level, Fasting Blood Sugar, Resting Electrocardiographic Results, Maximum Heart Rate, Angina, ST Depression, and Slope of the ST Segment. The goal is to categorize patients into four clusters based on chest pain types, a crucial symptom indicating disease severity. The computation concludes after the sixth iteration, revealing Cluster 1 (27 patients), Cluster 2 (135 patients), Cluster 3 (15 patients), and Cluster 4 (126 patients). Collaborative analysis with medical experts highlights that Cluster 1, mainly comprising older males, exhibits high-risk indicators. While this grouping aids in personalized treatment strategy development, further clinical validation involving more experts and datasets is imperative for enhanced reliability.*

***Keywords: Implementation, K-Means, Clustering, Grouping, Heart Disease***

**Abstrak**

Penyakit jantung menjadi permasalahan kesehatan serius diseluruh dunia. Pendeteksian dini dan pengelompokan pasien berdasarkan ciri-ciri khusus dapat mendukung manajemen penanganan penyakit jantung. Penelitian ini mengusulkan algoritma K-Means *clustering* untuk mengelompokkan pasien penyakit jantung dengan *dataset* medis sebanyak 303 pasien. *Dataset* mencakup atribut Umur, Jenis Kelamin, Jenis Nyeri Dada, Tekanan Darah, Kadar Serum Kolesterol, Gula Darah, Hasil Elektrokardiografi, Denyut Jantung Maksimum, Angina, Depresi ST, dan Kemiringan Segmen ST. Tujuan penelitian ini adalah mengelompokkan pasien penyakit jantung berdasarkan tingkat keparahan atau kegawatdaruratan pasien menggunakan algoritma K-Means *clustering*. Wawancara bersama ahli medis untuk pembagian kelompok menjadi empat *cluster* berdasarkan jenis nyeri dada yang merupakan gejala utama tingkat keparahan penyakit jantung. Interpretasi menghasilkan 5 *cluster* dengan *cluster k*1 berjumlah 27 pasien, *k*2 berjumlah 135 pasien, *k*3 berjumlah 15 pasien, dan *k*4 berjumlah 126 pasien. Analisis data menunjukan, *cluster* 1 (*k*1), cenderung terdiri dari pasien yang lebih tua, mayoritas laki-laki, menunjukan risiko tinggi dengan gejala nyeri dada parah, tekanan darah, dan kadar kolesterol tinggi. Sementara itu, *cluster k*2, *k*3, dan *k*4 menunjukkan risiko lebih rendah, dengan variasi respons terhadap aktivitas fisik. Pengelompokan ini memberikan dukungan kepada dokter dan peneliti dalam memahami pola penyakit jantung serta merancang strategi pengobatan yang lebih spesifik dan personal.

**Kata Kunci: Implementasi, K-Means, *Clustering*, Pengelompokkan, Penyakit Jantung**

## 1. PENDAHULUAN

Penyakit jantung merupakan penyebab utama kematian di seluruh dunia. Menurut World Health Organization tahun 2020 terdapat 17,9 juta kematian dan 80% disebabkan oleh penyakit arteri koroner dan *stroke* serebral (Ali et al., 2021). Jumlah kematian yang besar ini umum terjadi di negara-negara berpenghasilan rendah dan menengah (Shah et al., 2020). Penyakit jantung dapat disebabkan oleh berbagai faktor yang berkaitan dengan kebiasaan hidup, seperti merokok, penggunaan alkohol dan kafein secara berlebihan, stres, aktifitas fisik yang kurang. Sebab lain

adalah faktor-faktor fisiologis (obesitas, hipertensi, kolesterol, darah tinggi, dan kondisi jantung). Identifikasi pasien penyakit jantung dilakukan dengan melihat atribut terkait data pasien yang memiliki signifikansi besar untuk membantu dokter memberikan perawatan yang lebih terfokus (Singh & Kumar, 2020). Salah satu teknik yang dapat membantu dokter dalam mengidentifikasi dan mengelompokan pasien penyakit jantung adalah teknik *data mining*.

*Clustering* adalah teknik yang populer digunakan pada *data mining.* Teknik ini merupakan proses pengelompokan data menjadi beberapa *cluster* data berdasarkan kemiripan atribut-atribut yang dimiliki data (Haris Kurniawan et al., 2020). Pada penelitian ini, digunakan algoritma K-Means *clustering*. K-Means adalah teknik pengelompokan data di mana atribut data dikelompokkan ke dalam partisi set data, kemudian ditetapkan ke dalam kelompok yang berbeda (Ikotun et al., 2023). Penelitian ini relavan dengan beberapa penelitian sebelumnya dalam penerapan K-Means *clustering* sebagai sumber referensi dan perbandingan hasil penelitian.

Penelitian Ariefandi et al., menggunakan *k-medoids clustering* untuk klasterisasi wilayah terinfeksi kasus *covid*-19 di DKI Jakarta. Penelitian ini menghasilkan 3 *cluster* dengan kasus yang paling tertinggi yakni *cluster* 0 terdiri dari 31 kelurahan sedangkan *cluster* paling rendah diketahui *cluster* 2 terdiri 66 kelurahan (Arifandi et al., 2021). Kemudian penelitian Purba et al., menggunakan K-Means *clustering* untuk mengelompokkan penyebab penyakit ISPA. Menghasilkan 2 *cluster*, di mana *cluster* 1 memberikan rekomendasi tinggi berjumlah 10 Kabupaten, *cluster* 2 memberikan rekomendasi rendah berjumlah 2 Kabupaten (Purba et al., 2021). Solechati & Jananto mengelompokan *profile* pasien. Dengan interpretasi menghasilkan 5 *cluster* dengan *cluster* 1 memiliki 228 *record*, pada *cluster* 2 memiliki 248 *record*, *cluster* 3 memiliki 1551 *record*, *cluster* 4 memiliki 2592 *record*, dan *cluster* 5 memiliki 362 *record* (Solechati & Jananto, 2023). Mashita et al., melakukan klasifikasi pada pasien penyakit jantung. penelitian ini menghasilkan akurasi yang diperoleh adalah *k*=7 dan *k*=9, yang merupakan hasil paling optimal karena memiliki akurasi tertinggi dibandingkan dengan nilai *k* lainnya, dengan akurasi sebesar 88% (Masitha et al., 2023). Novidianto el al., menggunakan Metode *k-prototypes cluster mix algorithm* untuk mengidentifikasi faktor kematian pada pasien gagal jantung. Hasil klasterisasi membentuk 2 *cluster* yang dianggap optimal berdasarkan nilai koefisien *silhouette* tertinggi sebesar 0,5777. Analisis hasil menunjukan bahwa *cluster* 1 adalah *cluster* pasien yang memiliki risiko rendah terhadap kemungkinan kematian akibat gagal jantung dan *cluster* 2 adalah *cluster* pasien dengan risiko tinggi terhadap kematian akibat gagal jantung (Novidianto et al., 2021)

Berdasarkan kajian literatur penelitian terdahulu di atas, maka penelitian ini dilakukan dengan tujuan untuk mengelompokkan pasien penyakit jantung berdasarkann keparahan atau kegawatdaruratan pasien menggunakan pendekatan K-Means *clustering*. Harapan dilakukan penelitian ini dapat memberikan wawasan alaternatif dalam pengelompokan pasien penyakit jantung dan berpotensi menjadi landasan untuk pengembangan strategi pengobatan yang lebih efektif di masa depan. Kontribusi penelitian ini terletak pada pengembangan strategi pengobatan yang lebih spesifik dan personal.

## 2. METODE PENELITIAN

### 2.1 Tahapan Penelitian

Penelitian ini melibatkan serangkaian langkah yang penting untuk mempersiapkan dan merencanakan studi secara menyeluruh. Lima langkah tahapan yang dilakukan dalam penelitian ini adalah studi pustaka, pengumpulan data, implementasi K-Means *clustering*, dan analisis hasil *cluster*. Adapun tahapan penelitian secara lengkap dapat dilihat pada Gambar 1.


**Gambar 1 Tahapan Penelitian**

### 2.1.1 Studi Pustaka

Langkah awal penting dalam melakukan penelitian pendahuluan adalah melakukan studi pustaka yang komprehens. Ini melibatkan peninjauan yang luas terhadap literatur yang berkaitan dengan topik penelitian yang akan dilakukan (Muslimah, 2024). Pada tahap ini dilakukan pengumpulan, peninjauan, dan analisis berbagai sumber informasi seperti jurnal ilmiah, buku, artikel, dan publikasi terkait lainnya. Studi pustaka bertujuan untuk menghimpun, menelaah, dan menganalisis literatur terkait yang relevan dengan topik penelitian yang sedang diselidiki. Melalui proses ini, dapat memperoleh pemahaman yang mendalam tentang status terkini dari penelitian yang sudah ada, mengidentifikasi pengetahuan yang telah dikembangkan, serta menemukan area-area di mana pengetahuan masih terbatas dan memerlukan penelitian lebih lanjut.

### 2.1.2 Pengumpulan Data

**Tabel 1 Deskripsi *Dataset Heart Disease* (Penyakit Jantung)**

| Atribut | Keterangan | Penjelasan |
|---|---|---|
| *Id* | Id Pasien | Kode pasien dari 1-303 |
| *Age* | umur pasien (tahun) | Minimal = 29, Maksimal = 77 |
| *Sex* | Jenis kelamin pasien | 1 = laki-laki, 0 = perempuan |
| *Cp* | Jenis nyeri dada | *Cp* (*Chest pain*) yaitu tipe nyeri dada yang diderita pasien. Atribut ini memiliki 4 nilai yaitu:<br>Nilai 1: tidak nyeri dada (*no chest pain*)<br>Nilai 2: nyeri dada ringan (*mild chest pain*)<br>Nilai 3: nyeri dada sedang (*moderate chest pain*)<br>Nilai 4: nyeri dada parah (*severe chest pain*) |
| *Trestbps* | Tekanan darah istirahat (mm Hg) | *Trestbps* (*Resting blood pressure*) yaitu tekanan darah pasien ketika dalam keadaan istirahat. Rendah < 120, normal = 120, tinggi > 120 |
| *Chol* | Serum kolesterol (mg/dl) | *Chol* (*Cholesterol*) yaitu kadar kolesterol dalam darah pasien. Rendah < 140, normal = 140, tinggi > 140 |
| *Fbs* | Gula darah puasa > 120 mg/dl | *Fbs (Fasting blood sugar)* yaitu kadar gula darah pasien, atribut fbs ini memiliki 2 nilai yaitu 1 jika kadar gula darah pasien melebihi 120 mg/dl, dan 0 jika tidak melebihi atau sama dengan 120 mg/dl. |
| *Restecg* | Hasil elektrokardiografi istirahat | *Resting electrocardiographic* memiliki 3 nilai yaitu nilai 0 = normal, nilai 1 = ST-T *wave abnormality* nilai 2 = ventricular kiri mengalami hipertrop |
| *Thalach* | Denyut jantung maksimum | Tingkat detak jantung maksimum yang dicapai. Jika nilai “thalac” semakin tinggi dapat dianggap sebagai tanda risiko yang lebih tinggi untuk penyakit jantung |
| *Exang* | Angina yang dipicu oleh Latihan | *Exang* (*Exercise-induced angina*) keadaan dimana pasien akan mengalami nyeri dada apabila berolah raga, 0 = tidak nyeri, dan 1 = menyebabkan nyeri. |
| *Oldpeak* | Depresi ST | Depresi ST yang diinduksi oleh latihan relatif terhadap istirahat. Penurunan ST akibat olahraga. Nilai "*Oldpeak*" yang tinggi dapat dianggap sebagai tanda risiko yang lebih tinggi untuk penyakit jantung |
| *Slope* | Kemiringan segmen ST latihan puncak. | *Slope* dari puncak ST setelah berolah raga. Atribut ini memiliki 3 nilai yaitu 0 untuk *downsloping*, 1 untuk flat, dan 2 untuk *upsloping* |

Data yang digunakan merupakan data sekunder yang di ambil dari internet situs Kaggle, *dataset* Penyakit Jantung (*heart disease*) oleh Awan (2020) sebagai objek penelitian. Penjelasan lebih

spesifik *dataset* penyakit jantung dilakukan bersama ahli medis (Dokter Spesialis Jantung dan Pembuluh Darah) dan sumber referensi dari beberapa artikel jurnal. Berikut pada Tabel 1 penjelasan setiap atribut *dataset* (Ali et al., 2021; Shah et al., 2020; Singh & Kumar, 2020; V. Ramalingam et al., 2018).

### 2.1.3 *Preprocessing*

Setelah melakukan pengumpulan data, tahap selanjutnya adalah *pre-processing* data. Tahap ini meliputi proses pengecekan *missing value* dan normalisasi data. *Missing value* mengindikasikan ketiadaan informasi untuk suatu variabel pada observasi tertentu. Pentingnya pengecekan *missing value* dalam analisis data karena hal tersebut dapat membantu mencegah adanya bias dalam penarikan kesimpulan (Han & Kang, 2023). Normalisasi bertujuan untuk membuat skala variabel dalam *dataset* menjadi seragam, sehingga setiap variabel memiliki kontribusi yang seimbang dalam analisis (Mishra et al., 2020). Metode normalisasi yang akan dilakukan pada penelitian ini yaitu *feature scaling*. *Feature scaling* dilakukan dengan tujuan untuk membandingkan atau mengintegrasikan data dari berbagai sumber atau variabel yang memiliki rentang nilai yang berbeda-beda. *Feature scaling* mengubah nilai-nilai yang diperkirakan ke dalam rentang yang lebih kecil atau seragam memiliki skala, Di mana nilai-nilai diperkirakan dikonversi ke rentang antara 0 sampai 1. Normalisasi *feature scaling* dapat dilakukan menggunakan Pers. (1) (Sun & Yu, 2021). Dalam rumus *feature scaling*, $X_{baru}$ adalah nilai atribut baru setelah normalisasi, $X_{awal}$ adalah nilai atribut asli yang akan dinormalisasi, dan $X_{max}$ adalah nilai maksimum dari semua data pada atribut yang sama.

$$X_{baru} = \frac{X_{awal}}{X_{max}} \quad (1)$$

### 2.1.4 Penerapan K-Means *Clustering*

Proses selanjutnya adalah penerapan algoritma K-Means *clustering*. K-Means merupakan salah satu teknik pengelompokan yang paling *prominent* dalam ilmu dan teknologi (Das et al., 2023). Tujuan utama dari K-Means *clustering* adalah untuk membagi *dataset* menjadi kelompok-kelompok yang homogen, di mana setiap kelompok memiliki kesamaan internal yang tinggi dan perbedaan yang signifikan antar kelompok (Qi et al., 2023). *Flowchart* K-Means *clustering* dapat dilihat pada Gambar 2 (Rizki et al., 2020).

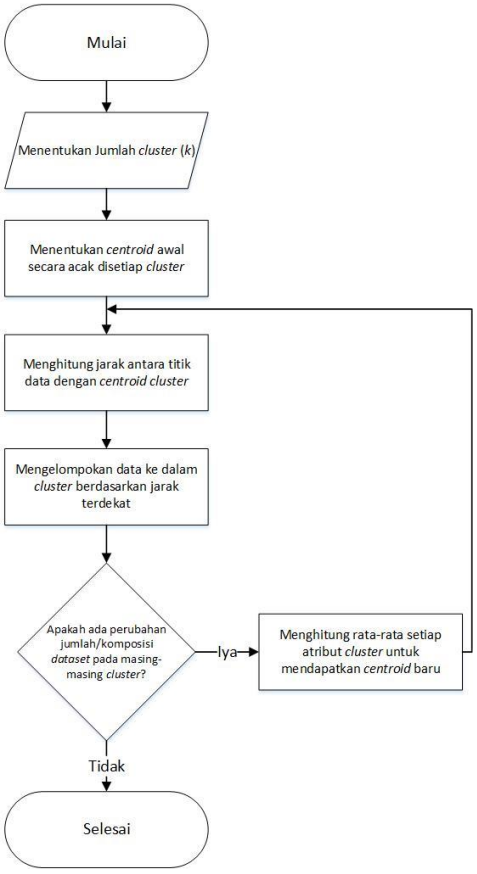
Berikut penjelasan tahapan *flowchart* K-Means *clustering* pada Gambar 2:

1) Menentukan jumlah *cluster* (*k*) merupakan tahap pertama dalam penentuan jumlah *cluster* yang optimal untuk data yang akan dikelompokan.
2) Menentukan *centroid* awal secara acak di setiap *cluster*. Pemilihan awal *centroid* dilakukan dengan cara mengambil secara acak titik data yang ada dalam *dataset*. Titik data yang akan menjadi *centroid* awal dipilih tanpa mempertimbangkan distribusi atau karakteristik khusus dari data. Secara praktis, setiap titik dalam *dataset* memiliki kesempatan yang sama untuk dipilih sebagai *centroid* awal.
3) Menghitung jarak antara titik data dengan setiap *centroid cluster* yaitu proses dalam K-Means *clustering* di mana jarak antara setiap titik data dengan setiap *centroid cluster* dihitung, menghitung jarak digunakan rumus Euclidean *distance*. Untuk menghitung jarak antara data *x* baris ke-*i* (*i*=1,2,3,…*n*), data *c* baris ke-*h* (*h*=1,2,3,…*k*) yang disimbolkan $d\ (x_i, c_h)$, dengan *n* merupakan jumlah total baris data, *m* adalah jumlah atribut, dan *k* adalah jumlah *cluster*. Rumus jarak $d\ (x_i, c_h)$ ditampilkan pada Pers. (2). Di mana $x_{ij}$ adalah atribut ke *j* dari data ke *i* dan $c_{hj}$ adalah atribut ke *j* dari *cluster h*. Jarak terkecil dari data ke *i* ke *cluster h* menunjukan bahwa data ke *i* masuk dalam *cluster h.* Jika jarak dari data ke 5 paling kecil adalah dengan *cluster* 3, maka data 5 dikelompokkan dalam *cluster* 3.

$$d\ (x_i, c_h) = \sqrt{\sum_{j=1}^{m}\sum_{h=1}^{k}(x_{ij} - c_{hj})^2} \quad (2)$$

$$d\ (x_i, c_h) = \sqrt{(x_{i1} - c_{h1})^2 + (x_{i2} - c_{h2})^2 + (x_{i3} - c_{h3})^2 + \ldots + (x_{im} - c_{hm})^2}$$

4) Kemudian memeriksa apakah terjadi perubahan dari *centroid* baru terhadap *centroid* sebelumnya setelah pengelompokan data ke dalam *cluster*. Jika terjadi perubahan pada nilai *centroid*, maka menunjukan bahwa proses masih berjalan dan pengelompokan data harus terus dilakukan pada iterasi berikutnya.
5) Tahap terakhir yaitu jika terjadi perubahan pada nilai *centroid*, maka lanjut ke tahap selanjutnya yaitu menghitung nilai rata-ratanya untuk menghasilkan *centroid* baru pada *cluster* tersebut. Kemudian ulangi langkah 3, dan 4 pada iterasi berikutnya sampai tidak ada perubahan lagi pada *centroid* setiap *cluster*. Jika tidak terjadi perubahan pada *centroid* maka proses *clustering* dinyatakan selesai.




**Gambar 2 *Flowchart* K-Means *Clustering***

### 2.1.5 Analisis Hasil *Cluster*

Pada tahap analisis *cluster*, dilakukan pemilihan kelompok yang diprioritaskan untuk penanganan dalam pengobatan penyakit jantung. Pemilihan ini didasarkan pada hasil wawancara dengan ahli medis, yang memberikan wawasan terkait kelompok pasien dengan tingkat risiko tertinggi. Dengan menggunakan teknik *cluster*ing, data pasien dibagi ke dalam beberapa kelompok berdasarkan karakteristik medis mereka, seperti usia, riwayat penyakit, dan faktor risiko lainnya. Kelompok yang diidentifikasi sebagai prioritas merupakan fokus utama dalam pemberian

intervensi medis untuk meningkatkan efektivitas pengobatan dan mengurangi risiko komplikasi penyakit jantung.

## 3. HASIL DAN PEMBAHASAN

### 3.1 *Dataset* Penyakit Jantung

*Dataset* penelitian ini, yaitu data pasien berpenyakit jantung (*heart disease)* yang diambil dari data repositori Kaggle sebanyak 303 titik data *(data point*). Masing-masing titik data memiliki 12 atribut. Sebelum diproses *dataset* ini disortir terlebih dahulu berdasarkan *Cp* (*Chest pain*) karena hasil wawancara bersama ahli medis bahwa *Cp* merupakan gejala utama resiko penyakit jantung. Berikut pada Tabel 2 disajikan *row dataset*.

**Tabel 2 *Row Dataset* Sebelum Normalisasi**

| *Id* | *Age* | *Sex* | *Cp* | *Trestbps* | *Chol* | *Fbs* | *Restecg* | *Thalach* | *Exang* | *Oldpeak* | *Slope* |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 63 | 1 | 1 | 145 | 233 | 1 | 2 | 150 | 0 | 2.3 | 3 |
| 21 | 64 | 1 | 1 | 110 | 211 | 0 | 2 | 144 | 1 | 1.8 | 2 |
| 22 | 58 | 0 | 1 | 150 | 283 | 1 | 2 | 162 | 0 | 1 | 1 |
| 28 | 66 | 0 | 1 | 150 | 226 | 0 | 0 | 114 | 0 | 2.6 | 3 |
| 31 | 69 | 0 | 1 | 140 | 239 | 0 | 0 | 151 | 0 | 1.8 | 1 |
| 42 | 40 | 1 | 1 | 140 | 199 | 0 | 0 | 178 | 1 | 1.4 | 1 |
| 60 | 51 | 1 | 1 | 125 | 213 | 0 | 2 | 125 | 1 | 1.4 | 1 |
| 102 | 34 | 1 | 1 | 118 | 182 | 0 | 2 | 174 | 0 | 0 | 1 |
| 113 | 52 | 1 | 1 | 118 | 186 | 0 | 2 | 190 | 0 | 0 | 2 |
| 125 | 65 | 1 | 1 | 138 | 282 | 1 | 2 | 174 | 0 | 1.4 | 2 |
| 142 | 59 | 1 | 1 | 170 | 288 | 0 | 2 | 159 | 0 | 0.2 | 2 |
| 151 | 52 | 1 | 1 | 152 | 298 | 1 | 0 | 178 | 0 | 1.2 | 2 |
| … | … | … | … | … | … | … | … | … | … | … | … |
| 298 | 57 | 0 | 4 | 140 | 241 | 0 | 0 | 123 | 1 | 0.2 | 2 |
| 300 | 68 | 1 | 4 | 144 | 193 | 1 | 0 | 141 | 0 | 3.4 | 2 |
| 301 | 57 | 1 | 4 | 130 | 131 | 0 | 0 | 115 | 1 | 1.2 | 2 |

### 3.2 *Preprocessing*

Pada tahap *pre-processing* dalam penelitian ini, dilakukan pengecekan *missing value* dan normalisasi pada *dataset*. Hasil pengecekan terhadap *missing value*, tidak terdeteksi adanya *missing value* dalam *dataset* sehingga jumlah data yang diproses tetap 303 *record*. Selanjutnya dilakukan proses normalisasi data, sehingga nilai-nilai dalam *dataset* tersebut berada dalam rentang skala 0 sampai 1. Pada Tabel 3 disajikan row *dataset* setelah proses normalisasi.

**Tabel 3 *Row Dataset* Setelah Normalisasi**

| *Age* | *Sex* | *Cp* | *Trestbps* | *Chol* | *Fbs* | *Restecg* | *Thalach* | *Exang* | *Oldpeak* | *Slope* |
|---|---|---|---|---|---|---|---|---|---|---|
| 0.818 | 1 | 0.25 | 0.725 | 0.413 | 1 | 1 | 0.743 | 0 | 0.371 | 1.000 |
| 0.831 | 1 | 0.25 | 0.550 | 0.374 | 0 | 1 | 0.713 | 1 | 0.290 | 0.667 |
| 0.753 | 0 | 0.25 | 0.750 | 0.502 | 1 | 1 | 0.802 | 0 | 0.161 | 0.333 |
| 0.857 | 0 | 0.25 | 0.750 | 0.401 | 0 | 0 | 0.564 | 0 | 0.419 | 1.000 |
| 0.896 | 0 | 0.25 | 0.700 | 0.424 | 0 | 0 | 0.748 | 0 | 0.290 | 0.333 |
| 0.519 | 1 | 0.25 | 0.700 | 0.353 | 0 | 0 | 0.881 | 1 | 0.226 | 0.333 |
| 0.662 | 1 | 0.25 | 0.625 | 0.378 | 0 | 1 | 0.619 | 1 | 0.226 | 0.333 |
| 0.442 | 1 | 0.25 | 0.590 | 0.323 | 0 | 1 | 0.861 | 0 | 0.000 | 0.333 |
| 0.675 | 1 | 0.25 | 0.590 | 0.330 | 0 | 1 | 0.941 | 0 | 0.000 | 0.667 |
| 0.844 | 1 | 0.25 | 0.690 | 0.500 | 1 | 1 | 0.861 | 0 | 0.226 | 0.667 |
| 0.766 | 1 | 0.25 | 0.850 | 0.511 | 0 | 1 | 0.787 | 0 | 0.032 | 0.667 |
| 0.675 | 1 | 0.25 | 0.760 | 0.528 | 1 | 0 | 0.881 | 0 | 0.194 | 0.667 |
| … | … | … | … | … | … | … | … | … | … | … |
| 0.740 | 0 | 1 | 0.700 | 0.427 | 0 | 0 | 0.609 | 1 | 0.032 | 0.667 |
| 0.883 | 1 | 1 | 0.720 | 0.342 | 1 | 0 | 0.698 | 0 | 0.548 | 0.667 |
| 0.740 | 1 | 1 | 0.650 | 0.232 | 0 | 0 | 0.569 | 1 | 0.194 | 0.667 |

### 3.3 Implementasi K-Means *Clustering*

#### 3.3.1 Menentukan jumlah *cluster* (*k*)

Berdasarkan hasil wawancara dengan ahli medis, penentuan jumlah *cluster* (*k*) dalam analisis K-Means *cluster*ing dilakukan dengan mempertimbangkan atribut *Cp* (*Chest Pain*), yang merupakan faktor risiko utama penyakit jantung. Untuk tahap awal, diputuskan untuk menggunakan 4 *cluster*, sesuai dengan empat tingkat nyeri dada yang terdefinisi pada atribut *Cp* (*k*=4). Sebelum implementasi algoritma K-Means, *dataset* disusun terlebih dahulu dengan cara menyortir data berdasarkan urutan nilai atribut *Cp*. Proses penyortiran ini ditampilkan pada Tabel 4 dan dilakukan untuk memastikan bahwa pengelompokan data lebih terarah dan relevan dengan risiko penyakit jantung yang dikaitkan dengan nyeri dada.

#### 3.3.2 Menentukan titik pusat *cluster* awal

Tahap ini merupakan tahap iterasi 1 dengan penentuan titik pusat *cluster* atau *centroid* awal. Penentuan *centroid* awal dilakukan secara acak pada *dataset* penyakit jantung yang berjumlah 303 pasien. *Pada centroid* (*c*1) merupakan titik pusat pada *cluster* (*k*1), *centroid* (*c*2) merupakan titik pusat pada *cluster* (*k*2), *centroid* (*c*2) merupakan titik pusat pada *cluster* (*k*2), *centroid* (*c*3) merupakan titik pusat pada *cluster* (*k*3), *centroid* (*c*4) merupakan titik pusat pada *cluster* (*k*4). *Centroid* awal setiap *cluster* disajikan pada Tabel 4.

**Tabel 4 *Centroid* Awal**

| *Centroid* | *Age* | *Sex* | *Cp* | *Trestbps* | *Chol* | *Fbs* | *Restecg* | *Thalach* | *Exang* | *Oldpeak* | *Slope* |
|---|---|---|---|---|---|---|---|---|---|---|---|
| c1 | 0.844 | 1 | 0.25 | 0.69 | 0.500 | 1 | 1 | 0.861 | 0 | 0.226 | 0.667 |
| c2 | 0.818 | 0 | 0.5 | 0.7 | 0.346 | 0 | 0 | 0.886 | 0 | 0 | 0.333 |
| c3 | 0.623 | 1 | 0.75 | 0.62 | 0.452 | 1 | 0 | 0.866 | 0 | 0 | 0.333 |
| c4 | 0.844 | 1 | 1 | 0.55 | 0.440 | 0 | 1 | 0.782 | 0 | 0.097 | 0.333 |

#### 3.3.3 Menghitung jarak data ke *centroid* setiap *cluster*

Perhitungan jarak data pertama dengan titik *centroid* awal (Tabel 4) menggunakan Pers. (2). Pada tahap ini, jarak data pertama dihitung terhadap masing-masing *centroid* dari setiap *cluster*. Pertama, dihitung jarak data pertama dengan *centroid* pertama (*cluster* 1) yang dinyatakan sebagai $d(1,1)$. Selanjutnya, jarak data pertama dengan *centroid* kedua (*cluster* 2) dihitung sebagai $d(1,2)$, diikuti dengan jarak ke *centroid* ketiga (*cluster* 3) yang ditunjukkan oleh $d(1,3)$. Terakhir, jarak data pertama dengan *centroid* keempat (*cluster* 4) dihitung sebagai $d(1,4)$. Proses ini memastikan bahwa setiap jarak antara data dan *centroid* dapat dianalisis untuk menentukan *cluster* yang paling relevan bagi data tersebut.

$$d\,(x_i, c_h) = \sqrt{(x_{i1} - c_{h1})^2 + (x_{i2} - c_{h2})^2 + (x_{i3} - c_{h3})^2 + \ldots + (x_{im} - c_{hm})^2}$$

$$d(1,1) = \sqrt{(0.818 - 0.844)^2 + (1-1)^2 + (0.25 - 0.25)^2 + (0.725 - 0.69)^2 + (0.413 - 0.500)^2 + (1-1)^2 + (1-1)^2 + (0.743 - 0.861)^2 + (0-0)^2 + (0.371 - 0.226)^2 + (1 - 0.667)^2}$$

$$d(1,1) = 0.66$$

$$d(1,2) = \sqrt{(0.818 - 0.818)^2 + (1-0)^2 + (0.25 - 0.5)^2 + (0.725 - 0.7)^2 + (0.413 - 0.346)^2 + (1-0)^2 + (1-0)^2 + (0.743 - 0.886)^2 + (0-0)^2 + (0.371 - 0)^2 + (1 - 0.333)^2}$$

$$d(1,2) = 1.75$$

$$d(1,3) = \sqrt{(0.818 - 0.623)^2 + (1-1)^2 + (0.25 - 0.75)^2 + (0.725 - 0.62)^2 + (0.413 - 0.452)^2 + (1-1)^2 + (1-0)^2 + (0.743 - 0.886)^2 + (0-0)^2 + (0.371 - 0.0)^2 + (1 - 0.333)^2}$$

$$d(1,3) = 1.18$$

$$d(1,4) = \sqrt{\begin{matrix}(0.818-0.844)^2+\ (1-1)^2+\ (0.25-1)^2+\ (0.725-0.55)^2+(0.413-0.440)^2+(1-0)^2+(1-1)^2+\\ (0.743-0.782)^2+(0-0)^2+\ (0.371-0.097)^2+(1-0.333)^2\end{matrix}}$$

$$d(1,4) = 1.05$$

Perhitungan jarak ($d$) dari data pertama terhadap *centroid* awal (*c1*, *c2*, *c3*, dan *c4*) yang terdiri dari 11 atribut menunjukkan bahwa data pertama dikelompokkan ke dalam *cluster* 1. Hal ini disebabkan oleh jarak antara data pertama dengan *centroid* pertama *cluster* 1, yaitu $d(1,1)$, yang menghasilkan nilai terdekat sebesar 0,66. Sementara itu, jarak data pertama dengan *centroid* kedua *cluster* 2 $d(1,2)$ bernilai 1,75, jarak dengan *centroid* ketiga *cluster* 3 $d(1,3)$ bernilai 1,18, dan jarak dengan *centroid* keempat *cluster* $d(1,4)$ bernilai 1,05. Proses perhitungan jarak untuk data kedua hingga data terakhir dilakukan dengan menggunakan metode yang sama seperti pada perhitungan $d(1,1)$ hingga $d(1,4)$. Hasil dari *cluster* pada iterasi pertama berdasarkan perhitungan jarak ditampilkan pada Tabel 5.

**Tabel 5 Hasil *Cluster* Iterasi 1**

| *Id* | *Age* | *Sex* | *Cp* | *Trestbps* | *Chol* | *Fbs* | *Restecg* | *Thalach* | *Exang* | *Oldpeak* | *Slope* | *Cluster* |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 117 | 0.753 | 1 | 0.75 | 0.7 | 0.374 | 1 | 1 | 0.817 | 0 | 0 | 0.333 | 1 |
| 125 | 0.844 | 1 | 0.25 | 0.25 | 0.5 | 1 | 1 | 0.861 | 0 | 0.226 | 0.667 | 1 |
| 140 | 0.662 | 1 | 0.75 | 0.625 | 0.434 | 1 | 1 | 0.822 | 0 | 0.387 | 0.667 | 1 |
| … | … | … | … | … | … | … | … | … | … | … | … | … |
| 262 | 0.753 | 0 | 0.5 | 0.68 | 0.566 | 1 | 1 | 0.752 | 0 | 0 | 0.333 | 1 |
| 214 | 0.857 | 0 | 1 | 0.89 | 0.404 | 1 | 0 | 0.817 | 1 | 0.161 | 0.667 | 2 |
| 6 | 0.727 | 1 | 0.5 | 0.6 | 0.418 | 0 | 0 | 0.881 | 0 | 0.129 | 0.333 | 2 |
| 14 | 0.571 | 1 | 0.5 | 0.6 | 0.466 | 0 | 0 | 0.856 | 0 | 0 | 0.333 | 2 |
| … | … | … | … | … | … | … | … | … | … | … | … | … |
| 295 | 0.818 | 0 | 1 | 0.62 | 0.349 | 0 | 0 | 0.673 | 1 | 0 | 0.667 | 2 |
| 32 | 0.779 | 1 | 1 | 0.585 | 0.408 | 1 | 0 | 0.792 | 1 | 0.226 | 0.333 | 3 |
| 40 | 0.792 | 1 | 0.75 | 0.75 | 0.431 | 1 | 0 | 0.678 | 1 | 0.161 | 0.667 | 3 |
| 267 | 0.675 | 1 | 1 | 0.64 | 0.362 | 1 | 0 | 0.722 | 1 | 0.161 | 0.667 | 3 |
| … | … | … | … | … | … | … | … | … | … | … | … | … |
| 300 | 0.883 | 1 | 1 | 0.72 | 0.342 | 1 | 0 | 0.698 | 0 | 0.548 | 0.667 | 3 |
| 2 | 0.870 | 1 | 1 | 0.8 | 0.507 | 0 | 1 | 0.535 | 1 | 0.242 | 0.667 | 4 |
| 3 | 0.870 | 1 | 1 | 0.6 | 0.406 | 0 | 1 | 0.639 | 1 | 0.419 | 0.667 | 4 |
| 9 | 0.818 | 1 | 1 | 0.65 | 0.450 | 0 | 1 | 0.728 | 0 | 0.226 | 0.667 | 4 |
| … | … | … | … | … | … | … | … | … | … | … | … | … |
| 254 | 0.662 | 0 | 0.75 | 0.6 | 0.523 | 0 | 1 | 0.777 | 0 | 0.097 | 0.333 | 4 |

Berdasarkan hasil *cluster* pada iterasi pertama, dilakukan perhitungan rata-rata untuk setiap *cluster* pada 11 atribut, yang disajikan dalam Tabel 6. Proses ini menghasilkan *centroid* baru untuk masing-masing *cluster*. Hasilnya menunjukkan bahwa *cluster* 1 (*k*1) terdiri dari 27 pasien, *cluster* 2 (*k*2) berjumlah 96 pasien, *cluster* 3 (*k*3) memiliki 15 pasien, dan *cluster* 4 (*k*4) mencakup 165 pasien. Penentuan jumlah pasien dalam setiap *cluster* ini penting untuk memahami distribusi data dan karakteristik masing-masing kelompok dalam analisis penyakit jantung.

**Tabel 6 Hasil *Centroid* Iterasi 1**

| *k* | *Age* | *Sex* | *Cp* | *Trestbps* | *Chol* | *Fbs* | *Restecg* | *Thalach* | *Exang* | *Oldpeak* | *Slope* | *Jumlah Data* |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0.748 | 0.704 | 0.731 | 0.717 | 0.467 | 0.963 | 1 | 0.716 | 0.444 | 0.210 | 0.617 | 27 |
| 2 | 0.688 | 0.354 | 0.667 | 0.652 | 0.424 | 0.042 | 0.120 | 0.775 | 0.188 | 0.099 | 0.490 | 96 |
| 3 | 0.716 | 1 | 0.767 | 0.665 | 0.391 | 1 | 0 | 0.798 | 0.2 | 0.146 | 0.511 | 15 |
| 4 | 0.711 | 0.836 | 0.873 | 0.652 | 0.445 | 0 | 0.676 | 0.720 | 0.4 | 0.203 | 0.547 | 165 |

Tabel 6 menunjukan data *centroid* setiap *cluster* (*k*) beserta jumlah data. Perhitungan iterasi kedua menggunakan proses yang sama seperti pada tahap pertama dengan perhitungan jarak setiap data dengan *centroid* baru pada Tabel 6. Perhitungan K-Means *clustering* berakhir pada iterasi keenam karena *centroid* baru pada iterasi ini tidak berubah dari *centroid* sebelumnya seperti yang terlihat pada tabel 8. Hasil *cluster* ditunjukkan pada Tabel 7.

Perhitungan rata-rata setiap atribut pada iterasi keenam menghasilkan *centroid* baru dengan menggunakan rumus AVERAGE di Excel, yang berfungsi untuk menghitung rata-rata. Hasil perhitungan tersebut ditampilkan dalam Tabel 8, yang menunjukkan data *centroid* untuk setiap *cluster* (*k*) beserta jumlah pasien dalam masing-masing *cluster*. Dalam hasil ini, *cluster* 1 (*k*1) tediri 27 pasien, *cluster* 2 (*k*2) berjumlah 135 pasien, *cluster* 3 (*k*3) memiliki 15 pasien, *cluster* 4

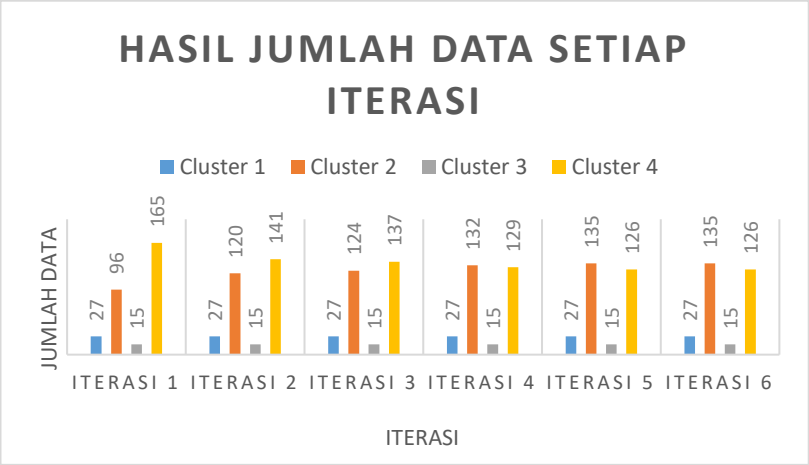
(*k*4) mencakup 126 pasien. Hasil *centroid* dan jumlah data pada *cluster* iterasi 5 sama dengan hasil iterasi 6 sehingga perhitungan dihentikan dan dinyatakan telah selesai. Jumlah data pada setiap iterasi ditampilkan dalam diagram batang yang disajikan pada Gambar 3.

**Tabel 7 Hasil *Cluster* Iterasi 6**

| *Id* | *Age* | *Sex* | *Cp* | *Trestbps* | *Chol* | *Fbs* | *Restecg* | *Thalach* | *Exang* | *Oldpeak* | *Slope* | *Cluster* |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 117 | 0.753 | 1 | 0.75 | 0.7 | 0.374 | 1 | 1 | 0.817 | 0 | 0 | 0.333 | 1 |
| 125 | 0.844 | 1 | 0.25 | 0.25 | 0.5 | 1 | 1 | 0.861 | 0 | 0.226 | 0.667 | 1 |
| 140 | 0.662 | 1 | 0.75 | 0.625 | 0.434 | 1 | 1 | 0.822 | 0 | 0.387 | 0.667 | 1 |
| … | … | … | … | … | … | … | … | … | … | … | … | … |
| 262 | 0.753 | 0 | 0.5 | 0.68 | 0.566 | 1 | 1 | 0.752 | 0 | 0 | 0.333 | 1 |
| 4 | 0.481 | 1 | 0.75 | 0.65 | 0.443 | 0 | 0 | 0.926 | 0 | 0.565 | 1 | 2 |
| 6 | 0.727 | 1 | 0.5 | 0.6 | 0.418 | 0 | 0 | 0.881 | 0 | 0.129 | 0.333 | 2 |
| 11 | 0.740 | 1 | 1 | 0.7 | 0.340 | 0 | 0 | 0.733 | 0 | 0.065 | 0.667 | 2 |
| … | … | … | … | … | … | … | … | … | … | … | … | … |
| 122 | 0.818 | 0 | 1 | 0.75 | 0.722 | 0 | 1 | 0.762 | 0 | 0.645 | 0.667 | 2 |
| 32 | 0.779 | 1 | 1 | 0.585 | 0.408 | 1 | 0 | 0.792 | 1 | 0.226 | 0.333 | 3 |
| 40 | 0.792 | 1 | 0.75 | 0.75 | 0.431 | 1 | 0 | 0.678 | 1 | 0.161 | 0.667 | 3 |
| 267 | 0.675 | 1 | 1 | 0.64 | 0.362 | 1 | 0 | 0.722 | 1 | 0.161 | 0.667 | 3 |
| … | … | … | … | … | … | … | … | … | … | … | … | … |
| 300 | 0.883 | 1 | 1 | 0.72 | 0.342 | 1 | 0 | 0.698 | 0 | 0.548 | 0.667 | 3 |
| 283 | 0.714 | 0 | 1 | 0.64 | 0.363 | 0 | 0.5 | 0.644 | 1 | 0.323 | 0.667 | 4 |
| 2 | 0.870 | 1 | 1 | 0.8 | 0.507 | 0 | 1 | 0.535 | 1 | 0.242 | 0.667 | 4 |
| 3 | 0.870 | 1 | 1 | 0.6 | 0.406 | 0 | 1 | 0.639 | 1 | 0.419 | 0.667 | 4 |
| … | … | … | … | … | … | … | … | … | … | … | … | … |
| 62 | 0.597 | 0 | 0.75 | 0.71 | 0.314 | 0 | 1 | 0.792 | 1 | 0.226 | 1 | 4 |

**Tabel 8 Hasil *Centroid* Iterasi 6**

| *k* | *Age* | *Sex* | *Cp* | *Trestbps* | *Chol* | *Fbs* | *Restecg* | *Thalach* | *Exang* | *Oldpeak* | *Slope* | *Jumlah Data* |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0.751 | 0.667 | 0.759 | 0.717 | 0.464 | 1 | 0.963 | 0.719 | 0.481 | 0.191 | 0.605 | 27 |
| 2 | 0.694 | 0.415 | 0.757 | 0.645 | 0.437 | 0.022 | 0.226 | 0.763 | 0.059 | 0.118 | 0.489 | 135 |
| 3 | 0.716 | 1 | 0.767 | 0.665 | 0.391 | 1 | 0 | 0.798 | 0.200 | 0.146 | 0.511 | 15 |
| 4 | 0.711 | 0.929 | 0.833 | 0.659 | 0.438 | 0 | 0.742 | 0.714 | 0.595 | 0.219 | 0.569 | 126 |




**Gambar 3 Hasil Jumlah Data Setiap Iterasi**

Pada Gambar 3, terlihat bahwa proses iterasi dari 1 hingga 6 menunjukkan jumlah data yang konsisten pada *cluster* 1, yaitu sebanyak 27 pasien, dan pada *cluster* 3, yang terdiri dari 15 pasien. Namun, terdapat perubahan jumlah pasien pada *cluster* 2 dan *cluster* 4. Pada akhir iterasi, jumlah pasien di *cluster* 2 mencapai 135 pasien, sedangkan *cluster* 4 berjumlah 126 pasien. Perubahan ini mencerminkan dinamika pengelompokan data selama proses iterasi, di mana *cluster* 2 dan *cluster* 4 mengalami penyesuaian jumlah pasien sesuai dengan perhitungan *centroid* yang dilakukan.

### 3.4 Analisis Hasil *Cluster*

Pada hasil iterasi keenam, telah diperoleh *cluster* dan *centroid* berdasarkan data yang telah dinormalisasi. Namun, untuk melakukan analisis yang lebih komprehensif, penting untuk menerjemahkan hasil *cluster* kembali ke dalam bentuk data awal atau data sebelum dinormalisasi. Tabel 9 menyajikan terjemahan dari data hasil *cluster*, sementara Tabel 10 menampilkan terjemahan *centroid* pada iterasi keenam.

**Tabel 9 Terjemahan Data Hasil *Cluster* Iterasi 6**

| *Id* | *Age* | *Sex* | *Cp* | *Trestbps* | *Chol* | *Fbs* | *Restecg* | *Thalach* | *Exang* | *Oldpeak* | *Slope* | *Cluster* |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 117 | 58 | 1 | 3 | 140 | 211 | 1 | 2 | 165 | 0 | 0 | 1 | 1 |
| 125 | 65 | 1 | 1 | 138 | 282 | 1 | 2 | 174 | 0 | 1.4 | 1.4 | 1 |
| 140 | 51 | 1 | 3 | 125 | 245 | 1 | 2 | 166 | 0 | 2.4 | 2.4 | 1 |
| … | … | … | … | … | … | … | … | … | … | … | … | … |
| 262 | 58 | 0 | 2 | 136 | 319 | 1 | 2 | 152 | 0 | 0 | 0 | 1 |
| 4 | 37 | 1 | 3 | 130 | 250 | 0 | 0 | 187 | 0 | 3.5 | 3.5 | 2 |
| 6 | 56 | 1 | 2 | 120 | 236 | 0 | 0 | 178 | 0 | 0.8 | 0.8 | 2 |
| 11 | 57 | 1 | 4 | 140 | 192 | 0 | 0 | 148 | 0 | 0.4 | 2 | 2 |
| … | … | … | … | … | … | … | … | … | … | … | … | … |
| 122 | 63 | 0 | 4 | 150 | 407 | 0 | 2 | 154 | 0 | 4 | 2 | 2 |
| 32 | 60 | 1 | 4 | 117 | 230 | 1 | 0 | 160 | 1 | 1.4 | 1 | 3 |
| 40 | 61 | 1 | 4 | 150 | 243 | 1 | 0 | 137 | 1 | 1 | 2 | 3 |
| 267 | 52 | 1 | 3 | 128 | 204 | 1 | 0 | 156 | 1 | 1 | 2 | 3 |
| … | … | … | … | … | … | … | … | … | … | … | … | … |
| 300 | 68 | 1 | 4 | 144 | 193 | 1 | 0 | 141 | 0 | 3.4 | 2 | 3 |
| 283 | 55 | 0 | 4 | 128 | 205 | 0 | 1 | 130 | 1 | 2 | 2 | 4 |
| 2 | 67 | 1 | 4 | 160 | 286 | 0 | 2 | 108 | 1 | 1.5 | 2 | 4 |
| 3 | 67 | 1 | 4 | 120 | 229 | 0 | 2 | 129 | 1 | 2.6 | 2 | 4 |
| … | … | … | … | … | … | … | … | … | … | … | … | … |
| 62 | 46 | 0 | 3 | 142 | 177 | 0 | 2 | 160 | 1 | 1.4 | 13 | 4 |

**Tabel 10 Terjemahan Data Hasil *Centroid* Iterasi 6**

| *k* | *Age* | *Sex* | *Cp* | *Trestbps* | *Chol* | *Fbs* | *Restecg* | *Thalach* | *Exang* | *Oldpeak* | *Slope* | *Jumlah Data* |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 58 | 1 | 3 | 143 | 262 | 1 | 2 | 145 | 0 | 1.2 | 2 | 27 |
| 2 | 53 | 0 | 3 | 129 | 246 | 0 | 0 | 154 | 0 | 0.7 | 1 | 135 |
| 3 | 55 | 1 | 3 | 133 | 220 | 1 | 0 | 161 | 0 | 0.9 | 2 | 15 |
| 4 | 55 | 1 | 3 | 132 | 247 | 0 | 1 | 144 | 1 | 1.4 | 2 | 126 |

Interpertasi data hasil *cluster* pada Tabel 9 dan Tabel 10 yang dilakukan bersama ahli medis dihasilkan bahwa *cluster* 1 (*k*1) memiliki rata-rata usia 58 tahun, mayoritas laki-laki, dengan tipe nyeri dada (*Cp*) yang menunjukkan gejala nyeri dada parah. Pasien dalam kelompok ini cenderung memiliki tekanan darah istirahat (*Trestbps*) dan kadar *cholesterol* (*Chol*) yang tinggi, serta *fasting blood sugar* (*Fbs*) yang lebih dari 120 mg/dl. *Resting electrocardiographic* (*Restecg*) dan *exercise-induced angina* (*Exang*) menunjukkan tingkat abnormalitas yang cukup signifikan. Tingkat detak jantung maksimum (*Thalach*) cenderung rendah, dan nilai *Oldpeak* yang tinggi mengindikasikan depresi ST yang mungkin menjadi tanda risiko lebih tinggi untuk penyakit jantung. *Slope* puncak ST setelah berolah raga (*Slope*) cenderung meningkat seiring dengan peningkatan tingkat nyeri dada.

*Cluster* 2 (*k*2), dengan rata-rata usia 53 tahun, mayoritas perempuan, menunjukkan tipe nyeri dada yang cenderung parah. Pasien dalam kelompok ini memiliki tekanan darah (*Trestbps*) dan kadar *cholesterol* (*Chol*) yang relatif normal, *fasting blood sugar* (*Fbs*) rendah, serta *resting electrocardiographic* (*Restecg*) dan *exercise-induced angina* (*Exang*) yang cenderung normal. Tingkat detak jantung maksimum (*Thalach*) dan nilai *Oldpeak* yang rendah menunjukkan adanya respon yang lebih baik terhadap aktivitas fisik. *Slope* puncak ST setelah berolah raga (*Slope*) cenderung datar.

*Cluster* 3 (*k*3), dengan rata-rata usia 55 tahun, mayoritas laki-laki, menunjukkan tipe nyeri dada yang cenderung parah. Pasien dalam kelompok ini memiliki tekanan darah (*Trestbps*) dan kadar *cholesterol* (*Chol*) yang relatif normal, *fasting blood sugar* (*Fbs*) tinggi, serta *resting electrocardiographic* (*Restecg*) yang normal. *Exercise-induced angina* (*Exang*) cenderung rendah. Tingkat detak jantung maksimum (*Thalach*) dan nilai *Oldpeak* menunjukkan respon yang baik terhadap aktivitas fisik. *Slope* puncak ST setelah berolah raga (*Slope*) cenderung meningkat.

*Cluster* 4 (*k*4), dengan rata-rata usia 55 tahun, mayoritas laki-laki, menunjukkan tipe nyeri dada yang cenderung parah. Pasien dalam kelompok ini memiliki tekanan darah (*Trestbps*) dan kadar *cholesterol* (*Chol*) yang relatif normal, *fasting blood sugar* (*Fbs*) rendah, serta *resting electrocardiographic* (*Restecg*) dan *exercise-induced angina* (*Exang*) yang cenderung tinggi. Tingkat detak jantung maksimum (*Thalach*) dan nilai *Oldpeak* menunjukkan respon yang beragam, sementara *Slope* puncak ST setelah berolah raga (*Slope*) cenderung rendah.

Berdasarkan hasil tersebut, dapat disarankan bahwa *cluster k*1 menunjukan tingkat risiko penyakit jantung yang lebih tinggi karena pasien dalam kelompok ini memiliki resiko sangat tinggi terhadap penyakit jantung, pasien menunjukan gejala serius seperti nyeri dada (*Cp*) tinggi, tekanan darah (*Trestbps*) tinggi, *cholesterol* (*Chol*) tinggi, *fasting blood sugar* (*Fbs*) tinggi, *electrocardiographic* (*Restecg*) tinggi dan faktor resiko lainnya, sementara *k*2, *k*3, dan *k*4 menunjukkan risiko yang lebih rendah, dengan variasi respon terhadap aktivitas fisik. Pengelompokan ini dapat memberikan informasi awal untuk merancang strategi pengelolaan dan perawatan yang lebih spesifik sesuai dengan karakteristik dari setiap kelompok. Tetapi, perlu diingat bahwa validasi klinis lebih lanjut dan pertimbangan medis lebih mendalam tetap diperlukan untuk penanganan pasien secara lebih tepat dan efektif.

## 4. KESIMPULAN

Kesimpulan dari penelitian ini menunjukkan bahwa *cluster k*1 memiliki profil risiko yang paling tinggi. Pasien dalam kelompok *k*1 sebagian besar adalah laki-laki dengan rata-rata usia lebih tua yang menunjukkan gejala nyeri dada parah, tekanan darah tinggi, kadar kolesterol tinggi. Sementara itu, *cluster k*2, *k*3, dan *k*4 menunjukkan profil risiko yang lebih rendah dengan parameter kesehatan yang lebih normal dan respon yang baik terhadap aktivitas fisik, meskipun nyeri dada tetap menjadi gejala yang dominan.
Dalam sintesis hasil penelitian, dapat disimpulkan bahwa pengelompokan pasien berdasarkan karakteristik klinis dan demografis dapat memberikan wawasan penting untuk strategi pengelolaan dan perawatan yang lebih terarah. *Cluster k*1 memerlukan intervensi medis yang lebih intensif dan pemantauan ketat untuk mengelola faktor risiko yang tinggi, sedangkan *cluster k*2, *k*3, dan *k*4 memerlukan pendekatan yang lebih disesuaikan dengan profil risiko masing-masing. Pendekatan yang berbeda ini memungkinkan penyedia layanan kesehatan untuk memberikan perawatan yang lebih efektif dan efisien, serta meningkatkan kualitas hidup pasien melalui pengelolaan penyakit yang lebih personal dan tepat sasaran. Validasi klinis lebih lanjut diperlukan untuk memastikan bahwa pendekatan ini dapat diterapkan secara luas dan memberikan manfaat yang maksimal bagi pasien.

## DAFTAR PUSTAKA

Ali, M. M., Paul, B. K., Ahmed, K., Bui, F. M., Quinn, J. M. W., & Moni, M. A. (2021). Heart disease prediction using supervised machine learning algorithms: Performance analysis and comparison. *Computers in Biology and Medicine*, *136*, 104672. https://doi.org/10.1016/j.compbiomed.2021.104672

Arifandi, M., Hermawan, A., Hermawan, A., Avianto, D., & Avianto, D. (2021). Implementasi Algoritma K-Medoids untuk Clustering Wilayah Terinfeksi Kasus Covid-19 di DKI Jakarta. *JTT (Jurnal Teknologi Terapan)*, *7*(2), 120–128. https://doi.org/10.31884/jtt.v7i2.353

Awan, A. A. (2020). *Heart Disease patients*. Kaggle. https://www.kaggle.com/datasets/kingabzpro/heart-disease-patients

Das, D., Kayal, P., & Maiti, M. (2023). A K-means clustering model for analyzing the Bitcoin extreme value returns. *Decision Analytics Journal*, *6*(2022), 100152. https://doi.org/10.1016/j.dajour.2022.100152

Han, J., & Kang, S. (2023). Optimization of missing value imputation for neural networks. *Information Sciences*, *649*, 119668. https://doi.org/10.1016/j.ins.2023.119668

Haris Kurniawan, Sarjon Defit, & Sumijan. (2020). Data Mining Menggunakan Metode K-Means Clustering Untuk Menentukan Besaran Uang Kuliah Tunggal. *Journal of Applied Computer Science and Technology*, *1*(2), 80–89. https://doi.org/10.52158/jacost.v1i2.102

Ikotun, A. M., Ezugwu, A. E., Abualigah, L., Abuhaija, B., & Heming, J. (2023). K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data. *Information Sciences*, *622*, 178–210. https://doi.org/10.1016/j.ins.2022.11.139

Masitha, A., Biddinika, M. K., & Herman, H. (2023). K Value Effect on Accuracy Using the K-NN for Heart Failure Dataset. *MATRIK : Jurnal Manajemen, Teknik Informatika Dan Rekayasa Komputer*, *22*(3), 593–604. https://doi.org/10.30812/matrik.v22i3.2984

Mishra, P., Biancolillo, A., Roger, J. M., Marini, F., & Rutledge, D. N. (2020). New data preprocessing trends based on ensemble of multiple preprocessing techniques. *TrAC Trends in Analytical Chemistry*, *132*, 116045. https://doi.org/10.1016/j.trac.2020.116045

Muslimah, V. (2024). Implementing Bayes' Theorem Method in Expert System to Determine Infant Disease. *Khazanah Informatika : Jurnal Ilmu Komputer Dan Informatika*, *10*(1), 1–14. https://doi.org/10.23917/KHIF.V10I1.4837

Novidianto, R., Wibowo, H., & Chandranegara, D. R. (2021). ClusterMix K-Prototypes Algorithm to Capture Variable Characteristics of Patient Mortality With Heart Failure. *Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control*, *6*(2), 109–116. https://doi.org/10.22219/kinetik.v6i2.1209

Purba, N., Poningsih, P., & Tambunan, H. S. (2021). Penerapan Algoritma K-Means Clustering Pada Penyebaran Penyakit Infeksi Saluran Pernapasan Akut (ISPA) di Provinsi Riau. *Journal of Information System Research (JOSH)*, *2*(3), 220–226. https://ejurnal.seminar-id.com/index.php/josh/article/view/736

Qi, K.-T., Zhang, H.-S., Zheng, Y.-G., Zhang, Y., & Ding, L.-Y. (2023). Stripe segmentation of oceanic internal waves in SAR images based on Gabor transform and K-means clustering. *Oceanologia*, *65*(4), 548–555. https://doi.org/10.1016/j.oceano.2023.06.006

Rizki, B., Ginasta, N. G., Tamrin, M. A., & Rahman, A. (2020). Customer Loyality Segmentation on Point of Sale System Using Recency-Frequency-Monetary (RFM) and K-Means. *Jurnal Online Informatika*, *5*(2), 130–136. https://doi.org/10.15575/join.v5i2.511

Shah, D., Patel, S., & Bharti, S. K. (2020). Heart Disease Prediction using Machine Learning Techniques. *SN Computer Science*, *1*(6), 345. https://doi.org/10.1007/s42979-020-00365-y

Singh, A., & Kumar, R. (2020). Heart Disease Prediction Using Machine Learning Algorithms. *2020 International Conference on Electrical and Electronics Engineering (ICE3)*, 452–457. https://doi.org/10.1109/ICE348803.2020.9122958

Solechati, R. G., & Jananto, A. (2023). Penerapan Algoritma K-Means Clustering pada Data Brain Stroke untuk Pengelompokan Profile Pasien. *SemanTIK*, *9*(1), 39–46. https://doi.org/10.55679/semantik.v9i1.29446

Sun, F., & Yu, J. (2021). Improved energy performance evaluating and ranking approach for office buildings using Simple-normalization, Entropy-based TOPSIS and K-means method. *Energy Reports*, *7*, 1560–1570. https://doi.org/10.1016/j.egyr.2021.03.007

V. Ramalingam, V., Dandapath, A., & Karthik Raja, M. (2018). Heart disease prediction using machine learning techniques : a survey. *International Journal of Engineering & Technology*, *7*(2.8), 684–687. https://doi.org/10.14419/ijet.v7i2.8.10557

**JISKA (Jurnal Informatika Sunan Kalijaga)**
**Vol. 9, No. 3, SEPTEMBER, 2024, Pp. 217 – 229**
**ISSN: 2527 – 5836 (print) | 2528 – 0074 (online)**

# Pelabelan Sentimen Berbasis *Semi-Supervised Learning* menggunakan Algoritma LSTM dan GRU

**Puji Ayuningtyas [(1)*], Siti Khomsah [(2)], Sudianto [(3)]**
[1,3] Teknik Informatika, Fakultas Informatika, Institut Teknologi Telkom Purwokerto, Banyumas, Indonesia
[2] Sains Data, Fakultas Informatika, Institut Teknologi Telkom Purwokerto, Banyumas, Indonesia
e-mail : {20102122,siti,sudianto}@ittelkom-pwt.ac.id.
* Penulis korespondensi.
Artikel ini diajukan 9 Mei 2024, direvisi 25 Juni 2024, diterima 26 Juni 2024, dan dipublikasikan 25 September 2024.

***Abstract***

*In the sentiment analysis research process, there are problems when still using manual labeling methods by humans (expert annotation), which are related to subjectivity, long time, and expensive costs. Another way is to use computer assistance (machine annotator). However, the use of machine annotators also has the research problem of not being able to detect sarcastic sentences. Thus, the researcher proposed a sentiment labeling method using Semi-Supervised Learning. Semi-supervised learning is a labeling method that combines human labeling techniques (expert annotation) and machine labeling (machine annotation). This research uses machine annotators in the form of Deep Learning algorithms, namely the Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) algorithms. The word weighting method used in this research is Word2Vec Continuous Bag of Word (CBoW). The results showed that the GRU algorithm tends to have a better accuracy rate than the LSTM algorithm. The average accuracy of the training results of the LSTM and GRU algorithm models is 0.904 and 0.913. In contrast, the average accuracy of labeling by LSTM and GRU is 0.569 and 0.592, respectively.*

***Keywords: Annotation, Deep Learning, GRU, LSTM, Semi-Supervised Learning, Word2Vec***

**Abstrak**

Dalam proses penelitian analisis sentimen, terdapat permasalahan yaitu ketika masih menggunakan metode pelabelan secara manual oleh manusia (*expert annotation*), yaitu terkait subjektivitas, waktu yang lama dan biaya yang mahal. Cara yang lain adalah dengan menggunakan bantuan komputer (*machine annotator*). Tetapi, penggunaan *machine annotator* juga memiliki permasalahan penelitian yaitu kurang mampu mendeteksi kalimat sarkas. Sehingga, peneliti mengusulkan metode pelabelan sentimen menggunakan *semi-supervised learning*. *Semi-supervised learning* adalah metode pelabelan di mana akan menggabungkan teknik pelabelan manusia (*expert annotation*) dan pelabelan mesin (*machine annotation*). Dalam penelitian ini menggunakan *machine annotator* berupa algoritma *deep learning* yaitu algoritma Long Short-Term Memory (LSTM) dan Gated Recurrent Unit (GRU). Metode pembobotan kata digunakan dalam penelitian ini adalah Word2Vec Continuous Bag of Word (CBoW). Hasil penelitian menunjukkan bahwa algoritma GRU cenderung memiliki tingkat akurasi yang lebih baik daripada algoritma LSTM. *Dataset* yang digunakan sebanyak 43.825 baris data dengan perbandingan pembagian data latih dan data uji sebesar 80:20. Akurasi rata-rata hasil pelatihan model algoritma LSTM dan GRU adalah 0,904 dan 0,913. Sedangkan akurasi rata-rata pelabelan oleh LSTM dan GRU masing-masing adalah sebesar 0,569 dan 0,592.

**Kata Kunci: Anotasi, *Deep Learning*, GRU, LSTM, *Semi-Supervised Learning*, Word2Vec**

## 1. PENDAHULUAN

Pada awal abad 21, perkembangan teknologi semakin masif di mana percepatan implementasi teknologi cerdas atau revolusi industri 4.0 sudah diterapkan di seluruh dunia termasuk Indonesia. Perkembangan teknologi informasi dan komunikasi ini juga mempengaruhi perubahan pola pada bidang pembayaran dari pembayaran tunai berubah menjadi non tunai. Di Indonesia, perkembangan alat pembayaran terus mengalami perubahan bentuk, mulai dari uang logam,

uang kertas, hingga mengalami evolusi berupa uang yang ditempatkan dalam media elektronik yang disebut dompet digital (Janah & Setiyawan, 2022).

Bank Indonesia mencatat lebih dari 38 aplikasi dompet digital atau *e-wallet*. Menurut laporan E-Wallet Industry Outlook 2023 dari Insight Asia, dari 1.300 warga perkotaan yang disurvei, 74% di antaranya sudah pernah menggunakan dompet digital. Persebaran popularitas aplikasi *e-wallet* pada survei tersebut adalah pada peringkat pertama aplikasi GoPay dengan pengguna sebanyak 71%, peringkat kedua yaitu aplikasi OVO dengan 70%, dan peringkat ketiga teratas yaitu aplikasi Dana dengan 61% pengguna (Setiyawan et al., 2023).

Perkembangan media teknologi informasi yang masif saat ini selaras dengan peningkatan kuantitas data yang tersedia. Banyaknya data yang tersedia seringkali memiliki kelemahan sehingga membutuhkan berbagai pemrosesan tambahan sebelum dilakukan tahap selanjutnya (Rahma & Suadaa, 2023). Salah satunya kelemahan bahwa data yang tersedia tidak banyak yang memiliki kelas label/kategori. Ketersediaan data yang tidak memiliki label banyak ditemukan pada data teks.

Pada data teks, metode pemberian kelas label dapat dilakukan secara manual, seperti yang dilakukan dalam penelitian Khomsah & Aribowo (2020). Teknik pelabelan opini pada penelitian tersebut menggunakan pelabelan manual oleh manusia. Sedangkan dalam penelitian Zhafira et al. (2021), dijelaskan bahwa pelabelan manual menggunakan *human annotator* memiliki kelemahan terkait subjektivitas. Berdasarkan penelitian tersebut, subjektivitas dapat diminimalisir dengan menambah jumlah *annotator*. Pada penelitian tersebut menggunakan 3 *annotator* yang berasal dari bidang ilmu bahasa, ilmu psikologi, dan teknik komputer. Hal tersebut membutuhkan waktu yang lama dan biaya yang mahal.

Cara yang lain adalah dengan menggunakan bantuan komputer (*machine annotator*). Dalam penelitian Bandhakavi et al. (2017) yang menggunakan metode *sentiment lexicons* dan General-Purpose Emotion Lexicons (GPELs) seperti WordNet-Affect2 sebagai *machine annotator*, dijelaskan bahwa *machine annotator* memiliki permasalahan di mana dalam media sosial (misal: Twitter) manusia lebih memilih menggunakan kosakata informal dan emoji untuk menyampaikan emosi, daripada menggunakan kosakata formal seperti pada GPEL. Selain itu, hubungan antara kata-kata dan emosi bervariasi dari satu domain ke domain lain, ini lebih dikenal dengan disambiguasi kontekstual.

Berdasarkan permasalahan yang telah dijabarkan, diperlukan sebuah solusi di mana anotasi dapat dilakukan otomatis sehingga dapat meminimalisir terjadinya disambiguasi kontekstual, sekaligus dapat mengurangi durasi dalam proses anotasi. Maka, penelitian ini bertujuan mengusulkan pemodelan anotasi (pelabelan) dengan "Pelabelan Sentimen Berbasis Semi-Supervised Learning menggunakan Deep Learning."

Semi-Supervised Learning (SSL) merupakan gabungan antara anotasi menggunakan *human annotator* dan *machine annotator.* SSL pernah dilakukan penelitian, misalnya pada penelitian Wisnalmawati et al. (2022) menggunakan metode pelabelan manual, dengan pakar sebagai manusia yang menentukan label dalam korpus. Tetapi, untuk membuat korpus berlabel lengkap dengan kualitas tinggi memerlukan banyak usaha, waktu, dan biaya, serta dapat menjadi tugas yang berat. Tujuan dilakukan penelitian ini adalah untuk mengetahui bagaimana kombinasi antara Term Frequency–Inverse Document Frequency (TF-IDF) dengan Random Forest (RF) untuk meningkatkan akurasi ketika digunakan pada SSL. Penelitian ini menjabarkan hasil penelitian bahwa pada Data1 dengan jumlah kelas data sebanyak 3, Random Forest memiliki F1-*score* 0,65 sedangkan Naive Bayes (NB) 0,62. Hal ini menunjukkan bahwa RF bekerja lebih baik daripada NB pada data 3 kelas. Sedangkan pada Data2 yang memiliki 2 kelas, NB memiliki F1-*score* 0,76 sedangkan RF 0,71.

Selanjutnya pada penelitian Anggraini et al. (2021) bertujuan untuk menerapkan metode untuk memberikan pelabelan sentimen berdasarkan kata kunci dengan tetap memperhatikan konteks

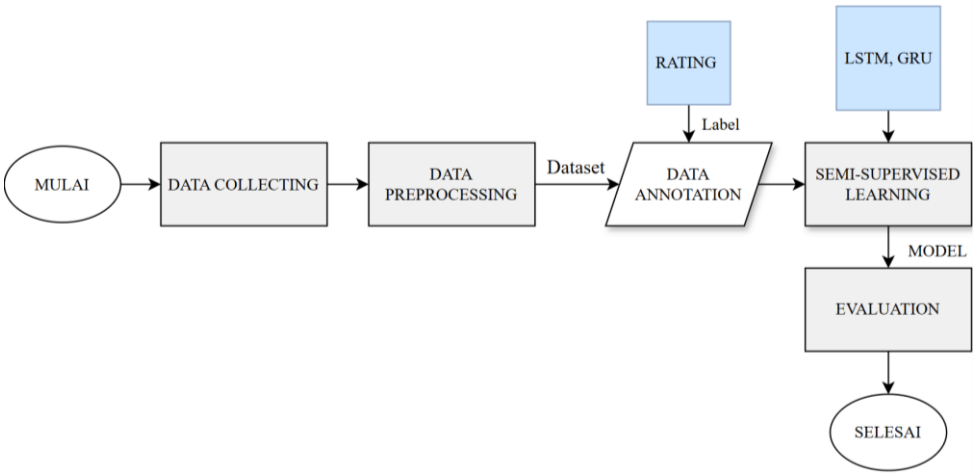
kalimat. Penelitian ini dilatarbelakangi oleh penggunaan metode Lexicon dalam pelabelan *dataset* hanya memberikan label berdasarkan makna setiap kata penyusunnya. Sehingga tidak dapat mengetahui arah konteks komentar, dan kurang dapat efektif untuk mengatasi kalimat sarkas. Metode penelitian yang diusulkan oleh peneliti menggunakan metode TF-IDF untuk *word embedding*, dan algoritma Latent Semantic Index (LSI)/Latent Semantic Analysis (LSA) untuk pemaknaan kata terhadap konteks kalimat. Pada hasil penelitian mengkombinasikan metode menggunakan TF-IDF dan LSA mampu menemukan detail permasalahan. Contohnya kata "vaksin" di TF-IDF menempati rangking pertama positif, negatif, maupun netral dengan masing-masing memiliki bobot 0,69, 0,77, dan 0,69.

Dalam penelitian ini menggunakan *machine annotator* berupa algoritma *deep learning* yaitu algoritma Long Short-Term Memory (LSTM) dan algoritma Gated Recurrent Unit (GRU). LSTM dan GRU merupakan algoritma turunan dari Recurrent Neural Network (RNN) yang banyak digunakan dalam penelitian penambangan data teks. Performa Algoritma LSTM dijelaskan melalui penelitian (Seabe et al., 2023), yang bertujuan membandingkan performa algoritma LSTM dengan algoritma algoritma Gated Reccurent Unit (GRU) yang diterapkan pada permasalahan *forecasting*. Hasil penelitian didapatkan bahwa algoritma LSTM memiliki performa yang lebih baik daripada GRU dengan hasil RMSE LSTM sebesar 0,039 dan MAPE 1031,34. Sedangkan algoritma GRU memiliki nilai RMSE sebesar 0,057 dan MAPE 1274,17.

Alasan digunakannya algoritma LSTM adalah kemampuan algoritma LSTM dalam memproses data sekuensial yang panjang sehingga mampu mengatasi permasalahan *vanishing gradient* (Oktaviani & Hustinawati, 2021), dan memiliki tingkat akurasi klasifikasi cenderung lebih baik (Ezen-Can, 2020; Romadhoni & Holle, 2022). Algoritma GRU merupakan algoritma *deep learning* yang memiliki struktur model mirip dengan Algoritma LSTM dengan versi yang lebih sederhana sehingga memiliki waktu komputasi yang labih singkat. Misalnya, pada penelitian Nosouhian et al. (2021) dengan membandingkan algoritma GRU dan LSTM dengan waktu komputasi GRU yang lebih singkat daripada LSTM pada seluruh skenario.

## 2. METODE PENELITIAN

Metode yang digunakan dalam penelitian ini ditunjukkan pada Gambar 1. Proses dimulai dengan pengumpulan data, diikuti oleh pra pemrosesan data untuk membersihkan dan menyiapkan data sebelum analisis lebih lanjut. Setelah itu, dilakukan data annotation untuk memberikan label pada data yang diperlukan. Selanjutnya, dataset dibagi menjadi beberapa bagian untuk pelatihan dan pengujian model. Proses *semi-supervised learning* diterapkan untuk memanfaatkan data berlabel dan tidak berlabel guna meningkatkan akurasi model. Akhirnya, evaluasi model dilakukan untuk menilai kinerja dan efektivitas dari algoritma yang digunakan. Metodologi ini memastikan bahwa setiap tahap dilakukan secara sistematis untuk mencapai hasil yang optimal dalam penelitian.




**Gambar 1 *Flowchart* Penelitian**

### 2.1 Pengumpulan Data (*Data Colecting*)

Data dikumpulkan melalui proses scraping aplikasi yang terdapat pada Google Play Store. Proses ini menggunakan *library* yang tersedia pada bahasa pemrograman Python, yaitu google-play-scraper. *Dataset* yang digunakan ada 3 macam, yaitu *dataset* komentar *review e-wallet* Dana, Ovo, dan GoPay dari Google PlayStore. *Dataset* yang digunakan sebanyak 43.825 baris data dengan rincian *dataset* pertama, menggunakan data yang diambil dari PlayStore pada komentar aplikasi Dana sebanyak 18.353 data. *Dataset* kedua diambil dari PlayStore pada komentar *review e-wallet* Ovo sebanyak 23.880 data. *Dataset* ketiga diambil dari PlayStore pada komentar aplikasi GoPay sebanyak 1.592 data.

### 2.2 Pra Pemrosesan Data (*Data Preprocessing*)

Pra pemrosesan data sangat penting dilakukan karena data yang diperoleh sering kali masih mengandung *noise*, seperti data *null*, data duplikat, dan berbagai ketidaksesuaian lainnya. Oleh karena itu, proses pembersihan data perlu dilakukan untuk memastikan kualitas dan keakuratan data yang akan digunakan dalam analisis. Dalam penelitian ini, data *preprocessing* terdiri dari enam tahap, yaitu:

#### 2.2.1 *Drop null* dan *drop duplicate*

*Null* dapat memengaruhi kinerja algoritma dalam melakukan proses selanjutnya. Sehingga adanya nilai kosong (*null*) pada suatu *dataset* harus dilakukan penanganan. Terdapat 2 (dua) cara yang dapat dilakukan, yaitu dengan menghapus *null*, dan mengisi *null* dengan nilai tertentu (Khatri & P, 2020). *Dataset* OVO terdapat 217 data duplikat, dan tidak memiliki data kosong (null). *Dataset* DANA terdapat 16 data duplikat, dan tidak memiliki data kosong (null). Sedangkan *dataset* GOPAY tidak terdapat data duplikat maupun data kosong, sehingga jumlah baris data tidak berkurang setelah melewati proses ini.

#### 2.2.2 *Cleaning data*

Cleaning data merupakan proses penting untuk membersihkan data dari komponen tertentu yang tidak diperlukan, seperti URL, *username*, dan *hashtags*, guna memastikan kualitas dan relevansi informasi yang akan dianalisis (Arsi & Waluyo, 2021). Proses ini membantu menghilangkan elemen yang dapat menyebabkan kebisingan dalam dataset, sehingga analisis yang dilakukan menjadi lebih akurat. Hasil dari cleaning data dapat dilihat pada Tabel 1.

**Tabel 1 Hasil *Cleaning Data***

| Sebelum *Cleaning* | Setelah *Cleaning* |
|---|---|
| Woi aplikasi gak jelas lu , susah amat dibuka, gimana mau chat | Woi aplikasi gak jelas lu susah amat dibuka gimana mau chat |

#### 2.2.3 *Case folding*

*Case folding* adalah proses *preprocessing* data teks yang bertujuan untuk mengubah seluruh huruf dalam *dataset* menjadi huruf kecil atau *lowercase*, sehingga mengurangi variasi data yang disebabkan oleh perbedaan penggunaan huruf besar dan kecil (Af'idah et al., 2021; Ayuningtyas & Tantyoko, 2024). Proses ini penting dalam analisis teks karena membantu menyederhanakan data dan memastikan bahwa istilah yang sama tidak diperlakukan sebagai entitas yang berbeda hanya karena perbedaan kapitalisasi. Hasil dari proses *case folding* dapat dilihat pada Tabel 2, yang menunjukkan perbedaan antara data sebelum dan setelah proses ini dilakukan.

**Tabel 2 Hasil *Case Folding***

| Sebelum *Case Folding* | Setelah *Case Folding* |
|---|---|
| Woi aplikasi gak jelas lu susah amat dibuka gimana mau chat | woi aplikasi gak jelas lu susah amat dibuka gimana mau chat |

### 2.2.4 *Tokenizing*

Merupakan tahap pemisahan kalimat dalam *dataset* berdasarkan tiap kata penyusunnya. Kata yang telah dipisah dari rangkaian kalimat disebut *token* atau *term* (Ayuningtyas & Tantyoko, 2024). *Term* ini akan diberikan bobot kata pada tahap pembobotan kata (Romadhoni & Holle, 2022). Hasil *tokenizing* dapat dilihat pada Tabel 3, yang menunjukkan perbedaan sebelum dan setelah proses tokenisasi.

**Tabel 3 Hasil Tokenisasi**

| Sebelum *Tokenizing* | Setelah *Tokenizing* |
|---|---|
| woi aplikasi gak jelas lu susah amat dibuka gimana mau chat | ['woi', 'aplikasi', 'gak', 'jelas', 'lu', 'susah', 'amat', 'dibuka', 'gimana', 'mau', 'chat'] |

### 2.2.5 *Slang word removal*

Penghapusan kata *slang* (*slang word removal*) bertujuan untuk mengubah kata-kata tidak baku menjadi kata baku, sehingga meningkatkan kualitas dan formalitas data yang akan dianalisis (Gifari et al., 2022). Proses ini sangat penting dalam konteks analisis teks, terutama ketika data berasal dari sumber yang tidak terstruktur, seperti media sosial atau forum *online*. Hasil dari proses *slang word removal* dapat dilihat pada Tabel 4.

**Tabel 4 Hasil *Slang Word Removal***

| Sebelum *Slang Word Removal* | Setelah *Slang Word Removal* |
|---|---|
| ['woi', 'aplikasi', 'gak', 'jelas', 'lu', 'susah', 'amat', 'dibuka', 'gimana', 'mau', 'chat'] | ['', 'aplikasi', 'tidak', 'jelas', 'kamu', 'susah', 'amat', 'dibuka', 'bagaimana', 'mau', 'pesan'] |

### 2.2.6 *Under sampling*

*Under sampling* merupakan salah satu metode yang dapat digunakan untuk mengatasi ketidakseimbangan kuantitas data antarkelas. Cara kerja *under sampling* adalah dengan membuang sampel (secara acak) dari kelas mayoritas, sehingga kuantitas dari kelas minoritas dan mayoritas akan sama (Magnolia et al., 2022). Hasil dari proses *under sampling*, diperoleh banyak *dataset* Ovo pada *rating* 1, 2, 4, dan 5 masing-masing 4.943 data, dan banyak *dataset* Ovo pada *rating* 1, 2, 4, dan 5 masing-masing 4.000 data.

### 2.3 *Data Annotation*

Setelah dilakukan *under sampling*, masuk pada tahap pengelompokan kategori kelas berdasarkan *rating* komentar, yaitu *rating* 1 dan 2 termasuk pada kategori negatif, *rating* 4 dan 5 termasuk kategori positif. Hasil yang diperoleh pada tahap pelabelan ini adalah *dataset* Ovo memiliki banyak data pada kelas positif dan negatif masing-masing sebanyak 9.886 data. Sedangkan *dataset* Dana memiliki banyak data pada kelas positif dan negatif masing-masing sebanyak 8.000 data.

### 2.4 Pembagian *Dataset*

Setelah dilakukan pengelompokan kelas menjadi kelas positif dan negatif, tahap selanjutnya adalah menggabungkan *dataset* Ovo dan Dana menjadi satu *dataset* gabungan. Penggabungan ini bertujuan untuk menyatukan data dari kedua *platform e-wallet* sehingga analisis dapat dilakukan secara lebih komprehensif. *Dataset* gabungan ini telah melalui proses *preprocessing*, seperti pembersihan data dan pengelompokan kategori kelas. Contoh *dataset* yang telah diproses dan dikelompokkan ke dalam kategori kelas positif dan negatif dapat dilihat pada Tabel 5.

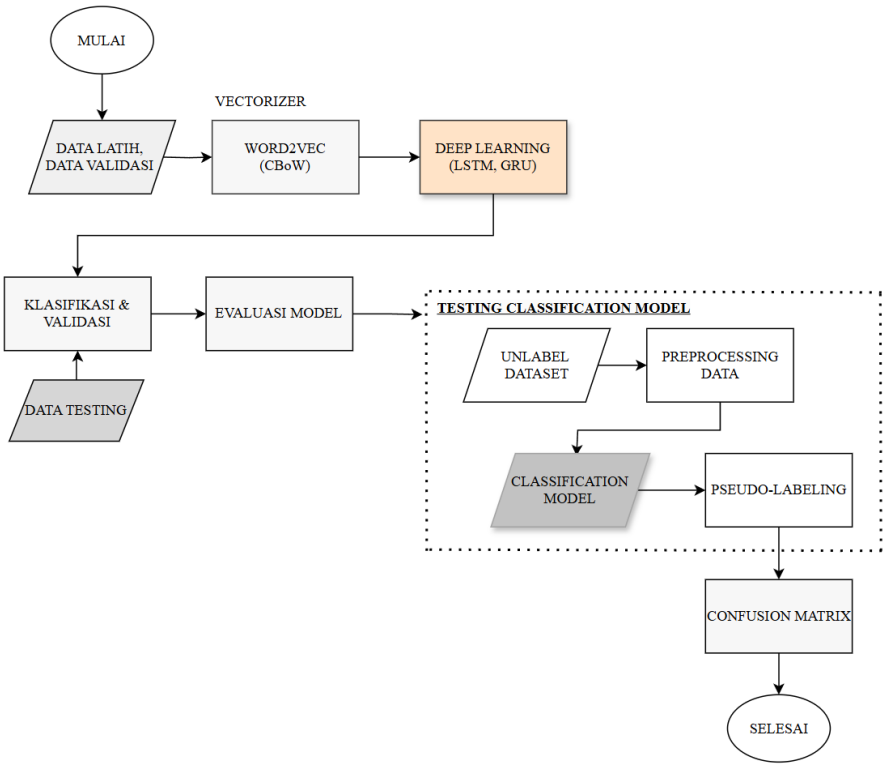
Pembagian *dataset* (*splitting data*) dilakukan setelah proses pelabelan data, yaitu dengan membagi *dataset* yang digunakan dengan perbandingan yang telah ditentukan. *Splitting data*

dengan membagi data sebanyak 35.772 ke dalam 3 (tiga) kategori, yaitu data latih, data validasi, dan data uji. Langkah pertama adalah mengambil 386 data pada masing-masing kelas untuk menjadi data uji, atau total sebanyak 772 data. Kemudian sebanyak 35.000 data dibagi menjadi data latih dan data validasi dengan perbandingan 80:20. Membagi data menjadi 80% untuk pelatihan dan 20% untuk pengujian memberikan cukup data untuk melatih model sambil menyisakan cukup data untuk menguji kinerjanya secara independen. Ini berguna dalam menentukan perbandingan yang efisien secara praktis, untuk mengantisipasi model mengalami *overfitting* (terlalu cocok dengan data latih) dan juga *underfitting* (tidak cukup belajar dari data latih). Sebelum dilakukan pembagian ke data latih dan data validasi, data dilakukan cek terhadap data duplikat dan data *null*, serta *under sampling*. Diperoleh banyak data yang akan dibagi ke data latih dan data validasi sebanyak 34.884 data. Dengan data tersebut, didapatkan data latih sebanyak 27.907 data, dan data validasi sebanyak 6.977 data.

**Tabel 5 Contoh *Dataset***

| No. | Data | *Score* |
|---|---|---|
| 1 | pelayanan buruk yang pernah saya temui di aplikasi transaksi uang masak saya transfer dana dari jam pagi jam sore tidak masuk pas ke lagi di proses terus tidak sudah aplikasi macam apa ini tolong main toko ditegur lah apa di baned sekali bukan dikit saya melakukan transaksi uang padahal itu kan uang saya sendiri dan transfer juga ke rekening saya sendiri | 0 |
| 2 | halo min biasanya transaksi dengan ovo sangat aman dan mudah tapi semalam mungkin ada gangguan jaringan dan saya ada coba isi pulsa tapi kok sampai sekarang belum masuk pulsanya tapi saldonya kepotong saya harus bagaimana iya mohon solusinya terima kasih | 0 |
| 3 | terima kasih buat aplikasi ovo tinkatkan layanan untuk kedepan yang lebih baik | 1 |
| 4 | akhir ini server sering error padahl sinyal wifi penuh isi paket data pun sering gagal | 1 |

## 2.5 *Semi-Supervised Learning*




**Gambar 2 Alur Metode *Semi-Supervised Learning***

Semi-Supervised Learning (SSL) merupakan *framework* untuk memberikan label pada sejumlah besar data yang tidak berlabel (Zhou, 2021). Teknik SSL dapat meningkatkan kinerja model pada tugas *machine learning*, misalnya klasifikasi teks, terjemahan mesin, klasifikasi gambar. Cara kerja SSL adalah dengan mengambil *dataset* berlabel yang sudah ada dan hanya menggunakan sebagian kecil data pelatihan sebagai data berlabel, sementara memperlakukan sisa data sebagai *dataset* tidak berlabel (Ouali et al., 2020). Alur metode *semi-supervised learning* pada penelitian ini terdapat pada Gambar 2.

Tahap *semi-supervised learning* dimulai dengan menentukan *dataset* yang akan digunakan sebagai masukan (input). *Dataset* yang telah melewati tahap *preprocessing* dan tahap *binary labeling*, kemudian dibagi ke dalam data latih, data validasi, dan data uji. Ketiga kategori data tersebut, masing-masing masuk dalam tahap vektorisasi yang mengubah kata menjadi bentuk vektor numerik. Tahap ini menggunakan metode Word2Vec, dengan model Continuous Bag of Word (CBoW). Model CBoW dilatih dengan 120 *epoch* dengan *hyperparameter* vector_size=100, window=5, min_count=1, sg=0. Lebih lanjut skenario pemodelan algoritma berdasarkan *hyperparameter* yang sudah ditentukan, yaitu untuk skenario pemodelan algoritma LSTM dapat dilihat pada Tabel 6, dan skenario pemodelan algoritma GRU dapat dilihat pada Tabel 7.

**Tabel 6 Skenario Pemodelan Algoritma LSTM**

| Learning Rate (LR) | Koefisien Regularisasi (l2) | *Epoch* | *Batch Size* | Skenario |
|---|---|---|---|---|
| 0.002 | 0.001 dan 0.001 | 50 | 128 | *0.002-LSTM-50-128* |
| | | | 256 | *0.002-LSTM-50-256* |
| | | | 512 | *0.002-LSTM-50-512* |
| | | 100 | 128 | *0.002-LSTM-100-128* |
| | | | 256 | *0.002-LSTM-100-256* |
| | | | 512 | *0.002-LSTM-100-512* |
| 0.001 | 0.005 dan 0.01 | 50 | 128 | *0.001-LSTM-50-128* |
| | | | 256 | *0.001-LSTM-50-256* |
| | | | 512 | *0.001-LSTM-50-512* |
| | | 100 | 128 | *0.001-LSTM-100-128* |
| | | | 256 | *0.001-LSTM-100-256* |
| | | | 512 | *0.001-LSTM-100-512* |

**Tabel 7 Skenario Pemodelan Algoritma GRU**

| Learning Rate (LR) | Koefisien Regularisasi (l2) | *Epoch* | *Batch Size* | Skenario |
|---|---|---|---|---|
| 0.002 | 0.001 dan 0.001 | 50 | 128 | *0.002-GRU-50-128* |
| | | | 256 | *0.002-GRU-50-256* |
| | | | 512 | *0.002-GRU-50-512* |
| | | 100 | 128 | *0.002-GRU-100-128* |
| | | | 256 | *0.002-GRU-100-256* |
| | | | 512 | *0.002-GRU-100-512* |
| 0.001 | 0.005 dan 0.01 | 50 | 128 | *0.001-GRU-50-128* |
| | | | 256 | *0.001-GRU-50-256* |
| | | | 512 | *0.001-GRU-50-512* |
| | | 100 | 128 | *0.001-GRU-100-128* |
| | | | 256 | *0.001-GRU-100-256* |
| | | | 512 | *0.001-GRU-100-512* |

Mencoba berbagai skenario memungkinkan evaluasi kinerja model di bawah kondisi yang berbeda, membantu menemukan konfigurasi yang menghasilkan akurasi dan efisiensi terbaik. Eksperimen dengan skenario yang berbeda memungkinkan peneliti untuk mengoptimalkan model secara menyeluruh, termasuk mengurangi *overfitting* atau *underfitting*. Dengan

memvariasikan *learning rate*, koefisien regularisasi, *epoch*, dan *batch size*, penelitian dapat mengidentifikasi kombinasi parameter yang optimal untuk model LSTM dan GRU.

### 2.6 Evaluasi Model

*Confusion matrix* merupakan salah satu metode yang umum digunakan dalam metode evaluasi algoritma yang melakukan komputasi data berlabel (*supervised learning*). Cara kerja *confusion matrix* adalah dengan mengolah data dengan tujuan membandingkan hasil prediksi dengan data sesungguhnya. Terdapat empat macam bagian evaluasi dari *confusion matrix*, yaitu akurasi, *recall, precision,* dan F1 *score*. *Confusion matrix* dapat dilihat pada Tabel 8.

**Tabel 8 *Confusion Matrix***

| Aktual | Prediksi | |
|---|---|---|
| | Positif | Negatif |
| Positif | True Positive (TP) | False Negative (FN) |
| Negatif | False Positive (FP) | True Negative (TN) |

1) Akurasi, yaitu ukuran tingkat kedekatan antara nilai sebenarnya dengan nilai hasil prediksi model (Rolangon et al., 2023). Hasil model dapat dikatakan semakin baik apabila menghasilkan nilai akurasi yang mendekati 100%. Rumus akurasi dituliskan pada Pers. (1).

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

2) *Recall*, atau disebut dengan sebut *sensitivity* merupakan metode yang digunakan untuk mengevaluasi seberapa baik model mengelompokkan data berlabel positif secara keseluruhan (Rolangon et al., 2023). Rumus *recall* dituliskan pada Pers. (2).

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

3) *Precision*, digunakan untuk mengukur tingkat ketepatan suatu model yang dibangun dalam mengklasifikasikan suatu data kedalam kelas positif (Suryati et al., 2023). Rumus *precision* dituliskan pada Pers. (3).

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

4) F1 *score*, adalah perbandingan rata-rata *recall* dan presisi (*precision*) (Ayuningtyas & Tantyoko, 2024), dengan rumus seperti pada Pers. (4).

$$F1\ score = 2 \times \frac{Recall \times Precision}{Recall + Precision} \times 100\% \quad (4)$$

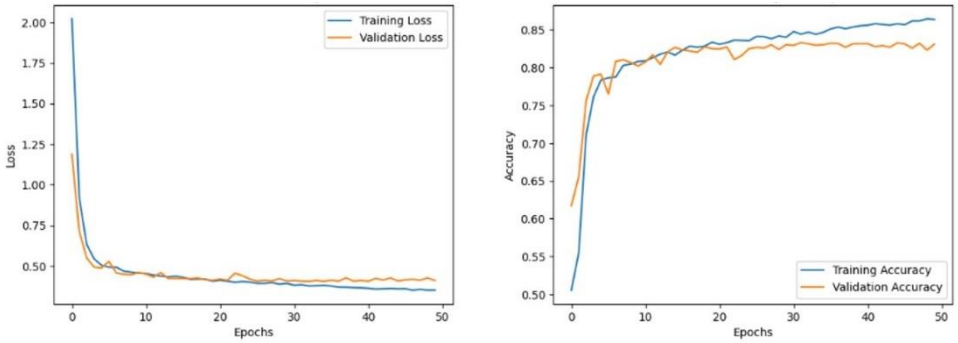
## 3. HASIL DAN PEMBAHASAN

### 3.1 Pemodelan Algoritma *Deep Learning*

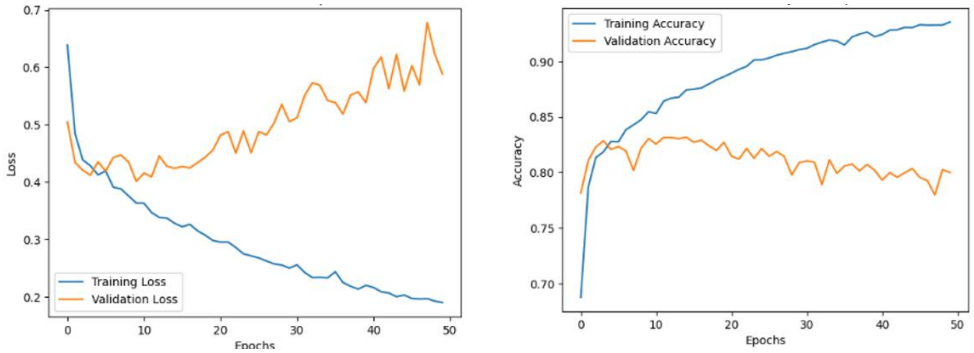
#### 3.1.1 Hasil Pelatihan Algoritma LSTM

Pada skenario *0.001-LSTM-50-512,* model dilatih menggunakan Algoritma LSTM dengan *hyperparameter* Learning Rate Optimasi ADAM sebesar 0,001, *epoch* 50, Koefisien Regularisasi (l2) *layer* pertama dan kedua masing-masing 0,005 dan 0,01, dan *batch size* 512. Algoritma LSTM dapat mencapai kondisi *good fit* pada susunan hyperparameter ini. Sedangkan pada skenario *0.002-LSTM-50-512,* dengan *hyperparameter* Learning Rate Optimasi ADAM sebesar 0,002,

*epoch* 50, Koefisien Regularisasi (l2) *layer* pertama dan kedua masing-masing 0,001 dan 0,001, dan *batch size* 512. Algoritma LSTM berada pada kondisi *overfit.* Perbandingan performa kedua skenario LSTM dapat dilihat pada Gambar 3 dan Gambar 4.



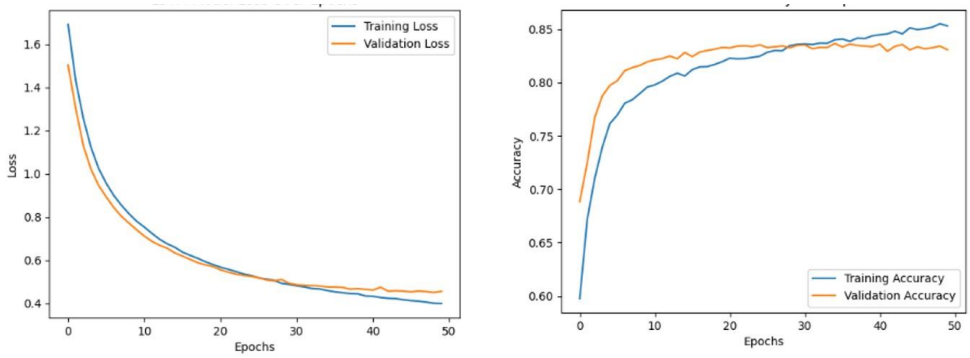

**Gambar 3 Performa Skema *0.001-LSTM-50-512***



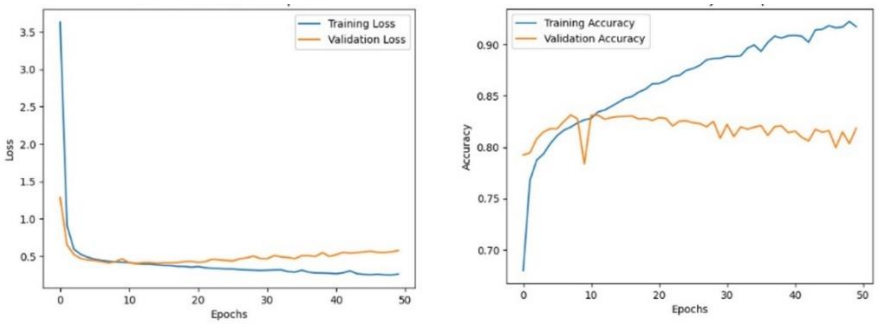

**Gambar 4 Performa Skema *0.002-LSTM-50-512***

### 3.1.2 Hasil Pelatihan Algoritma GRU

Pada skenario *0.002-GRU-50-128,* model dilatih menggunakan Algoritma GRU dengan *hyperparameter* Learning Rate (LR) Optimasi ADAM sebesar 0,002, *epoch* 50, Koefisien Regularisasi (l2) *layer* pertama dan kedua masing-masing 0,001, dan *batch size* 128. Algoritma GRU dapat mencapai kondisi *good fit* pada susunan *hyperparameter* ini. Sedangkan pada skenario *0.001-GRU-50-128,* dengan *hyperparameter* Learning Rate Optimasi ADAM sebesar 0,001, *epoch* 50, Koefisien Regularisasi (l2) *layer* pertama dan kedua masing-masing 0,005 dan 0,01, dan *batch size* 128. Algoritma GRU berada pada kondisi *overfit.* Perbandingan performa kedua skenario GRU dapat dilihat pada Gambar 5 dan Gambar 6.




**Gambar 5 Performa Skema *0.002-GRU-50-128***




**Gambar 6 Performa Skema *0.001-GRU-50-128***

### 3.1.3 Evaluasi Pelatihan Algoritma *Deep Learning*

Hasil evaluasi model algoritma LSTM menggunakan data uji terdapat pada Tabel 9. Sedangkan, hasil evaluasi model algoritma GRU menggunakan data uji dapat dilihat pada Tabel 10. Dari kedua tabel tesebut, dapat diketahui bahwa algoritma LSTM memiliki nilai akurasi, presisi, *recall*, dan F1-*score* tertinggi masing-masing sebesar 0,94, 0,966, 0,943, dan 0,939. Sedangkan GRU memiliki nilai akurasi, presisi, *recall*, dan F1-*score* tertinggi masing-masing sebesar 0,952, 0,951, 0,961, dan 0,952. Dengan demikian, pada penelitian ini, algoritma GRU memiliki kemampuan klasifikasi yang lebih baik daripada LSTM.

**Tabel 9 Hasil Evaluasi Algoritma LSTM**

| Skenario LSTM | Akurasi | Presisi | *Recall* | F1-*Score* |
|---|---|---|---|---|
| *0.002-LSTM-50-128* | 0,922 | 0,95 | 0,891 | 0,92 |
| *0.002-LSTM-50-256* | 0,885 | 0,845 | 0,943 | 0,891 |
| *0.002-LSTM-50-512* | 0,903 | 0,959 | 0,842 | 0,897 |
| *0.002-LSTM-100-128* | 0,929 | 0,949 | 0,907 | 0,927 |
| *0.002-LSTM-100-256* | 0,94 | 0,955 | 0,925 | 0,939 |
| *0.002-LSTM-100-512* | 0,938 | 0,947 | 0,927 | 0,937 |
| *0.001-LSTM-50-128* | 0,891 | 0,942 | 0,834 | 0,885 |
| *0.001-LSTM-50-256* | 0,883 | 0,911 | 0,85 | 0,879 |
| *0.001-LSTM-50-512* | 0,876 | 0,818 | 0,966 | 0,886 |
| *0.001-LSTM-100-128* | 0,896 | 0,875 | 0,925 | 0,899 |
| *0.001-LSTM-100-256* | 0,895 | 0,932 | 0,852 | 0,89 |
| *0.001-LSTM-100-512* | 0,889 | 0,936 | 0,834 | 0,882 |

**Tabel 10 Hasil Evaluasi Algoritma GRU**

| Skenario GRU | Akurasi | Presisi | *Recall* | F1-*Score* |
|---|---|---|---|---|
| *0.002-GRU-50-128* | 0,872 | 0,877 | 0,865 | 0,871 |
| *0.002-GRU-50-256* | 0,931 | 0,924 | 0,940 | 0,932 |
| *0.002-GRU-50-512* | 0,946 | 0,932 | 0,961 | 0,946 |
| *0.002-GRU-100-128* | 0,900 | 0,889 | 0,915 | 0,902 |
| *0.002-GRU-100-256* | 0,887 | 0,919 | 0,850 | 0,883 |
| *0.002-GRU-100-512* | 0,873 | 0,873 | 0,873 | 0,873 |
| *0.001-GRU-50-128* | 0,921 | 0,945 | 0,894 | 0,919 |
| *0.001-GRU-50-256* | 0,902 | 0,928 | 0,870 | 0,898 |
| *0.001-GRU-50-512* | 0,891 | 0,913 | 0,865 | 0,888 |
| *0.001-GRU-100-128* | 0,942 | 0,945 | 0,938 | 0,941 |
| *0.001-GRU-100-256* | 0,942 | 0,945 | 0,938 | 0,941 |
| *0.001-GRU-100-512* | 0,952 | 0,951 | 0,953 | 0,952 |

### 3.2 Pengujian Model *Semi-Supervised Learning*

Setelah data dibersihkan melalui tahap *preprocessing*, langkah berikutnya adalah melakukan pengujian model *supervised learning* menggunakan *dataset* tanpa label (*unlabeled dataset*). Pengujian ini bertujuan untuk mengevaluasi performa model *deep learning* dalam menangani data yang belum diberi label. Hasil pengujian model Long Short-Term Memory (LSTM) terhadap dataset tanpa label dapat dilihat pada Tabel 11, yang memperlihatkan kinerja model dalam mengenali pola dari data. Sementara itu, hasil pengujian model Gated Recurrent Unit (GRU) terhadap *dataset* tanpa label disajikan pada Tabel 12. Kedua tabel ini menampilkan hasil perbandingan akurasi dan performa masing-masing model dalam mengolah *dataset* yang sama, sehingga dapat diidentifikasi model mana yang lebih efektif dalam skenario ini.

**Tabel 11 Tabel Hasil Percobaan Pelabelan (SSL) Algoritma LSTM**

| Skenario LSTM | Akurasi | Presisi | *Recall* | F1-*Score* |
|---|---|---|---|---|
| *0.002-LSTM-50-128* | 0,538 | 0,537 | 0,554 | 0,545 |
| *0.002-LSTM-50-256* | 0,566 | 0,602 | 0,388 | 0,472 |
| *0.002-LSTM-50-512* | 0,568 | 0,556 | 0,668 | 0,607 |
| *0.002-LSTM-100-128* | 0,560 | 0,559 | 0,569 | 0,564 |
| *0.002-LSTM-100-256* | 0,571 | 0,566 | 0,607 | 0,586 |
| *0.002-LSTM-100-512* | 0,553 | 0,556 | 0,556 | 0,544 |
| *0.001-LSTM-50-128* | 0,588 | 0,580 | 0,641 | 0,609 |
| *0.001-LSTM-50-256* | 0,602 | 0,603 | 0,595 | 0,599 |
| *0.001-LSTM-50-512* | 0,575 | 0,559 | 0,707 | 0,625 |
| *0.001-LSTM-100-128* | 0,546 | 0,568 | 0,389 | 0,462 |
| *0.001-LSTM-100-256* | 0,581 | 0,588 | 0,540 | 0,563 |
| *0.001-LSTM-100-512* | 0,579 | 0,568 | 0,653 | 0,608 |

**Tabel 12 Tabel Hasil Percobaan Pelabelan (SSL) Algoritma GRU**

| Skenario GRU | Akurasi | Presisi | *Recall* | F1-*Score* |
|---|---|---|---|---|
| *0.002-GRU-50-128* | 0,638 | 0,653 | 0,588 | 0,619 |
| *0.002-GRU-50-256* | 0,583 | 0,592 | 0,533 | 0,561 |
| *0.002-GRU-50-512* | 0,584 | 0,586 | 0,570 | 0,578 |
| *0.002-GRU-100-128* | 0,599 | 0,622 | 0,608 | 0,559 |
| *0.002-GRU-100-256* | 0,606 | 0,602 | 0,624 | 0,613 |
| *0.002-GRU-100-512* | 0,601 | 0,606 | 0,573 | 0,589 |
| *0.001-GRU-50-128* | 0,572 | 0,567 | 0,609 | 0,587 |
| *0.001-GRU-50-256* | 0,573 | 0,571 | 0,590 | 0,581 |
| *0.001-GRU-50-512* | 0,607 | 0,605 | 0,617 | 0,611 |
| *0.001-GRU-100-128* | 0,574 | 0,575 | 0,570 | 0,573 |
| *0.001-GRU-100-256* | 0,594 | 0,595 | 0,585 | 0,590 |
| *0.001-GRU-100-512* | 0,574 | 0,580 | 0,535 | 0,557 |

Dari Tabel 11 dan Tabel 12, dapat diketahui bahwa algoritma LSTM memiliki nilai akurasi, presisi, *recall*, dan F1-*score* tertinggi masing-masing sebesar 0,602, 0,603, 0,707, dan 0,625. Sedangkan GRU memiliki nilai akurasi, presisi, *recall*, dan F1-*score* tertinggi masing-masing sebesar 0,638, 0,653, 0,624, dan 0,619. Hasil pelatihan dan pengujian model bergantung pada kualitas dan kuantitas data yang digunakan. Pada penelitian ini, akurasi pengujian model SSL masih kurang optimal. Hal ini disebabkan data yang digunakan masih terdapat beberapa kelemahan. Salah satunya tidak konsisten antara kecenderungan komentar yang dituliskan oleh pengguna dan *rating* yang diberikan. Contohnya pada Tabel 5 nomor 4, di mana kecenderungan komentar mengarah ke kategori negatif, namun oleh pengguna diberikan *rating* positif. Ini menyebabkan kebingungan model dalam mempelajari suatu pola klasifikasi, sehingga model memiliki akurasi yang rendah ketika melakukan pelabelan terhadap data baru tanpa label.

## 4. KESIMPULAN

Algoritma LSTM mencapai *goodfit* ketika menggunakan skema *0.001-LSTM-50-512*, dan mengalami *overfitting* pada skema *0.002-LSTM-50-512.* Parameter *learning rate* pada LSTM dengan nilai 0,001 memiliki hasil evaluasi model yang lebih baik daripada nilai 0,002. Sedangkan Algoritma GRU mencapai *goodfit* ketika menggunakan skema *0.002-GRU-50-128* dan mengalami *overfitting* pada skema *0.001-GRU-50-128.* Parameter *learning rate* pada GRU dengan nilai 0,002 memiliki hasil evaluasi model yang lebih baik daripada nilai 0,001. Algoritma LSTM dan GRU memiliki titik *goodfit* dan *overfit* masing-masing yang dapat memengaruhi hasil evaluasi model, baik ketika pelatihan maupun pengujian model. Hasil pelatihan dan pengujian model bergantung pada kualitas dan kuantitas data yang digunakan.

Penelitian selanjutnya diharap mengoptimalkan pembersihan data, *preprocessing data*, dan melakukan cek ulang terkait kesesuaian komentar dan label sehingga dapat memberikan hasil akurasi pelabelan yang lebih optimal. Menggunakan nilai *hyperparameter* yang lain, misal menambahkan jumlah epoch, mengubah fungsi aktivasi algoritma, menaikkan nilai *learning rate*, dan yang lainnya. Menggunakan metode *word embedding* lain, misal FastText, Glove, atau BERT.

## DAFTAR PUSTAKA

Af'idah, D. I., Dairoh, D., Handayani, S. F., & Pratiwi, R. W. (2021). Pengaruh Parameter Word2Vec terhadap Performa Deep Learning pada Klasifikasi Sentimen. *Jurnal Informatika: Jurnal Pengembangan IT*, *6*(3), 156–161. https://doi.org/10.30591/jpit.v6i3.3016

Anggraini, N., Harahap, E. S. N., & Kurniawan, T. B. (2021). Text Mining - Analisis Teks Terkait Isu Vaksinasi COVID-19 (Text Mining - Text Analysis Related to COVID-19 Vaccination Issues). *JURNAL IPTEKKOM Jurnal Ilmu Pengetahuan & Teknologi Informasi*, *23*(2), 141–153. https://doi.org/10.17933/iptekkom.23.2.2021.141-153

Arsi, P., & Waluyo, R. (2021). Analisis Sentimen Wacana Pemindahan Ibu Kota Indonesia Menggunakan Algoritma Support Vector Machine (SVM). *Jurnal Teknologi Informasi Dan Ilmu Komputer*, *8*(1), 147. https://doi.org/10.25126/jtiik.0813944

Ayuningtyas, P., & Tantyoko, H. (2024). Comparison of the Word2vec Skipgram Model Method Linkaja Application Review using Bidirectional LSTM Algorithm and Support Vector Machine. *Jurnal Sistem Dan Teknologi Informasi (JustIN)*, *12*(1), 189. https://doi.org/10.26418/justin.v12i1.72530

Bandhakavi, A., Wiratunga, N., Massie, S., & Padmanabhan, D. (2017). Lexicon Generation for Emotion Detection from Text. *IEEE Intelligent Systems*, *32*(1), 102–108. https://doi.org/10.1109/MIS.2017.22

Ezen-Can, A. (2020). *A Comparison of LSTM and BERT for Small Corpus*. http://arxiv.org/abs/2009.05451

Gifari, O. I., Adha, Muh., Freddy, F., & Durrand, F. F. S. (2022). Analisis Sentimen Review Film Menggunakan TF-IDF dan Support Vector Machine. *Journal of Information Technology*, *2*(1), 36–40. https://doi.org/10.46229/jifotech.v2i1.330

Janah, L. N., & Setiyawan, S. (2022). Dampak Pandemi Covid-19 Terhadap Penggunaan Dompet Digital Di Indonesia. *Journal of Educational and Language Research*, *1*(7), 709–716. https://doi.org/https://doi.org/10.53625/joel.v1i7.1463

Khatri, A., & P, P. (2020). Sarcasm Detection in Tweets with BERT and GloVe Embeddings. *Proceedings of the Second Workshop on Figurative Language Processing*, 56–60. https://doi.org/10.18653/v1/2020.figlang-1.7

Khomsah, S., & Aribowo, A. S. (2020). Text-Preprocessing Model Youtube Comments in Indonesian. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, *4*(4), 648–654. https://doi.org/10.29207/resti.v4i4.2035

Magnolia, C., Nurhopipah, A., & Kusuma, B. A. (2022). Penanganan Imbalanced Dataset untuk Klasifikasi Komentar Program Kampus Merdeka Pada Aplikasi Twitter. *Edu Komputika Journal*, *9*(2), 105–113. https://doi.org/10.15294/edukomputika.v9i2.61854

Nosouhian, S., Nosouhian, F., & Khoshouei, A. K. (2021). A Review of Recurrent Neural Network Architecture for Sequence Learning: Comparison between LSTM and GRU. *Preprints*, 1–7. https://doi.org/https://doi.org/10.20944/preprints202107.0252.v1

Oktaviani, A., & Hustinawati. (2021). Prediksi Rata-Rata Zat Berbahaya di DKI Jakarta Berdasarkan Indeks Standar Pencemar Udara Menggunakan Metode Long Short-Term Memory. *Jurnal Ilmiah Informatika Komputer*, *26*(1), 41–55. https://doi.org/10.35760/ik.2021.v26i1.3702

Ouali, Y., Hudelot, C., & Tami, M. (2020). *An Overview of Deep Semi-Supervised Learning*. http://arxiv.org/abs/2006.05278

Rahma, I. A., & Suadaa, L. H. (2023). Penerapan Text Augmentation untuk Mengatasi Data yang Tidak Seimbang pada Klasifikasi Teks Berbahasa Indonesia. *Jurnal Teknologi Informasi Dan Ilmu Komputer*, *10*(6), 1329–1340. https://doi.org/10.25126/jtiik.1067325

Rolangon, A., Weku, A., & Sandag, G. A. (2023). Perbandingan Algoritma LSTM Untuk Analisis Sentimen Pengguna Twitter Terhadap Layanan Rumah Sakit Saat Pandemi Covid-19. *TeIKa*, *13*(01), 31–40. https://doi.org/10.36342/teika.v13i01.3063

Romadhoni, Y., & Holle, K. F. H. (2022). Analisis Sentimen Terhadap PERMENDIKBUD No.30 pada Media Sosial Twitter Menggunakan Metode Naive Bayes dan LSTM. *Jurnal Informatika: Jurnal Pengembangan IT*, *7*(2), 118–124. https://doi.org/10.30591/jpit.v7i2.3191

Seabe, P. L., Moutsinga, C. R. B., & Pindza, E. (2023). Forecasting Cryptocurrency Prices Using LSTM, GRU, and Bi-Directional LSTM: A Deep Learning Approach. *Fractal and Fractional*, *7*(2), 203. https://doi.org/10.3390/fractalfract7020203

Setiyawan, D. A., W, S. K., Diana, A. L., W, I. A. H., Yusuf, M., & Krisnando, K. (2023). Penyuluhan Pemahaman Digital Wallet, Digital Perbankan Dan Pajak Penghasilan Bagi Pengusaha Kecil Untuk Meningkatkan Omzet Penjualan. *Jurnal Pengabdian Mandiri*, *2*(9), 1955–1962. https://bajangjournal.com/index.php/JPM/article/view/6615

Suryati, E., Styawati, S., & Aldino, A. A. (2023). Analisis Sentimen Transportasi Online Menggunakan Ekstraksi Fitur Model Word2vec Text Embedding Dan Algoritma Support Vector Machine (SVM). *Jurnal Teknologi Dan Sistem Informasi*, *4*(1), 96–106. https://doi.org/10.33365/jtsi.v4i1.2445

Wisnalmawati, W., Aribowo, A. S., & Herawati, Y. (2022). Semi-supervised Learning Models for Sentiment Analysis on Marketplace Dataset. *International Journal of Artificial Intelligence & Robotics (IJAIR)*, *4*(2), 78–85. https://doi.org/10.25139/ijair.v4i2.5267

Zhafira, D. F., Rahayudi, B., & Indriati, I. (2021). Analisis Sentimen Kebijakan Kampus Merdeka Menggunakan Naive Bayes dan Pembobotan TF-IDF Berdasarkan Komentar pada Youtube. *Jurnal Sistem Informasi, Teknologi Informasi, Dan Edukasi Sistem Informasi*, *2*(1). https://doi.org/10.25126/justsi.v2i1.24

Zhou, Z. H. (2021). Machine Learning. In *Machine Learning*. Springer Nature. https://doi.org/10.1007/978-981-15-1967-3/COVER

**JISKA (Jurnal Informatika Sunan Kalijaga)**
**Vol. 9, No. 3, SEPTEMBER, 2024, Pp. 230 – 243**
**ISSN: 2527 – 5836 (print) | 2528 – 0074 (online)**

# Integrating Retrieval-Augmented Generation with Large Language Model Mistral 7b for Indonesian Medical Herb

**Diash Firdaus [(1)*], Idi Sumardi [(2)], Yuni Kulsum [(3)]**
[1] Informatika, Fakultas Teknologi Industri, Institut Teknologi Nasional, Bandung, Indonesia
[2] Teknik Informatika, STMIK Jawa Barat, Bandung, Indonesia
[3] Laboratorium Terpadu, UIN Sunan Gunung Djati, Bandung, Indonesia
e-mail : diash@itenas.ac.id, idis@stmikjabar.ac.id, yunikulsum05@gmail.com.
* Penulis korespondensi.
Artikel ini diajukan 11 Mei 2024, direvisi 28 Agustus 2024, diterima 5 September 2024, dan dipublikasikan 25 September 2024.

***Abstract***

*Large Language Models (LLMs) are advanced artificial intelligence systems that use deep learning, particularly transformer architectures, to process and generate text. One such model, Mistral 7b, featuring 7 billion parameters, is optimized for high performance and efficiency in natural language processing tasks. It outperforms similar models, such as LLaMa2 7b and LLaMa 1, across various benchmarks, especially in reasoning, mathematics, and coding. LLMs have also demonstrated significant advancements in addressing medical queries. This research leverages Indonesia's rich biodiversity, which includes approximately 9,600 medicinal plant species out of the 30,000 known species. The study is motivated by the observation that LLMs, like ChatGPT and Gemini, often rely on internet data of uncertain validity and frequently provide generic answers without mentioning specific herbal plants found in Indonesia. To address this, the dataset for pre-training the model is derived from academic journals focusing on Indonesian medicinal herbal plants. The research process involves collecting these journals, preprocessing them using Langchain, embedding models with sentence transformers, and employing Faiss CPU for efficient searching and similarity matching. Subsequently, the Retrieval-Augmented Generation (RAG) process is applied to Mistral 7b, allowing it to provide accurate, dataset-driven responses to user queries. The model's performance is evaluated using both human evaluation and ROUGE metrics, which assess recall, precision, F1 measure, and METEOR scores. The results show that the RAG Mistral 7b model achieved a METEOR score of 0.22%, outperforming the LLaMa2 7b model, which scored 0.14%.*

***Keywords: LLM, Generative AI, LLAMA2, Retrieval-Augmented Generation, Deep Learning***

**Abstrak**

Large Language Models (LLM) adalah sistem kecerdasan buatan canggih yang menggunakan pembelajaran mendalam, khususnya arsitektur transformator, untuk memproses dan menghasilkan teks. Salah satu model tersebut, Mistral 7b, yang memiliki 7 miliar parameter, dioptimalkan untuk kinerja tinggi dan efisiensi dalam tugas pemrosesan bahasa alami. Model ini mengungguli model serupa, seperti LLaMa2 7b dan LLaMa 1, di berbagai tolok ukur, terutama dalam penalaran, matematika, dan pengkodean. LLM juga telah menunjukkan kemajuan yang signifikan dalam menjawab pertanyaan-pertanyaan medis. Penelitian ini memanfaatkan keanekaragaman hayati Indonesia yang kaya, yang mencakup sekitar 9.600 spesies tanaman obat dari 30.000 spesies yang diketahui. Penelitian ini dilatarbelakangi oleh pengamatan bahwa LLM, seperti ChatGPT dan Gemini, sering kali mengandalkan data internet yang validitasnya tidak pasti dan sering kali memberikan jawaban umum tanpa menyebutkan tanaman herbal tertentu yang ditemukan di Indonesia. Untuk mengatasi hal ini, *dataset* untuk pra-pelatihan model ini berasal dari jurnal-jurnal akademis yang berfokus pada tanaman herbal obat Indonesia. Proses penelitian melibatkan pengumpulan jurnal-jurnal ini, *preprocessing* menggunakan Langchain, menanamkan model dengan *transformer* kalimat, dan menggunakan CPU Faiss untuk pencarian dan pencocokan kemiripan yang efisien. Selanjutnya, proses Retrieval-Augmented Generation (RAG) diterapkan pada Mistral 7b, yang memungkinkannya untuk memberikan respons yang akurat dan berbasis *dataset* terhadap pertanyaan pengguna. Kinerja model dievaluasi dengan menggunakan evaluasi manusia dan metrik ROUGE, yang menilai

*recall*, presisi, F1 *measure*, dan skor METEOR. Hasilnya menunjukkan bahwa model RAG Mistral 7b mencapai skor METEOR 0,22%, mengungguli model LLaMa2 7b, yang mendapat skor 0,14%.

**Kata Kunci: LLM, AI Generatif, LLAMA2, *Retrieval-Augmented Generation*, *Deep Learning***

## 1. INTRODUCTION

The domain of Natural Language Processing (NLP) and Artificial Intelligence (AI) has witnessed a remarkable breakthrough with the introduction of Large Language Models (LLMs). This innovative development has dramatically amplified the capacity of machines to comprehend and produce human-like language, thereby revolutionizing the field of language processing and AI research (Hadi et al., 2024; Jain et al., 2023; Kaddour et al., 2023). LLMs have consistently demonstrated outstanding performance in a wide range of tasks. However, their exceptional abilities present notable challenges due to their large-scale and intensive computational demands (Zhu et al., 2023).

Some frequently used Large Language Model (LLM) models are LLaMa (Touvron, Lavril, et al., 2023), GPT-4(OpenAI et al., 2023), LLaMa2 (Touvron, Martin, et al., 2023), PANGU-Σ (Ren et al., 2023)and so on. Considering the remarkable achievements of pre-training methods in general LLM, researchers have begun to explore the potential application of these techniques in the biomedical and health domain. However, it has been observed that simply applying these models directly to the biomedical domain does not yield satisfactory results. The underwhelming performance can be attributed to the significant differences and distinct characteristics between the general and specialized biomedical domains, known as domain shift (K. Zhang et al., 2024). Several studies have used the llm model to be implemented in the field of health and medicine, such as Med-PaLM (Singhal, Azizi, et al., 2023), BioGPT (Luo et al., 2022), Med-PaLM 2 (Singhal, Tu, et al., 2023), Flan-PaLM (Chung et al., 2022), and so on.

The utilization of medicinal plants is of utmost importance in acting as the primary healthcare system, particularly in underserved regions of developing nations. These areas often rely solely on herbal remedies as the sole medication accessible to them (Fathir et al., 2021). Indonesia is known to be a country with biodiversity resources spread from Sabang to Merauke Region, where there are around 30,000 plant species, of which 9600 are medicinal plants. The utilization of herbal medicine, commonly referred to as “jamu” within the Indonesian culture, holds significant importance in healthcare for a staggering majority of the population, approximately 80%. Derived from the Javanese tribal language, the term “jamu” embodies the concept of traditional herbal treatments. Essentially, jamu encompasses the utilization of plant-based substances that are intricately crafted to serve medicinal purposes. Over time, the term “jamu” has seamlessly integrated itself into the Indonesian language, bearing a resemblance to its original definition (Elfahmi et al., 2014; Sianipar, 2021). The application of NLP in the field of Herbal Medicine has been carried out by several previous researchers, such as (T. Zhang et al., 2022). However, it is mentioned that in developing NLP herbal medicine, a collaboration between many experts, such as biologists and computer experts, is required. The next issue is that the currently available models, such as ChatGPT, Google Gemini, Claude, and others, provide answers based on internet data whose validity is uncertain (Ray, 2023). Often, the responses include herbs that are not available in Indonesia.

Mistral 7b is a language model equipped with 7 billion parameters, engineered for superior performance and efficiency in tasks associated with natural language processing. It surpasses larger models such as LLaMa2 7b, which has 13 billion parameters, and LLaMa 1, with 34 billion parameters, across multiple benchmarks, particularly excelling in reasoning, mathematics, and code generation tasks. (Jiang et al., 2023). RAG, or Retrieval-Augmented Generation, is a technique used in natural language processing that combines a retrieval system with a generative model. This approach enhances the generative model’s ability to produce relevant and accurate responses by first retrieving relevant information from a large dataset or knowledge base before generating an answer. In the RAG setup, when a question or input is presented to the system,

the retrieval component first searches through a database to find relevant documents or pieces of text. These retrieved texts are then provided as additional context to the generative model, which uses this context to generate a more informed and precise answer. This method helps in improving the quality of the output by grounding the responses in factual content retrieved from the dataset (Radeva et al., 2024).

In this study, we propose a specific Retrieval-Augmented Generation for Herbal Medicine text generation and mining. Herbal Medicine LLM follows the backbone of the Sentence-Transformer language model and Mistral 7b for the Large Language Model. For QA, we used nine international journals on herbal medicine in Indonesia. At the time of evaluation, we used the results issued by RAG Mistral, which were compared with the answers from the experts.

## 2. RESEARCH METHOD

Our research introduces Herbal Medicine using RAG Mistral 7b, an innovative and specialized language model specifically tailored for various Herbal medical applications. With a unique approach, we have successfully trained this model using a diverse and extensive collection of Journal Medical Herb in Indonesia. This abundant dataset enables the infusion of domain-specific knowledge into the model, enhancing its efficacy. Moreover, we have meticulously devised a comprehensive evaluation framework that encompasses question-answering.

### 2.1 Research Stage

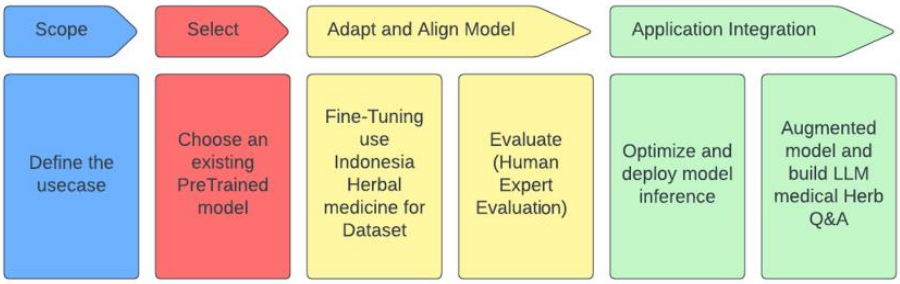



**Figure 1 Research Stage of Medical Herb**

The research stages of medical herbs using Mistral 7b are shown in Figure 1. In the first stage, we identified the problem, followed by the process of selecting an LLM model. In this study, we chose the Mistral 7b model because previous research has shown that the RAG on the LLM Mistral 7b model provides better results in terms of evaluation time, loading duration, prompt evaluation count, total duration, and tokens per second compared to the LLM LLaMa2 7b and Orca2 7b models. The LLM Mistral 7b RAG model also delivered the best results for METEOR, ROUGE 1 Recall, ROUGE Precision, and BLUE Score when compared to LLaMa2 7b and Orca2 7b (Radeva et al., 2024).

The process depicted in the diagram involves several stages to develop a language model tailored specifically for medical herb Q&A using Indonesian herbal medicine data. First, define the use case clearly to establish the specific application or problem the model will address, focusing on the context of Indonesian herbs. Next, select an appropriate pre-trained model as a foundation for further customization. Then, the model can be adapted by training it with a specialized dataset containing information about Indonesian herbs, ensuring the model learns domain-relevant knowledge. Evaluate the fine-tuned model with the help of human experts, such as ROUGE and METEOR, to assess its performance. Afterward, optimize and deploy the model for real-world application by ensuring efficient inference and accessibility. Finally, the enhanced model will be integrated into an application framework designed for Q&A, focusing on medical herb-related queries. This process ensures that the language model is not only accurate and relevant but also effectively integrated into practical applications. When finished, each answer to the question

posed will be evaluated by an expert. On this occasion, our expert is assisted by an expert in the field of ethnobotany from Sunan Gunung Djati University, Bandung, namely Dr. Tri Cahyanto, M.Si. After the evaluation process is complete, the model will be deployed with the final goal of a web-based application.

### 2.2 RAG LLM Herb Medicine Architecture

Figure 2 is the architecture of LLM Herbal Medicine using Mistral 7b, where, at the initial stage, we will extract data and then preprocess, including quality filtering, de-duplication, and so on. The next process is to divide the data into chunks, where each chunk will be embedded, and a semantics index will be built. Then, the data will be stored in the DB vector data in the form of Knowledge-based Data. In the last stage, the user will give questions to the Knowledge Database, and the Knowledge Database will reply to user questions based on the answers ranking.

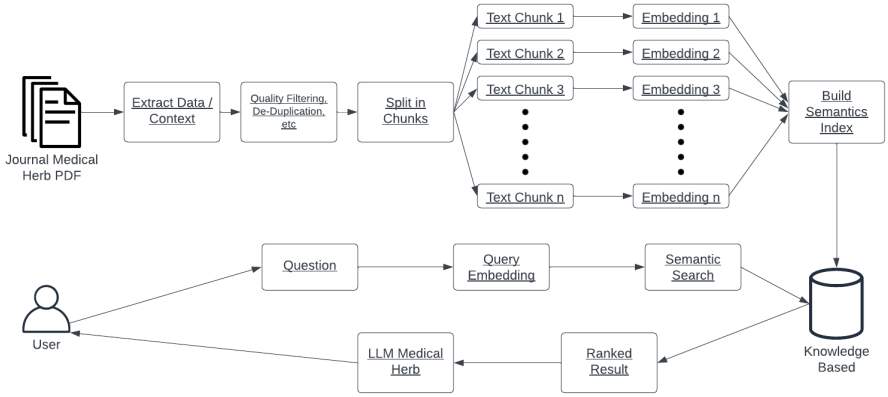



**Figure 2 Architecture of RAG LLM Mistral 7b Herbal Medicine**

In the dataset section, we utilized nine journal sources focused on herbal medicine in Indonesia, which are detailed in Table 1. These journals were specifically selected for their relevance and suitability concerning various herbal plants found in the region. This selection provides a comprehensive foundation for our research on the efficacy of these plants.

**Table 1 Journal Medical Herb Dataset**

| Journal | Title |
|---|---|
| (Sianipar, 2021) | The Potential of Indonesian Traditional Herbal Medicine as Immunomodulatory Agents: A Review |
| (Putri et al., 1970) | Ethnobotanical study of herbal medicine in Ranggawulung Urban Forest, Subang District, West Java, Indonesia |
| (Fathir et al., 2021) | Ethnobotanical study of medicinal plants used for maintaining stamina in Madura ethnic, East Java, Indonesia |
| (Sholikhah, 2016) | Indonesian medicinal plants as sources of secondary metabolites for pharmaceutical industry |
| (Ardiyanto et al., 2021) | The use of hyperuricemia herbs at “Hortus Medicus” herbal medicine clinic Tawangmangu |
| (Elfahmi et al., 2014) | Jamu: Indonesian traditional herbal medicine towards rational phytopharmacological use |
| (Arozal et al., 2020) | Selected Indonesian Medicinal Plants for the Management of Metabolic Syndrome: Molecular Basis and Recent Studies |
| (Kartini et al., 2019) | Standardization of Some Indonesian Medicinal Plants Used in “Scientific Jamu” |
| (Sumarni et al., 2019) | The scientification of jamu: a study of Indonesian’s traditional medicine |

### 2.3 Environment Setup

In this step, the necessary environment for the project is established. This involves configuring tools, software, hardware, and other resources essential for conducting research or development activities. The details of the Environment Setup are provided in Table 2.

**Table 2 Environment Setup**

| No. | Name | Version |
|---|---|---|
| 1 | Operating System | Windows 11 |
| 2 | Programming | Python 11.2 |
| 3 | Supporting Tools | Library Chainlit |
| | | Library Huggingface |
| | | Library Langchain |
| 4 | Hardware | CPU AMD Ryzen 7 5800 |
| | | RAM 16 GB |
| | | VGA RADEON RX 5500M |
| 5 | Model | Mistral 7b |
| | | LLaMa 7b |
| | | 500 Chunk |
| | | 512 Tokens |
| | | 0.5 Temperatur |
| | | Mistral 7b |

### 2.4 Metric Evaluation

For Metric Evaluation, we use several evaluation models such as METEOR (Metric for Evaluation of Translation with Explicit Ordering), ROUGE (Recall-Oriented Understudy for Gisting Evaluation) (Radeva et al., 2024), and human evaluation (Wang et al., 2023)

#### 2.4.1 METEOR (Metric for Evaluation of Translation with Explicit Ordering)

The METEOR score is a metric used to evaluate the quality of machine-generated text by comparing it to one or more reference texts. The calculation involves several steps:

1) Word Alignment: Align words between candidate and reference translations based on exact matches, stems, synonyms, and paraphrases, ensuring each word in the candidate and reference sentences is used only once to maximize the overall match.
2) Precision and Recall Calculation:
   a) Precision measures the proportion of matched words in the candidate compared to the total number of words in the candidate.
   b) Recall measures the proportion of matched words in the candidate against the total number of words in the reference.

$$P = \frac{m}{w_c}, R = \frac{m}{w_r} \quad (1)$$

   The formulas for precision and recall are presented in Equation (1). In this context, $m$ represents the number of unigrams in the candidate translation that match those in the reference, $w_c$ denotes the total number of unigrams in the candidate translation, and $w_r$ indicates the total number of unigrams in the reference translation(s).
3) The penalty accounts for chunkiness, which refers to the arrangement and fluency of matched chunks, represented by the formula in Equation (2), where $c$ defines the number of contiguous matched unigrams.

$$Penalty = 0.5 \left(\frac{c}{m}\right)^3 \quad (2)$$

4) The final METEOR score is computed using the harmonic mean of precision and recall, adjusted by the penalty factor. The formula is shown in Equation (3).

$$M_{Score} - F_{Mean}\, X\, (1 - Penalty) \tag{3}$$

Where

$$F_{Mean} = \frac{10PR}{R + 9P}$$

The calculations for Equations (1) to (3) are implemented using the NLTK library's *single_meteor_score* function, specifically at line 58 in the Python script. This pseudocode outlines the process of dividing two texts into individual words and computing the METEOR score for them. In the context of RAG models, the METEOR score is used to assess the quality of generated responses. A high METEOR score signifies that the generated response closely aligns with the reference text, indicating the model's effectiveness in accurately retrieving and generating responses. On the other hand, a low METEOR score may highlight areas where the model's performance could be enhanced.

### 2.4.2 ROUGE (Recall-Oriented Understudy for Gisting Evaluation)

ROUGE is a collection of metrics used to assess automatic summarization and machine translation. It evaluates by comparing a machine-generated summary or translation with one or more reference summaries (typically human-generated). ROUGE includes different variants, including ROUGE-N, ROUGE-L, and ROUGE-W.

ROUGE-N measures the overlap of n-grams (sequences of n words) between the system-generated summary and the reference summaries. It is calculated using recall, precision, and F1 score:

1) **Recall for ROUGE-N** is the ratio of overlapping n-grams between the system summary and the reference summaries to the total n-grams in the reference summaries. The formula is presented in Equation (4).

$$Recall_{ROUGE-N} = \frac{\sum_{S \in \{Reference\ Summaries\}} \sum_{gram_n \in S} Count_{match}(gram_n)}{\sum_{S \in \{Reference\ Summaries\}} \sum_{gram_n \in S} Count(gram_n)} \tag{4}$$

2) **Precision for ROUGE-N** is the ratio of overlapping n-grams in the system summary to the total n-grams in the system summary itself. The formula is presented in Equation (5).

$$Precision_{ROUGE-N} = \frac{\sum_{S \in \{System\ Summaries\}} \sum_{gram_n \in S} Count_{match}(gram_n)}{\sum_{S \in \{System\ Summaries\}} \sum_{gram_n \in S} Count(gram_n)} \tag{5}$$

3) **F1 Score for ROUGE-N** is the harmonic mean of precision and recall. The formula is presented in Equation (6).

$$F1_{ROUGE-N} = 2\frac{Precision_{ROUGE-N} \times Recall_{ROUGE-N}}{Precision_{ROUGE-N} + Recall_{ROUGE-N}} \tag{6}$$

ROUGE-L emphasizes the longest common subsequence (LCS) between the generated summary and reference summaries. The LCS is the longest sequence of words that appear in both texts in the same order, but not necessarily consecutively. The parameters for ROUGE-L include:

1) **Recall for ROUGE-L** is calculated by dividing the length of the LCS by the total number of words in the reference summary. This evaluates how well the generated summary captures the content of the reference summaries. The formula is presented in Equation (7).

$$Recall_{ROUGE-L} = \frac{LCS(System\ Summary, Reference\ Summary)}{Length\ of\ Reference\ Summary} \quad (7)$$

2) **Precision for ROUGE-L** is determined by dividing the length of the LCS by the total number of words in the generated summary. This assesses how much of the generated summary's content appears in the reference summaries. The formula is presented in Equation (8).

$$Precision_{ROUGE-L} = \frac{LCS(System\ Summary, Reference\ Summary)}{Length\ of\ System\ Summary} \quad (8)$$

3) **F1 for ROUGE-L** is the harmonic mean of LCS-based precision and recall. The formula is presented in Equation (9).

$$F1_{ROUGE-L} = 2\frac{Precision_{ROUGE-L} \times Recall_{ROUGE-L}}{Precision_{ROUGE-L} + Recall_{ROUGE-L}} \quad (6)$$

ROUGE-W extends ROUGE-L by assigning more weight to longer sequences of matching words. However, in this context, ROUGE-W is not utilized.

The calculations for Equations (4) to (9) are implemented using the rouge library, specifically the *rouge.get_scores* function, located on line 65 of the Python script. The following pseudocode outlines the steps to initialize a ROUGE object and compute ROUGE scores between two texts:

1) Assign 'hypotheses' to the reference text and 'ref' to the candidate text.
2) Create a ROUGE object.
3) Use the *get_scores* method of the ROUGE object to compute ROUGE scores between 'hypotheses' and 'ref.'

## 3. RESULT AND DISCUSSION

In this section, we present the experimental design and implementation details of LLM Medical Herb, highlighting its superior performance evaluated by Dr. Tri Cahyanto, M.Si., an expert in ethnobotany. We intentionally selected datasets from reputable journals to demonstrate the promising generalization and validity of our approach. This thorough evaluation underscores the effectiveness of our method in assessing herbal medicine applications.

### 3.1 Implementation Details

At the time of implementation, the chunk size is 500 with a maximum number of tokens of 512, along with a temperature of 0.5. The training model used is Mistral 7b, where this model has 7 billion parameters. In the application implementation stage, we use a web-based application that utilizes Chainlit. Chainlit is an open-source package that has tremendous speed when it comes to developing and distributing LLM applications. It provides a seamless experience by allowing users to incorporate the Chainlit API into their existing code. We can create a user interface similar to ChatGPT.

The testing process will be conducted under the guidance of expert Dr. Tri Cahyanto, M.Si. Tables 3 to 5 present six questions posed to the Herb Medical Bot, along with the corresponding answers provided by experts. This setup facilitates a comprehensive evaluation of the bot's performance and effectiveness in delivering accurate information.

Table 6 presents the results of the evaluation matrix, comparing the responses generated by the RAG model with those provided by the experts. This comparison highlights the effectiveness of the RAG model in generating accurate and relevant answers in relation to expert opinions, providing insights into its performance and areas for improvement.

Figure 3 presents a question posed to the Mistral 7b model, asking, "What's an herb for diabetes?" The model responds as shown in the Q2 Mistral 7b Answer. Figure 4 displays the results of a question directed to the RAG Medical Chatbot utilizing the LLaMa2 7b model. In this case, the question asked was, "What's an herb for diabetes?" The model responded that Guazuma ulmifolia leaves and Trigonella foenum-graceum seeds are known for their potential to manage diabetes mellitus through their anti-diabetic effects.

**Table 3 Human Evaluation Question and Answer**

| No. | Question Prompt | Answer | | |
|---|---|---|---|---|
| | | LLaMa2 7b | Mistral 7b | Expert |
| Q1 | Herb for headache | Cinnamon is an herb that can be used to treat headaches, including migraines. It has been shown to have anti-inflammatory properties and can help to reduce inflammation in the body, which may contribute to headache pain. | Herbs are a plant or part of a plant that is used to make medicine. They can also be eaten as food, such as cinnamon. Some herbs have been used medicinally for thousands of years.<br>The properties of some herbs may be beneficial in relieving pain and muscle tension, but there is limited clinical evidence about how effective they are at treating headaches. However, it is thought that these herbs can help to reduce the symptoms of a headache if you use them as part of a regular treatment plan alongside other lifestyle changes.<br>There are many different types of herbal remedies available on the market today, but only a few have been tested scientifically for effectiveness against headache relief. Some may contain harmful ingredients, so it's important to check with your doctor before taking any natural supplement or medication.<br>A common home remedy for headaches is peppermint oil, which is applied topically around the temple area where most people feel their pain when they experience this condition. This method works well because peppermint contains menthol, which has been shown in studies to have anti-inflammatory properties that help reduce swelling and inflammation associated with headaches caused by stress or allergies. | Cinnamon is a spice that has anti-inflammatory and neuroprotective properties. Researchers were therefore interested in studying whether cinnamon could help reduce migraine attacks and inflammation. For example, this journal describes about that (Zareie et al., 2020). |
| Q2 | Herb For Diabetes | Guazuma ulmifolia leaves and Trigonella foenum-graceum seeds are known to have the potential to manage diabetes mellitus through their anti-diabetic effects. | The most popular herbs in the treatment of diabetes are bitter melon, fenugreek, bilberry, cinnamon, ginger, chromium, bitter gourd, gymnema, zinc, psyllium husk, aloe vera, magnesium, ginseng, and garlic.<br>Bitter melon (Momordica charantia), which is also known as bitter gourd or bitter squash, has been used in Chinese and Ayurvedic medicine for centuries to treat diabetes mellitus (12). In recent years, laboratory studies have demonstrated that the plant extracts lower blood sugar levels.<br>The anti-diabetic effects of bitter melon are related to its ability to increase insulin production by pancreatic beta cells and reduce glucose absorption in the intestine. The hypoglycemic effect of bitter melon is also associated with increased insulin sensitivity at the cellular level, which results in more efficient uptake of glucose by muscle and fat tissues (13). | Trigonella foenum-graecum is one of the important medicinal plants in the management of diabetes mellitus.<br>Several studies, such as (Geberemeskel et al., 2019), have investigated the effect of Trigonella foenum-graecum seed powder on the lipid profile of newly diagnosed type II diabetic patients. |

**Table 4 Human Evaluation Question and Answer (Continued)**

| No. | Question Prompt | Answer | | |
|---|---|---|---|---|
| | | LLaMa2 7b | Mistral 7b | Expert |
| Q3 | What Herb for hypertension | C. asiatica<br>The total triterpenoid fraction of C. asiatica appears to have a vasodilatory effect, which may contribute to its antihypertensive activity. | Curcuma longa is a tropical plant native to Southern Asia, and its root has been used in traditional medicine in India, China, Burma, and other parts of Southeast Asia. The rhizome of Curcuma longa, commonly known as turmeric, contains 3-5% curcumin, which has been shown to have anti-inflammatory activity.<br>The most active compound identified from Curcuma longa is curcumin (diferuloylmethane), a yellow-colored substance. Its chemical structure is similar to that of resveratrol. Curcumin was found to have antioxidant properties; it is an inhibitor of both cyclooxygenase and lipoxygenase activity, and it can attenuate the inflammatory response.<br>The main active ingredient in turmeric is curcumin, which has been used to treat arthritis pain for centuries. Turmeric supplements are available as tablets, capsules, powders, teas, and extracts. It can also be found in products like curry powder, mustard, and some yellow cheeses. | Hypertension is a disease that is quite high in Indonesia, a major risk factor for cardiovascular disease (CVD). One of the herbs used is *Centella asiatica (L) Urb.* belongs to the Apiaceae (*Umbelliferae*) plant family, which has high triterpenoids and flavonoids and has antioxidant properties and is involved in the renin-angiotensin-aldosterone system, which is an important hormonal system for blood pressure regulation. |
| Q4 | Medical herb for fever | The medicinal plant species found in the area of Ranggawulung's Urban Forest (RUF), Sudang, Indonesia, that are known to be used for treating fever include Abelmoschus manihot (L.) Medik. (Malvaceae) with its leaves used as a decoction for oral consumption, and Ageratum conyzoides L. (Compos (Arecourse with its leaves which has its leaves used as a Composita (Compositive (Compossession (Compositing (Compos (Composit (Compositive leaves (Compositing (Composit (Composica composita Babad Ot Composita (Compossession (Compos. (Arecceae(Composited with its leaves used as a Composita. (Compos. (Compos (Compos (Compos (Composited as a (Compos (Compositive (Compos (Compositive (Compositing (Compositive (Compos (Composit (Arecaceae) (Compositing the leaves used as a Composita (Compossession. (Compositing (Compositive (Compossession (Composit (Com | The plant you are looking for is Abelmoschus manihot (L.) Medik. from Malvaceae family. It is called Daun nedi/Daun mujarab, and the leaves can be used to treat fever.<br>To use it, you have to make a juice of leaves, mix it with water, and drink it orally.<br>The plant is found in Ranggawulung's Urban Forest (RUF), Sudang, Indonesia. | Some journals have reported that A. manihot and its bioactive constituents have a wide range of biological properties, including anti-diabetic nephropathy, antioxidant, antiadipogenic, anti-inflammatory, analgesic, anticonvulsant, antidepressant, antiviral, antitumor, cardioprotective, antiplatelet, neuroprotective activity, immunomodulatory, and hepatoprotective. And A.manihot can be used for fever. However, further studies and clinical trials are still needed to confirm these findings. |

**Table 5 Human Evaluation Question and Answer (Continued)**

| No. | Question Prompt | Answer | | |
|---|---|---|---|---|
| | | LLaMa2 7b | Mistral 7b | Expert |
| Q5 | Medical herb for rheumatism | Cynamaldehide from Cinnamomum verum can enhance the immune response in Rheumatoid Arthritis (RA) by inhibiting the proteins involved in RA progression and ins in RA. | Curcumin is a chemical compound found in turmeric, which has been shown to have anti-inflammatory and antioxidant properties. It is thought that curcumin may help reduce pain and stiffness associated with rheumatism. However, more research is needed to confirm its effectiveness as a treatment for rheumatism.<br>Several medical herbs have been traditionally used in the treatment of rheumatism. These include turmeric, ginger, and willow bark. Turmeric has anti-inflammatory and antioxidant properties, while ginger can help reduce pain and stiffness. Willow bark contains salicylic acid, which is a compound that is similar to aspirin and can also help reduce inflammation and pain.<br>Curcumin has been shown to have anti-inflammatory and antioxidant properties. It may help reduce pain and stiffness associated with rheumatism. However, more research is needed to confirm its effectiveness as a treatment for rheumatism. | Rheumatoid arthritis is one part of rheumatic disease. In Indonesia, some herbs that are often used for rheumatism are jambe, jackfruit, and several other examples, such as cinnamon, curcumin, African tree, and so on. |
| Q6 | Medical herb for Hartburn | The herb commonly used to treat heartburn is ginger. Ginger has natural anti-inflammatory properties that can help reduce inflammation in the esophagus and alleviate symptoms of heartburn. | Medicinal plants have been used in Ayurveda, traditional Chinese medicine, the Unani system of medicine, folk medicine, and home remedies since ancient times. In India, about 80 percent of people still depend on herbal medicines for their primary healthcare needs. In the past two decades, research has been focused on identifying and testing the active ingredients in traditional medicinal plants. This research is aimed at developing safe and effective drug products based on natural compounds that are able to treat specific diseases. According to a 2008 report, approximately 75 percent of the population in rural areas of India rely solely on traditional medicines for their primary healthcare needs. Traditional medicine systems, including Ayurveda, Siddha, Unani, and naturopathy, were included in the Indian Constitution in 1970 under Schedule H of the Drugs and Cosmetic Act. | Some sources have been researched, such as Harvard Medical School, which states that ginger root is a popular herbal remedy for heartburn. It has been used for centuries to relieve the symptoms of heartburn, such as a burning sensation in the chest. |

**Table 6 Metric Evaluation**

| Question | Model | ROUGE Precision | ROUGE Recall | ROUGE F-Measure | METEOR |
|---|---|---|---|---|---|
| Q1 | Mistral 7b | 0.09 | 0.29 | 0.13 | 0.23 |
| | LLaMa2 7b | 0.31 | 0.17 | 0.22 | 0.13 |
| Q2 | Mistral 7b | 0.05 | 0.33 | 0.09 | 0.26 |
| | LLaMa2 7b | 0.17 | 0.31 | 0.22 | 0.24 |
| Q3 | Mistral 7b | 0.09 | 0.20 | 0.12 | 0.24 |
| | LLaMa2 7b | 0.18 | 0.06 | 0.09 | 0.06 |
| Q4 | Mistral 7b | 0.09 | 0.10 | 0.10 | 0.16 |
| | LLaMa2 7b | 0.06 | 0.12 | 0.08 | 0.11 |
| Q5 | Mistral 7b | 0.05 | 0.22 | 0.09 | 0.21 |
| | LLaMa2 7b | 0.16 | 0.11 | 0.13 | 0.08 |
| Q6 | Mistral 7b | 0.07 | 0.25 | 0.11 | 0.22 |
| | LLaMa2 7b | 0.21 | 0.15 | 0.17 | 0.24 |
| Average | Mistral 7b | 0.07 | 0.23 | 0.10 | 0.22 |
| | LLaMa2 7b | 0.18 | 0.15 | 0.15 | 0.14 |

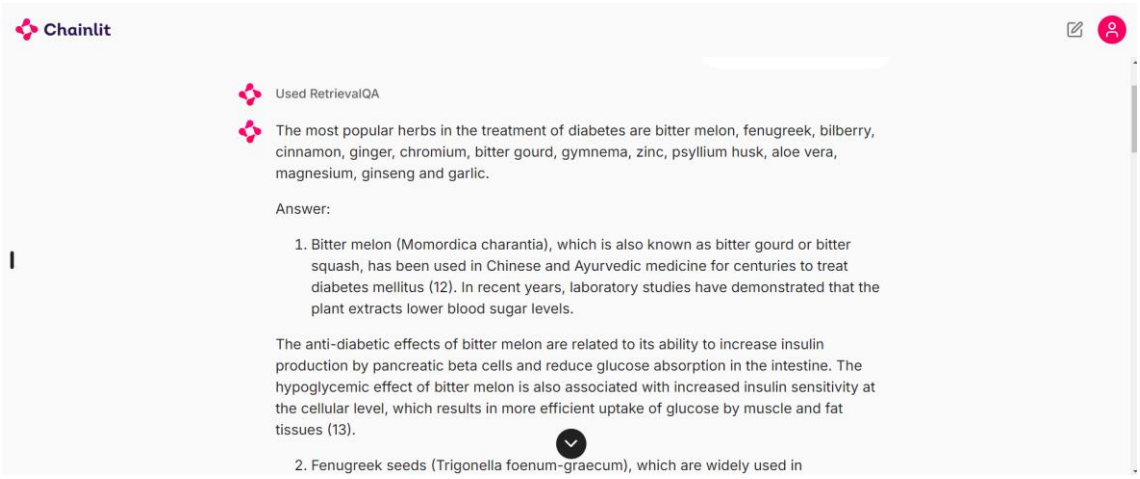



**Figure 3 Question and Answer Medical Herb with Mistral 7b**

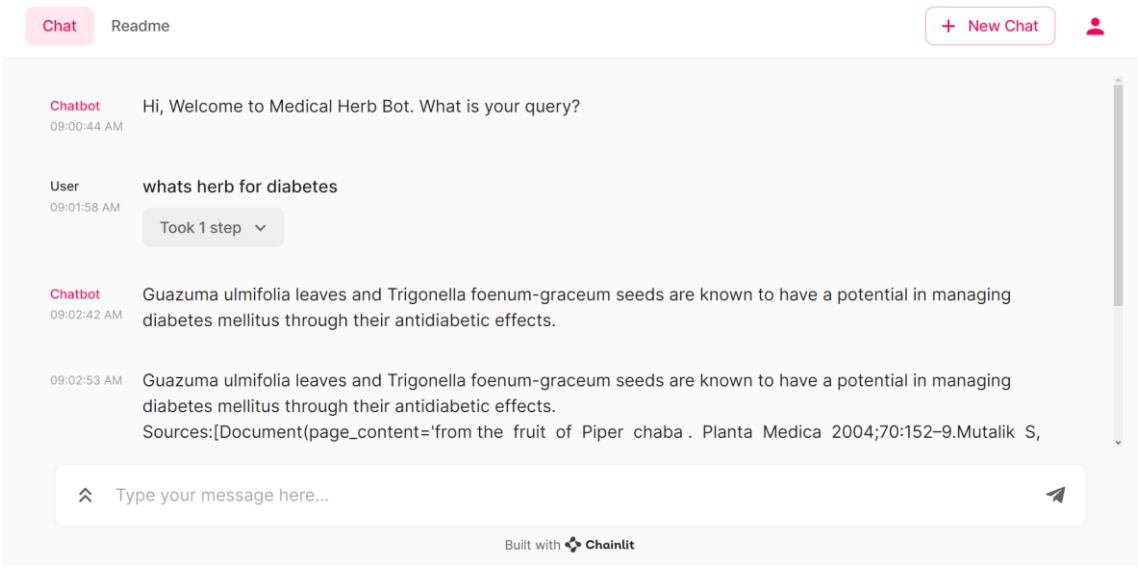



**Figure 4 Question and Answer Medical Herb with LLaMa2 7b**

### 3.2 Discuss

In the Metric Evaluation at Precision section, Mistral 7b has a low percentage because the output generated by Mistral 7b exhibits a high level of creativity, even with the same temperature setting of 0.5. However, when evaluated using METEOR, Mistral 7b shows a high percentage, making the Mistral model arguably better than the LLaMa2 7b model. However, if you prefer a model with higher precision in text, you might choose LLaMa2 7b because it has higher text precision.

## 4. CONCLUSION

Indonesia has a large and abundant number of plants, but their utilization is still derived from customs only. Several studies have been conducted to explore the potential of plants in Indonesia, and they have been stored in accredited journals, which will be used as datasets. With RAG, we can obtain answers where each response is valid based on journals and contains data on herbal plants from Indonesia. Based on the results of the table above, the use of RAG Mistral 7b as an LLM model for Question Answering Medical Herb can be stated quite well. When viewed from six questions, the answers from the medical herb LLM and the solutions from the experts are by the

average Score from METEOR is 0.22% where the score given is higher than LLaMa2 7b, which only received 0.14%.

Furthermore, the precision score for Mistral 7b is only 0.07%, compared to 0.18% for LLaMa2 7b. This is because the Mistral 7b model answers questions creatively, as seen in Table 1. The creative responses from Mistral 7b reduce the precision score compared to expert answers. Next, one of the answers from LLaMa2 7b, specifically for question Q4, contains repetitive content, indicating that the model did not perform RAG correctly and effectively. Meanwhile, the answer from Mistral 7b does not contain repetitive content.

For further research, tests should be conducted with several experts and several questions related to herbal medicines for several diseases. Furthermore, journal sources should be added to the dataset so that the answers from LLM Medical Herb are more valid and qualified. This approach would not only improve the model's reliability but also contribute to a deeper understanding of the therapeutic potential of various herbs.

## REFERENCES

Ardiyanto, D., Triyono, A., Nisa, U., Fitriani, U., Astana, P. R., Novianto, F., & Zulkarnain, Z. (2021). The use of hyperuricemia herbs at “Hortus Medicus” herbal medicine clinic Tawangmangu. *Jurnal Kedokteran Dan Kesehatan Indonesia*. https://doi.org/10.20885/JKKI.Vol12.Iss2.art9

Arozal, W., Louisa, M., & Soetikno, V. (2020). Selected Indonesian Medicinal Plants for the Management of Metabolic Syndrome: Molecular Basis and Recent Studies. *Frontiers in Cardiovascular Medicine*, *7*. https://doi.org/10.3389/fcvm.2020.00082

Chung, H. W., Hou, L., Longpre, S., Zoph, B., Tay, Y., Fedus, W., Li, Y., Wang, X., Dehghani, M., Brahma, S., Webson, A., Gu, S. S., Dai, Z., Suzgun, M., Chen, X., Chowdhery, A., Castro-Ros, A., Pellat, M., Robinson, K., … Wei, J. (2022). *Scaling Instruction-Finetuned Language Models: Vol. 1?54* (H. W. Chung, S. Longpre, B. Zoph, A. Castro-ros, A. Yu, & A. Dai, Eds.). http://arxiv.org/abs/2210.11416

Elfahmi, Woerdenbag, H. J., & Kayser, O. (2014). Jamu: Indonesian traditional herbal medicine towards rational phytopharmacological use. *Journal of Herbal Medicine*, *4*(2), 51–73. https://doi.org/10.1016/j.hermed.2014.01.002

Fathir, A., HAIKAL, MOCH., & Wahyudi, D. (2021). Ethnobotanical study of medicinal plants used for maintaining stamina in Madura ethnic, East Java, Indonesia. *Biodiversitas Journal of Biological Diversity*, *22*(1), 386–392. https://doi.org/10.13057/biodiv/d220147

Geberemeskel, G. A., Debebe, Y. G., & Nguse, N. A. (2019). Antidiabetic Effect of Fenugreek Seed Powder Solution ( *Trigonella foenum-graecum L* .) on Hyperlipidemia in Diabetic Patients. *Journal of Diabetes Research*, *2019*, 1–8. https://doi.org/10.1155/2019/8507453

Hadi, M. U., Al-tashi, Q., Qureshi, R., Shah, A., Muneer, A., Irfan, M., Zafar, A., Shaikh, M. B., Akhtar, N., Hassan, S. Z., Shoman, M., Wu, J., Mirjalili, S., & Shah, M. (2024). Large Language Models: A Comprehensive Survey of its Applications, Challenges, Limitations, and Future Prospects. *TechRxiv*, 1–47. https://doi.org/10.36227/techrxiv.23589741.v2

Jain, N., Saifullah, K., Wen, Y., Kirchenbauer, J., Shu, M., Saha, A., Goldblum, M., Geiping, J., & Goldstein, T. (2023). *Bring Your Own Data! Self-Supervised Evaluation for Large Language Models*. http://arxiv.org/abs/2306.13651

Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., Casas, D. de las, Bressand, F., Lengyel, G., Lample, G., Saulnier, L., Lavaud, L. R., Lachaux, M.-A., Stock, P., Scao, T. Le, Lavril, T., Wang, T., Lacroix, T., & Sayed, W. El. (2023). *Mistral 7B: Vol. 7b.? 1?9*. http://arxiv.org/abs/2310.06825

Kaddour, J., Harris, J., Mozes, M., Bradley, H., Raileanu, R., & McHardy, R. (2023). *Challenges and Applications of Large Language Models*. http://arxiv.org/abs/2307.10169

Kartini, K., Jayani, N. I. E., Octaviyanti, N. D., Krisnawan, A. H., & Avanti, C. (2019). Standardization of Some Indonesian Medicinal Plants Used in “Scientific Jamu.” *IOP Conference Series: Earth and Environmental Science*, *391*(1), 012042. https://doi.org/10.1088/1755-1315/391/1/012042

Luo, R., Sun, L., Xia, Y., Qin, T., Zhang, S., Poon, H., & Liu, T.-Y. (2022). BioGPT: generative pre-trained transformer for biomedical text generation and mining. *Briefings in Bioinformatics*, *23*(6). https://doi.org/10.1093/bib/bbac409

OpenAI, Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., Avila, R., Babuschkin, I., Balaji, S., Balcom, V., Baltescu, P., Bao, H., Bavarian, M., Belgum, J., … Zoph, B. (2023). *GPT-4 Technical Report*. *4*, 1–100. http://arxiv.org/abs/2303.08774

Putri, L. S. E., Dasumiati, D., Kristiyanto, K., Mardiansyah, M., Malik, C., Leuvinadrie, L. P., & Mulyono, E. A. (1970). Ethnobotanical study of herbal medicine in Ranggawulung Urban Forest, Subang District, West Java, Indonesia. *Biodiversitas Journal of Biological Diversity*, *17*(1), 172–176. https://doi.org/10.13057/biodiv/d170125

Radeva, I., Popchev, I., Doukovska, L., & Dimitrova, M. (2024). Web Application for Retrieval-Augmented Generation: Implementation and Testing. *Electronics*, *13*(7), 1361. https://doi.org/10.3390/electronics13071361

Ray, P. P. (2023). ChatGPT: A comprehensive review on background, applications, key challenges, bias, ethics, limitations and future scope. *Internet of Things and Cyber-Physical Systems*, *3*, 121–154. https://doi.org/10.1016/j.iotcps.2023.04.003

Ren, X., Zhou, P., Meng, X., Huang, X., Wang, Y., Wang, W., Li, P., Zhang, X., Podolskiy, A., Arshinov, G., Bout, A., Piontkovskaya, I., Wei, J., Jiang, X., Su, T., Liu, Q., & Yao, J. (2023). *PanGu-Σ: Towards Trillion Parameter Language Model with Sparse Heterogeneous Computing*. http://arxiv.org/abs/2303.10845

Sholikhah, E. N. (2016). Indonesian medicinal plants as sources of secondary metabolites for pharmaceutical industry. *Journal of the Medical Sciences (Berkala Ilmu Kedokteran)*, *48*(04), 226–239. https://doi.org/10.19106/JMedSci004804201606

Sianipar, E. A. (2021). The Potential of Indonesian Traditional Herbal Medicine as Immunomodulatory Agents: A Review. *International Journal of Pharmaceutical Sciences and Research*, *12*(10), 5229–5237. https://doi.org/10.13040/IJPSR.0975-8232.12(10).5229-37

Singhal, K., Azizi, S., Tu, T., Mahdavi, S. S., Wei, J., Chung, H. W., Scales, N., Tanwani, A., Cole-Lewis, H., Pfohl, S., Payne, P., Seneviratne, M., Gamble, P., Kelly, C., Babiker, A., Schärli, N., Chowdhery, A., Mansfield, P., Demner-Fushman, D., … Natarajan, V. (2023). Large language models encode clinical knowledge. *Nature*, *620*(7972), 172–180. https://doi.org/10.1038/s41586-023-06291-2

Singhal, K., Tu, T., Gottweis, J., Sayres, R., Wulczyn, E., Hou, L., Clark, K., Pfohl, S., Cole-Lewis, H., Neal, D., Schaekermann, M., Wang, A., Amin, M., Lachgar, S., Mansfield, P., Prakash, S., Green, B., Dominowska, E., Arcas, B. A. y, … Natarajan, V. (2023). *Towards Expert-Level Medical Question Answering with Large Language Models*. http://arxiv.org/abs/2305.09617

Sumarni, W., Sudarmin, S., & Sumarti, S. S. (2019). The scientification of jamu: a study of Indonesian's traditional medicine. *Journal of Physics: Conference Series*, *1321*(3), 032057. https://doi.org/10.1088/1742-6596/1321/3/032057

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., & Lample, G. (2023). *LLaMA: Open and Efficient Foundation Language Models*. http://arxiv.org/abs/2302.13971

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., Bikel, D., Blecher, L., Ferrer, C. C., Chen, M., Cucurull, G., Esiobu, D., Fernandes, J., Fu, J., Fu, W., … Scialom, T. (2023). *Llama 2: Open Foundation and Fine-Tuned Chat Models*. http://arxiv.org/abs/2307.09288

Wang, T., Yu, P., Tan, X. E., O'Brien, S., Pasunuru, R., Dwivedi-Yu, J., Golovneva, O., Zettlemoyer, L., Fazel-Zarandi, M., & Celikyilmaz, A. (2023). *Shepherd: A Critic for Language Model Generation*. https://arxiv.org/abs/2308.04592v1

Zareie, A., Sahebkar, A., Khorvash, F., Bagherniya, M., Hasanzadeh, A., & Askari, G. (2020). Effect of cinnamon on migraine attacks and inflammatory markers: A randomized double-blind placebo-controlled trial. *Phytotherapy Research*, *34*(11), 2945–2952. https://doi.org/10.1002/ptr.6721

Zhang, K., Zhou, R., Adhikarla, E., Yan, Z., Liu, Y., Yu, J., Liu, Z., Chen, X., Davison, B. D., Ren, H., Huang, J., Chen, C., Zhou, Y., Fu, S., Liu, W., Liu, T., Li, X., Chen, Y., He, L., … Sun, L. (2024). BiomedGPT: A Generalist Vision-Language Foundation Model for Diverse Biomedical Tasks. *Nature Medicine*. https://doi.org/10.1038/s41591-024-03185-2

Zhang, T., Huang, Z., Wang, Y., Wen, C., Peng, Y., & Ye, Y. (2022). Information Extraction from the Text Data on Traditional Chinese Medicine: A Review on Tasks, Challenges, and Methods from 2010 to 2021. *Evidence-Based Complementary and Alternative Medicine*, *2022*, 1–19. https://doi.org/10.1155/2022/1679589

Zhu, X., Li, J., Liu, Y., Ma, C., & Wang, W. (2023). *A Survey on Model Compression for Large Language Models*. https://arxiv.org/abs/2308.07633v4


9 772527 583007