PROPOSAL TUGAS AKHIR

KOMPARASI AKURASI KLASIFIKASI DATA TEKS MENGGUNAKAN JARAK MANHATTAN DAN JARAK JACCARD PADA ALGORITMA KNN

KOMPETENSI STATISTIKA

LUH SUKMA MULYANI

2108541027

PROGRAM STUDI MATEMATIKA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS UDAYANA

BUKIT JIMBARAN

2025

LEMBAR JUDUL

KOMPARASI AKURASI KLASIFIKASI DATA TEKS MENGGUNAKAN JARAK MANHATTAN DAN JARAK JACCARD PADA ALGORITMA KNN

KOMPETENSI STATISTIKA

LUH SUKMA MULYANI

2108541027

PROGRAM STUDI MATEMATIKA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS UDAYANA

BUKIT JIMBARAN

2025

LEMBAR PENGESAHAN

Judul	: Komparasi Akurasi Klasifikasi Data Teks Menggunakan Jarak Manhattan dan Jarak Jaccard pada Algoritma KNN
kompetensi	: Statistika
Nama	: Luh Sukma Mulyani
NIM	: 2108541027
Tanggal Seminar	: 28 Februari 2025

Disetujui oleh:

Pembimbing II	Pembimbing I
Ni Ketut Tari Tastrawati, S.Si., M.Si	Ir. I Komang Gde Sukarsa, M.Si
NIP 197405282002122002	NIP 196501051991031004

Mengetahui:
Komisi Tugas Akhir
Program Studi Matematika FMIPA Unud
Ketua,

I Wayan Sumarjaya, S.Si., M.Stats
NIP 197704212005011001

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa atas segala rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan proposal tugas akhir yang berjudul "Komparasi Akurasi Klasifikasi Data Teks Menggunakan Jarak Manhattan dan Jarak Jaccard pada Algoritma KNN" tepat pada waktunya. Pada kesempatan ini, penulis mengungkapkan rasa terima kasih kepada berbagai pihak yang telah memberikan dukungan dan bantuan, baik secara langsung maupun tidak langsung, sehingga proposal tugas akhir ini dapat tersusun dengan baik, antara lain:

1. Ibu I Gusti Ayu Made Srinadi, S.Si., M.Si., selaku Koordinator Program Studi Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Udayana.
2. Bapak I Wayan Sumarjaya, S.Si., M.Stats., selaku Ketua Komisi Tugas Akhir Program Studi Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Udayana.
3. Dosen Pembimbing I, Ir. I Komang Gde Sukarsa, M.Si., yang telah memberikan bimbingan dan arahan dalam penyusunan proposal tugas akhir ini.
4. Dosen Pembimbing II, Ni Ketut Tari Tastrawati, S.Si., M.Si., yang telah dengan sabar memberikan bimbingan, dukungan, dan arahan dalam proses penyusunan proposal tugas akhir ini.
5. Para dosen yang berada di lingkungan Program Studi Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Udayana.

6. Keluarga tercinta yang selalu memberikan motivasi, dukungan moral, dan dana dalam proses penyusunan proposal tugas akhir ini.
7. Teman-teman yang telah memberikan dukungan moral yang sangat berarti selama penyelesaian proposal tugas akhir ini.

Penulis menyadari bahwa apa yang telah disampaikan dalam proposal tugas akhir ini masih jauh dari kesempurnaan. Oleh karena itu, penulis sangat mengharapkan kritik dan saran yang membangun guna memperbaiki dan meningkatkan diri menjadi lebih baik lagi.

Bukit Jimbaran, 7 Maret 2025

Penulis

DAFTAR ISI

Halaman

LEMBAR JUDUL i

LEMBAR PENGESAHAN ii

KATA PENGANTAR iii

DAFTAR ISI v

DAFTAR TABEL vii

DAFTAR GAMBAR viii

DAFTAR LAMPIRAN ix

BAB I PENDAHULUAN 1

1.1 Latar Belakang 1

1.2 Rumusan Masalah 4

1.3 Batasan Masalah 5

1.4 Tujuan Penelitian 5

1.5 Manfaat Penelitian 5

BAB II TINJAUAN PUSTAKA 7

2.1 Penelitian Sebelumnya 7

2.2 Landasan Teori 8

2.2.1 Natural Language Processing 8

2.2.2 *Text Mining* 9

2.2.3 Klasifikasi 11

2.2.4 Klasifikasi Teks 12

2.2.5 *Term Frequency Inverse Document Frequency* (TF-IDF) 13

2.2.6 *K Nearest Neighbors* 14

2.2.7 Jarak Pada KNN 18

2.2.8 *Confusion Matrix* 20

2.2.9 Uji Hipoteis 21

BAB III METODE PENELITIAN 24

3.1 Jenis dan Sumber Data 24

3.2 Variabel Penelitian 24

3.3 Pelaksanaan Penelitian 25

DAFTAR PUSTAKA..... 32

DAFTAR TABEL

Tabel Halaman
2. 1 Confusion Matrix 21

3. 1 Variabel Penelitian 25

DAFTAR GAMBAR

Gambar Halaman

2. 1 Ilustrasi Pemilihan Nilai k pada KNN 17

2. 2 Ilustrasi Jarak Manhattan 18

2. 3 Ilustrasi Jaccard Similarity 20

3. 1 Diagram Alur Penelitian 25

3. 2 Alur Data Preprocessing 27

DAFTAR LAMPIRAN

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pada proses analisi data, pemilihan motode klasifikasi yang tepat sangat bergantung pada bagaimana cara algoritma menghitung kedekatan antar data. Dengan berkembangnya ilmu pengetahuan, berbagai pendekatan telah dikembangkan untuk menyelesaikan masalah klasifikasi, salah satunya dengan menggunakan konsep perhitungan jarak (Uddin et al., 2022). Jarak berperan penting dalam mengukur kedekatan antar titik data di ruang multidimensi dan menjadi komponen utama dalam banyak algoritma pembelajaran mesin. Salah satu algoritma yang memanfaatkan konsep perhitungan jarak sebagai dasar proses klasifikasi adalah *K Nearest Neighbors* (KNN).

Algoritma *K Nearest Neighbors* (KNN) merupakan salah satu metode pembelajaran mesin terawasi (*supervised learning*) yang memanfaatkan pendekatan jarak untuk menentukan hasil klasifikasi atau prediksi. Konsep dasar KNN adalah mengasumsikan bahwa data yang berdekatan dalam ruang fitur memiliki karakteristik yang serupa, sehingga algoritma ini menentukan kelas atau nilai baru berdasarkan mayoritas label dari sejumlah tetangga terdekat yang telah diketahui sebelumnnya. Algoritma ini bersifat non-parametrik, yang berarti tidak memiliki asumsi-asumsi kaku dalam distribusi data (Prasath et al., 2019). Selain itu, KNN memiliki prinsip pendekatan "*lazy learning*", di mana algoritma ini menunda proses pembelajaran hingga saat klasifikasi atau prediksi dilakukan (Jo, 2019). Prinsip ini memberikan fleksibilitas yang tinggi serta memungkinkan model

untuk beradaptasi dengan mudah terhadap data baru tanpa perlu pelatihan ulang yang intensif.

Keberhasilan algoritma KNN sangat bergantung pada dua faktor utama: pemilihan jumlah tetangga terdekat (k) dan pemilihan metrik jarak yang digunakan untuk mengukur kedekatan antar data. Penentuan jumlah tetangga terdekat (k) pada algoritma KNN merupakan langkah yang sangat penting, karena nilai k yang terlalu kecil dapat menyebabkan algoritma menjadi sensitif terhadap *outlier*, sedangkan jika nilai k terlalu besar dapat mengaburkan kejelasan dalam klasifikasi kelas (Halder et al., 2024). Untuk menentukan nilai k yang optimal, dapat digunakan metode *cross-validation* atau *grid search* guna memastikan keseimbangan antara sensitivitas terhadap pola data dan kemampuan model dalam melakukan generalisasi (Zhongguo et al., 2017). Selain nilai k, penentuan jarak juga penting dalam mengukur kedekatan antara data (Cha, 2007). Jarak seperti Euclidean, Manhattan, Cosine, Minkowski, Mahalanobis, Jaccard dapat dipilih berdasarkan karakteristik data dan tujuan analisis, karena penggunaan jarak yang tidak sesuai dapat menurunkan performa model.

Pada KNN, metrik jarak berperan penting dalam menentukan tetangga terdekat yang akan memengaruhi hasil klasifikasi. Secara matematis, jarak sering direpresentasikan melalui metrik yang memenuhi aksioma dasar non-negativitas, identitas, simetri, dan pertidaksamaan segitiga untuk mengukur kedekatan antara pasangan data dalam ruang multidimensi. Masing-masing jarak memiliki karakteristik unik yang membuatnya cocok untuk berbagai jenis aplikasi seperti klasifikasi, klasterisasi, atau pengenalan pola. Pemilihan jarak yang tepat

memastikan hasil klasifikasi lebih akurat, sedangkan jarak yang tidak sesuai dapat mengurangi efektivitas model. Dengan memahami pentingnya nilai k dan jarak, KNN dapat diterapkan secara efektif untuk berbagai jenis data dan tujuan analisis.

Penelitian yang berkaitan dengan metode KNN dilakukan oleh Wahyono et al. (2019) yang membahas tentang perbandingan beberapa metode perhitungan jarak pada KNN dalam klasifikasi data tekstual. Penelitian tersebut menunjukkan bahwa jarak Euclidean dan Minkowski memberikan akurasi tertinggi (85,5%) pada $k = 3$. Jarak Manhattan mencapai akurasi yang sedikit lebih rendah, yaitu 85,05%, sementara Chebyshev menunjukkan kinerja yang jauh lebih rendah pada 61,87%. Berdasarkan penelitian yang dilakukan oleh Prasath et al. (2019), jarak yang digunakan pada penelitian Wahyono et al. (2020) termasuk ke dalam *family* L_p Minkowski, dimana *family* ini menekankan pengukuran perbedaan geometris antar data.

Salah satu tantangan dalam klasifikasi data teks adalah bagaimana cara merepresentasikan teks ke dalam format numerik yang dapat diproses oleh algoritma *machine learning*. Salah satu metode yang umum digunakan adalah *Term Frequency-Inverse Document Frequency* (TF-IDF). Menurut Jo (2019), TD-IDF adalah metode yang digunakan untuk mengukur seberapa penting sebuah kata dalam sebuah dokumen relatif terhadap seluruh korpus, dengan memberikan bobot lebih tinggi pada kata-kata yang sering muncul di satu dokumen namun jarang ditemukan dalam dokumen lainnya. Metode ini menghasilkan vektor berdimensi tinggi yang cenderung bersifat *sparse*, yaitu memiliki bannyak elemen bernilai nol.

Karakteristik *sparse* pada representasi TF-IDF melatar belakangi perlunya eksplorasi terhadap jarak alternatif yang lebih sesuai. Salah satunya adalah keluarga *inner product distance measure*, yang diwakili oleh jarak Jaccard. Jarak dalam keluarga ini dihitung berdasarkan hasil perkalian pasangan nilai dari kedua vektor, sehingga lebih sensitif terhadap kesamaan pola antar vektor yang jarang memiliki nilai nol-nol bersamaan (Prasath et al., 2019). Penelitian ini bertujuan untuk mengeksplorasi apakah jarak Jaccard dapat memberikan kinerja yang lebih baik dibandingkan dengan jarak Manhattan dalam konteks klasifikasi data teks.

Untuk memastikan validasi hasil penelitian ini, uji statistik akan digunakan untuk menentukan apakah perbedaan akurasi yang diamati antara kedua jarak tersebut signifikan secara statistik atau hanya karena kebetulan. Evaluasi dilakukan dengan metode pembagian data latih dan data uji secara acak sebanyak 20 kali, serta dengan variasi nilai k pada algoritma KNN dari $k = 1$ hingga $k = 19$ untuk bilangan ganjil. Dimana hasil akurasi akan dievaluasi menggunakan uji t beda berpasangan untuk menentukan signifikansi perbedaan antara kedua jarak. Dengan pendekatan tersebut, diharapkan penelitian ini memberikan kontribusi dalam pemilihan jarak yang optimal untuk klasifikasi data teks, serta memperkaya pemahaman tentang pengaruh metrik jarak dalam algoritma KNN.

1.2 Rumusan Masalah

Berdasarkan uraian latar belakang masalah, maka rumusan masalah pada penelitian ini adalah:

1. Bagaimana perbandingan akurasi algoritma KNN menggunakan jarak Manhattan dengan Jaccard dalam klasifikasi data teks?

2. Apakah perbedaan akurasi klasifikasi antara kedua jarak signifikan secara statistik?

1.3 Batasan Masalah

Untuk memperjelas permasalahan serta mempertimbangkan keterbatasan yang ada pada peneliti, maka batasan masalah dari penelitian ini adalah sebagai berikut:

1. Data teks direpresentasikan sebagai vektor numerik menggunakan metode TF-IDF.
2. Penelitian ini difokuskan pada perbandingan performa dua jarak, yaitu Manhattan dan Jaccard;
3. Nilai k dalam algoritma KNN akan diuji dengan nilai ganjil, mulai dari k =1 hingga k =19.

1.4 Tujuan Penelitian

Berdasarkan latar belakang dan rumusan masalah yang telah disampaikan, adapun tujuan dari penelitian ini adalah:

1. Menganalisis perbandingan akurasi klasifikasi data teks antara jarak Manhattan dengan jarak Jaccard dalam algoritma KNN;
2. Untuk menguji signifikansi statistik dari perbedaan akurasi klasifikasi antara kedua jarak.

1.5 Manfaat Penelitian

Adapun manfaat penelitian ini adalah sebagai berikut:

1. Bagi Penulis

   Untuk mengimplementasikan ilmu yang sudah diperoleh selama perkuliahan dan melatih kemampuan menganalisa dan mengolah data berbasis teks.

2. Bagi Pembaca

   Penelitian ini dapat digunakan sebagai sumber pengetahuan dan sebagai referensi yang dapat dibandingkan untuk bidang studi yang serupa, khususnya di bidang klasifikasi data teks dan algoritma KNN.

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Sebelumnya

Penelitian yang dilakukan oleh Wahyono et al. (2020) membahas tentang perbandingan beberapa metode perhitungan jarak pada *K Nearest Neighbor* (KNN) dalam klasifikasi data tekstual. Jarak yang digunakan pada penelitian tersebut adalah jarak Euclidean, Chebyshev, Manhattan, dan Minkowski. Data set yang digunakan berjumlah 448 amatan dengan CONTENT sebagai variabel independent dan CLASS (SPAM or HAM) sebagai variabel dependen, serta representasi kalimat dengan *Bag of Word*. Pengujian dilakukan dengan membagi data menjadi dua yaitu data latih sebesar 90% dan data pengujian sebesar 10%, serta nilai k ganjil dari 1 hingga 19. Hasil penelitian tersebut menunjukkan bahwa jarak Euclidean dan Minkowski memiliki akurasi terbaik, masing-masing 85,5% pada k =3, sementara Manhattan menghasilkan akurasi sebesar 85,05%, dan Chebyshev memiliki tingkat akurasi paling kecil, yaitu sebesar 61,87%.

Penelitian yang dilakukan oleh (Prasath et al., 2019) membahas tentang pengaruh pemilihan metode pengukuran jarak terhadap performa algoritma *K Nearest Neighbors* (KNN) pada klasifikasi data dengan menguji 54 jarak dari delapan keluarga besar. Uji dilakukan pada 28 dataset nyata dari UCI *Machine Learning Repository*, dengan setiap dataset dibagi menjadi 66% data pelatihan dan 34% data pengujian, serta nilai $k = 1$ untuk evaluasi akurasi, presisi, dan recall. Hasil penelitian menunjukkan bahwa performa KNN sangat dipengaruhi oleh pemilihan jarak, dengan Hassanat *Distance* mencatatkan performa terbaik. Jarak

Manhattan dan Minkowski dalam keluarga L_p Minkowski menunjukkan performa yang stabil, sedangkan jarak Cosine dan Jaccard dalam keluarga *Inner Product* sangat efektif pada dataset teks. Penelitian ini juga menemukan bahwa beberapa jarak lebih toleran terhadap noise hingga Tingkat 90%, termasuk Hassanat dan Manhattan, sementara keluarga *Squared* L_2 dan *Shannon Entropy* lebih sensitif terhadap jenis data tertentu.

Penelitian yang dilakukan oleh (Jain et al., 2020) membahas tentang perbandingan beberapa metode perhitungan jarak pada algoritma KNN dalam klasifikasi data tekstual. Jarak yang digunakan pada penelitian tersebut adalah Euclidean, Manhattan, Chebyshev, Cosine, Dice, Jaccard, dan Inner Product. Dataset yang digunakan berasal dari ulasan pelanggan Amazon dengan jumlah data sebanyak 2000 amatan dan menerapkan *preprocessing* dengan RapidMiner untuk membersihkan data. Hasil eksperimen menunjukkan bahwa jarak Manhattan memberikan akurasi tertinggi sebesar 97,03%, diikuti oleh Cosine, Dice, dan Jaccard dengan akurasi 96,43%. Sementara itu, Euclidean dam Chebyshev menunjukkan performa yang lebih rendah dibandingkan dengan Manhattan. Studi ini menyimpulkan bahwa pemilihan jarak sangat mempengaruhi efektivitas klasifikasi teks menggunakan KNN.

2.2 Landasan Teori

2.2.1 Natural Language Processing

Natrual Language Processing merupakan bidang penelitian dalam ilmu komputer dan kecerdasan buatan yang berfokus pada hubungan antara komputer dan bahasa manusia (Cole et al., 2019). Pemrosesan ini umumnya melibatkan

penerjemahan bahasa alami yang dapat digunakan komputer untuk memahami, menafsirkan, serta menghasilkan bahasa alami secara bermakna. Karena bahasa manusia bersifat ambigu dan tidak terstruktur seperti bahasa pemrograman, umumnya NLP dirancang dalam bentuk pipeline yang terdiri dari beberapa tahapan, seperti *tokenization*, *stemming, lemmatization, stopword remover*, *part-of-speech tagging*, *parsing*, hingga *sematic analysis*. Melalui tahapan tersebut, bahasa alami diterjemahkan ke dalam bentuk data numerik agar dapat diproses lebih lanjut oleh algoritma pembelajaran mesin. Dengan NLP, komputer dapat melakukan tugas seperti menerjemahkan teks, menjawab pertanyaan, hingga berkomunikasi secara interaktif.

2.2.2 *Text Mining*

Text mining merupakan salah satu cabang dari *data mining* yang berfokus pada proses ekstraksi informasi atau pengetahuan tersembunyi dari kumpulan data berbentuk teks yang tidak terstruktur. Menurut (Jo, 2019), *text mining* didefinisikan sebagai proses untuk mengekstrak pengetahuan yang tidak secara eksplisit tersedia dalam data teks. Pengetahuan yang dihasilkan text mining bersifat baru dan dapat digunakan secara langsung dalam pengambilan keputusan, menjadikan text mining sebagai jenis khusus dari data mining. Teks dalam konteks ini merujuk pada kumpulan kalimat dan paragraf yang disusun dalam bahasa alami, bukan kode atau ekspresi matematis. Tugas utama dari *text mining* mencakup klasifikasi, klasterisasi dan Asosisasi.

Text mining membutuhkan proses awal berupa *text indexing*, yaitu mengubah teks tidak terstruktur menjadi representasi yang dapat diolah oleh komputer. Karena

teks dalam bentuk mentah tidak dapat langsung di proses secara numerik, maka diperlukan tahap *indexing* untuk mengubah teks menjadi daftar kata yang terstruktur (Jo, 2019). Terdapat tiga langkah dasar dalam proses *text indexing* menurut Jo (2019):

1. Tokenisasi

   Tokenisasi merupakan tahapan awal dalam *text indexing* yang bertujuan untuk memecahkan teks menjadi unit-unit kata yang disebut sebagai token. Menurut Jo (2019), tokenisasi dilakukan dengan memisahkan teks berdasarkan spasi atau tanda baca, kemudian dilanjutkan dengan pembersihan token dari karakter yang tidak relevan dan konversi menjadi huruf kecil. Dalam implementasi, tahapan berikut dimulai dengan:

   i. *Lower Case*, yaitu mengubah seluruh huruf menjadi huruf kecil untuk menyamakan bentuk kata yang berbedahanya karena kapitalisasi.
   ii. Penghapusan karakter khusus dan angka, seperti symbol (!, %, *, #, $) dan angka karena dianggap tidak memiliki nilai semantic penting.
   iii. Segmentasi teks, yaitu pemisahan teks menjadi kata-kata menggunakan spasi atau tanda baca.

   Hasil dari tokenisasi berupa daftar kata yang telah dibersihkan dan dinormalisasi.

2. *Stemming*

   Stemming bertujuan untuk menggembalikan setiap kata ke bentuk dasarnya. Misalnya, bentuk *plural*, bentuk kata kerja lampau atau bentuk kata kerja

progresif akan dikembalikan ke bentuk dasar seperti "*run*" dari "*running*", "*ran*", dan "*runs*".

3. *Stopword Removal*

Stopword merupakan kata-kata yang hanya memiliki fingsi grammatical dan tidak membawa malna signifikan, seperti "*and*", "*of*", atau "*the*". Penghapusan stopword membantu meningkatkan efisiensi dan mengurangi redundansi dalam representasi teks.

2.2.3 Klasifikasi

Menurut Han et al. (2012) klasifikasi merupakan salah satu analisis data yang bertujuan untuk membangun model (*classifer*) yang bisa mendeskripsikan label kelas berdasarkan data baru dengan pola yang ditemukan dalam data sebelumnya. Label sebuah kelas dapat bersifat karegorikal, yang mana label bernilai diskret tanpa memiliki urutan tertentu. Klasifikasi termasuk ke dalam metode pembelajaran terawasi (*supervised learning*), yang mana model akan dibangun berdasarkan data latih yang sudah memiliki label kelas. Proses pembelajaran terawasi terdiri dari dua tahapan yaitu, *learning phase* dan *classification phase.* Pada tahap *learning phase* (tahap pembelajaran) algoritma akan mempelajari pola dari data latih yang terdiri dari atribut-atribut dan label kelas dengan hasil pada proses ini adalah sebuah model yang merepresentasikan fungsi antara atribut-atribut tersebut dengan label kelas. Selanjutnya, pada tahap klasifikasi, model yang telah dibentuk akan digunakan untuk memprediksi label kelas pada data baru yang terdapat pada data uji.

2.2.4 Klasifikasi Teks

Klasifikasi teks merupakan suatu pengelompokan dokumen teks ke dalam label tertentu berdasarkan isi yang terkandung di dalamnya. Dalam bidang pengambilan informasi dan teks mining, klasifikasi teks merupakan fondasi penting yang sering digunakan untuk menangani data yang berukuran besar yang berasal dari berbagai sumber (Muliono & Tanzil, 2018). Proses ini sering digunakan dalam berbagai aplikasi, seperti fitur spam pada email, analisis sentimen, pengelompokan artikel berita, dan sistem rekomendasi. Dalam penerapannya, klasifikasi teks membutuhkan data latih yang sudah diberi label untuk menghasilkan model yang mampu mengelompokkan dokumen baru secara otomatis. Dengan demikian, klasifikasi teks mampu mengolah dan menganalisis data dalam jumlah besar dengan lebih mudah.

Sebelum data dapat diklasifikasi, terdapat beberapa tahapan yang harus dilakukan guna mempermudah perhitungan, yaitu tahapan *pre-processing* teks dan representasi teks. Tahapan *pre-processing* bertujuan untuk membersihkan elemen-elemen teks dari elemen-elemen yang tidak relevan, adapun beberapa tahapannya seperti mengubah teks menjadi huruf kecil (*lower case*), membersihkan teks dari karakter-karakter yang tidak relevan, seperti tanda baca, angka, spasi berlebihan, atau karakter khusus (*teks cleaning*), tokenisasi, penghapusan *stopword* dan stemming. Setelah dilakukan tahapan *pre-processing*, tahapan selanjutnya adalah representasi teks. Representasi teks bertujuan untuk menggambarkan relevansi kata dalam dokumen terhadap kumpulan dokumen lainya. Salah satu teknik representasi

teks yang sering digunakan, yaitu TF-IDF (*Term Frequency-Inverse Document Frequency*).

2.2.5 *Term Frequency Inverse Document Frequency* (TF-IDF)

Term Frequency-Inverse Document Frequency (TF-IDF) merupakan sebuah metode numerik yang digunakan untuk menentukan bobot atau tingkat kepentingan setiap kata dalam sebuah dokumen. Metode ini sering diterapkan di dalam *Natural Language Processing* (NLP), pengambilan informasi (*information retrieval*), dan *text mining*. TF-IDF mengevaluasi pentingnya sebuah kata dari kumpulan dokumen dengan mempertimbangkan dua aspek utama, yaitu frekuensi kemunculan kata (*Term Frequency*) dan seberapa jarang kata tersebut muncul di seluruh dokumen dalam koleksi (*Inverse Document Frequency*). Berikut adalah tahapan dalam melakukan pembobotan kata menggunakan TF-IDF (Farhan AlShammari, 2023):

Term frequency digunakan untuk mengukur seberapa sering sebuah kata ke-i muncul dalam sebuah dokumen j dibandingkan dengan jumlah total kata dalam dokumen tersebut. Nilai yang diperoleh dari TF menggambarkan tingkat kepentingan suatu kata dalam dokumen tersebut. Cara menghitung TF adalah sebagai berikut:

$$TF_{ij} = \frac{f_{ij}}{X_i} \tag{2.1}$$

Dengan, f_{ij} merupakan jumlah kemunculan kata (*term*) ke-i dalam dokumen j, dan X_i merupakan total jumlah kata dalam dokumen j.

Inverse Document Frequency (IDF) merupakan nilai logaritma dari jumlah total dokumen dalam kumpulan data (*corpus*) dibagi dengan jumlah dokumen yang

mengandung kata tertentu. IDF digunakan untuk menghitung seberapa penting suatu kata dalam *corpus* secara keseluruhan. Rumus unruk menghitung IDF sebagai berikut:

$$IDF_i = \log\left(\frac{X}{df_i}\right) \tag{2.2}$$

Dengan, X merupakan jumlah total dokumen keseluruhan dan df_i merupakan jumlah dokumen yang mengandung kata (*term)* i. Semakin jarang sebuah kata muncul dalam dokumen, maka nilai IDF-nya akan semakin tinggi, ini menandakan bahwa kata tersebut lebih unik dalam kumpulan dokumen. Sebaliknya, jika kata sering muncul dalam dokumen maka akan memiliki nilai IDF yang lebih rendah karena dianggap kurang informatif.

Term Frequency-Inverse Document Frequency (TF-IDF) adalah hasil perkalian antara TF dan IDF. TF-IDF digunakan untuk mengukur tingkat kepentingan suatu kata dalam sebuah dokumen relatif terhadap keseluruhan dokumen. Rumus perhitungan TF-IDF adalah sebagai berikut:

$$n_{ij} = TF_{ij} \times IDF_i \tag{2.3}$$

Dengan, n_{ij} merupakan bobot dari kata (term) i terhadap dokumen ke- j, TF_{ij} merupakan frekuensi kemunculan kata (term) i pada dokumen ke- j, dan IDF_i merupakan nilai bobot IDF dari term i.

2.2.6 *K Nearest Neighbors*

K Nearest Neighbors diperkenalkan pertama kali pada tahun 1951 oleh Fix dan Hodges, kemudian KNN mengalami modifikasi secara signifikan pada tahun 1967 oleh Cover dan Hart (Cover & Hart, 1967), dimana merupakan salah satu

algoritma tradisional dalam pembelajaran mesin yang didasakan pada teori kedekatan dalam ruang vektor $\mathbb{R}^n$. Setiap data $x_i \in \mathbb{R}^n$ direpresentasikan sebagai vektor fitur berdimensi n, dengan ruang sampel data latih dinyatakan sebagai $x = \{x_1, x_2, x_3, \dots, x_i\}$, dimana i merupakan jumlah data latih. Data latih akan didefinisikan dengan label $y = \{y_1, y_2, y_3, \dots, y_i\}$, dimana $y_i \in C$, dimana C merupakan himpunan kelas diskret pada kasus klasifikasi.

Karakteristik non-parametrik yang dimiliki oleh algoritma *K Nearest Neighbors* (KNN) memberikan fleksibilitas yang signifikan dalam menangani dataset yang kompleks dan beragam, tanpa memerlukan asumsi distribusi tertentu. Hal ini memungkinkan KNN untuk beradaptasi dengan berbagai jenis data serta pola yang sekiranya sulit ditangani oleh algoritma parametrik (Weinberger & Saul, 2009). Selain itu, algoritma ini menggunakan prinsip "*lazy learning*" atau pembelajaran malas, di mana proses pembelajaran ditunda hingga tahap klasifikasi atau prediksi diperlukan. Pendekatan ini tidak hanya memberikan fleksibilitas tinggi, tetapi juga memungkinkan model beradaptasi dengan mudah terhadap data baru tanpa memerlukan proses pelatihan ulang yang intensif.

Secara matematis, jarak sering direpresentasikan melalui metrik jarak, yang mendefinisikan kedekatan antara dua data dalam ruang multidimensi. Metrik jarak $d : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R} \geq 0$, harus memenuhi aksioma dasar untuk memastikan perhitungan kedekatan antar data dapat dilakukan secara matematis dengan benar. Aksioma-aksioma tersebut, yang dijelaskan oleh Deza & Deza (2009), adalah sebagai berikut:

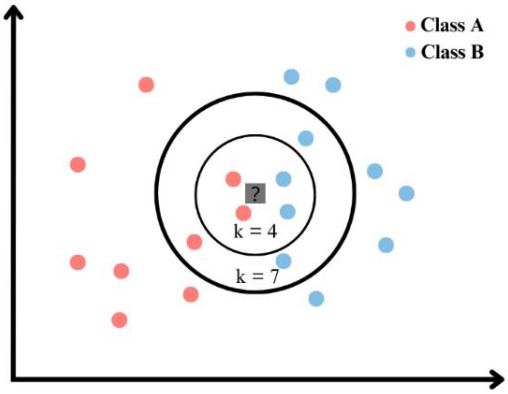
1. *Non-negativity*: $d(x, y) \geq 0$

2. *Indentity of Indiscernibles*: $d(x, y) = 0 \iff x = y$
3. *Symmetry*: $d(x, y) = d(y, x)$
4. *Triangle Inequality*: $d(x, y) \leq d(x, z) + d(z, y)$

Berdasarkan aksioma-aksioma tersebut, berbagai jarak dapat digunakan dalam KNN, tergantung pada sifat data yang dianalisi. Pemilihan jarak yang sesuai bergantung pada karakteristik data dan tujuan dalam analisis, karena penggunaan jarak yang tidak tepat dapat menurunkan akurasi model KNN.

Jarak antara data dalam KNN dapat direpresentasikan ke dalam bentuk matriks jarak $D \in \mathbb{R}^{n \times n}$, dimana elemen D_{ij} merupakan jarak antar data latih x_i dan x_j. Matriks jarak ini digunakan untuk mencari tetangga terdekat untuk setiap data uji x_{test}. Tetangga terdekat ditentukan dengan mentukan nilai k, yang merupakan parameter terpenting dalam menentukan performa klasifikasi. Salah satu pertimbangan utamanya adalah memilih k ganjil, terutama dalam kasus dua kelas (Richard O. Duda et al., 2001). Hal ini bertujuan untuk menghidari hasil seri (*ties*) dalam proses voting, situasi dimana jumlah tetangga dari dua kelas sama besar.

Dapat dilihat pada Gambar 2.1 yang menunjukkan bagaimana pemilihan jumlah k dapat mempengaruhi hasil klasifikasi, terutama jika k genap dapat menyebabkan hasil seri. Dengan memilih k ganjil, algoritma dapat secara langsung menentukan label mayoritas tanpa memerlukan aturan tambahan untuk memberikan label pada data baru.



Gambar 2. 1 Ilustrasi Pemilihan Nilai k pada KNN

Dapat dilihat bahwa ketika $k = 4$, terdapat dua titik merah dan dua titik biru dalam lingkaran, yang menyebabkan kebingungan dalam penentuan kelas. Namun, dengan memilih $k = 7$, mayoritas label dapat ditentukan dengan lebih jelas, sehingga algoritma dapat menghasilkan prediksi yang lebih stabil.

Selain itu, pemilihan nilai k yang tepat sangat penting, karena algoritma ini sensitif terhadap *outlier* ketika nilai k terlalu kecil. Sebaliknya, jika nilai k terlalu besar akan mengakibatkan pemilihan kelas menjadi tidak jelas atau "kabur" (Halder et al., 2024). Oleh karena itu, penentuan nilai k yang optimal sangat penting untuk menjaga keseimbangan antara sensitivitas terhadap pola lokal dan kemampuan generalisasi model. Nilai k yang optimal dapat di tentukan dengan menggunakan metode *cross-validation* dan *grid search* (Zhongguo et al., 2017). Penentuan nilai k yang tepat akan membantu dalam meningkatkan akurasi prediksi dan memastikan bahwa model KNN dapat bekerja secara efektif pada berbagai tipe data.

Selanjutnya kelas data uji y_{test} ditentukan berdasarkan mayoritas label dari k tetangga terdekat, yaitu dengan menggunakan aturan voting mayoritas:

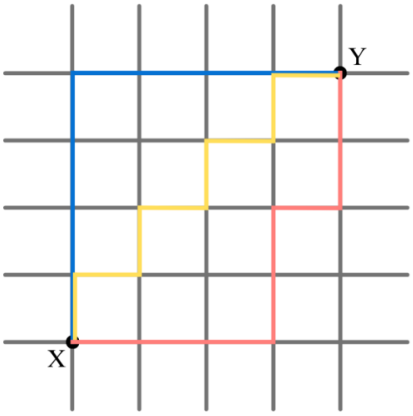
$$y_{test} = mode(\{y_1, y_2, y_3, \dots, y_k\})$$

Dimana $\{y_1, y_2, y_3, \dots, y_k\}$ merupakan label dari k tetangga terdekat data uji x_{test}.

2.2.7 Jarak Pada KNN

Pada penelitian kali ini, peneliti akan menggunakan dua jenis jarak dalam algoritma *K Nearest Neighbors* (KNN) untuk mengklasifikasi data teks, yaitu jarak Manhattan dan Jaccard dalam mengklasifikasi data teks. Jarak ini digunakan untuk mengukur kedekatan antara data yang ada, dengan tujuan untuk menentukan kelas dari data yang belum diketahui berdasarkan tetangga terdekatnya.

Jarak Manhattan, yang dikenal juga sebagai *City-block distance*, *Taxicab distance*, atau L1-norm, merupakan salah satu jarak dalam ruang $\mathbb{R}^n$ yang digunakan untuk mengukur jarak antara dua titik berdasarkan perbedaan nilai setiap koordinatnya (Deza & Deza, 2009). Jarak Manhattan diasumsikan bahwa perpindahan antara dua titik hanya dapat dilakukan sepanjang sumbu koordinat dalam struktur berbasis *grid* (Aha, 1997), sebagaimana ditunjukkan pada Gambar 2.2.



Gambar 2. 2 Ilustrasi Jarak Manhattan

Jarak Manhattan antara dua vektor $x, y \in \mathbb{R}^n$, didefinisikan sebagai berikut:

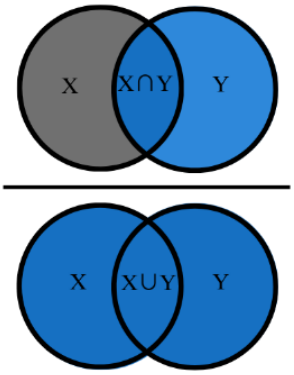
$$d(x,y) = \sum_{i=1}^{n} |x_i - y_i| \tag{2.4}$$

dengan $x_i, y_i \in \mathbb{R}$, di mana x_i menyatakan nilai fitur ke-i dari data uji dan y_i menyatakan nilai fitur ke-i dari data latih.

Jarak Jaccard digunakan untuk mengukur tingkat ketidaksamaan (*dissimilarity*) antara dua himpunan. Dalam konteks klasifikasi teks, jarak ini sering digunakan untuk membandingkan dua dokumen dengan menghitung kesamaan kata-kata yang terdapat di dalamnya. Secara matematis, *Jaccard Similarity* antara dua himpunan X dan Y didefinisikan sebagai Deza & Deza (2009):

$$Jaccard(X,Y) = \frac{|X \cap Y|}{|X \cup Y|} \tag{2.5}$$

dengan X dan Y merupakan himpunan kata dari dua dokumen, $|X \cap Y|$ menyatakan jumlah kata yang muncul di kedua dokumen, sedangkan $|X \cup Y|$ menyatakan jumlah total kata unik dalam kedua dokumen. Jarak ini menghasilkan nilai dalam rentang 0 hingga 1, di mana nilai 1 menunjukkan kesamaan sempurna antara dua dokumen sedangkan nilai 0 menunjukkan bahwa kedua dokumen tidak memiliki kesamaan kata sama sekali. Konsep *Jaccard Similarity* dapat dilihat pada Gambar 2.3, di mana area yang diarsir menunjukkan bagian irisan, sedangkan seluruh area dalam lingkaran merepresentasikan gabungan.



Gambar 2. 3 Ilustrasi *Jaccard Similarity*

Gambar 2.3 menunjukkan bahwa semakin besar irisan antara dua himpunan dibandingkan dengan gabungannya, semakin tinggi pula nilai *Jaccard Similarity*. Selanjutnya, jarak Jaccard (*Jaccard distance*) merupakan komplemen dari *Jaccard similarity*, yang didefinisikan sebagai Deza & Deza (2009):

$$d_{Jaccard(X,Y)} = 1 - Jaccard(X,Y) \tag{2.6}$$

Pada penelitian ini, jarak Jaccard digunakan untuk menghitung kedekatan antar dokumen teks berdasarkan kesamaan kata yang terkandung dalam dokumen tersebut. Nilai jarak Jaccard berada dalam rentang [0,1], dimana nilai 0 menunjukkan bahwa kedua himpunan sangat identik, sedangkan nilai 1 menunjukkan bahwa kedua himpunan sangat berbeda.

2.2.8 *Confusion Matrix*

Untuk menilai sejauh mana akurasi dari model klasifikasi yang telah dibagun sebelumnya, akan dilakukan perhitungan ketepatan klasifikasi. Perhitungan ini biasanya disajikan dalam bentuk matriks berukuran $n \times n$ yang disebut dengan *confusion matrix. Confusion matrix* merupakan metode evaluasi yang digunakan untuk mengukur kinerja atau tingkat akurasi dari proses klasifikasi.

Pada kasus klasifikasi dengan dua kelas, *confusion matrix* yang terbentuk akan memiliki ukuran 2×2 seperti berikut (Sokolova & Lapalme, 2009):

Tabel 2. 1 *Confusion Matrix*

Kelas Aktual (y)	Kelas Prediksi $(f(x))$	
	-1	+1
-1	*True Negative* (TN)	*False Positive* (FP)
+1	*False Negative* (FN)	*True Positive* (TP)

Kinerja model klasifikasi yang baik dapat dinilai berdasarkan tingkat akurasinya. Nilai akurasi ini dapat dihitung menggunakan persamaan berikut:

$$Akurasi = \frac{TP + TN}{(TP + TN + FP + FN)} \times 100\% \quad (2.7)$$

2.2.9 Uji Hipoteis

Untuk mengetahui seberapa signifikan perbedaan akurasi model dalam mengklasifikasi teks, peneliti akan menggunakan uji statistik untuk memvalidasi apakah perbedaan tersebut benar-benar signifikan atau hanya terjadi secara kebetulan. Salah satu metode yang digunakan adalah uji t beda berpasangan. Uji ini bertujuan untuk mengevaluasi apakah perbedaan rata-rata akurasi yang dihasilkan oleh kedua jarak tersebut menunjukkan hasil yang signifikan secara statistik. Uji t beda berpasangan dilakukan dengan menggunakan data akurasi dari pasangan pengamatan, di mana setiap pasangan terdiri atas hasil klasifikasi data teks. Langkah-langkah utama dalam uji t beda berpasangan menurut (Snedecor & Cochran, 1967)meliputi:

1. Menyusun Hipotesis Penelitian

a. H_0 (Hipotesis nol): Tidak ada perbedaan yang signifikan antara rata-rata perbedaan akurasi yang dihasilkan oleh kedua jarak.

$$H_0: \mu_d = 0$$

Di mana μ_d adalah rata-rata selisih akurasi antara kedua jarak.

b. H_1 (Hipotesis alternatif): Ada perbedaan yang signifikan antara rata-rata perbedaan akurasi yang dihasilkan oleh kedua jarak.

$$H_1: \mu_d \neq 0$$

2. Menghitung Statistik Uji

Statistik uji dalam uji t beda berpasangan dirumuskan berdasarkan selisih antara pasangan pengamatan serta ukuran sampel. Statistik uji dirumuskan sebagai:

$$t = \frac{\bar{d} - 0}{\frac{s_d}{\sqrt{n}}} \tag{2.8}$$

Di mana, $\bar{d}$ merupakan rata-rata perbedaan akurasi antara pasangan pengamatan, n merupakan jumlah pasangan pengamatan, s_d merupakan standar deviasi dari perbedaan pasangan yang dihitung dengan menggunakan:

$$s_d = \sqrt{\frac{\sum_{i=1}^{n}\left(d_i - \bar{d}\right)^2}{n-1}} \tag{2.9}$$

Dengan d_i menunjukkan selisih antara pasangan ke- i ($d_i = x_1 - x_2$), $\bar{d}$ menunjukkan rata-rata dari semua perbedaan $\left(\bar{d} = \frac{\sum d_i}{n}\right)$, dan n menunjukkan jumlah pasangan pengamatan.

3. Membandingkan Nilai t_{hitung} dan t_{kritis}

Setelah nilai t_{hitung} diperoleh, selanjutnya akan dibandingkan dengan nilai t_{kritis} dari tabel distribusi t dengan derajat kebebasan $(df = n - 1)$. Jika nilai $t_{hitung} > t_{kritis}$ atau $p - value < \alpha$, maka hipotesis nol ditolak, yang berarti terdapat perbedaan signifikan antara kedua jarak. Sebaliknya jik nilai $t_{hitung} \leq t_{kritikal}$ atau p$-value \geq \alpha$, maka hipotesis nol gagal ditolak, yang berarti tidak ada cukup bukti untuk menyatakan bahwa kedua metode memiliki perbedaan yang signifikan.

BAB III

METODE PENELITIAN

3.1 Jenis dan Sumber Data

Penelitian ini menggunakan data sekunder berupa data komentar pengguna pada video youtube milik Eminem yang diperoleh dari kaggle. Data yang diperoleh berjumlah 448 dari periode 6 Mei 2015 hingga 29 Mei 2015 dan disimpan dengan format *file* .csv. Dataset ini terdiri dari 5 atribut, yaitu COMMENT_ID, AUTHOR, DATE, CONTENT, dan CLASS. Namun, pada penelitian ini peneliti akan menggunakan 2 atribut saja, yaitu CONTENT dan CLASS. Data diklasifikasikan ke dalam dua kelas, yaitu kelas SPAM sebanyak 245 komentar dan kelas HAM sebanyak 203 komentar. Data ini akan dibagi menjadi data train untuk melatih model dan data test untuk menguji model.

3.2 Variabel Penelitian

Pada penelitian ini, terdapat dua jenis variabel yang digunakan, yaitu variabel prediktor (X) dan variabel target (Y). Variabel prediktor (X) merupakan variabel yang digunakan sebagai input untuk memprediksi variabel target. Pada penelitian ini, variabel prediktornya adalah CONTENT, yang berisi teks komentar yang ditinggalkan oleh pengguna pada video youtube. Variabel CONTENT berperan untuk memberikan informasi mengenai isi komentar yang akan dianalisis untuk menentukan apakah komentar tersebut termasuk ke dalam kategori SPAM atau HAM.

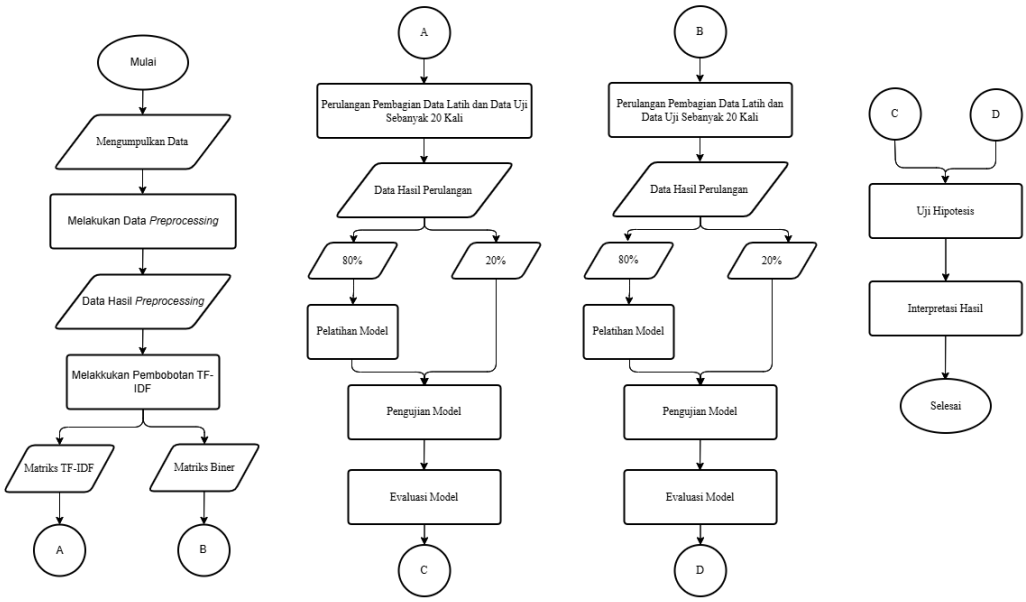
Sementara, CLASS adalah variabel target (Y), yang menunjukkan kategori dari komentar, yakni SPAM atau HAM. Variabel ini berfungsi sebagai label yang

akan diprediksi oleh model berdasarkan teks yang ada pada variabel CONTENT. Kategori SPAM mencakup komentar-komentar yang tidak relevan atau tidak diinginkan, seperti iklan atau promosi, sementara kategori HAM mencakup komentar-komentar yang relevan dan sesuai dengan konteks video. Dalam penelitian ini, model akan dilatih untuk memprediksi kelas CLASS berdasarkan teks komentar yang ada pada variabel CONTENT.

Tabel 3. 1 Variabel Penelitian

No	Nama Variabel	Deskripsi	Skala Pengukuran	kategori
1	Y = *CLASS*	Kategori komentar	Nominal	0: HAM 1: SPAM
2	X_p= *CONTENT*	Jumlah kata dari komentar youtube Eminem	Rasio	

3.3 Pelaksanaan Penelitian



Gambar 3. 1 Diagram Alur Penelitian

Dalam pelaksanaan penelitian, peneliti menggunakan *software Visual Studio Code* dengan bahasa pemrograman Python dalam seluruh tahapan analisis data dalam mengimplementasikan algoritma. Platform ini dipilih karena kemudahan pada aksesnya, kemampuan komputasi yang mendukung proses pengolahan data dalam skala besar, serta kompatibilitasnya dengan berbagai pustaka pendukung analisis data. Penelitian ini memanfaatkan pustaka seperti *scikit-learn* untuk mengimplementasikan algoritma KNN, `pandas` dan `NumPy` untuk pengolahan data, serta `NLTk` untuk *preprocessing* teks. Seluruh proses analisis dilakukan dengan sistematis dimulai dari tahapan preprocessing data teks hingga evalusai performa algoritma berdasarkan jarak yang ditentukan.

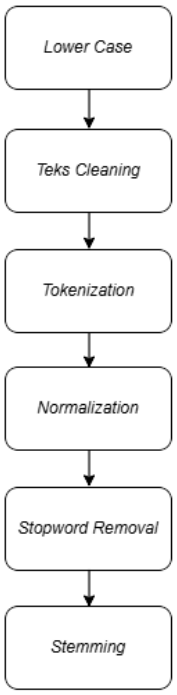
Berikut tahapan-tahapan penelitian yang akan dilakukan, meliputi:

1. Pengumpulan Data

   Peneliti menggunakan data berupa komentar teks yang diperoleh dari website kaggle. Data yang dikumpulkan terdiri dari 448 komentar youtube yang relevan dengan topik yang diteliti. Pemilihan data ini didasarkan pada keberagaman isi komentar yang dapat memberikan variasi yang cukup untuk melakukan analisis klasifikasi teks. Data komentar youtube dipilih karena banyaknya interaksi yang terjadi dalam bentuk teks, yang mencakup berbagai perspektif.

2. Data *Preprocessing*

Tahapan ini bertujuan untuk membersihan data teks dari elemen-elemen yang tidak relevan sebelum dilakukannya analisis lebih lanjut. Proses *preprocessing* meliputi:



Gambar 3. 2 Alur Data *Preprocessing*

a. *Lower case,* berfungsi untuk mengubah seluruh huruf pada teks komentar menjadi huruf kecil;
b. *Teks Cleaning,* berfungsi untuk menghapus karakter yang tidak relevan, seperti tanda baca, angka, spasi berlebihan, atau karakter khusus;
c. *Tokenization,* berfungsi untuk memecahkan teks menjadi unit-unit kata yang disebut sebagai token;
d. *Normalization,* berfungsi untuk mengubah kata-kata tidak baku, menjadi kata baku;

e. *Stopword Removal,* berfungsi untuk menghapus kata-kata umum sperti "*the*", "*is*", "*and*" yang tidak relevan dalam analisis teks;

f. *Stemming,* berfungsi untuk mengubah kata ke bentuk dasar.

3. Pembobotan TF-IDF

Setelah melalui tahap *preprocessing* data, langkah selanjutnya adalah membentuk matriks fitur menggunakan metode *Term Frequency-Inverse Document Frequency* (TF-IDF). Proses ini diawali dengan menghitung *Term Frequency* (TF) menggunakan Persamaan (2.1), selanjutnya dilakukan perhitungan *Inverse Document Frequency* (IDF) berdasarkan Persamaan (2.2). Setelah kedua nilai tersebut diperoleh, bobot TF-IDF dihitung dengan mengalikan TF dan IDF sebagaimana dinyatakan dalam Persamaan (2.3). Hasil akhir dari proses ini berupa sebuah matriks fitur yang merepresentasikan bobot setiap kata dalam dokumen berdasarkan pentingnya dalam keseluruhan korpus. Matriks TF-IDF yang dihasilkan akan digunakan dalam perhitungan Jarak Manhattan. Selain itu, untuk perhitungan Jarak Jaccard, matriks TF-IDF diubah menjadi matriks biner, di mana kata dengan bobot TF-IDF > 0 diberi nilai 1, dan kata dengan bobot 0 diberi nilai 0.

4. Pembagian Data Latih dan Data Uji (20 Kali)

Data akan dibagi menjadi dua bagian, yaitu data latih untuk melatih model dan data uji untuk mengevaluasi kinerjanya. Pembagian data akan dilakukan secara acak menjadi 80% data latih dan 20% data uji. Pembagian

ini dilakukan sebanyak 20 kali untuk melihat variabilitas dan stabilitas model.

5. Penerapan Metode KNN

Setelah menyelesaikan seluruh tahapan sebelumnya, tahapan selanjutnya adalah melaksanakan proses klasifikasi dengan menggunakan algoritma *K Nearest Neighbors* (KNN) berbasis dua jarak: Manhattan dengan menggunakan matriks TF-IDF dan Jaccard dengan menggunakan matriks biner. Proses klasifikasi mencangkup beberapa langkah penting, di antaranya:

a. Model akan dilatih menggunakan data latih dengan menggunakan dua jarak, yaitu Manhattan dan Jaccard.
b. Pada setiap iterasi, model dilatih menggunakan berbagai nilai k (jumlah tetangga) yang mencakup k = 1, 3, 5, 7, 9, 11, 13, 15, 17, 19.
c. Model yang telah dilatih akan diuji dengan menggunakan data uji. Hasil pengujian dievaluasi dengan menggunakan *confusion matrix* untuk mengukur kinerja model, serta menghitung akurasi berdasarkan data uji.

6. Evaluasi Model Dengan Menggunakan Uji Hipotesis

Untuk memastikan bahwa perbedaan antara jarak Manhattan dan Jaccard signifikan secara statistik, dilakukan Uji t beda berpasangan. Proses

ini dilakukan setelah melakukan perulangan sebanyak 20 kali untuk mendapatkan data akurasi yang lebih stabil.

a. Pertama hipotesis awal akan dibentuk sebagai berikut:
    i. Hipotesis nol (H_0) : Tidak terdapat perbedaan yang signifikan dalam selisih rata-rata akurasi antara jarak Manhattan dan Jaccard.
    ii. Hipotesis alternatif H_1: Terdapat perbedaan yang signifikan dalam rata-rata akurasi antara jarak Manhattan dan Jaccard.
b. Setelah hipotesis terbentuk, Uji t beda berpasangan akan dilakukan untuk membandingkan rata-rata akurasi dari kedua jarak tersebut.

7. Interpretasi Hasil

Berdasarkan hasil evaluasi model menggunakan *K Nearest Neighbors* (KNN) dengan dua jarak, yaitu Manhattan dan Jaccard, diperoleh perbandingan akurasi pada data uji. Model yang menggunakan jarak Manhattan menunjukkan akurasi rata-rata sebesar X%, sedangkan model dengan jarak Jaccard menghasilkan akurasi rata-rata sebesar Y%. Dari hasil tersebut, terlihat bahwa salah satu jarak memberikan performa yang lebih baik dalam mengklasifikasi data teks. Jika X > Y, maka jarak Manhattan lebih efektif dalam klasifikasi data teks, namun jika Y > X, jarak Jaccard lebih unggul dalam akurasi.

Untuk memastikan apakah perbedaan akurasi antara kedua jarak tersebut signifikan secara statistik, dilakukan Uji t beda berpasangan. Hasil p-value dari uji tersebut menunjukkan nilai Z. Jika $p - value < \alpha$, maka

tolak hipotesis nol dan menyimpulkan bahwa perbedaan akurasi antara kedua jarak adalah signifikan secara statistik. Sebaliknya, jika $p - value \geq \alpha$, maka tidak ada perbedaan signifikan antara kedua jarak. Dengan demikian, hasil evaluasi dan uji statistik ini memberikan dasar untuk memilih jarak yang lebih optimal.

DAFTAR PUSTAKA

Aha, D. W. (1997). Lazy Learning. In *Lazy Learning*. Springer Netherlands. https://doi.org/10.1007/978-94-017-2053-3

Cha, S.-H. (2007). 2007_Comprehensive Survey on Distance or Similarity. *International Journal Of Mathematical Models and Methods In Applied Sciences*, *1*(4), 300–307. https://api.semanticscholar.org/CorpusID:15506682

Cole, H. L., Hannes, H., & Hapke, M. (2019). *Natural Language Processing IN ACTION*.

Cover, T. M., & Hart, P. E. (1967). Nearest Neighbor Pattern Classification. In *IEEE TRANSACTIONS ON INFORMATION THEORY* (Vol. 24, Issue 1). https://doi.org/doi.org/10.1109/TIT.1967.1053964

Deza, M. M., & Deza, E. (2009). Encyclopedia of Distances. In *Encyclopedia of Distances*. Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-00234-2

Farhan AlShammari, A. (2023). Implementation of Keyword Extraction using Term Frequency-Inverse Document Frequency (TF-IDF) in Python. In *International Journal of Computer Applications* (Vol. 185, Issue 35). https://doi.org/doi.org/10.5120/ijca2023923137

Halder, R. K., Uddin, M. N., Uddin, M. A., Aryal, S., & Khraisat, A. (2024). Enhancing K-nearest neighbor algorithm: a comprehensive review and performance analysis of modifications. *Journal of Big Data*, *11*(1). https://doi.org/10.1186/s40537-024-00973-y

Han, J., Kamber, M., & Pei, J. (2012). *Data Mining Concepts and Techniques*. https://doi.org/https://doi.org/10.1016/C2009-0-61819-5

Jain, S., Jain, Dr. S. C., & Vishwakarma, Dr. S. K. (2020). A Proposed Similarity Measure for Text-Classification. *International Journal of Innovative Technology and Exploring Engineering*, *9*(6), 2232–2235. https://doi.org/10.35940/ijitee.D1939.049620

Jo, T. (2019). *Text Mining*. https://doi.org/doi.org/10.1007/978-3-319-91815-0

Muliono, Y., & Tanzil, F. (2018). A Comparison of Text Classification Methods k-NN, Naïve Bayes, and Support Vector Machine for News Classification. *Jl. Kh. Syahdan*, *03*(02). https://doi.org/doi.org/10.30591/jpit.v3i2.828

Prasath, V. B. S., Alfeilat, H. A. A., Hassanat, A. B. A., Lasassmeh, O., Tarawneh, A. S., Alhasanat, M. B., & Salman, H. S. E. (2019). *Effect of Distance Measures Choice on KNN Classifier Performance -- A Review*. https://doi.org/10.1089/big.2018.0175

Richard O. Duda, Peter E. Hart, & David G. Stork. (2001). *Pattern classification* (2nd edition). Wiley-Interscience. https://www.researchgate.net/publication/228058014_Pattern_Classification

Snedecor, G. W., & Cochran, W. G. (1967). *Statistical Methods* (6th Edition). Iowa State University Press.

Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing and Management*, *45*(4), 427–437. https://doi.org/10.1016/j.ipm.2009.03.002

Uddin, S., Haque, I., Lu, H., Moni, M. A., & Gide, E. (2022). Comparative performance analysis of K-nearest neighbour (KNN) algorithm and its different variants for disease prediction. *Scientific Reports*, *12*(1). https://doi.org/10.1038/s41598-022-10358-x

Wahyono, W., Trisna, I. N. P., Sariwening, S. L., Fajar, M., & Wijayanto, D. (2019). 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

Weinberger, K. Q., & Saul, L. K. (2009). Distance Metric Learning for Large Margin Nearest Neighbor Classification. In *Journal of Machine Learning Research* (Vol. 10).

Zhongguo, Y., Hongqi, L., Liping, Z., Qiang, L., & Ali, S. (2017). A case based method to predict optimal k value for k-NN algorithm. *Journal of Intelligent and Fuzzy Systems*, *33*(1), 55–65. https://doi.org/10.3233/JIFS-161062