BAB II
TINJAUAN PUSTAKA DAN DASAR TEORI

2.1. Tinjauan Pustaka

Pada penelitian yang dirancang oleh Xin Lei, Sichang Yang dan Longsheng Zhou (Lei, Yan, & Zhou, 2023), menjelaskan mengenai perbandingan *distance* pada metode Local Mean KNN. Jenis *metrics* yang digunakan pada penelitian adalah *Euclidean, Minkowski, Chebyshev dan Cityblock*. Hasil yang didapat dari penelitian tersebut adalah bahwa perhitungan jarak dengan minkowski memiliki akurasi yang lebih tinggi dibandingkan dengan perhitungan lainnya.

Pada tahun 2023, dilakukan penelitian yang dilakukan oleh Neli Kalcheva, Maya Todorova, dan Ivaylo Penev (Kalcheva, Todorova, & Penev, 2023). Penelitian ini membahas mengenai perbandingan *metrics* untuk menghitung jarak antar tetangga pada metode KNN. Dataset yang digunakan berupa data *text*. Hasil penelitian diatas didapatkan suatu kesimpulan bahwa penggunaan *metrics Cosine Similarity* memiliki perolehan akurasi yang cukup jauh diatas perhitungan jarak lainnya hingga selisih 10% dibandingkan dengan perhitungan jarak *Euclidean*, *Minkowski*, dan *Manhattan*.

Penelitian yang dirancang dan dilakukan oleh Shengpeng Guo, Liyuan Guo, Seyed Mohammad Ali Zeinolabedin dan Christian Mayer dilakukan pada tahun 2022 (Guo, Guo, Zeinolabedin, & Mayr, 2022). Penelitian ini membahas mengenai evaluasi berbagai perhitungan jarak (*distance metrics*) yang dilakukan pada klasifikasi lonjakan syaraf pada manusia. Penelitian ini menggunakan 3 perhitungan jarak, yaitu *Manhattan, Euclidean dan Mahalanobis*. Metode yang digunakan pada penelitian adalah metode *K-Means*. Hasil yang didapat pada penelitian yang dilakukan adalah bahwa perhitungan *Mahalanobis* lebih unggul dari performansi keseluruhan dibandingkan dengan *Euclidean* dan *Manhattan*.

Penelitian yang dilakukan Akhiladevi M, Anitha K, Amrutha K, Amrutha M, dan Chandanashree K R menjelaskan mengenai klasifikasi kecelakaan di jalan raya dengan menggunakan algoritma KNN (Akhiladevi, Anitha, Amrutha, & Chandanashree, 2022). Prosedur penelitian dilakukan dengan pengumpulan data,

preprocessing data, dan melakukan perhitungan dengan *Euclidean distance* untuk menentukan jarak tiap tetangganya. Nilai K yang digunakan adalah 5. Pada hasil penelitian didapatkan nilai akurasi dari algoritma KNN sebesar 88,25%. Peneliti juga membandingkan dengan algoritma lain seperti *decision tree*, *random forest* dan *logistic regression*. Namun algoritma KNN ini mamiliki tingkat akurasi yang tertinggi dibandingkan dengan metode lainnya.

Pada tahun 2022, penelitian yang dikerjakan oleh Shivani Wadhwa, Deepak Kumar, Shivani Gupta, dan Vinay Kukreja membahas mengenai pengenalan aksara urdu (Wadhwa, Kumar, Gupta, & Kukreja, 2022). Penelitian ini dilakukan dengan menggunakan *image compression* pada preparasi datanya dan mengggunakan algoritma KNN sebagai metode untuk mengklasifikasikan aksara urdu. Perhitungan jarak tetangganya menggunakan *Euclidean distance*. Hasil dari penelitian ini menunjukkan bahwa penggunaan *image compression* dan KNN sebagai algoritma klasifikasi mendapatkan nilai akurasi sebesar 97,57%.

Penelitian pada tahun 2022 yang dilakukan oleh Tsvetelina Mladenova dan Irena Valova membahas tentang perbandingan kinerja dari metode KNN biasa dengan metode KNN yang sudah dilakukan perhitungan bobot (Mladenova & Valova, 2022). Penelitian ini diterapkan ke 18 dataset yang berbeda. Hasil dari penelitian ini menunjukkan bahwa perhitungan dengan menggunakan KNN biasa menghasilkan nilai akurasi 73%. Sedangkan performa metode KNN yang sudah ditambahkan perhitungan bobot tiap tetangganya memiliki nilai akurasi sebesar 91%.

Penelitian yang dilakukan oleh Yuslena Sari, Endi Gunawan, Mutia Maulida dan Johan Wahyudi pada tahun 2021 membahas tentang penggunaan metode KNN pada *website* BAZNAZ (Sari, Gunawan, Maulida, & Wahyudi, 2021). Dataset yang digunakan merupakan dataset berjenis numerik. Penelitian ini menghasilkan nilai akurasi yang didapat sebesar 94,32%. Pada tahun 2023, penelitian mengenai identifikasi kanker paru-paru yang dilakukan oleh Kotesh Kumar dan Priyanka menggunakan algoritma KNN dan *Logistic Regression* (Kumar & Priyanka, 2023). Dataset yang digunakan marupakan data pada pasien penderita kanker paru-paru.

Hasil pada penelitian ini hasil akurasi yang didapat pada metode KNN adalah 89,56% sedangkan pada *logistic regression* sebesar 80,11%.

Sai Charan, Saravanan dan Surendran melakukan suatu penelitian dengan menggunakan algoritma KNN dan SVM pada tahun 2023 (Charan, M. S., & Surendran, 2023). Penelitian ini diimplementasikan pada aktifitas fungsi kognitif pada manusia berdasarkan rentang umurnya. Penelitian ini menghasilkan nilai akurasi sebesar 93,05% pada metode KNN dan 91,30% pada SVM. Kemudian penelitian pada tahun 2021 yang dilaksanakan oleh Savira Rohwinasakti, Budhi Irawan dan Casi Setianingsih menggunakan metode KNN untuk mengidentfikasi analisis sentimen pada jasa transportasi *online* (Rohwinasakti, Irawan, & Setianingsih, 2021). Hasil dari penelitian ini menunjukkan nilai akurasi sebesar 94%.

Dari semua tinjauan pustaka yang diambil dari penelitian terdahulu, dapat diringkas dan dirangkum menjadi suatu informasi sebagai dasar dilakukan penelitian kali ini, ditunjukkan pada Tabel 2.1.

Tabel 2. 1 Tinjauan Pustaka

No	Penulis	Judul	Metode	Hasil
1	Xin Lei, Sichang Yang dan Longsheng Zhou	*Extensions of local-mean K-nearest neighbor classifier with various distance metrics*	LMKNN (*Euclidean, Minkowski, Chebyshev* dan *Cityblock)*	*Minkowski* paling tinggi
2	Neli Kalcheva, Maya Todorova, dan Ivaylo Penev	*Study of the K-Nearest Neighbors Method with Various Features Text Classification in Machine Learning*	KNN (*Minkowski, Manhattan, Euclidean*, dan *Cosine Similarity*)	*Cosine Similarity* paling tinggi

Tabel 2. 1 Tinjauan Pustaka (lanjutan)

No	Penulis	Judul	Metode	Hasil
3	Shengpeng Guo, Liyuan Guo, Seyed Mohammad Ali Zeinolabedin dan Christian Mayer	*Various Distance Metrics Evaluation on Neural Spike Classification*	*K-Means (Manhattan, Euclidean* and*, Mahalanobis*)	*Mahalanobis* paling tinggi
4	Akhiladevi M, Anitha K, Amrutha K, Amrutha M, dan Chandanashree K. R.	*Accident Prediction Using KNN Algorithm*	KNN, *Decision Tree, Random Forest,* dan *Logistic Regression*	KNN (88,25%), *Decision Tree* (78,01%), *Random Forest* (86,84%), *Logistic Regression* (88,24%)
5	Shivani Wadhwa, Deepak Kumar, Shivani Gupta, dan Vinay Kukreja	*Dissected Urdu Dots Recognition Using Image Compression and KNN Classifier*	KNN	97,57%
6	Tsvetelina Mladenova dan Irena Valova	*Comparative analysis between the traditional KNearest Neighbor and Modifications with Weight-Calculation*	KNN dan WKNN	KNN (73%), WKNN(91%)

Tabel 2. 1 Tinjauan Pustaka (lanjutan)

No	Penulis	Judul	Metode	Hasil
7	Yuslena Sari, Endi Gunawan, Mutia Maulida dan Johan Wahyudi	*Artificial Intelligence Approach For BAZNAS Website Using K-Nearest Neighbor (KNN)*	KNN	94,32%
8	Kotesh Kumar dan Priyanka	*Lung Cancer Identification System to Improve the Accuracy using Novel K-Nearest Neighbour in Comparison with Logistic Regression Algorithm*	KNN dan *Logistic Regression*	KNN (89,56%), *Logistic Regression* (80,11%)
9	Sai Charan, Saravanan dan Surendran	*Prediction of Insufficient Accuracy for Human Activity Recognition with Limited Range of Age using K-Nearest Neighbor.*	KNN dan SVM	KNN (93,05%), SVM (91,30%)

Tabel 2. 1 Tinjauan Pustaka (lanjutan)

No	Penulis	Judul	Metode	Hasil
10	Savira Rohwinasakti, Budhi Irawan dan Casi Setianingsih	*Sentiment Analysis on Online Transportation Service Products Using K-Nearest Neighbor Method*	KNN	94,40%

Penelitian ini didasarkan pada identifikasi dua kesenjangan utama dalam literatur sebelumnya, yaitu kesenjangan metodologi (*methodology gap*) dan kesenjangan data (*data gap*). Pada aspek metodologi, sebagian besar penelitian sebelumnya hanya menggunakan metode klasifikasi konvensional seperti *K-Nearest Neighbor (KNN)* dengan implementasi distance metrics yang terbatas, misalnya *Euclidean* dan *Cosine Similarity*. Penelitian ini memperluas pendekatan tersebut dengan menambahkan perhitungan bobot tetangga menjadi *Weighted K-Nearest Neighbor (WKNN)* yang memanfaatkan *Gaussian Kernel* untuk perhitungan bobot, sekaligus membandingkan kinerja berbagai *distance metrics*, termasuk *Minkowski* dan *Jaccard*, yang jarang dibahas secara mendalam dalam penelitian serupa.

Selain itu, pada aspek data, penelitian sebelumnya cenderung berfokus pada analisis data dengan dua label kelas (*biner*), sehingga kurang mengeksplorasi kompleksitas yang muncul pada data dengan lebih dari dua kelas. Penelitian ini mengatasi keterbatasan tersebut dengan menggunakan dataset yang terdiri dari tiga label kelas, sehingga memungkinkan evaluasi yang lebih komprehensif terhadap performa metode WKNN, khususnya dalam konteks analisis sentimen di media sosial.

2.2. Dasar Teori

2.2.1. Media Sosial

Media sosial adalah *platform* daring yang memungkinkan pengguna untuk berinteraksi, berbagi konten, dan terhubung dengan orang lain secara global. Mereka menyediakan ruang untuk menyampaikan pendapat, mengekspresikan diri, dan memperluas jaringan sosial. Dengan berbagai fitur seperti postingan teks, gambar, dan video, media sosial memfasilitasi komunikasi yang cepat dan mudah. Selain itu, mereka juga menjadi tempat bagi promosi bisnis, kampanye sosial, dan diskusi publik tentang berbagai topik. X, salah satu platform media sosial terkemuka, menonjol dengan format postingannya yang singkat dan cakupan berita yang luas.

2.2.2. Platform X

Twitter atau *X* adalah platform media sosial yang terkenal dengan ciri khasnya yang memungkinkan pengguna untuk membuat postingan dengan panjang terbatas hingga 280 karakter. Dikenal sebagai "*tweet*", pesan-pesan ini sering digunakan untuk berbagi pemikiran, berita terkini, atau mengungkapkan pendapat dalam format yang singkat dan langsung. Selain itu, *Twitter* juga memfasilitasi interaksi antar pengguna melalui fitur seperti *retweet*, *like*, dan balasan, yang memungkinkan konten untuk dengan cepat menyebar dan menjadi viral. Dengan jutaan pengguna aktif di seluruh dunia, platform ini menjadi tempat yang penting untuk mendapatkan informasi terbaru, berpartisipasi dalam percakapan global, dan membangun jejaring sosial yang luas. Selain individu, banyak perusahaan, tokoh publik, dan organisasi menggunakan *Twitter* sebagai alat untuk berkomunikasi dengan audiens mereka secara langsung dan secara *real-time*.

2.2.3. Analisis Sentimen

Analisis sentimen merupakan proses untuk mengidentifikasi dan mengklasifikasikan opini atau emosi yang terkandung dalam suatu teks, seperti perasaan positif, negatif, atau netral. Proses ini bertujuan untuk memahami bagaimana individu mengekspresikan pendapat mereka terhadap suatu topik, produk, atau peristiwa tertentu. Dengan memanfaatkan teknik pemrosesan bahasa alami (*Natural Language Processing/NLP*), analisis sentimen dapat mengolah data

teks dalam jumlah besar dari berbagai sumber, seperti media sosial, ulasan produk, dan artikel berita. Hasil analisis ini dapat digunakan untuk pengambilan keputusan strategis, baik dalam bidang bisnis, politik, maupun sosial, karena memungkinkan penggunanya untuk memahami kecenderungan emosi publik dan respons masyarakat terhadap isu-isu yang relevan.

Dalam penerapan analisis sentimen, kualitas hasil sangat bergantung pada bagaimana data teks diproses sebelum dianalisis. Teks mentah yang diambil dari sumber seperti media sosial sering kali mengandung elemen-elemen yang tidak relevan, seperti tanda baca, simbol khusus, atau kata-kata yang tidak bermakna signifikan dalam konteks analisis. Oleh karena itu, diperlukan tahap *preprocessing* text yang bertujuan untuk membersihkan dan menyusun data agar lebih terstruktur dan dapat diolah dengan baik oleh model analisis. Tahap ini meliputi beberapa langkah, seperti penghapusan *noise*, *stemming*, tokenisasi, dan normalisasi, yang semuanya berperan penting dalam meningkatkan akurasi dan efisiensi model analisis sentimen.

2.2.4. *Text Preprocessing*

Text processing adalah proses manipulasi dan analisis teks menggunakan berbagai teknik komputasional. Berikut adalah langkah-langkah umum dalam *text processing*:

1. Pemrosesan Teks Awal (*Preprocessing*): Langkah pertama dalam text *processing* adalah membersihkan dan mempersiapkan teks untuk analisis lebih lanjut. Ini melibatkan tahap seperti menghilangkan tanda baca yang nantinya hanya akan menjadi *noise* dan penggunaan spasi yang berlebih. Tahapan ini dimanakan juga dengan *punctuation removal*.
2. Tokenisasi: Setelah teks dibersihkan, teks tersebut perlu dipecah menjadi unit-unit yang lebih kecil, disebut token. Tokenisasi mengubah teks menjadi potongan-potongan seperti kata, frasa, atau kalimat agar dapat diolah lebih lanjut.
3. *Stopword Removal*: *Stopword* adalah kata-kata umum yang sering muncul dalam teks tetapi tidak memberikan informasi signifikan, seperti "dan", "atau",

"di", dan sebagainya. Menghapus *stopword* dari teks dapat membantu memfokuskan analisis pada kata-kata yang lebih bermakna.

4. *Stemming* atau *Lemmatization*: Langkah ini bertujuan untuk mengubah kata-kata menjadi bentuk dasarnya. Stemming menghapus akhiran kata secara *heuristik*, sedangkan *lemmatization* menggunakan aturan gramatikal untuk mengubah kata-kata ke bentuk dasarnya (lematis).

2.2.5. TF-IDF

TF-IDF (Term Frequency-Inverse Document Frequency) adalah metode yang digunakan dalam pemrosesan teks untuk mengevaluasi seberapa penting suatu kata dalam sebuah dokumen dalam korpus dokumen. Metode ini mengkombinasikan dua komponen: frekuensi kemunculan kata dalam dokumen (TF) dan *invers* dari frekuensi kemunculan kata di seluruh dokumen dalam korpus (IDF).

1. *Term Frequency (TF): Term frequency* adalah rasio dari jumlah kemunculan suatu kata dalam dokumen terhadap total jumlah kata dalam dokumen tersebut.
2. *Inverse Document Frequency (IDF): Inverse document frequency* adalah logaritma dari rasio total jumlah dokumen dalam korpus terhadap jumlah dokumen yang mengandung term tertentu. Ini membantu dalam memberikan bobot lebih besar untuk kata-kata yang jarang muncul di dalam korpus, karena kata-kata tersebut mungkin lebih informatif.
3. *TF-IDF Score: TF-IDF score* adalah perkalian antara *term frequency (TF)* dan *inverse document frequency (IDF)* untuk setiap term dalam dokumen. Berikut adalah rumus dari TF-IDF:

$$TF(t,d) = \frac{\text{jumlah kemunculan term t dalam d}}{\text{total term dalam d}} \quad (2.1)$$

$$IDF\ (t) = \log \frac{N}{1+df} \quad (2.2)$$

$$TF\text{-}IDF(t,d) = TF(t,d)\ x\ IDF(t) \quad (2.3)$$

Berdasarkan Rumus (2.1) menunjukkan perhitungan *Term Frequency (TF)* dimana t merupakan *term*/kata dan d adalah *document*. Setelah dilakukan

perhitungan TF, dilakukan pencarian IDF seperti pada Rumus (2.2) sehingga nantinya dapat dilakukan penggabungan perhitungan TF-IDF yang ditunjukkan pada Rumus (2.3)

2.2.6. *K-Nearest Neighbor*

K-Nearest Neighbor (KNN) adalah algoritma pembelajaran mesin yang digunakan untuk klasifikasi dan regresi. Algoritma *K-Nearest Neighbors* (KNN) berperan sebagai metode pembelajaran berbasis *instance* yang non-parametrik, yang sering digunakan dalam *supervised learning*, termasuk klasifikasi dan regresi (Halder, Uddin, Uddin, Aryal, & Khraisat, 2024). Dalam klasifikasi, KNN mencari k tetangga terdekat dari titik data yang diberikan, Kemudian menentukan label kelasnya berdasarkan mayoritas kelas dari tetangga-tetangga tersebut. Algoritma KNN membuat klasifikasi berdasarkan umpan balik yang paling sering muncul dari K titik data terdekat dengan titik uji (Duan, 2024). Dalam regresi, algoritma ini menghitung rata-rata atau *median* dari nilai target dari k tetangga terdekat untuk memklasifikasi nilai target untuk titik data yang diberikan (Lei, Zhu, Fang, Li, & Liu, 2020). Algoritma KNN memiliki banyak keunggulan, seperti kemudahan dalam implementasi dan kemampuan generalisasi yang baik (Yunneng, 2020). Berikut adalah rumus perhitungan jarak default Euclidean dan 3 jarak yang akan menjadi komparator yaitu *Minkowski*, *Cosine Similarity* dan *Jaccard*:

1. *Euclidean* : Rumus perhitungan jarak *Euclidean* adalah metode untuk mengukur jarak antara dua titik dalam ruang *Euclidean*. Ini adalah metode yang paling umum digunakan dalam analisis data dan pembelajaran mesin.

$$d(A, B) = \sqrt{(A1-B1)^2+(A2-B2)^2} \quad (2.4)$$

   Rumus (2.4) merupakan perhitungan dari *Euclidean distance*, dimana p dan q merupakan dua titik A = (A1, A2, …, An) dan B = (B1, B2, …, Bn) yang titik-titik ini berada pada dimensi n.

2. *Minkowski* : Disini, Ai dan Bi adalah koordinat dari titik data dalam dimensi ke-i, sedangkan n merupakan jumlah dimensi dan p menunjukkan parameter yang mengontrol jenis jarak.

$$\text{minkowski} = \left(\sum_{i=1}^{n} |\text{Ai-Bi}|^{p}\right)^{\frac{1}{p}} \quad (2.5)$$

Rumus (2.5) menunjukkan perhitungan jarak *minkowski* dimana Ai dan Bi merupakan 2 titik yang terdapat pada dimensi n, Kemudian p merupakan parameter orde yang menentukan jenis jarak. Nilai p ini apabila bernilai 2 maka akan menjadi rumus perhitungan *Euclidean*. Dikarenakan *Euclidean* sudah masuk ke dalam perhitungan pembanding maka diambil nilai p yang digunakan pada penelitian kali ini adalah 3.

3. *Cosine Similarity* : *Cosine similarity* adalah metode untuk mengukur kesamaan antara dua *vector* dalam ruang berdimensi banyak. Metode ini sering digunakan dalam pemrosesan bahasa alami, pengelompokan dokumen, dan pencarian informasi. Rumus untuk menghitung *cosine similarity* antara dua *vector* sebagai berikut.

$$\text{cosine similarity(A,B)} = \frac{\text{A . B}}{\|A\|\|B\|} \quad (2.6)$$

Berdasarkan Rumus (2.6), didalam perhitungan rumus ini terdapat A dan B yang merupakan dua buah *vector* yang terdapat dalam ruang berdimensi n. Kemudian $A \,.\, B$ menunjukkan hasil *dot* dari 2 *vector* dan tanda mutlak yang adalah norma dari *vector*, yang dihitung sebagai akar kuadrat dari jumlah kuadrat masing-masing komponen *vector*.

4. *Jaccard* : Jarak *Jaccard* adalah metode untuk mengukur kesamaan antara dua himpunan. Ini sering digunakan dalam bidang seperti pengelompokan data, pertandingan string, dan analisis teks. Berikut adalah rumus perhitungannya.

$$\text{J(A,B)} = 1 - \frac{|\text{A}\cap\text{B}|}{|\text{A}\cup\text{B}|} \quad (2.7)$$

Pada Rumus (2.7) terdapat A dan B yang merupakan himpunan, dimana perhitungan ini membagi irisan himpunan dengan gabungan dari himpunan A dan B.

2.2.7. *Weighted K-Nearest Neighbor*

Weighted K-Nearest Neighbor (KNN) adalah variasi dari algoritma KNN yang memberikan bobot yang berbeda untuk setiap tetangga terdekat dalam proses pengambilan keputusan. Dalam *Weighted KNN*, bobot diberikan berdasarkan jarak antara titik data yang sedang diklasifikasi dengan tetangga-tetangganya. Semakin jauh jarak antara titik data yang sedang diklasifikasi dengan tetangga, semakin kecil bobotnya, dan sebaliknya. Dalam fungsi *gaussian kernel*, parameter σ mengontrol lingkup efek lokal dari fungsi *kernel* (Siyu, Chongnan, Mingjuan, & Linna, 2021).

Bobot yang digunakan pada penelitian kali adalah menggunakan *Gaussian Kernel* yang dirumuskan pada Rumus (2.8).

$$weight = e^{-\frac{d^2}{2\sigma^2}} \tag{2.8}$$

Dimana :

e = bilangan *euler*

d = jarak antara titik data yang sedang diklasifikasi dengan tetangga

σ = parameter *bandwidth* yang mengontrol distribusi dari bobot.

Sigma (σ) merupakan parameter dalam *Gaussian Kernel* yang menentukan lebar atau penyebaran fungsi kernel. Dalam konteks metode *Weighted K-Nearest Neighbor (WKNN), Gaussian Kernel* digunakan untuk menghitung bobot setiap tetangga berdasarkan jaraknya ke data uji. Semakin besar jaraknya, bobot yang diberikan semakin kecil. Fungsi dari sigma ini sendiri adalah sebagai pengontrol penurunan bobot dan mempengaruhi sensitivitas model. *Sigma* yang terlalu kecil membuat model terlalu sensitif terhadap data terdekat (*overfitting*), sedangkan *sigma* yang terlalu besar membuat model kurang mampu membedakan tetangga berdasarkan jaraknya (*underfitting*).

Sementara itu, *gaussian kernel* adalah fungsi yang digunakan dalam pembelajaran mesin dan model statistik yang menerapkan distribusi *gaussian* (normal) untuk memberikan bobot pada titik data berdasarkan jaraknya dari titik pusat. Titik yang lebih dekat mendapatkan bobot lebih tinggi, sementara titik yang lebih jauh mendapatkan bobot lebih rendah, sehingga menghaluskan pengaruh dari

titik-titik yang jauh (Lai & Xu, 2020). Bobot ini kemudian digunakan dalam proses pengambilan keputusan untuk memberikan kontribusi yang lebih besar dari tetangga-tetangga yang lebih dekat dalam menentukan kelas atau nilai target dari titik data yang sedang diklasifikasi. *Weighted KNN* membantu dalam meningkatkan akurasi klasifikasi, terutama ketika beberapa tetangga yang lebih dekat memiliki pengaruh yang lebih besar dalam menentukan label kelas atau nilai target daripada tetangga yang lebih jauh.

2.2.8. Evaluasi Model Sistem

Evaluasi model sistem adalah proses penting dalam menilai kinerja model prediktif atau klasifikasi. Melalui tahapan inilah kita bisa menilai seberapa bagus efektivitas suatu metode setelah dilakukan pengujian menggunakan dataset tertentu. Evaluasi model sistem ini meliputi *confusion matrix* yang nantinya dari *confusion matriks* kita bisa mengetahui nilai akurasi, presisi dan *recall* dari performa suatu metode.

1. *Confusion Matriks : Confusion matrix* adalah tabel yang menggambarkan kinerja model pada set data uji, yang membandingkan klasifikasi yang benar dan yang salah. Ini terdiri dari empat indikator: *True Positive (TP), True Negative (TN), False Positive (FP),* dan *False Negative (FN)*.
2. Akurasi (*accuration*) : Akurasi mengukur seberapa sering model memberikan klasifikasi yang benar dari semua klasifikasi yang dilakukan. Ini dihitung dengan membagi jumlah klasifikasi yang benar oleh total jumlah klasifikasi. Perhitungan nilai akurasi terdapat pada Rumus (2.9).

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{2.9}$$

3. Presisi (*precision*) : Presisi mengukur seberapa akurat model dalam memklasifikasi kelas positif dari semua klasifikasi positif yang dibuat. Ini dihitung dengan membagi *True Positive* oleh jumlah semua klasifikasi positif. Nilai presisi dapat dilakukan perhitungan berdasarkan Rumus (2.10).

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2.10}$$

4. *Recall* (*sensitivity* atau *true positive rate*) : *Recall* mengukur seberapa baik model dapat mengidentifikasi kelas positif dari semua *instance* yang sebenarnya positif. Berdasarkan Rumus (2.11) nilai *recall* dihitung dengan membagi *True Positive* oleh jumlah semua *instance* positif.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{2.11}$$

Metrik-metrik evaluasi ini membantu dalam memahami kinerja model dalam mendeteksi dan mengklasifikasikan *instance-instance* dari kelas yang berbeda, dan mereka dapat memberikan informasi untuk meningkatkan model.

BAB III
METODE PENELITIAN

3.1. Bahan dan Alat Penelitian

Di dalam suatu penelitian, diperlukan alat dan bahan untuk menunjang proses penelitian agar bisa berjalan dengan lancar dan baik. Berikut ini adalah bahan dan alat yang dibutuhkan dalam melakukan penelitian ini.

3.1.1. Bahan Penelitian

Adapun bahan yang diperlukan di dalam penelitian ini adalah dataset yang berisi ujaran ujaran pada sosial media baik itu yang memiliki emosi atau bersifat netral. Dataset ini nantinya akan terdiri dari 3 label. Label pertama akan menggunakan label bernama positif yang merepresentasikan emosi-emosi yang positif seperti perasaan senang, gembira, terharu, bersemangat dan sebagainya. Kemudian label kedua akan dinamai netral yang artinya didalam ujaran yang diposting tidak memiliki emosi dan cenderung flat. Kemudian label terakhir akan dilabeli dengan label negatif yang mewakili emosi negatif, seperti contohnya sedih, marah, takut, khawatir, cemas, dan sebagainya. Dataset yang nantinya digunakan pada penelitian kali ini adalah dataset yang ada pada situs *Kaggle*, dimana dataset tersebut sudah sesuai spesifikasi dan diambil dari sosial media X. Di dalam dataset tersebut terdapat 4091 data dimana dari keseluruhan data tersebut mengacu pada satu topik yang sedang hangat untuk dibahas, yaitu sentimen terkait dengan pemilihan presiden tahun 2024.

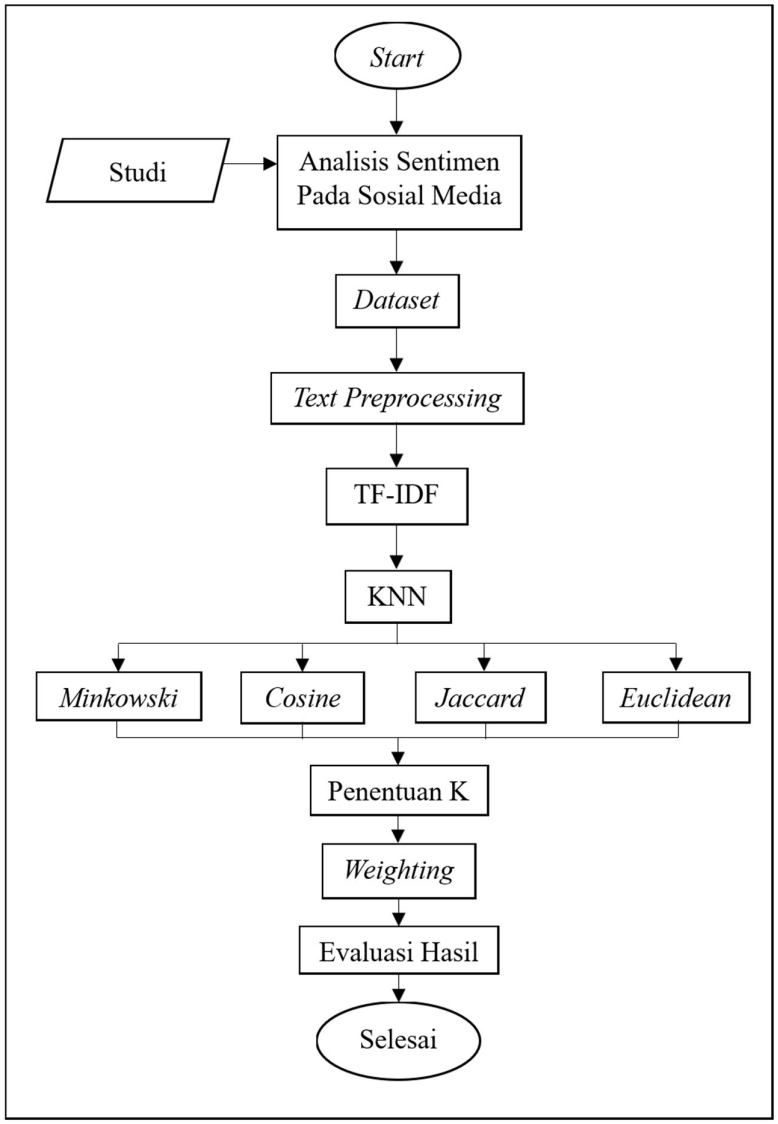
3.1.2. Alat Penelitian

Perancangan sistem pada penelitian ini dilakukan implementasi metode WKNN, dimana nantinya akan menggunakan algoritma KNN dengan 3 jenis perhitungan distance dan akan ditambahkan perhitungan bobot tetangga (*weighting*). Proses implementasi ini akan dilakukan pada Bahasa pemrograman *python*. *Software* yang digunakan adalah *Pycharm* dengan kontribusi library yang dapat dimanfaatkan untuk menunjang proses Pembangunan sistem. Adapun *software* lain yang digunakan adalah *Windows* 11 sebagai sistem operasi, *Microsoft Word* tahun 2021 untuk membuat laporannya, dan *Mendeley* untuk mempermudah

melakukan manajemen referensi. Selain itu, akses internet juga dibutuhkan untuk mencari penelitian-penelitian terkait. Adapun *hardware* atau perangkat keras yang digunakan untuk melakukan proses penelitian menggunakan satu unit laptop dengan spesifikasi *intel core* i7-9750H, 32GB RAM dan 1TB harddisk.

3.2. Prosedur Penelitian

Pada penelitian ini, nantinya akan dibangun suatu sistem yang mampu mengklasifikasikan suatu kalimat ini nantinya masuk ke dalam label tertentu sesuai dengan yang sudah ditetapkan sebelumnya. Data dikumpulkan dari postingan pada sosial media *twitter* mengenai sentimen pemilihan presiden 2024. Kemudian dilakukan klasifikasi dengan metode WKNN. Tahapan dari prosedur penelitian terdapat pada Gambar 3.1.



Gambar 3. 1 Bagan Prosedur Penelitian

Berikut ini adalah penjelasan dari Gambar 3.1 mengenai setiap tahapan dari prosedur umum penelitian sesuai dengan gambar diatas.

1. Studi Literatur dan Rumusan Masalah

   Penelitian dilakukan dengan adanya studi literatur penelitian sebelumnya yang terkait, dengan mengambil tema *data mining* yang diharapkan akan menjadi penelitian yang bermanfaat untuk kebutuhan perkembangan teknologi. Setelah menemukan referensi yang ada, diangkatlah suatu rumusan masalah yang nantinya dapat dikembangkan menjadi sebuah

sistem yang berjudul "Pengaruh Distance Metriks Terhadap Performansi *Weighted K-Nearest Neighbor (WKNN)* Pada Analisis Sentimen Di Sosial Media".

2. Dataset

   Data yang akan digunakan pada penelitian kali ini bersumber pada salah satu aplikasi sosial media yang cukup terkenal, yaitu *Twitter* atau yang sekarang ini disebut dengan X. Tema yang diambil sebagai batasan topik adalah mengenai pemilihan presiden 2024 yang saat laporan ini disusun masih sangat hangat dan viral.

3. *Text preprocessing*

   Pada tahapan ini dataset yang sudah ada akan dilakukan proses sedemikian rupa agar data yang diolah sesuai dengan yang diinginkan, meliputi proses penghilangan noise, hingga membuat data lebih terstruktur. Hal ini dilakukan dengan harapan proses penelitian dapat berjalan lebih lancar dan meminimalisir adanya pencilan. Adapun tahapan pada *preprocessing* ini adalah *stopword removal, case folding*, dan tokenisasi.

4. TF-IDF

   Proses ini merupakan tahapan untuk mengubah tipe data teks menjad numerik agar nantinya dapat dilakukan proses komputasi untuk proses selanjutnya.

5. KNN

   Pada tahapan ini, data yang sudah melalui proses pembersihan dan pengolahan yang tadinya merupakan data teks menjadi data numerik, selanjutnya adalah pengimplementasian dengan menggunakan algoritma KNN.

6. *Minkowski, Cosine Similarity, dan Jaccard*

   Pada tahapan ini akan dilakukan masing-masing perhitungan dataset yang akan diterapkan dengan 3 jenis perhitungan distance *metrics*. Nantinya masing-masing dari hasil perhitungan akan dilakukan perbandingan dan komparasi dengan perhitungan jarak yang popular digunakan yaitu perhitungan *euclidean* untuk dianalisa lebih lanjut.

7. Penentuan K

   Setelah dilakukan perhitungan jarak antar tetangga, selanjutnya adalah penentuan nilai K. Nilai K yang akan diambil pada penelitian kali ini adalah 1, 3, 5, 7, 11 dan 13. Pengambilan ini dilakukan pada angka ganjil dengan tujuan agar nantinya menghindari nilai sama pada kelas yang akan ditentukan.

8. *Weighting*

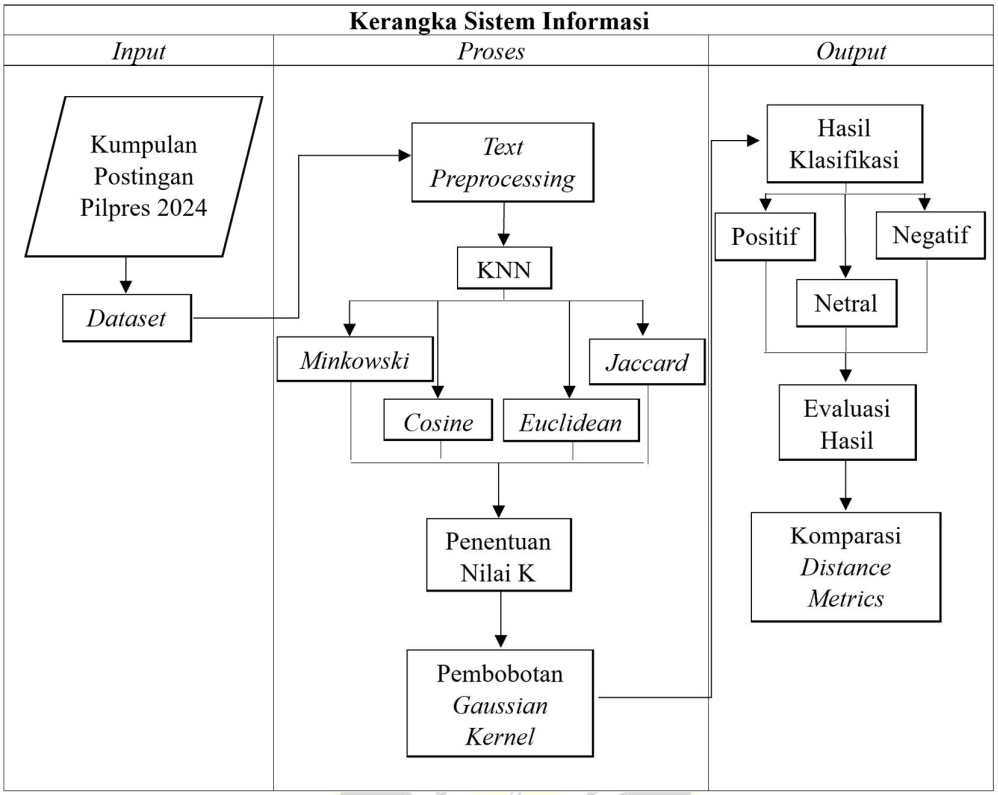
   Setelah melakukan penentuan nilai K, maka tahapan selanjutnya adalah melakukan perhitungan bobot. Perhitungan bobot dilakukan dengan menggunakan perhitungan *Gaussian Kernel* yang bertujuan untuk menentukan bobot dan kontribusi tiap-tiap tetangga terhadap data *testing* yang akan diklasifikasikan.

9. Evaluasi Hasil

   Pada bagian ini, seluruh hasil perhitungan yang sudah dilakukan pada tahap-tahap sebelumnya akan diukur nilai akurasi, presisi dan *recall* nya sehingga nanti akan dapat dilakukan perbandingan dan komparasi. Nantinya diharapkan dapat menentukan efektifitas dari algoritma WKNN dengan variasi nilai K dan variasi perhitungan distance metriks nya.

3.3. Kerangka Sistem Informasi

Berdasarkan penelitian ini, kerangka sistem informasi merupakan pedoman yang digunakan sebagai tujuan untuk mengilustrasikan tiap-tiap proses yang akan dibangun. Berikut ini adalah kerangka sistem informasi yang digambarkan pada Gambar 3.1 sebagai berikut.



Gambar 3. 2 Bagan Kerangka Sistem Informasi

Adapun setiap langkah pada Gambar 3.2 kerangka sistem informasi akan dijelaskan sebagai berikut ini :

1. *Input*

   Berdasarkan dari bahan penelitian yang nantinya akan dilakukan. Data yang akan digunakan sebagai bahan penelitian adalah postingan-postingan yang terdapat pada platform sosial media X. Agar nantinya dimensi bahasan bisa terfokus dan tidak terlalu luas, maka penulis akan membatasi topik terkait dengan pemilihan presiden 2024 yang mana topik tersebut merupakan topik yang cukup panas untuk dibahas akhir-akhir ini. Disini peneliti menggunakan dataset yang sudah jadi yang terdapat pada situs *Kaggle*. Dataset ini terdiri dari 4091 data dimana dari keseluruhan data ini diklasifikasikan pada 3 jenis label, yaitu label positif, negatif dan netral.

2. *Prosess*

   Adapun pada tahapan ini sistem mulai melakukan proses. Dataset yang sudah terbentuk tersebut akan melalui proses *text prepossessing* untuk melakukan

pembersihan dataset terhadap *noise* yang ada pada data. Kemudian akan dilakukan klasifikasi KNN dengan menggunakan 3 macam perhitungan jarak yaitu *Minkowski, Cosine Similarity, dan Jaccard*. Data *training* dan data *testing* yang akan digunakan pada 3 jenis perhitungan *distance* akan disamakan sehingga nantinya dapat diketahui efektivitas dari metode KNN dengan menggunakan ketiga jenis perhitungan jarak. Setelah itu kemudian dilakukan pengambilan jumlah K atau jumlah tetangga sesuai dengan prosedur pengklasifikasian KNN. Tahapan selanjutnya adalah modifikasi pengembangan dari metode KNN biasa yaitu dilakukan pembobotan tiap tetangga yang sudah dipilih dengan alasan agar dapat mengetahui kontribusi tiap tetangganya terhadap data testing. Algoritma pembobotan yang digunakan adalah *Gaussian Kernel*. Sehingga proses klasifikasi ini menjadi WKNN yang merupakan *Weighted K-Nearest Neighbor*. Setelah itu adalah tahapan pengujian dengan data *testing* yang nantinya akan dievaluasi performanya menggunakan *confusion matriks* untuk mengetahui nilai akurasi, presisi dan *recall* nya. Langkah terakhir adalah melakukan komparasi pada performa dari WKNN dengan *Minkowski*, WKNN dengan *Cosine Similarity* dan WKNN dengan *Jaccard* untuk melihat mana yang paling efektif diantara ketiga jenis tersebut.

3. *Output*

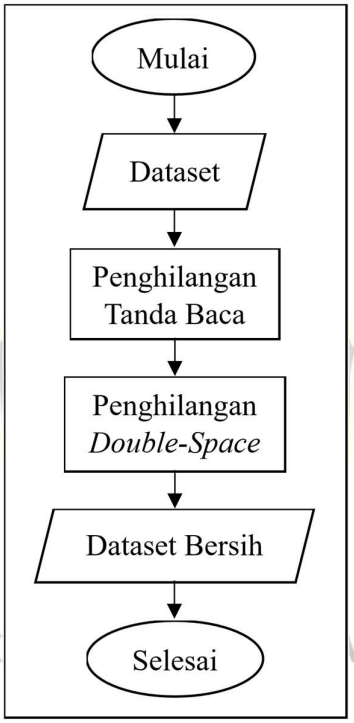
   Setelah seluruh rangkaian dilakukan, meliputi proses *input* dan proses *output*. Maka nantinya akan mendapatkan hasil *output* yang menunjukkan bahwa data testing yang dilakukan tes apakah akan menghasilkan label positif, negatif ataupun netral.

3.4. *Preprocessing Text*

Pengumpulan data adalah proses mengumpulkan informasi dan dataset yang relevan yang penting untuk analisis dan pengembangan model penelitian (Barwal & Raheja, 2022). Untuk meningkatkan kualitas dan kesesuaian data, langkah-langkah *preprocessing* perlu dilakukan (Islami, Sumijan, & Defit, 2024). Text *preprocessing* adalah langkah penting dalam pemrosesan data teks sebelum analisis lebih lanjut dilakukan, proses ini bertujuan untuk membersihkan data *text* sebelum

dilakukan pemrosesan lebih lanjut. Tujuan utama *text preprocessing* adalah untuk membersihkan dan mengorganisir data mentah agar lebih mudah diproses dan diolah oleh algoritma komputer.

Langkah pertama dalam melakukan *preprocessing text* yaitu dengan *punctuation removal*, yaitu menghilangkan adanya tanda baca seperti (“?”, “!”, “()”, “{}”) dan penggunaan spasi yang berlebihan. *Flowchart* diagram terkait dengan *punctuation removal* dapat dilihat pada Gambar 3.3.



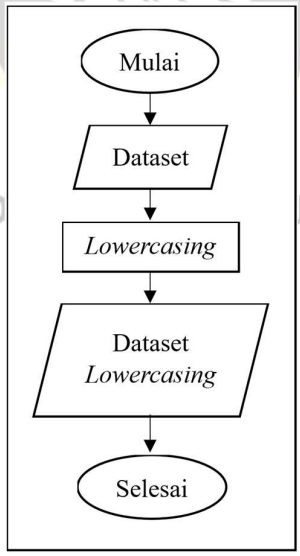
Gambar 3. 3 Prosedur Punctuation Removal

Pada Tabel 3.1 menunjukkan proses sebelum data itu dilakukan *punctuation removal* dan sesudahnya.

Tabel 3. 1 *Punctuation Removal*

Sebelum	Sesudah
Tunggu saja!!! meja kehormatan (Hakim MK) sedang bekerja karena legitimasi adalah segalanya!! ingat pkpu kpu membutuhkan kejelasan hukum, sebelum melaksanakan pemilihan umum supaya Pemilu tidak #CACAT HUKUM	Tunggu saja meja kehormatan Hakim MK sedang bekerja karena legitimasi adalah segalanya ingat pkpu kpu membutuhkan kejelasan hukum sebelum melaksanakan pemilihan umum supaya pemilu tidak CACAT HUKUM

Kemudian selanjutnya adalah proses *lowercasing* atau *case folding*. Proses ini merupakan konversi untuk membuat semua dataset *text* yang tadinya terdiri dari huruf kapital menjadi huruf kecil (*lowercase*). *Case folding* dibutuhkan untuk membuat semua data dokumen menjadi bentuk standarnya. *Flowchart* diagram yang menjelaskan mengenai tahapan dari *case folding* akan ditunjukkan pada Gambar 3.4.



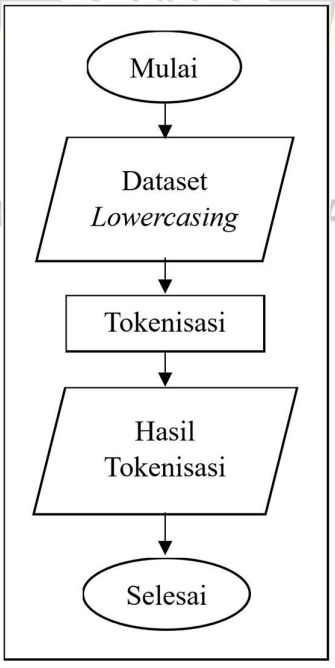
Gambar 3. 4 Prosedur *Case Folding*

Tabel 3.2 menampilkan suatu dokumen sebelum dan sesudah dilakukan *case folding/lowercasing*.

Tabel 3. 2 *Case Folding*

Sebelum	Sesudah
Tunggu saja meja kehormatan Hakim MK sedang bekerja karena legitimasi adalah segalanya ingat pkpu kpu membutuhkan kejelasan hukum sebelum melaksanakan pemilihan umum supaya pemilu tidak CACAT HUKUM	tunggu saja meja kehormatan hakim mk sedang bekerja karena legitimasi adalah segalanya ingat pkpu kpu membutuhkan kejelasan hukum sebelum melaksanakan pemilihan umum supaya pemilu tidak cacat hukum

Langkah selanjutnya adalah tokenisasi, merupakan proses pemecahan teks menjadi unit-unit kecil yang disebut token, di mana setiap token dapat berupa kata, frasa, atau bahkan karakter individual. Dengan melakukan tokenisasi, komputer dapat memahami struktur teks dan memprosesnya dalam bentuk yang lebih mudah dikelola oleh algoritma. *Flowchart* mengenai tokenisasi ini akan ditampilkan pada Gambar 3.5.



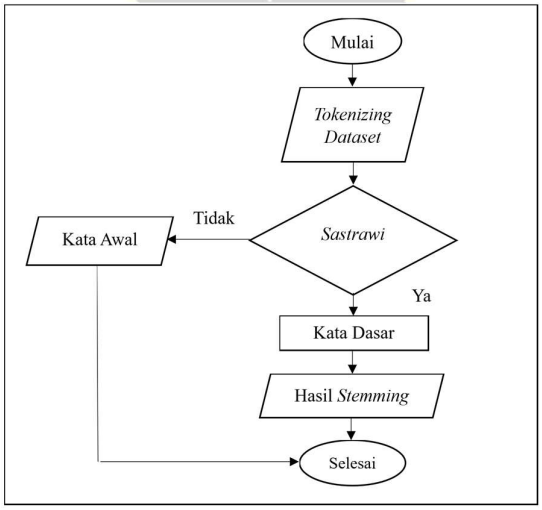
Gambar 3. 5 Prosedur Tokenisasi

Tabel 3.3 merupakan tabel yang menunjukkan bagaimana data teks yang sudah mengalami proses tokenisasi.

Tabel 3. 3 Tokenisasi

Sebelum	Sesudah
tunggu saja meja kehormatan hakim mk sedang bekerja karena legitimasi adalah segalanya ingat pkpu kpu membutuhkan kejelasan hukum sebelum melaksanakan pemilihan umum supaya pemilu tidak cacat hukum	[‘tunggu’, ‘saja’, ‘meja’, ‘kehormatan’, ‘hakim’, ‘mk’, ‘sedang’, ‘bekerja’, ‘karena’, ‘legitimasi’, ‘adalah’, ‘segalanya’, ‘ingat’, ‘pkpu’, ‘kpu’, ‘membutuhkan’, ‘kejelasan’, ‘hukum’, ‘sebelum’, ‘melaksanakan’, ‘pemilihan’, ‘umum’, ‘supaya’, ‘pemilu’, ‘tidak’, ‘cacat’, ‘hukum’]

Stemming adalah proses dalam *preprocessing text* yang bertujuan untuk mengurangi kata-kata turunan ke bentuk dasarnya atau kata dasar (*stem*). Proses ini dilakukan dengan menghilangkan imbuhan seperti awalan, akhiran, atau bentuk kata jamak, sehingga kata-kata yang berbeda bentuknya tetapi memiliki arti dasar yang sama dapat diwakili oleh satu kata. Tahapan dalam *stemming* akan disajikan pada Gambar 3.6.



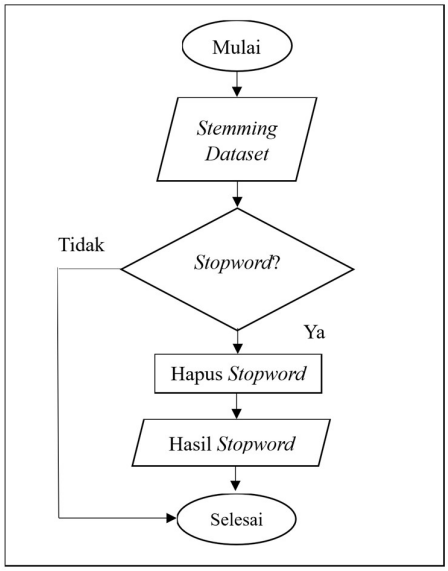
Gambar 3. 6 Prosedur *Stemming*

Berikut Tabel 3.4 menunjukkan proses sebelum dan sesudah *stemming*. Terlihat bahwa dokumen sudah mengalami penghilangan imbuhan seperti awalan, akhiran, atau bentuk kata jamak.

Tabel 3. 4 *Stemming*

Sebelum	Sesudah
[‘tunggu’, ‘saja’, ‘meja’, ‘kehormatan’, ‘hakim’, ‘mk’, ‘sedang’, ‘bekerja’, ‘karena’, ‘legitimasi’, ‘adalah’, ‘segalanya’, ‘ingat’, ‘pkpu’, ‘kpu’, ‘membutuhkan’, ‘kejelasan’, ‘hukum’, ‘sebelum’, ‘melaksanakan’, ‘pemilihan’, ‘umum’, ‘supaya’, ‘pemilu’, ‘tidak’, ‘cacat’, ‘hukum’]	[‘tunggu’, ‘saja’, ‘meja’, ‘hormat’, ‘hakim’, ‘mk’, ‘sedang’, ‘bekerja’, ‘karena’, ‘legitimasi’, ‘adalah’, ‘segalanya’, ‘ingat’, ‘pkpu’, ‘kpu’, ‘butuh’, ‘jelas’, ‘hukum’, ‘sebelum’, ‘melaksanakan’, ‘pilih’, ‘umum’, ‘supaya’, ‘pemilu’, ‘tidak’, ‘cacat’, ‘hukum’]

Stopword removal merupakan tahapan yang digunakan untuk menghapus kata-kata umum yang sering muncul dalam teks tetapi tidak memberikan informasi penting dalam analisis, seperti "dan", "di", "yang", atau "dengan". Kata-kata ini disebut *stopwords* dan biasanya tidak memberikan kontribusi signifikan terhadap pemahaman atau interpretasi makna dari teks. Dengan menghilangkan *stopwords*, ukuran data yang diproses menjadi lebih kecil dan lebih efisien, sehingga algoritma akan berjalan dengan lebih optimal dan dapat meningkatkan hasil akurasi yang didapat. Gambar 3.7 merupakan tahapan proses dalam *stopword removal*.



Gambar 3. 7 Proses *Stopword Removal*

Data hasil dari proses ini dapat dilihat pada Tabel 3.5. Terlihat bahwa dokumen sudah melewati proses penghapusan kata-kata umum yang sering muncul dalam teks tetapi tidak memberikan informasi penting dalam analisis, seperti "dan", "di", "yang", atau "dengan".

Tabel 3. 5 *Stopword Removal*

Sebelum	Sesudah
[‘tunggu’, ‘saja’, ‘meja’, ‘hormat’, ‘hakim’, ‘mk’, ‘sedang’, ‘bekerja’, ‘legitimasi’, ‘karena’, ‘legitimasi’, ‘adalah’, ‘segalanya’, ‘ingat’, ‘pkpu’, ‘kpu’, ‘butuh’, ‘jelas’, ‘hukum’, ‘sebelum’, ‘melaksanakan’, ‘pilih’, ‘umum’, ‘pemilu’, ‘tidak’, ‘cacat’, ‘supaya’, ‘hukum’]	[‘tunggu’, ‘saja’, ‘meja’, ‘hormat’, ‘hakim’, ‘mk’, ‘bekerja’, ‘legitimasi’, ‘adalah’, ‘segalanya’, ‘ingat’, ‘pkpu’, ‘kpu’, ‘butuh’, ‘jelas’, ‘hukum’, ‘sebelum’, ‘melaksanakan’, ‘pilih’, ‘umum’, ‘pemilu’, ‘tidak’, ‘cacat’, ‘hukum’]

3.5. TF – IDF

TF-IDF (Term Frequency - Inverse Document Frequency) adalah salah satu metode yang umum digunakan untuk menilai seberapa penting suatu kata dalam sebuah dokumen relatif terhadap kumpulan dokumen atau korpus. Metode ini

membantu mengidentifikasi kata-kata yang relevan dalam sebuah dokumen dengan cara memberikan bobot lebih tinggi kepada kata-kata yang sering muncul dalam dokumen tetapi jarang muncul dalam dokumen lain di korpus.

Proses perhitungan dengan menggunakan metode TF IDF ini akan dijelaskan dengan proses dibawah ini. Sebagai contoh terdapat 2 dokumen yang disajikan pada Tabel 3.6. Yang pertama dilakukan adalah menghitung *Term Frequency (TF)* yaitu frekuensi kemunculan setiap kata dalam suatu dokumen.

Tabel 3. 6 Dokumen TF-IDF

Dokumen 1	Dokumen 2
['namanya', 'menjelang', 'pemilu', 'kaya', 'ga', 'tau']	['sudah', 'mulai', 'pemanasan', 'pemilu', 'ga', 'terlalu']

- Dokumen 1

  TF (namanya,d1) = $\frac{1}{6}$ = 0,167

  TF (menjelang,d1) = $\frac{1}{6}$ = 0,167

  TF (pemilu,d1) = $\frac{1}{6}$ = 0,167

  TF (kaya,d1) = $\frac{1}{6}$ = 0,167

  TF (ga,d1) = $\frac{1}{6}$ = 0,167

  TF (tau,d1) = $\frac{1}{6}$ = 0,167

- Dokumen 2

  TF (sudah,d2) = $\frac{1}{6}$ = 0,167

  TF (mulai,d2) = $\frac{1}{6}$ = 0,167

  TF (pemanasan,d2) = $\frac{1}{6}$ = 0,167

  TF (pemilu,d2) = $\frac{1}{6}$ = 0,167

  TF (ga,d2) = $\frac{1}{6}$ = 0,167

  TF (terlalu,d2) = $\frac{1}{6}$ = 0,167

Selanjutnya adalah melakukan perhitungan pada *Inverse Document Frequency (IDF)* untuk setiap kata pada masing-masing dokumen.

IDF (namanya, d1&d2) = $\log\left(\frac{2}{1}\right) = \log 2 = 0{,}301$

IDF (menjelang, d1&d2) = $\log\left(\frac{2}{1}\right) = \log 2 = 0{,}301$

IDF (pemilu, d1&d2) = $\log\left(\frac{2}{2}\right) = \log 1 = 0$

IDF (kaya, d1&d2) = $\log\left(\frac{2}{1}\right) = \log 2 = 0{,}301$

IDF (ga, d1&d2) = $\log\left(\frac{2}{2}\right) = \log 1 = 0$

IDF (tau, d1&d2) = $\log\left(\frac{2}{1}\right) = \log 2 = 0{,}301$

IDF (sudah, d1&d2) = $\log\left(\frac{2}{1}\right) = \log 2 = 0{,}301$

IDF (mulai, d1&d2) = $\log\left(\frac{2}{1}\right) = \log 2 = 0{,}301$

IDF (pemanasan, d1&d2) = $\log\left(\frac{2}{1}\right) = \log 2 = 0{,}301$

IDF (terlalu, d1&d2) = $\log\left(\frac{2}{1}\right) = \log 2 = 0{,}301$

Setelah diketahui masing-masing nilai dari perhitungan TF dan IDF dari setiap kata pada dokumen, selanjutnya adalah melakukan perhitungan TF-IDF yaitu dengan mengalikan nilai TF dan nilai IDF pada setiap kata. Data akan ditampilkan pada Tabel 3.7.

Tabel 3. 7 Proses TF-IDF

Kata	TF (D1)	TF (D2)	IDF	TF-IDF (D1)	TF-IDF (D2)
namanya	0,167	0	0,301	0,050	0
menjelang	0,167	0	0,301	0,050	0
pemilu	0,167	0,167	0	0	0
kaya	0,167	0	0,301	0,050	0
ga	0,167	0,167	0	0	0
tau	0,167	0	0,301	0,050	0
sudah	0	0,167	0,301	0	0,050
mulai	0	0,167	0,301	0	0,050
pemanasan	0	0,167	0,301	0	0,050
terlalu	0	0,167	0,301	0	0,050

Tabel 3.7 merupakan akhir dari perhitungan proses TF-IDF. Proses ini bertujuan untuk merubah atau mengkonversikan data dalam bentuk teks menjadi angka yang nanti nya dapat diproses lebih lanjut untuk menghasilkan informasi.

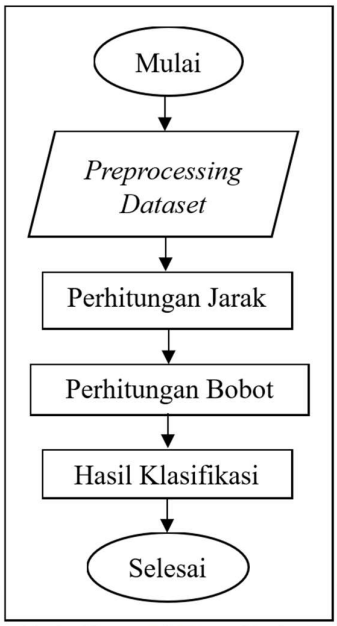
3.6. Pembagian Data

Pada penelitian kali ini dataset yang digunakan merupakan dataset berjenis *text*, yaitu kumpulan postingan pada sosial media X yang pada keseluruhan terdapat 4090 data. Dari keseluruhan dataset tersebut terbagi menjadi 3 kelas, yaitu kelas positif, kelas netral dan kelas negatif. Adapun jumlah dari masing masing kelas adalah kelas positif berjumlah 2.097 data, kelas netral ada 832 data dan kelas negatif berjumlah 1.161 data. Terkait dengan sumber daya peralatan dan komputer yang seadanya, maka penentuan resampling dibuat menjadi masing-masing berjumlah 800 data. Hal ini dimaksudkan agar proses komputasi dapat berjalan efektif dan tidak memakan waktu yang terlalu lama. Pada semua kelas akan dilakukan under-sampling, dimana hanya akan diambil 800 data secara acak. Kemudian selanjutnya untuk pembagian data latih dan data uji menggunakan perbandingan 80:20. (Aggarwal & Zhai, 2012) dalam bukunya menyarankan pembagian data yang memprioritaskan data training (lebih besar) agar model dapat mempelajari fitur teks dengan lebih baik. Kemudian menurut (Kohavi, 1995) dalam studinya tentang validasi silang (*cross-validation*), proporsi 80:20 memberikan keseimbangan yang baik antara data untuk pelatihan dan evaluasi. Dimana nantinya 80% dari dataset akan menjadi data uji dan 20% dari dataset akan menjadi data latihnya. Sehingga nantinya jumlah data latihnya sebesar 1920 data dan 480 data akan berperan sebagai data uji.

3.7. Klasifikasi *Weighted K-Nearest Neighbor*

Weighted K-Nearest Neighbors (WKNN) merupakan pengembangan dari algoritma *K-Nearest Neighbors (KNN)* yang memberikan bobot lebih besar kepada tetangga terdekat saat melakukan klasifikasi atau regresi. Dalam WKNN, tetangga yang lebih dekat dengan titik data uji diberikan pengaruh (bobot) lebih besar dibandingkan tetangga yang lebih jauh. Dalam KNN biasa, semua tetangga diperlakukan sama, terlepas dari seberapa dekat mereka dengan data uji. Hal ini bisa menyebabkan hasil yang tidak akurat jika tetangga yang jauh memberikan

pengaruh yang sama seperti tetangga yang lebih dekat. WKNN mengatasi masalah ini dengan memberikan bobot lebih besar pada tetangga yang lebih dekat, sehingga meningkatkan akurasi.



Gambar 3. 8 Prosedur Klasifikasi WKNN

Gambar 3.8 merupakan *flowchart* dari proses klasifikasi dengan algoritma WKNN. Langkah pertama dalam melakukan klasifikasi adalah dengan menetapkan terlabih dahulu dokumen latih dan dokumen uji. Tabel 3.8 akan menyajikan dokumen latih dan ujinya sebagai berikut.

Tabel 3. 8 Dokumen Latih dan Uji

No	Dokumen Latih	Label	Dokumen Uji
1	Saya suka belajar	Positif	Belajar itu menyenangkan
2	Belajar adalah kegiatan yang menyenangkan	Positif	
3	Tidak suka bekerja	Negatif	

Pada Tabel 3.8 terdapat 3 dokumen latih yang mana 2 diantaranya memiliki label positif dan satunya lagi memiliki label negatif. Disini akan dilakukan proses perhitungan dan klasifikasi sehingga nanti dokumen uji akan dapat dilabelkan

sesuai dengan dokumen latihnya. Setelah dilakukan preprocessing data dan konversi menjadi matriks angka didapatkan Tabel 3.9.

Tabel 3. 9 Konversi Matriks Numerik

No	Dokumen	Preprocessing	TF-IDF
1	Saya suka belajar	"saya", "suka", "belajar"	[1, 1, 1, 0, 0, 0, 0, 0]
2	Belajar adalah kegiatan yang menyenangkan	"belajar", "adalah", "kegiatan", "menyenangkan"	[0, 0, 1, 1, 1, 1, 0, 0]
3	Tidak suka bekerja	"suka", "tidak", "bekerja"	[0, 1, 0, 0, 0, 0, 1, 1]
4	Belajar itu menyenangkan	"belajar", "menyenangkan"	[0, 0, 1, 0, 0, 1, 0, 0]

Pada tabel 3.9 terdapat dokumen latih yaitu dokumen 1, 2 dan 3 serta dokumen uji pada dokumen 4. Kemudian dilakukan proses perhitungan jarak antara dokumen uji dengan masing-masing dokumen latih yang sudah dilakukan proses konversi menjadi tipe data numerik. Sebagai contoh disini metriks yang digunakan adalah menggunakan metriks *euclidean* distance.

- Jarak *Euclidean* dokumen uji dengan dokumen 1

$$d(D1, D4) = \sqrt{(D1_1 - D4_1)^2 + (D1_2 - D4_2)^2}$$

$$d(D1, D4) = \sqrt{\begin{matrix}(0 - 1)^2 + (0 - 1)^2 + (1 - 1)^2 + (0 - 0)^2 + \\ (0 - 0)^2 + (1 - 0)^2 + (0 - 0)^2 + (0 - 0)^2\end{matrix}}$$

$$d(D1, D4) = \sqrt{1 + 1 + 0 + 0 + 0 + 1 + 0 + 0}$$

$$d(D1, D4) = \sqrt{3} \approx 1{,}732$$

- Jarak *Euclidean* dokumen uji dengan dokumen 2

$$d(D2, D4) = \sqrt{(D2_1 - D4_1)^2 + (D2_2 - D4_2)^2}$$

$$d(D2, D4) = \sqrt{\begin{matrix}(0 - 0)^2 + (0 - 0)^2 + (1 - 1)^2 + (0 - 1)^2 + \\ (0 - 1)^2 + (1 - 1)^2 + (0 - 0)^2 + (0 - 0)^2\end{matrix}}$$

$$d(D2, D4) = \sqrt{0 + 0 + 0 + 1 + 1 + 0 + 0 + 0}$$

$$d(D2, D4) = \sqrt{2} \approx 1{,}414$$

- Jarak *Euclidean* dokumen uji dengan dokumen 3

$$d(D3, D4) = \sqrt{(D3_1 - D4_1)^2 + (D3_2 - D4_2)^2}$$

$$d(D3, D4) = \sqrt{\begin{matrix}(0 - 0)^2 + (0 - 1)^2 + (1 - 0)^2 + (0 - 0)^2 + \\ (0 - 0)^2 + (1 - 0)^2 + (0 - 1)^2 + (0 - 1)^2\end{matrix}}$$

$$d(D3, D4) = \sqrt{0 + 1 + 1 + 0 + 0 + 1 + 1 + 1}$$

$$d(D3, D4) = \sqrt{5} \approx 2{,}236$$

Selanjutnya adalah proses pengembangan dari metode KNN yaitu dengan menambahkan bobot pada setiap data uji sehingga dengan adanya pembobotan ini dapat mengetahui kontribusi dari setiap tetangga yang akan membuat metode WKNN menjadi optimal.

- Bobot untuk dokumen 1

$$W(D1,D4) = e^{-\frac{d^2}{2\sigma^2}}$$

$$W(D1,D4) = e^{-\frac{(1{,}732)^2}{2x1^2}}$$

$$W(D1,D4) = e^{-1.5} \approx 0{,}223$$

- Bobot untuk dokumen 2

$$W(D2,D4) = e^{-\frac{d^2}{2\sigma^2}}$$

$$W(D2,D4) = e^{-\frac{(1{,}414)^2}{2x1^2}}$$

$$W(D2,D4) = e^{-1} \approx 0{,}368$$

- Bobot untuk dokumen 3

$$W(D3,D4) = e^{-\frac{d^2}{2\sigma^2}}$$

$$W(D3,D4) = e^{-\frac{(2{,}236)^2}{2x1^2}}$$

$$W(D3,D4) = e^{-2.5} \approx 0{,}082$$

Setelah dilakukan pembobotan pada masing masing data latih maka dilakukan penjumlahan pada label yang sama dan dibandingkan nilai dari hasil pemjumlahan tiap label nya. Tabel 3.10 merupakan hasil pembobotan dari setiap dokumen latih.

Tabel 3. 10 Hasil Pembobotan

Dokumen	Label	Bobot
1	positif	0,223
2	positif	0,368
3	negatif	0,082

Dari Tabel 3.10 dapat terlihat bahwa hasil penjumlahan label positif adalah 0,223 dijumlahkan dengan 0,368 menghasilkan hasil 0,591 dan penjumlahan label negatif sejumlah 0,082. Terlihat bahwa nilai pada label positif lebih tinggi terhadap label negatif. Sehingga dapat disimpulkan bahwa klasifikasi kelas untuk dokumen uji diklasifikasikan menjadi kelas positif.

Berbeda dengan perhitungan dengan menggunakan *distance metrics Euclidean, minkowski* dan *cosine similarity*, perhitungan jarak *jaccard* tidak menggunakan perhitungan TF-IDF untuk melakukan perhitungan. Hal ini dikarenakan perhitungan *jaccard* menggunakan konsep himpunan sebagai dasar perhitungan, seperti penggunaan irisan dan gabungan (*union*). Berikut adalah contoh perhitungan WKNN dengan menggunakan perhitungan jarak *jaccard* dengan dokumen latih dan dokumen uji yang sama seperti pada Tabel 3.8.

- Jarak *Jaccard* dokumen uji dengan dokumen 1

$$J(D1,D4)=1-\frac{|D1\cap D4|}{|D1\cup D4|}$$

$$J(D1,D4)=1-\frac{|\{'belajar'\}|}{|\{saya','suka','belajar','itu','menyenangkan'\}|}$$

$$J(D1,D4)=1-\frac{1}{5}=0,80$$

- Jarak *Jaccard* dokumen uji dengan dokumen 2

$$J(D2,D4)=1-\frac{|D2\cap D4|}{|D2\cup D4|}$$

$$J(D2,D4)=1-\frac{|\{'belajar','menyenangkan'\}|}{|\{'adalah','kegiatan','yang','menyenangkan'\}| \quad \{suka','belajar','itu'\}}$$

$$J(D2,D4)=1-\frac{2}{6}=0{,}67$$

- Jarak *Jaccard* dokumen uji dengan dokumen 3

$$J(D3,D4)=1-\frac{|D3\cap D4|}{|D3\cup D4|}$$

$$J(D3,D4)=1-\frac{|\emptyset|}{|\{'tidak','suka','bekerja','menyenangkan'\}| \quad \{'belajar','itu'\}}$$

$$J(D3,D4)=1-0=1$$

Selanjutnya adalah proses pengembangan dari metode KNN yaitu dengan menambahkan bobot pada setiap data uji.

- Bobot untuk dokumen 1

$$W(D1,D4)=e^{-\frac{d^2}{2\sigma^2}}$$

$$W(D1,D4)=e^{-\frac{(0,80)^2}{2x1^2}}$$

$$W(D1,D4)=e^{-1.5}\approx 0{,}726$$

- Bobot untuk dokumen 2

$$W(D2,D4)=e^{-\frac{d^2}{2\sigma^2}}$$

$$W(D2,D4)=e^{-\frac{(0,67)^2}{2x1^2}}$$

$$W(D2,D4)=e^{-1}\approx 0{,}801$$

- Bobot untuk dokumen 3

$$W(D3,D4)=e^{-\frac{d^2}{2\sigma^2}}$$

$$W(D3,D4)=e^{-\frac{(1)^2}{2x1^2}}$$

$$W(D3,D4)=e^{-2.5}\approx 0{,}607$$

Setelah dilakukan pembobotan pada masing masing data latih maka dilakukan penjumlahan pada label yang sama dan dibandingkan nilai dari hasil pemjumlahan tiap label nya. Tabel 3.11 merupakan hasil pembobotan dari setiap dokumen latih.

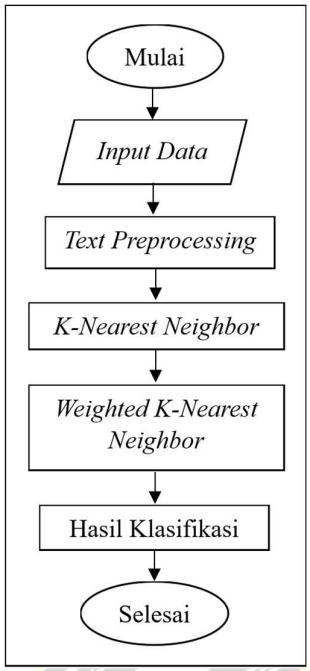
Tabel 3. 11 Pembobotan *Distance Jaccard*

Dokumen	Label	Bobot
1	positif	0,726
2	positif	0,801
3	negatif	0,607

Dari Tabel 3.11 dapat terlihat bahwa hasil penjumlahan label positif adalah 0,726 dijumlahkan dengan 0,801 menghasilkan hasil 1,527 dan penjumlahan label negatif sejumlah 0,607. Terlihat bahwa nilai pada label positif lebih tinggi terhadap label negatif. Sehingga dapat disimpulkan bahwa klasifikasi kelas untuk dokumen uji diklasifikasikan menjadi kelas positif.

3.8. Perancangan Sistem

Alur daripada perancangan proses dimulai dengan melakukan *input* data, user menginputkan data berupa kalimat yang nantinya akan diklasifikasikan menjadi label positif, negatif atau netral. Data yang sudah dimasukkan akan dilakukan *preprocessing data*, meliputi pembersihan dari *noise* dan merubah tipe data *string* menjadi numerik agar dapat dilakukan proses klasifikasi. Kemudian selanjutnya adalah dengan melakukan klasifikasi KNN dimana pada penelitian kali ini menggunakan 4 jenis perhitungan *distance metrics*, yaitu *euclidean, cosine similarity, minkowski* dan *jaccard*. Setelah melalui klasifikasi KNN selanjutnya adalah melakukan pembobotan dengan *Gaussian Kernel*, sehingga metode menjadi *Weighted K-Nearest Neighbor (WKNN)*. Tahap terakhir, sistem akan menampilkan data input awal dan hasil klasifikasi data dari proses analisis ini dan mendapatkan label kelas. Adapun alur dari perancangan sistem informasi terdapat pada Gambar 3.9.



Gambar 3. 9 Bagan Perancangan Sistem