

BAB II

Kerangka Teoritis

A. Landasan Teori

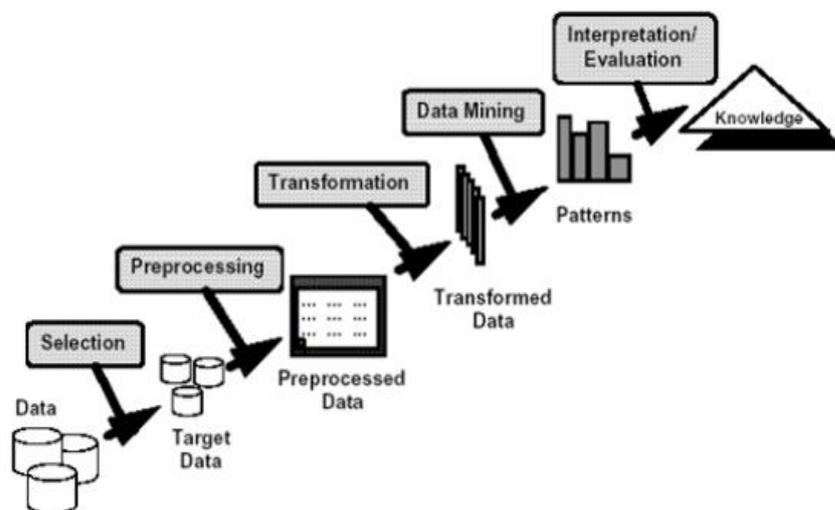
1. Bantuan Sosial

Menurut Kementerian Sosial (2011, p.15) bantuan sosial adalah bantuan yang sifatnya sementara yang diberikan kepada masyarakat miskin, dengan maksud agar mereka dapat meningkatkan kehidupannya secara wajar. Program bantuan sosial merupakan salah satu komponen program jaminan sosial yang menjadi bentuk realisasi tanggung jawab pemerintah atau pemerintah daerah yang sangat peduli terhadap kondisi Masyarakat yang miskin dan terlantar di tingkat bawah.

Secara umum, bantuan sosial adalah intervensi pemerintah dalam mengatasi kemiskinan dan meningkatkan kesejahteraan masyarakat melalui distribusi sumber daya ekonomi kepada mereka yang paling membutuhkan. Program seperti ini tidak hanya penting dalam membantu masyarakat miskin memenuhi kebutuhan sehari-hari, tetapi juga berperan dalam stabilisasi ekonomi nasional dengan meningkatkan daya beli dan mengurangi ketimpangan ekonomi.

2. Data Mining

Menurut Vlandari (2017, p.1), data mining adalah proses analisis untuk menemukan pengetahuan yang tersembunyi dalam *database*. Data mining digunakan untuk menggali pengetahuan dari basis data yang besar, sehingga sering disebut sebagai *Knowledge Discovery In Database* (KDD). Proses KDD melibatkan beberapa tahap, seperti gambar dibawah ini.



Gambar 2.1 Tahapan Proses KDD

Dari gambar 2.1 tentang tahapan proses kdd terdiri dari beberapa tahapan-tahapan KDD, yaitu :

1. *Data Selection*. Pada proses ini dilakukan pemilihan himpunan data, menciptakan himpunan data target, atau memfokuskan pada subset *variable* (*sampel data*) dimana penemuan (*discovery*) akan dilakukan. Hasil seleksi disimpan dalam suatu berkas yang terpisah dari basis data operasional.
2. *Pre-Processing* dan *Cleaning Data*. *Pre-Processing* dan *Cleaning Data* dilakukan membuang data yang tidak konsisten dan *noise*, duplikasi data, memperbaiki kesalahan data, dan bisa diperkaya dengan data eksternal yang relevan.
3. *Transformation*. Proses ini *mentransformasikan* atau menggabungkan data kedalam yang lebih tepat untuk melakukan proses *mining* dengan cara melakukan peringkasan (*agregasi*).
4. Data Mining. Proses Data Mining yaitu proses mencari pola atau informasi menarik dalam data terpilih dengan menggunakan teknik, metode atau algoritma tertentu sesuai dengan tujuan dari proses KDD secara keseluruhan.
5. *Interpretation* / Evaluasi. Proses untuk menerjemahkan pola-pola yang dihasilkan dari Data Mining. Mengevaluasi (menguji) apakah pola atau informasi yang ditemukan bersesuaian atau bertentangan dengan fakta atau hipotesa sebelumnya. Pengetahuan yang diperoleh dari pola-pola yang terbentuk dipresentasikan dalam bentuk visualisasi.

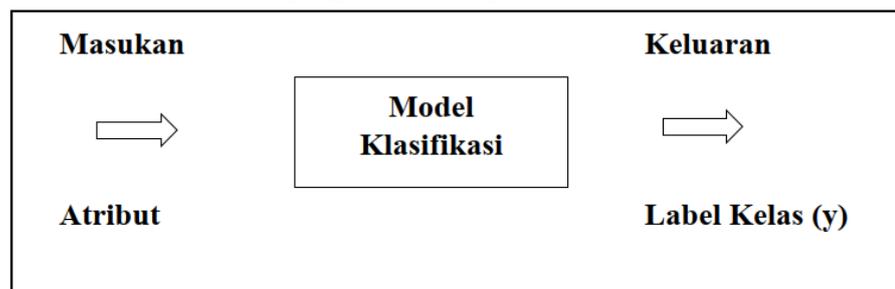
3. Supervised Learning

Supervised learning adalah pendekatan dalam pembelajaran mesin di mana model dilatih menggunakan data yang telah diberi label sebelumnya. Data latih ini terdiri dari pasangan input-output, dimana model belajar untuk memetakan input ke output yang benar. Setelah model dilatih, ia dapat digunakan untuk membuat prediksi pada data baru yang tidak diberi label. Goodfellow, Bengio, dan Courville (2016, p.137) dalam bukunya *Deep Learning* mengemukakan bahwa *supervised learning* adalah salah satu pendekatan yang paling banyak digunakan dalam pembelajaran mesin karena kesederhanaan dan efektivitasnya.

4. Klasifikasi

Menurut Hermawati (2013, p.55), Klasifikasi merupakan proses pembelajaran suatu fungsi tujuan (target) f yang memetakan tiap himpunan atribut x ke satu dari label kelas yang didefinisikan sebelumnya. Fungsi target disebut juga model klasifikasi.

Klasifikasi adalah salah satu metode yang sering digunakan dalam data mining setelah data melewati tahapan *preprocessing*. Metode ini bertujuan untuk mengekstraksi pola-pola yang mendeskripsikan kelas-kelas dalam data dan mengelompokkannya ke dalam salah satu kelas yang telah teridentifikasi sebelumnya. Proses klasifikasi melibatkan dua tahap utama. Pertama, tahap pembelajaran (*training*) di mana algoritma klasifikasi digunakan untuk menganalisis data latih dan mengubahnya menjadi aturan klasifikasi. Kedua, tahap klasifikasi itu sendiri, di mana data uji digunakan untuk mengevaluasi akurasi dari aturan klasifikasi yang telah dibangun.



Gambar 2.2 Konsep Klasifikasi

Ada dua jenis model klasifikasi, yaitu:

- a. Pemodelan deskriptif (*descriptive modelling*), yaitu model klasifikasi yang dapat berfungsi sebagai suatu alat penjelasan untuk membedakan objek-objek dalam kelas-kelas yang berbeda.
- b. Pemodelan prediktif (*predictive modelling*), yaitu klasifikasi yang dapat digunakan untuk memprediksi label kelas *record* yang tidak ketahui.

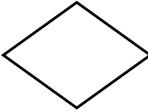
Dalam klasifikasi, data latih yang digunakan sudah memiliki *class label* yang diketahui, menjadikannya bagian dari *supervised learning*, yang bertujuan untuk memprediksi *class label* dari data yang ingin diklasifikasikan. Ini berbeda dengan *unsupervised learning*, atau *clustering*, dimana *class label* dari data latih tidak diketahui sebelumnya, dan jumlah kelompok yang akan dipelajari belum ditentukan (Han, Pei, & Tong, 2022, p.7). Klasifikasi pada dasarnya merupakan proses untuk menemukan pola yang dapat menggambarkan dan membedakan antara satu kelas dengan kelas lainnya, serta menentukan ke dalam label tertentu yang sudah teridentifikasi sebelumnya.

5. Flowchart

Menurut Albert dkk (2009, p.495) dalam bukunya *Buku Pintar Pekerja Sosial – Jilid 2*, *Flowchart* adalah sebuah metode yang digunakan untuk menyajikan serangkaian peristiwa atau tahapan dalam suatu proses atau layanan ke dalam bentuk diagram alir atau urutan yang aktual. Metode ini berfungsi untuk

memberikan gambaran visual yang jelas tentang alur suatu proses, sehingga memudahkan pemahaman dan analisis. Sebagai alat yang sangat fleksibel, *flowchart* memungkinkan anggota tim untuk menguraikan dan menganalisis proses yang kompleks dengan lebih terstruktur, yang pada gilirannya membantu dalam mengidentifikasi potensi masalah yang mungkin muncul serta menemukan berbagai peluang untuk melakukan peningkatan dan optimalisasi proses secara keseluruhan. Dengan *flowchart*, setiap tahap dalam proses dapat diperiksa secara mendetail, sehingga area-area yang membutuhkan perhatian khusus atau perbaikan dapat diidentifikasi dengan lebih akurat. Adapun berikut ini adalah simbol standar yang digunakan dalam diagram alur program di tabel 2.1.

Tabel 2.1 Simbol *Flowchart*

No.	Nama	Simbol	Keterangan
1	Terminator		Simbol yang menyatakan permulaan atau akhir suatu program
2	Input/Output Data		Simbol yang menyatakan proses input/output
3	Proses		Simbol yang menyatakan proses yang dilakukan komputer
4	Sub Program (<i>Predefined Process</i>)		Simbol untuk sub program/proses menjalankan sub program
5	Input/Output Dokumen		Simbol yang menyatakan input/output dalam bentuk dokumen
6	<i>Decision</i>		Simbol yang menunjukkan kondisi tertentu yang akan menghasilkan atau mengambil keputusan
7	<i>On Page Connector</i>		Simbol untuk menghubungkan bagianbagian dari diagram alur yang dilanjutkan pada

No.	Nama	Simbol	Keterangan
8	Garis Alir (<i>Flow Line</i>)		Simbol yang digunakan untuk menggabungkan antara simbol yang satu dengan simbol yang lain

Sumber: (Martin Fowler, 2003)

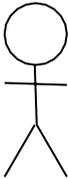
6. *Unified Modeling Language (UML)*

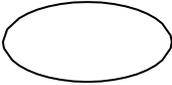
UML (*Unified Modeling Language*) adalah bahasa pemodelan standar yang digunakan untuk menentukan, memvisualisasikan, membangun, dan mendokumentasikan artefak sistem perangkat lunak. UML menyediakan serangkaian teknik notasi grafis untuk membuat model visual dari sistem perangkat lunak yang berbasis objek (Martin Fowler, 2003, p. 28):

a. *Use Case Diagram*

Diagram use case digunakan untuk menangkap kebutuhan fungsional dari sebuah sistem dengan memodelkan interaksinya dengan entitas eksternal, yang disebut aktor. Diagram ini berfokus pada apa yang harus dilakukan sistem dari perspektif pengguna, menunjukkan bagaimana pengguna (aktor) berinteraksi dengan sistem untuk mencapai tujuan mereka melalui *use case* tertentu (Martin Fowler, 2003, p. 119).

Tabel 2.2 Simbol *Use Case Diagram*

SIMBOL	NAMA	KETERANGAN
	ACTOR	Orang proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi dibuat itu sendiri, jadi walaupun simbol dari actor adalah gambar orang, biasanya dinyatakan menggunakan kata benda di awal frase nama actor

SIMBOL	NAMA	KETERANGAN
	<i>USECASE</i>	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau actor biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama usecase.
	<i>ASOSIASI/ ASSOCIATION</i>	Komunikasi antara actor dan usecase yang berpartisipasi pada usecase atau usecase memiliki interaksi dengan actor.
<i><<extend>></i> 	<i>EKSTENSI/ EXTEND</i>	Relasi usecase tambahan ke sebuah usecase dimana usecase yang ditambahkan dapat berdiri sendiri walau tanpa usecase tambahan memiliki nama depan yang sama dengan usecase yang ditambahkan.
	<i>GENERALISASI/ GENERALIZATION</i>	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah usecase dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
<i><<include>></i> 	<i>MENGGUNAKAN INCLUDE</i>	Relasi usecase tambahan ke sebuah usecase untuk menjalankan fungsional

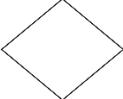
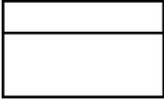
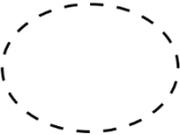
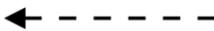
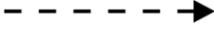
Sumber: (Martin Fowler, 2003)

b. *Class Diagram*

Diagram kelas menggambarkan jenis-jenis objek dalam sistem dan berbagai jenis hubungan statis yang ada di antara mereka. Diagram kelas juga menunjukkan properti dan operasi dari sebuah kelas serta batasan-

batasan yang berlaku pada cara objek dihubungkan. UML menggunakan istilah 'fitur' sebagai istilah umum yang mencakup properti dan operasi dari sebuah kelas (Martin Fowler, 2003, p. 61).

Tabel 2.3 Simbol *Class Diagram*

SIMBOL	NAMA	KETERANGAN
	<i>Generalization</i>	Hubungan dimana objek anak (descendent) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (ancestor)
	<i>Navy Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek
	<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama
	<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor.
	<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek
	<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (independent) akan mempengaruhi elemen yang bergantung pada elemen yang tidak mandiri.
	<i>Association</i>	Yang menghubungkan antara objek satu dengan objek lainnya.

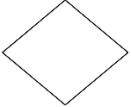
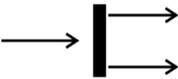
Sumber: (Martin Fowler, 2003)

c. *Activity Diagram*

Diagram aktivitas adalah teknik untuk menggambarkan logika prosedural, proses bisnis, dan alur kerja. Dalam banyak hal, diagram ini

memiliki peran yang mirip dengan diagram alir, tetapi perbedaan utamanya adalah bahwa diagram aktivitas mendukung perilaku paralel (Martin Fowler, 2003, p. 133).

Tabel 2.4 Simbol *Activity Diagram*

SIMBOL	NAMA	KETERANGAN
	Status Awal/Initial	Status awal aktivitas sistem
	Activity	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
	Decision	Asosiasi percabangan dimana lebih dari satu aktivitas digabungkan menjadi satu.
	Join	Asosiasi penggabungan dimana lebih dari satu aktivitas lebih dari satu.
	Status Akhir/Final	Status akhir yang dilakukan sistem.
	Swimlane	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

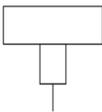
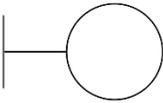
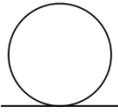
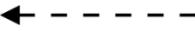
Sumber: (Martin Fowler, 2003)

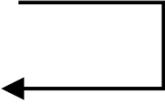
d. Sequence Diagram

Diagram interaksi menggambarkan bagaimana kelompok objek bekerja sama dalam suatu perilaku. UML mendefinisikan beberapa bentuk diagram interaksi, di antaranya yang paling umum adalah diagram urutan. Biasanya, diagram urutan menangkap perilaku dari satu skenario. Diagram ini

menunjukkan sejumlah objek contoh dan pesan-pesan yang dikirimkan antara objek-objek tersebut dalam kasus penggunaan (Martin Fowler, 2003, p. 79).

Tabel 2.5 Simbol Sequence Diagram

Simbol	Nama	Keterangan
	<i>Actor</i>	Mempresentasikan entitas yang berada diluar sistem dan berinteraksi diluar sistem.
	<i>Lifeline</i>	Menghubungkan objek selama sequence (message dikirim atau diterima).
	<i>General</i>	Merepresentasikan entitas tunggal dalam sequence
	<i>Boundary</i>	Berupa tepi dari sistem, seperti user interface dan alat yang berinteraksi dengan yang lain.
	<i>Control</i>	Elemen mengatur aliran dari informasi untuk sebuah scenario
	<i>Entity</i>	Elemen yang bertanggung jawab menyimpan informasi
	<i>Activation</i>	Suatu titik dimana sebuah objek mulai berpartisipasi dalam sebuah sequence yang menunjukkan sebuah objek atau mengirim atau menerima objek.
	<i>Message Entry</i>	Berfungsi untuk menggambarkan pesan/ hubungan antar objek yang menunjukkan urutan kejadian yang terjadi.
	<i>Message to Self</i>	Simbol ini menggambarkan pesan/hubungan objek itu sendiri, yang menunjukkan urutan kejadian yang terjadi.

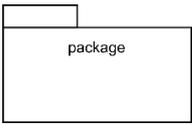
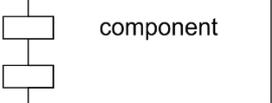
Simbol	Nama	Keterangan
	<i>Message Return</i>	Menggambarkan hasil dari pengiriman message yang digambarkan dengan arah dari kanan ke kiri.

Sumber: (Martin Fowler, 2003)

e. *Component Diagram*

Diagram komponen digunakan untuk memodelkan komponen-komponen fisik dalam sebuah sistem dan ketergantungan di antara mereka. Diagram ini menggambarkan arsitektur sistem pada tingkat abstraksi yang lebih tinggi daripada diagram kelas, menunjukkan bagaimana komponen-komponen saling berinteraksi melalui antarmuka dan bagaimana mereka saling terhubung. Diagram komponen membantu dalam memahami organisasi dan struktur dari komponen perangkat keras dan perangkat lunak sistem (Martin Fowler, 2003, p. 150). Berikut merupakan simbol-simbol yang dimiliki oleh *component diagram* :

Tabel 2.6 Simbol *Component Diagram*

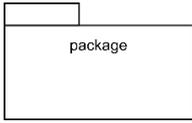
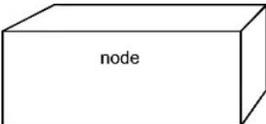
Simbol	Nama	Keterangan
	<i>Package</i>	<i>Package</i> merupakan sebuah bungkusan dari satu atau lebih komponen
	<i>Component</i>	Komponen sistem
	<i>Dependency</i>	Kebergantungan antar komponen, arah panah mengarah pada komponen yang dipakai.
	<i>Interface</i>	Sebagai antarmuka komponen agar tidak mengakses langsung komponen.
	<i>Link</i>	Relasi antar komponen

Sumber: (Martin Fowler, 2003)

f. *Deployment Diagram*

Diagram penyebaran menunjukkan tata letak fisik sebuah sistem, mengungkapkan perangkat lunak mana yang dijalankan pada perangkat keras apa. Diagram penyebaran sebenarnya sangat sederhana; oleh karena itu babnya pendek (Martin Fowler, 2003, p. 117). Berikut symbol-simbol yang digunakan untuk menggambarkan *deployment diagram*:

Tabel 2.7 Simbol *Deployment Diagram*

Simbol	Nama	Keterangan
	<i>Package</i>	<i>Package</i> merupakan sebuah bungkus dari satu atau lebih <i>node</i>
	<i>Node</i>	<i>Node</i> biasanya mengacu pada perangkat keras (<i>hardware</i>), perangkat lunak yang tidak dibuat sendiri (<i>software</i>), jika di dalam <i>node</i> disertakan komponen untuk mengkonsistenkan rancangan maka komponen yang diikutsertakan harus sesuai dengan komponen yang telah didefinisikan sebelumnya pada diagram komponen.
	<i>Dependency</i>	Kebergantungan antar <i>node</i> , arah panah mengarah pada <i>node</i> yang dipakai.
	<i>Link</i>	Relasi antar <i>node</i> .

Sumber: (Martin Fowler, 2003)

7. Metode Algoritma *K-Nearest Neighbors*

Metode *K-Nearest Neighbors* pertama kali diperkenalkan pada awal tahun 1950-an, namun baru mulai populer pada tahun 1960-an karena memerlukan sumber daya komputasi yang signifikan. Metode ini banyak digunakan dalam pengenalan pola dan bergantung pada konsep pembelajaran dengan analogi. Dalam metode ini, data uji dibandingkan dengan data pelatihan yang memiliki

kemiripan berdasarkan atribut tertentu. Setiap data pelatihan direpresentasikan sebagai titik dalam ruang berdimensi n, dan seluruh data pelatihan ini disimpan dalam ruang pola berdimensi n. Ketika diberikan data yang tidak diketahui, *K-Nearest Neighbors Classifier* akan mencari k data pelatihan terdekat dalam ruang pola tersebut. K data pelatihan terdekat ini disebut sebagai "tetangga terdekat". Metode ini telah digunakan secara luas dan terbukti efektif dalam pengenalan pola.

K-Nearest Neighbor (KNN) merupakan sebuah metode untuk melakukan klasifikasi terhadap obyek baru berdasarkan (K) tetangga terdekatnya (Han dkk, 2012, p.424). KNN termasuk algoritma *supervised learning*, yang mana hasil dari *query instance* baru, diklasifikasikan berdasarkan mayoritas dari kategori pada KNN. Kelas yang paling banyak muncul, yang akan menjadi kelas hasil klasifikasi. Pada algoritma KNN terdapat 5 (lima) cara, untuk mencari tetangga terdekat yaitu:

1. Jarak *Euclidean*
2. Jarak *Manhattan*
3. Jarak *Cosine*
4. Jarak *Correlation*
5. Jarak *Hamming*

Pada penelitian ini penulis hanya menggunakan jarak *Euclidean*, maka rumus perhitungan jarak dengan *Euclidean* seperti di bawah ini :

$$d(x, y) = \sqrt{\sum_{k=1}^n (x_k - y_k)^2}$$

Keterangan :

$d(i,j)$ = Jarak dokumen i ke j.

$x_i(n)$ = Kata ke-n pada dokumen i.

$x_j(n)$ = Kata ke-n pada dokumen j.

8. Confusion Matrix

Menurut Rahayu dkk (2024, p.103) *Confusion Matrix* merupakan sebuah matriks yang berfungsi untuk mengukur kinerja model pembelajaran terarah (*supervised learning*), terutama pada tugas-tugas klasifikasi dalam *machine learning*. Matriks ini memberikan gambaran rinci tentang bagaimana model melakukan prediksi dengan mengidentifikasi jumlah prediksi yang benar dan salah, sehingga memudahkan evaluasi performa model secara keseluruhan. Dalam mengukur tingkat akurasi sistem orientasi sentimen, maka digunakan tabel *confusion matrix*.

B. Studi Kasus Penerapan Algoritma *K-Nearest Neighbors*

Berdasarkan buku RapidMiner: *Data Mining Use Cases and Business Analytics Applications* oleh Markus Hofmann dan Ralf Klinkenberg, studi kasus terkait algoritma *K-Nearest Neighbor* (KNN) dijelaskan di Bab 3, halaman 33 hingga 44. Studi kasus ini bertujuan untuk mengklasifikasikan kinerja asisten pengajar berdasarkan data historis dari evaluasi mahasiswa. Algoritma *K-Nearest Neighbor* (KNN) digunakan untuk memprediksi kategori evaluasi berdasarkan fitur-fitur tertentu. KNN dipilih karena kesederhanaan algoritmanya dan kemampuannya menangani kasus klasifikasi dengan jumlah data kecil hingga menengah.

Dataset yang digunakan dalam studi kasus ini adalah *Teacher Assistant Evaluation* (TAE), yang terdiri dari beberapa atribut penting:

- **Class Size:** Ukuran kelas dalam jumlah mahasiswa.
- **Score:** Skor rata-rata yang diberikan oleh mahasiswa kepada asisten pengajar.
- **Evaluation Category:** Kategori evaluasi yang menjadi target klasifikasi, yaitu *Good*, *Average*, atau *Poor*.

1. Data Import dan Persiapan

Dataset dimuat menggunakan operator Read CSV di RapidMiner. Data ini diperiksa untuk memastikan tidak ada nilai kosong atau data anomali yang dapat mempengaruhi hasil analisis. Setelah itu, dilakukan normalisasi data menggunakan pendekatan Min-Max untuk menyamakan skala atribut sehingga jarak antar data lebih representatif. Rumus normalisasi yang digunakan adalah:

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

Sebagai contoh:

Jika *Class Size* memiliki rentang 25 hingga 45, nilai 35 akan dinormalisasi sebagai:

$$x' = \frac{35 - 25}{45 - 25} = 0.5$$

2. Pembagian Data

Dataset dibagi menjadi dua bagian:

- Data Latih (80% dari total data) untuk melatih model.
- Data Uji (20% dari total data) untuk menguji akurasi model.

3. Penerapan KNN

- Parameter *k*: Dalam studi kasus ini, nilai *k*=3 dipilih untuk menentukan tiga tetangga terdekat dari data uji.

- Metrik Jarak: Metrik yang digunakan adalah *Euclidean Distance*, yang dihitung dengan rumus:

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

Tabel 2.8 Sample Dataset

ID	Class Size	Score	Evaluation Category
1	35	85	Good
2	40	70	Average
3	30	65	Poor
4	45	90	Good
5	25	60	Poor

Data Uji:

Class Size = 32, Score = 75

- Normalisasi

Tabel 2.9 Tabel yang sudah dinormalisasi

ID	Class Size (Norm)	Score (Norm)	Evaluation Category
1	0,5	0,833	Good
2	0,75	0,5	Average
3	0,25	0,167	Poor
4	1	1	Good
5	0	0	Poor

Data Uji (Normalisasi):

$$\text{Class Size: } \frac{32 - 25}{45 - 25} = 0.35, \quad \text{Score: } \frac{75 - 60}{90 - 60} = 0.5$$

- Hitung Jarak Euclidean

Menggunakan rumus Euclidean Distance, hitung jarak antara data uji dan semua data latih. Contoh untuk ID 1:

$$d_1 = \sqrt{(0.35 - 0.5)^2 + (0.5 - 0.833)^2} = \sqrt{0.0225 + 0.111} \approx 0.371$$

Tabel 2.10 Pehitungan Jarak

ID	Jarak Euclidean (ddd)
1	0,371
2	0,4
3	0,361
4	0,771
5	0,5

- Tentukan k Tetangga Terdekat
Dengan $k = 3$, tetangga terdekat adalah:
 - ID 3 (Poor)
 - ID 1 (Good)
 - ID 2 (Average)
- Klasifikasi
Mayoritas tetangga (2 dari 3) adalah kelas "Good". Oleh karena itu, prediksi kategori evaluasi untuk data uji adalah **Good**.

4. Evaluasi Model

- Kinerja model dievaluasi menggunakan validasi silang (*cross-validation*) pada data latih.
- Operator Performance digunakan untuk menghitung metrik seperti akurasi, presisi, dan recall.

C. Tinjauan Studi

Adapun beberapa penelitian sejenis menggunakan algoritma *K-Nearest Neighbor* dengan masalah berbeda yang pernah di lakukan sebagai berikut:

1. **Metode *K-Nearest Neighbor* (KNN) Dalam Kelayakan Penerima Bantuan Sosial Berbasis Web** (Adawiyah & Desi, 2023). Penelitian ini mengeksplorasi penerapan algoritma *K-Nearest Neighbor* (K-NN) untuk menentukan kelayakan penerima bantuan sosial melalui platform berbasis web. Sistem ini dirancang untuk meningkatkan akurasi dan efisiensi dalam pengambilan keputusan, memanfaatkan K-NN untuk mengelompokkan data penerima berdasarkan kemiripan karakteristik. Platform berbasis web ini memungkinkan pembaruan data secara real-time dan memudahkan akses bagi administrator serta pemohon bantuan. Penelitian ini menekankan pentingnya antarmuka pengguna yang intuitif

dan proses klasifikasi yang cepat dalam mendukung keputusan yang lebih tepat sasaran. Dengan demikian, sistem ini tidak hanya meningkatkan akurasi prediksi kelayakan penerima bantuan, tetapi juga mempercepat proses pengambilan keputusan secara keseluruhan.

2. **Penerapan Seleksi Fitur Untuk Klasifikasi Penerima Bantuan Sosial Pangkalan Sesai Menggunakan Metode *K-Nearest Neighbor*** (Fauzan, Gusti, Jasril, & Pizaini, 2023). Penelitian ini berfokus pada penggunaan algoritma *K-Nearest Neighbor* (K-NN) untuk klasifikasi penerima bantuan sosial di Pangkalan Sesai, dengan penekanan khusus pada seleksi fitur sebagai upaya untuk meningkatkan akurasi model. Dalam klasifikasi, tidak semua fitur dalam dataset memberikan kontribusi yang signifikan, sehingga proses seleksi fitur menjadi krusial. Dengan memilih fitur-fitur yang paling relevan, penelitian ini bertujuan untuk menyaring variabel yang paling informatif dan mengurangi noise, sehingga algoritma K-NN dapat bekerja lebih efisien dan akurat. Hasil penelitian menunjukkan bahwa seleksi fitur yang tepat dapat meningkatkan kinerja model K-NN secara signifikan, baik dari segi akurasi maupun efisiensi komputasi, yang pada akhirnya membantu penyaluran bantuan sosial menjadi lebih tepat sasaran.
3. **Kombinasi K-NN Dan *Gradient Boosted Trees* Untuk Klasifikasi Penerima Program Bantuan Sosial** (Firasari, Khultsum, Winnarto, & Risnandar, 2020). Artikel ini menggabungkan algoritma *K-Nearest Neighbor* (K-NN) dengan *Gradient Boosted Trees* untuk meningkatkan akurasi dalam klasifikasi penerima program bantuan sosial. Kombinasi kedua metode ini memungkinkan model untuk menangkap berbagai variabel yang mempengaruhi kelayakan penerima bantuan dengan lebih baik. *Gradient Boosted Trees* digunakan untuk meningkatkan performa prediktif model dengan mengurangi *error* secara iteratif, sementara K-NN memastikan bahwa klasifikasi dilakukan berdasarkan kemiripan data. Hasil penelitian menunjukkan bahwa model gabungan ini memiliki akurasi yang lebih tinggi dibandingkan dengan penggunaan metode K-NN atau *Gradient Boosted Trees* secara terpisah, membuatnya lebih efektif dalam menentukan penerima bantuan yang benar-benar membutuhkan.
4. **Sistem Pendukung Keputusan Penerimaan Bantuan Langsung Tunai Dana Desa (BLT-DD) Dengan Metode KNN Pada Desa Ketowan** (Iman, Santoso, & Lidimilah, 2024). Artikel ini menjelaskan penerapan algoritma *K-Nearest Neighbor* (K-NN) dalam sistem pendukung keputusan untuk menentukan penerima Bantuan Langsung Tunai Dana Desa (BLT-DD) di Desa Ketowan. Sistem ini dikembangkan untuk membantu aparat desa dalam memilih penerima bantuan yang paling layak, berdasarkan kriteria tertentu yang telah ditetapkan. Algoritma K-NN dipilih karena

kemampuannya dalam mengklasifikasikan data berdasarkan kemiripan karakteristik dengan data lain yang telah diketahui kelasnya. Hasil penelitian menunjukkan bahwa penggunaan K-NN dapat meningkatkan ketepatan dalam penentuan penerima bantuan, sehingga bantuan dapat disalurkan secara lebih tepat sasaran.

5. **Penerapan Algoritma *K-Nearest Neighbor* Untuk Klasifikasi Warga Penerima Bantuan Sosial** (Pahrudin & Harianto, 2022). Penelitian ini mengeksplorasi penerapan algoritma *K-Nearest Neighbor* (K-NN) dalam mengklasifikasikan warga yang layak menerima bantuan sosial. Algoritma K-NN dipilih karena kemampuannya dalam mengelompokkan data berdasarkan kedekatan karakteristiknya dengan data lain yang sudah diklasifikasikan. Dalam penelitian ini, K-NN digunakan untuk memprediksi kelayakan warga menerima bantuan sosial berdasarkan sejumlah variabel seperti pendapatan, jumlah tanggungan, dan kondisi sosial-ekonomi lainnya. Hasil penelitian menunjukkan bahwa K-NN dapat membantu dalam menentukan penerima bantuan dengan lebih akurat, yang pada gilirannya dapat meningkatkan efektivitas penyaluran bantuan sosial.
6. **IMPLEMENTASI ALGORITMA *K-NEAREST NEIGHBOR* DENGAN NORMALISASI *Z-SCORE* DALAM KLASIFIKASI PENERIMA BANTUAN SOSIAL DESA SERUNAI** (Safitri, Kusnandar, & Martha, 2023). Penelitian ini mengkaji penggunaan algoritma KNN yang dikombinasikan dengan normalisasi *Z-score* dalam klasifikasi penerima bantuan sosial di Desa Serunai. Normalisasi *Z-score* digunakan untuk mengatasi masalah skala pada data, yang sering kali memengaruhi kinerja algoritma dalam proses klasifikasi. Dengan menerapkan normalisasi *Z-score*, setiap fitur dalam dataset distandarkan, sehingga mengurangi bias dalam perhitungan jarak antar data yang menjadi dasar bagi algoritma KNN. Penelitian ini menunjukkan bahwa penerapan *Z-score* secara signifikan meningkatkan akurasi prediksi KNN, menjadikan algoritma ini lebih andal dalam menentukan penerima bantuan sosial berdasarkan data yang ada.
7. **Klasifikasi Penentuan Penerima Bantuan Pangan Non Tunai Dengan Menggunakan Metode KNN (*K-Nearest Neighbor*)** (Sari, Suryanto, & Haryoko, 2024). Makalah ini membahas penggunaan metode *K-Nearest Neighbor* (K-NN) untuk menentukan penerima bantuan pangan non tunai. Penelitian ini bertujuan untuk meningkatkan keakuratan dan keadilan dalam penyaluran bantuan dengan menggunakan pendekatan berbasis data. Algoritma K-NN digunakan untuk mengelompokkan data penerima berdasarkan kemiripan karakteristiknya, seperti kondisi ekonomi, jumlah tanggungan, dan kebutuhan pangan. Hasil penelitian menunjukkan bahwa metode K-NN dapat membantu dalam membuat keputusan

yang lebih adil dan tepat sasaran, memastikan bahwa bantuan pangan disalurkan kepada mereka yang benar-benar membutuhkan.

8. **Klasifikasi Data Bantuan Sosial Pada Desa Sindangpano Dengan Menggunakan Algoritma *K-Nearest Neighbor*** (Suci, Nining, & Basysyar, 2022). Artikel ini menerapkan algoritma *K-Nearest Neighbor* (K-NN) untuk klasifikasi data bantuan sosial di Desa Sindangpano. Penelitian ini dilakukan untuk mengevaluasi efektivitas K-NN dalam membantu proses seleksi penerima bantuan sosial, yang didasarkan pada sejumlah kriteria yang telah ditentukan. Dengan menggunakan algoritma K-NN, data penerima dapat dikelompokkan berdasarkan kedekatan karakteristik dengan data yang sudah diketahui kelasnya. Hasil penelitian menunjukkan bahwa K-NN mampu memberikan hasil klasifikasi yang akurat dan dapat diandalkan, membantu pemerintah desa dalam membuat keputusan yang lebih baik terkait penyaluran bantuan.
9. **KLASIFIKASI PENERIMA BANTUAN BPNT MENGGUNAKAN ALGORITMA K-NN** (Syajida, Purnamasari, & Suprpti, 2024). Penelitian ini mengklasifikasikan penerima Bantuan Pangan Non Tunai (BPNT) menggunakan algoritma *K-Nearest Neighbor* (K-NN). Fokus dari penelitian ini adalah untuk menentukan sejauh mana K-NN dapat digunakan sebagai alat untuk mengidentifikasi penerima BPNT yang layak berdasarkan data yang tersedia. Dengan menggunakan K-NN, peneliti dapat mengelompokkan data penerima berdasarkan kedekatan karakteristiknya, yang memungkinkan proses seleksi penerima bantuan menjadi lebih akurat dan adil. Hasil penelitian menunjukkan bahwa K-NN dapat diandalkan dalam menentukan penerima BPNT, memastikan bahwa bantuan diberikan kepada mereka yang benar-benar membutuhkan.
10. **Implementasi Metode *K-Nearest Neighbor* Untuk Menentukan Klasifikasi Status Ekonomi Penerima Bantuan** (Fuansah, Meileni, & Novianti, 2023). Artikel ini mengkaji implementasi algoritma KNN untuk menentukan klasifikasi status ekonomi penerima bantuan. Fokus penelitian adalah untuk mengevaluasi sejauh mana algoritma KNN dapat mengkategorikan status ekonomi penerima dengan akurasi yang memadai. Algoritma KNN yang digunakan dalam penelitian ini didasarkan pada perhitungan jarak antara data baru dengan data lama yang sudah diketahui kelasnya. Penelitian ini menunjukkan bahwa KNN dapat digunakan untuk mengklasifikasikan status ekonomi dengan tingkat akurasi yang cukup tinggi. Hasil dari klasifikasi ini dapat digunakan untuk meningkatkan efisiensi dan ketepatan dalam penyaluran bantuan, sehingga bantuan dapat lebih tepat sasaran.

Tabel 2.11 Jurnal Referensi

NO	Peneliti/ Tahun	Judul	Sumber	Kontribusi
1	Adawiyah , Rabiatul, dan Efani Desi (2023)	Metode <i>K-Nearest Neighbor</i> (KNN) Dalam Kelayakan Penerima Bantuan Sosial Berbasis Web	Jurnal Rekayasa Sistem (JUREKSI) 1 (3): 1014–26 Link : https://kti.potensiutama.org/index.php/JUREKSI/article/view/1035	Memberikan dasar teori dan implementasi KNN dalam konteks bantuan sosial berbasis web, relevan untuk memahami pengembangan sistem berbasis web untuk prediksi penerima bantuan.
2	Fauzan, Muhammad, Siska Kurnia Gusti, Jasril Jasril, dan Pizaini Pizaini (2023)	Penerapan Seleksi Fitur Untuk Klasifikasi Penerima Bantuan Sosial Pangkalan Sesai Menggunakan Metode <i>K-Nearest Neighbor</i>	Jurnal Sistem Komputer Dan Informatika (JSON) 5 (1): 1 Link: https://ejurnal.stmik-budidarma.ac.id/index.php/JSON/article/view/6654	Menyediakan informasi tentang seleksi fitur yang dapat meningkatkan akurasi KNN.
3	Firasari, Elly, Umi Khultsum , Monikka Nur Winnarto, dan Risnanda r	Kombinasi K-NN Dan Gradient Boosted Trees Untuk Klasifikasi Penerima Program Bantuan Sosial	Jurnal Teknologi Informasi Dan Ilmu Komputer 7 (6): 1231 Link: http://jtiik.ub.ac.id/index.php/jtiik/article/view/3087	Menawarkan kombinasi metode KNN dan Gradient Boosted Trees untuk peningkatan kinerja klasifikasi, bisa menjadi referensi untuk eksperimen dengan metode kombinasi dalam skripsi.

NO	Peneliti/ Tahun	Judul	Sumber	Kontribusi
	Risnanda r (2020)			
4	Iman, Mohamm ad Fathor, Firman Santoso, dan Lukman Fakih Lidimilah (2024)	Sistem Pendukung Keputusan Penerimaan Bantuan Langsung Tunai Dana Desa (BLT- DD) Dengan Metode KNN Pada Desa Ketowan	G-Tech: Jurnal Teknologi Terapan 8 (2): 1251–60 Link : https://ejournal.uniramalang.ac.id/index.php/g-tech/article/view/4240	Contoh aplikasi KNN dalam sistem pendukung keputusan, bisa diadopsi untuk penerapan algoritma dalam skripsi Anda.
5	Pahrudin, Pajar, dan Kusno Harianto (2022)	Penerapan Algoritma <i>K- Nearest Neighbor</i> Untuk Klasifikasi Warga Penerima Bantuan Sosial	Building of Informatics, Technology and Science (BITS) 4 (3) Link: https://ejournal.semaminar-id.com/index.php/bits/article/view/2276	Menggambarkan penerapan praktis KNN dalam klasifikasi penerima bantuan sosial.
6	Safitri, Novi, Dadan Kusnandar, dan Shantika Martha (2023)	IMPLEMENTASI ALGORITMA <i>K-NEAREST NEIGHBOR</i> DENGAN NORMALISASI Z-SCORE	Bimaster: Buletin Ilmiah Matematika, Statistika Dan Terapannya 13 (1) Link:	Menyediakan metode normalisasi data dengan Z-Score untuk meningkatkan performa KNN, penting untuk pre- processing data dalam skripsi.

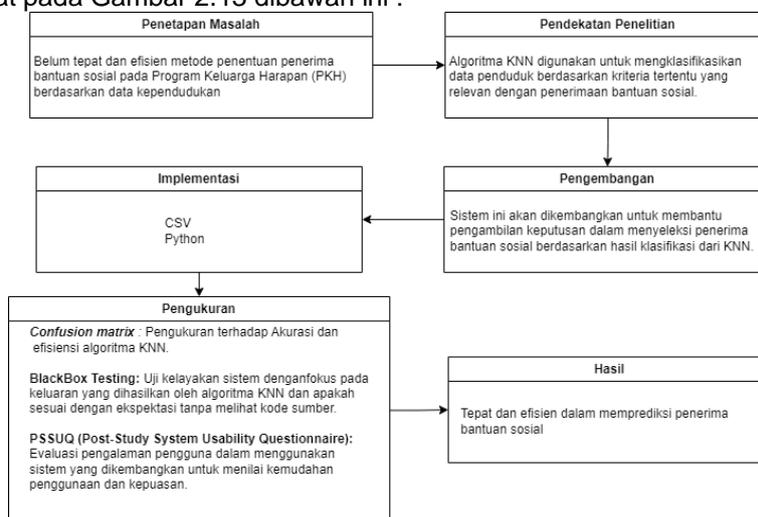
NO	Peneliti/ Tahun	Judul	Sumber	Kontribusi
		DALAM KLASIFIKASI PENERIMA BANTUAN SOSIAL DESA SERUNAI	https://jurnal.untan.ac.id/index.php/jbmstr/article/view/74063	
7	Sari, Susi Indah, Andik Adi Suryanto, dan Andy Haryoko (2024)	Klasifikasi Penentuan Penenerima Bantuan Pangan Non Tunai Dengan Menggunakan Metode Knn (<i>K-Nearest Neighbor</i>)	Curtina 5 (1): 10–21 Link: http://journal.unirow.ac.id/index.php/curtina/article/view/1284	Menyajikan studi kasus penggunaan KNN dalam klasifikasi bantuan pangan, dapat digunakan sebagai referensi untuk studi kasus yang serupa dalam skripsi.
8	Suci, Wulan, Nining, dan Fadhil M. Basysyar (2022)	Klasifikasi Data Bantuan Sosial Pada Desa Sindangpano Dengan Menggunakan Algoritma <i>K-Nearest Neighbor</i>	Jurnal Accounting Information System (AIMS) 5 (2): 167–74 Link: https://jurnal.masoemiversity.ac.id/index.php/aims/article/view/509	Memberikan contoh aplikasi KNN dalam klasifikasi data bantuan sosial di tingkat desa, relevan untuk konteks data kependudukan dalam skripsi.
9	Syajida, Hanna, Ade Irma Purnamasari, dan Tati	KLASIFIKASI PENERIMA BANTUAN BPNT MENGGUNAKAN	JATI (Jurnal Mahasiswa Teknik Informatika) 8 (1): 608–14 Link :	Studi tentang penerapan KNN untuk klasifikasi penerima bantuan BPNT, relevan sebagai referensi metode klasifikasi dalam skripsi.

NO	Peneliti/ Tahun	Judul	Sumber	Kontribusi
	Suprapti (2024)	ALGORITMA K-NN	https://ejournal.itn.ac.id/index.php/jati/article/view/8357	
10	Yohana Sicke Fuansah, Hetty Meileni, Leni Novianti (2023)	Implementasi Metode <i>K-Nearest Neighbor</i> Untuk Menentukan Klasifikasi Status Ekonomi Penerima Bantuan	JUPITER: Jurnal Penelitian Ilmu Dan Teknologi Komputer Link: https://jurnal.polsri.ac.id/index.php/jupiter/article/view/7374	Menyediakan informasi tentang implementasi KNN untuk klasifikasi status ekonomi, relevan untuk memahami aspek ekonomi dalam prediksi penerima bantuan.

Berdasarkan penelitian-penelitian sebelumnya, maka dapat dijabarkan orisinalitas penelitian ini terdapat pada letak penerapan dimana pada penelitian ini menerapkan algoritma *K-Nearest Neighbor* dalam memprediksi penerima PKH berdasarkan data kependudukan.

D. Kerangka Berfikir

Berikut merupakan kerangka pemikiran untuk memecahkan masalah penelitian ini dapat dilihat pada Gambar 2.13 dibawah ini :



Gambar 2.3 Kerangka Pemikiran

Kerangka pemikiran dapat dijelaskan sebagai berikut:

1. Penetapan masalah mencakup fenomena terkait prediksi penerima PKH berdasarkan data kependudukan belum tepat dan belum efektif dalam memprediksi.
2. Dari masalah tersebut, dapat diselesaikan dengan menggunakan pendekatan penelitian dengan penerapan *K-Nearest Neighbors*.
3. Pengembangan yang dilakukan yaitu menggunakan Sistem Pendukung Keputusan dan *Prototype* Aplikasi.
4. Pengukuran yang ditentukan yaitu dengan menggunakan *Confusion Matrix*, *PSSUQ*, dan *BlackBox Training*.
5. Hasil yang didapatkan yaitu berupa aplikasi yang menampilkan hasil prediksi penerima PKH berdasarkan data kependudukan diutamakan secara tepat dan efektif.

E. Hipotesis Penelitian

Algoritma *K-Nearest Neighbors* (KNN) adalah algoritma yang digunakan dalam klasifikasi data dalam data mining untuk mengidentifikasi kategori atau kelompok berdasarkan data terdekat dalam dataset. Algoritma KNN bekerja dengan cara membandingkan data baru dengan data yang sudah ada dan menentukan kategori berdasarkan kemiripan. Algoritma KNN telah dikenal dan banyak digunakan dalam klasifikasi penerima bantuan sosial di penelitian-penelitian sebelumnya yang terkait dengan permasalahan serupa. Penelitian yang telah dilakukan oleh Iman, Santoso, & Lidimilah (2024) dalam penelitiannya yang berjudul "*Sistem Pendukung Keputusan Penerimaan Bantuan Langsung Tunai Dana Desa (BLT-DD) Dengan Metode KNN Pada Desa Ketowan*" menunjukkan bahwa metode *K-Nearest Neighbors* (KNN) dapat secara efektif membantu dalam menentukan kelayakan penerima bantuan. Penelitian ini menghasilkan tingkat akurasi sebesar **95%**, dengan presisi **92%** dan recall **96%**, yang menunjukkan bahwa KNN dapat diandalkan untuk klasifikasi kelayakan penerima bantuan berdasarkan data sosial-ekonomi. Maka dapat diduga bahwa hipotesis pada penelitian ini adalah Algoritma *K-Nearest Neighbors* diduga dapat memprediksi penerima bantuan Program Keluarga Harapan berdasarkan data kependudukan.