BAB II
LANDASAN TEORI

2.1 Tinjauan Pustaka

Penelitian yang dilakukan oleh Kusuma & Nas, (2023) membahas pengembangan sistem pakar untuk mendiagnosa gangguan kesehatan mental menggunakan algoritma Dempster-Shafer. Sistem ini dirancang untuk mengatasi tantangan dalam mendiagnosa gangguan mental yang kompleks dan memakan waktu, dengan mengumpulkan data gejala dari 30 pasien yang mengalami enam jenis gangguan mental. Algoritma Dempster-Shafer digunakan untuk memproses data ini, menghasilkan tingkat kepercayaan hingga 97% dalam diagnosis. Sistem ini diimplementasikan dengan antarmuka berbasis web yang memungkinkan staf klinis untuk mengelola data pasien dan menampilkan hasil diagnosis. Studi ini menyoroti pentingnya teknologi dalam diagnosis kesehatan mental dan mengusulkan penelitian lanjutan untuk menggabungkan algoritma ini dengan metode lain guna meningkatkan akurasi.

Kemudian penelitian yang dilakukan oleh Anindita et al., (2023) membahas tentang pembuatan sistem pakar yang dirancang untuk mendiagnosis penyakit mental menggunakan metode *Forward Chaining* dan *Certainty Factor* dengan menggunakan 49 data. Penelitian ini merinci kemampuan sistem untuk menilai gejala kesehatan mental dan menghubungkannya dengan gangguan tertentu melalui serangkaian aturan yang diturunkan dari pakar. Sistem ini memiliki tingkat akurasi yang tinggi sebesar 95,918% jika dibandingkan dengan diagnosis pakar, yang divalidasi oleh seorang psikolog. Sistem ini memiliki antarmuka pengguna untuk pemilihan gejala dan hasil diagnosis, dan telah diuji untuk fungsionalitas dan akurasinya. Studi ini menekankan pentingnya diagnosis dini, khususnya di Indonesia, dan menyarankan penelitian di masa mendatang untuk memperluas basis pengetahuan sistem guna meningkatkan akurasi.

Penelitian yang dilakukan oleh Xu et al., (2022) membahas pendekatan CBR dengan *Matrix Learning* yang diawasi untuk mendiagnosis Penyakit Nodul Tiroid (TDN) berdasarkan pemeriksaan USG. Pendekatan yang diusulkan menggunakan kasus TDN historis untuk menghasilkan solusi diagnosis patologis untuk kasus baru, menggabungkan vektor fitur, diagnosis keseluruhan, dan diagnosis patologis untuk menentukan kesamaan antar kasus dan menghasilkan diagnosis untuk kasus baru. Hasil eksperimen menunjukkan bahwa pendekatan ini mengungguli model CBR dan pembelajaran mesin tradisional dalam mendiagnosis TDN, sehingga menunjukkan keefektifannya dalam tugas tersebut.

Kemudian penelitian yang dilakukan oleh Hassen et al., (2022) membahas pendekatan *Case-Based Reasoning* (CBR) untuk masalah routing dan penjadwalan *Home Health Care* (HHC) untuk menangani kejadian tak terduga secara efisien dengan mengklasifikasikan kasus menjadi empat kelompok utama, yaitu *Caregiver Information* (CI), Patient Information (PI), *Service Information* (SI), dan *Type of Perturbation* (TP). Fokusnya adalah menciptakan sistem pendukung keputusan yang menggunakan kasus-kasus masa lalu untuk bereaksi dengan cepat dan menjadwalkan ulang aktivitas pengasuh yang terganggu oleh kejadian seperti permintaan mendesak atau ketidakhadiran pengasuh. Tujuannya adalah untuk mengurangi keterlambatan dalam layanan pengasuh dan meningkatkan kepuasan pasien dengan memberikan dukungan pengambilan keputusan secara real-time di lingkungan HHC yang dinamis.

Penelitian lain tentang model CBR juga dilakukan oleh Bentaiba-Lagrid et al., (2020) membahas tentang pendekatan baru terhadap CBR di bidang medis, dengan fokus pada peningkatan akurasi dan efisiensi melalui pengacakan, pemilihan fitur, pembobotan, pembuatan aturan, dan proses validasi. Metode yang diusulkan mencakup proses validasi tiga lapis yang dihasilkan dari randomisasi, dan segmentasi basis kasus untuk pengambilan lebih cepat. Hasil eksperimen menunjukkan hasil yang menjanjikan dalam hal akurasi dan waktu resolusi, sehingga memposisikan pendekatan CBR sebagai metode kompetitif dalam mengklasifikasikan data ilmu kesehatan.

Penelitian tentang *K-Nearest Neighbor* yang dilakukan oleh Siddalingappa & Kanagaraj, (2022) menyatakan KNN bekerja sebagai algoritma untuk tugas klasifikasi dan regresi. Algoritma ini digunakan untuk memprediksi waktu bertahan hidup dan mengklasifikasikan berbagai tahap kanker mulut. KNN memberikan perhitungan untuk nilai terdekat. Algoritma ini menghitung jarak antara titik data baru/tidak dikenal dengan 'k' titik data menggunakan metrik pengukuran jarak seperti jarak *Euclidean*, jarak *Manhattan*, dan jarak *Minkowski*. KNN dapat memprediksi hasil kelangsungan hidup dan mengklasifikasikan pasien kanker mulut berdasarkan mutasi dan usia. Studi ini menekankan pentingnya prediksi yang akurat bagi dokter, pasien, dan peneliti di bidang prognosis kanker mulut. Model yang diusulkan melibatkan pengumpulan, analisis, dan klasifikasi data menggunakan algoritma KNN, dengan data yang hilang ditangani melalui imputasi atau penurunan nilai, dan standarisasi diterapkan untuk konsistensi data. Studi ini menunjukkan hasil yang menjanjikan dalam memprediksi waktu kelangsungan hidup dan mengklasifikasikan berbagai stadium kanker, menyoroti pentingnya mempertimbangkan usia dan mutasi dalam keputusan pengobatan.

Kemudian penelitian yang dilakukan oleh Sarkar & Leong, (2018) membahas tentang penerapan algoritma *K-Nearest Neighbors* (KNN) pada permasalahan diagnosis Kanker Payudara menggunakan dataset *Wisconsin-Madison Breast Cancer*. Algoritma ini terbukti memberikan hasil klasifikasi yang lebih baik dibandingkan teknik lainnya, dengan peningkatan sebesar 1,17%. Meskipun memiliki kelebihan seperti kesederhanaan, kemudahan implementasi, dan tidak memerlukan sesi pelatihan, algoritma KNN memiliki keterbatasan seperti kebutuhan untuk menyimpan semua data pelatihan dan komputasi jarak yang memakan waktu. Pekerjaan di masa depan disarankan untuk mengeksplorasi versi lanjutan dari algoritma KNN untuk meningkatkan hasil.

Selanjutnya KNN juga diteliti oleh Mushtaq et al., (2020) mempelajari penerapan algoritma *K-Nearest Neighbor* (KNN) dengan berbagai fungsi jarak dan teknik pemilihan fitur untuk mengklasifikasikan dataset kanker payudara. Fungsi jarak yang berbeda, seperti *Manhattan* dan *Canberra*, diuji, bersama dengan metode pemilihan fitur seperti seleksi berbasis Chi-kuadrat. Hasilnya menunjukkan

bahwa kombinasi pemilihan fitur berbasis Chi-kuadrat dengan fungsi jarak *Manhattan* atau *Canberra* menghasilkan tingkat akurasi tertinggi, yang menunjukkan efektivitas pendekatan KNN sederhana untuk klasifikasi kanker payudara. Kemudian hasil akhirnya menunjukkan bahwa pemilihan fitur Chi-kuadrat dengan fungsi jarak *Canberra* atau *Manhattan* yang memiliki rentang nilai k dari 1 hingga 9 merupakan teknik klasifikasi yang baik untuk kumpulan data WBC dan WDBC. Dengan menggunakan metodologi yang sama, algoritma KNN dapat bekerja lebih baik dibandingkan algoritma lain yang memakan waktu dan kompleks dalam masalah klasifikasi kanker payudara.

2.2 Dasar Teori

2.2.1 Sistem Diagnosis Kesehatan Mental

Diagnosis didefinisikan sebagai metode mengidentifikasi suatu penyakit dari tanda dan gejalanya hingga menyimpulkan patologinya, juga dapat didefinisikan sebagai metode untuk mengetahui penyakit mana yang didasarkan pada gejala dan tanda seseorang Kaur et al., (2020). Kesehatan mental didefinisikan oleh Organisasi Kesehatan Dunia (WHO) sebagai keadaan sejahtera di mana individu menyadari kemampuannya sendiri, dapat mengatasi tekanan hidup yang normal, dapat bekerja secara produktif dan bermanfaat, serta mampu memberikan kontribusi kepada komunitasnya (Fusar-Poli et al., 2020).

Sistem diagnosis kesehatan mental adalah sistem pakar yang memanfaatkan keahlian profesional di bidang tertentu dengan mengimplementasikan pengetahuan ke dalam program komputer, sehingga memungkinkan pengguna umum untuk membuat keputusan seperti para ahli. Sistem ini beroperasi dalam dua lingkungan utama: lingkungan pengembangan, yang menggabungkan pengetahuan ahli ke dalam program komputer, dan lingkungan konsultasi, yang memperoleh informasi atau pengetahuan dari ahli melalui program komputer (Kusuma & Nas, 2023).

2.2.2 Text Mining

Text mining merupakan suatu metode yang biasanya dilakukan oleh komputer dengan tujuan untuk mendapatkan informasi atau pengetahuan yang sebelumnya tidak diketahui Widjaja et al., (2023). Mengingat beragamnya jenis

sumber teks, maka informasi atau pengetahuan baru dapat diekstraksi secara otomatis menggunakan perangkat komputer. Tahap pertama dalam proses text mining adalah prapemrosesan teks, hal ini penting dilakukan karena objek text mining biasanya berasal dari basis data yang tidak terstruktur. Dengan menerapkan text preprocessing, maka dapat diperoleh data terstruktur yang akan digunakan pada tahap selanjutnya. Secara umum, tahap prapemrosesan ini meliputi beberapa langkah lanjutan, yaitu *cleaning, normalization, tokenizing, stop word removal*, dan *stemming*.

2.2.3 Klasifikasi dalam Data Mining

Data mining merupakan suatu proses analitis yang dirancang untuk tujuan melakukan eksplorasi data dalam jumlah yang besar untuk menemukan pengetahuan yang berharga, konsisten, dan tersembunyi. Proses ini akan melibatkan penggunaan metode analisis data tingkat lanjut, dimana dengan tujuan untuk menemukan pola dan tautan dalam kumpulan data besar yang sebelumnya tidak diketahui (Nugroho et al., 2022).

Dalam penambangan data, klasifikasi adalah metode pengelompokan data menurut batasan yang telah ditetapkan sebelumnya untuk mengidentifikasi suatu model De Wibowo Muhammad Sidik et al., (2020). Dalam penambangan data, pendekatan kategorisasi ini penting karena dapat mengelola spektrum data yang lebih besar daripada dalam regresi. Klasifikasi menjadi sangat populer digunakan sebagai alat analisis data yang canggih seperti model statistik, algoritma matematika, dan pendekatan pembelajaran mesin karena dapat menyelidiki dan memperkirakan tren dalam set data besar yang sebelumnya tidak tercatat. Dalam proses klasifikasi, data atau contoh dinyatakan oleh pasangan atribut dan nilainya, dan label atau keluaran data biasanya berupa nilai diskrit Putri & Wijayanto, (2022). Setiap metode klasifikasi memiliki kelebihan dan kekurangan, dan pemilihan metode yang tepat bergantung pada karakteristik data dan tujuan analisis.

2.2.4 Jenis Kesehatan Mental

Kesehatan Mental adalah istilah yang merujuk pada tingkat kesejahteraan emosional, psikologis, dan sosial Jean-Berluche, (2024). Kesehatan mental memengaruhi cara berpikir, merasakan, dan bertindak dalam kehidupan sehari-hari.

Kesehatan mental juga memengaruhi kemampuan seseorang untuk mengatasi stres, berinteraksi dengan orang lain, dan membuat keputusan. Kesehatan fisik dan kesehatan mental saling terkait erat. Kesehatan mental yang baik memungkinkan seseorang untuk menjalani kehidupan yang memuaskan, berpartisipasi dalam aktivitas sehari-hari, dan memberikan kontribusi positif bagi komunitasnya. Sebaliknya, masalah kesehatan mental dapat memengaruhi kemampuan seseorang untuk berfungsi dan menikmati hidup.

Stres merupakan reaksi fisiologis di mana berbagai mekanisme pertahanan tubuh berfungsi untuk menghadapi situasi yang mengancam atau menantang. Mengalami stres secara teratur dapat menyebabkan kerusakan pada berbagai sistem tubuh, yang berdampak negatif pada kesehatan mental dan fisik, tetapi stres memicu keadaan "lawan atau lari" melalui pelepasan hormon stres oleh sistem saraf Fujiwara et al., (2022).

Depresi adalah kondisi mental yang memiliki banyak sisi yang mencakup tindakan ekspresif, reaksi fisiologis, dan pengalaman subjektif. Pikiran, perasaan, dan perilaku sehari-hari seseorang dipengaruhi oleh emosinya, yang berdampak besar pada kesehatan mentalnya. Depresi seseorang dapat dipengaruhi oleh sejumlah hal, seperti perubahan biologis dan psikologis serta pengalaman hidup tertentu (Khoridah Maulidiya et al., 2024).

Kecemasan adalah perasaan takut atau khawatir yang sering kali tidak memiliki alasan yang jelas atau spesifik Ojala et al., (2021). Kecemasan merupakan reaksi emosional yang umum terhadap situasi yang dianggap mengancam atau menantang. Kecemasan, yang dapat memengaruhi kesehatan mental seseorang, dapat berkisar dari perasaan gelisah ringan hingga ketakutan yang parah. Kecemasan juga dapat menyebabkan respons fisik seperti detak jantung cepat, keringat berlebih, dan kesulitan bernapas.

Orang yang memiliki kesehatan mental yang normal mampu beraktivitas dengan baik dalam kehidupan sehari-hari karena mereka memiliki keseimbangan emosional dan psikologis M. Dhanabhakyam & Sarath M, (2023). Hal ini melibatkan memiliki tujuan dan makna dalam hidup, mengelola stres, dan

membentuk hubungan yang sehat. Emosi positif, kepuasan hidup, dan pengembangan pribadi juga merupakan komponen kesehatan mental yang baik.

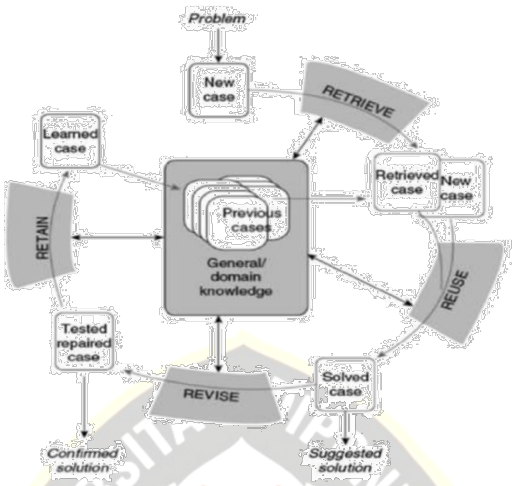
Ciri fisik dari kondisi depresi, cemas, dan stres dapat saling tumpang tindih, namun memiliki karakteristik masing-masing. Depresi ditandai dengan kelelahan berkepanjangan, perubahan pola tidur dan berat badan, serta gerakan tubuh yang melambat. Gangguan cemas memunculkan gejala fisik seperti jantung berdebar, otot tegang, dan pernapasan cepat. Sementara stres umumnya menyebabkan sakit kepala, gangguan tidur, dan ketegangan otot. Pemahaman terhadap gejala fisik ini penting dalam mendukung proses klasifikasi data dan validasi diagnosis berbasis teks yang digunakan dalam penelitian ini (Hendrawati et al., 2021; Surachman et al., 2024)

2.2.5 *Case Based Reasoning* (CBR)

Case Based Reasoning (CBR) merupakan metode yang digunakan untuk mengimplementasikan sistem diagnosa komputer ke dalam aplikasi di dunia nyata. CBR juga dapat digunakan untuk menganalisa suatu masalah sesuai dengan kasus yang dihadapi dan untuk selanjutnya mengklasifikasikan kasus tersebut berdasarkan pada pengalaman masa lalu pengklasifikasian. Kelebihan dari CBR yaitu memungkinkan penggunaan contoh kasus masa lalu untuk mengakuisisi pengetahuan dan akhirnya diketahui pokok permasalahannya. Selain itu CBR juga dapat mencari diagnosis dari permasalahan tersebut berdasarkan dari pengalaman kasus masa lalu sehingga segala permasalahan dapat diselesaikan untuk selanjutnya kasus serta diagnosisnya disimpan dan kemudian dapat digunakan kembali untuk memecahkan kasus baru, jika kasus tersebut hampir sama, atau mungkin sama dengan kasus terdahulu (Leśniak & Zima, 2018).

Struktur CBR yang diungkapkan oleh Oyelade & Ezugwu, (2020), selain CBR *cycle*, gambaran dari struktur CBR dapat lebih memperjelas proses dari CBR secara detail seperti tampak pada keempat proses yang telah dilakukan pada CBR cycle, semuanya masih dapat dipecah kembali menjadi beberapa bagian pekerjaan yang harus dilakukan, seperti keterangan berikut: Proses ini terdiri dari memilih informasi apa dari kasus yang akan disimpan, cara menyusun kasus agar mudah untuk menentukan masalah yang mirip, dan bagaimana mengintegrasikan kasus

baru pada struktur memori. Terdapat empat tipe *case-based reasoning*, yaitu *retrieve, reuse, revise, retain* Franciscatto et al., (2019). Adapun skema yang menggambarkan keempat tipe *case based reasoning* tersebut yaitu seperti ditunjukkan oleh Gambar 2.1



Gambar 2.1 *Case Based Reasoning* (CBR)

Sumber: (Mulyana et al, 2019)

Pada saat terjadi permasalahan baru, pertama-tama sistem akan melakukan proses *Retrieve.* Proses *Retrieve* akan menyalin, menyeleksi, dan melengkapi informasi yang akan digunakan dari persamaan masalah pada database. Setelah proses Retrieve selesai dilakukan, selanjutnya sistem akan melakukan proses Reuse. Di dalam proses *Reuse*, sistem akan menggunakan informasi permasalahan sebelumnya yang memiliki kesamaan untuk menyelesaikan permasalahan yang baru. Pada proses *Reuse* akan menyalin, menyeleksi, dan melengkapi informasi yang akan digunakan.

Selanjutnya pada proses *Revise*, informasi tersebut akan dikalkulasi, dievaluasi, dan diperbaiki kembali untuk mengatasi kesalahan-kesalahan yang terjadi pada permasalahan baru. Tentunya, permasalahan yang akan diselesaikan adalah permasalahan yang memiliki kesamaan dan mengekstrak diagnosis yang baru. Selanjutnya, diagnosis baru itu akan disimpan ke dalam *knowledge-base* untuk menyelesaikan permasalahan yang akan datang. Tentunya, permasalahan yang akan diselesaikan adalah permasalahan yang memiliki kesamaan dengannya.

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

Term Frequency-Inverse Document Frequency (TF-IDF) adalah metode statistik yang digunakan dalam sistem penerjemahan bahasa alami dan pencarian informasi untuk menghitung dua faktor signifikan: *Term frequency* (TF) dan *Inverse Document Frequency* (IDF). Dalam kerangka kumpulan makalah yang lebih besar, pendekatan ini diterapkan untuk mengevaluasi relevansi istilah (kata kunci) dalam sebuah dokumen. *Term frequency* atau TF, mengukur frekuensi kata dalam sebuah manuskrip. TF umumnya dihitung dengan membagi jumlah total kata dalam sebuah dokumen dengan freku Septiani & Isabela, (2022)ensi kata di dalam dokumen tersebut. *Inverse Document Frequency* (IDF) mengukur kepentingan relatif sebuah kata dalam sekumpulan dokumen tertentu. Kata-kata yang menunjukkan frekuensi lebih sedikit di sekumpulan makalah dapat memiliki IDF yang lebih tinggi. IDF dihitung dengan membagi jumlah total dokumen yang dikumpulkan dengan jumlah dokumen yang memuat kata tersebut. Kemudian, logaritma digunakan untuk menghaluskan skala. Dalam konteks kumpulan dokumen yang lebih luas, TF-IDF menghasilkan bobot atau skor yang menunjukkan signifikansi istilah dalam dokumen tertentu. Relevansi dokumen dalam sistem pencarian informasi dapat diukur dan dibandingkan menggunakan bobot ini. Dokumen dengan skor TF-IDF yang lebih tinggi untuk suatu istilah biasanya lebih terkait dengan pencarian atau permintaan pengguna yang menyertakan istilah tersebut. Pendekatan TF-IDF juga memiliki manfaat karena dokumen dengan skor TF-IDF yang lebih tinggi sering kali lebih tepat dalam mencocokkan pencarian atau permintaan pengguna yang menggunakan istilah tersebut.

TF mengukur seberapa sering sebuah kata muncul dalam sebuah dokumen. Ini dihitung dengan membagi jumlah kemunculan kata tertentu dalam dokumen dengan jumlah total kata dalam dokumen tersebut Albin Pranata et al., (2024).

Rumusnya adalah:

$$tf_{t.d} = \frac{jumlah\ kemunculan\ kata\ t\ dalam\ dokumen\ d}{jumlah\ total\ kata\ dalam\ dokumen\ d} \qquad (2.10)$$

IDF memberikan Bobot yang lebih tinggi pada kata-kata yang jarang dianggap lebih informatif, karena kata-kata yang jarang dianggap lebih informatif. Sebaliknya, kata-kata yang sering muncul di banyak dokumen (seperti "dan", "atau", "adalah") memberikan nilai IDF rendah, karena kata kata tersebut kurang memberikan yang berarti.

$$idf_d = log \frac{N}{n_t} \quad (2.11)$$

Setelah kita menghitung nilai TF dan IDF untuk setiap kata dalam dokumen, kita menggabungkan kedua nilai ini untuk mendapatkan nilai TF-IDF. Nilai ini memberikan gambaran yang lebih lengkap tentang pentingnya suatu kata dalam konteks dokumen tertentu.

Rumus untuk TF-IDF:

$$tfidf_{t.d} = tf_{t.d}\ x\ idf_d \quad (2.12)$$

Keterangan :

t = Kata kunci, term

d = Dokumen

t.d = nilai TF-IDF untuk kata t dalam dokumen d

Tf = Banyaknya t (kata) yang dicari dalam dokumen

Idf = Banyak t kebalikan dari kata yang dicari

2.2.7 Oversampling SMOTE

Salah satu metode untuk mengatasi ketidakseimbangan kelas dalam kumpulan data adalah *oversampling*. Ketika jumlah sampel dalam satu kelas jauh lebih sedikit daripada di kelas lain, terjadi ketidakseimbangan kelas, yang dapat menyebabkan prediksi kelas mayoritas yang lebih mungkin oleh model prediktif. *Oversampling* meningkatkan jumlah sampel kelas minoritas sehingga distribusi kelas menjadi lebih seimbang, sehingga mengatasi masalah ini Agung et al., (2020). *Synthetic Minority Oversampling Technique* (SMOTE) adalah salah satu teknik *oversampling* yang sering digunakan. SMOTE didukung oleh pembuatan data sintetis untuk kelas minoritas. Pendekatan ini menghasilkan data sintetis antara titik-titik yang dipilih dan tetangganya menggunakan teknik interpolasi setelah memilih data acak dari kelas minoritas. Dalam hal ini, SMOTE dapat menurunkan

risiko *overfitting* yang terkadang diakibatkan hanya dengan menduplikasi data minoritas.

Rumus perhitungan SMOTE

$$x_{syn} = x_i + rand(0,1)x(x_{knn} - x_i) \quad (2.13)$$

Keterangan

x_{syn} = Data sintetik hasil replikasi

x_i = Data yang akan direplikasi

$rand(0,1)$ = Bilangan random antara 0 dan 1

x_{knn} = Data terdekat dengan data yang akan direplikasi

2.2.8 Cosine Similarity

Cosine similarity digunakan untuk menemukan tingkat kesamaan antara dua hal yang dinyatakan dalam bentuk vektor. Dasar dari pendekatan ini adalah ukuran kesamaan ruang vektor, metrik kesamaan Rio Feriangga Kurniawan, (2022). Dalam penulisan, objek-objek ini sering merujuk pada dokumen atau pertanyaan yang dinyatakan sebagai vektor kata kunci.

Cosine Similarity bukanlah metode berbasis jarak, melainkan metode yang mengukur tingkat kemiripan antara dua vektor berdasarkan sudut (*cosinus*) di antara keduanya. Fokus utama dari *Cosine Similarity* adalah orientasi atau arah vektor, bukan panjang atau besarnya, sehingga sangat berguna untuk membandingkan dokumen yang memiliki ukuran atau panjang berbeda namun topik yang serupa. Berbeda dengan itu, metode jarak seperti *Euclidean Distance* mengukur jarak absolut antara dua titik dalam ruang, sehingga sangat dipengaruhi oleh panjang vektor atau jumlah kata dalam dokumen (Dzikri et al., 2023).

Perhitungan *Cosine similarity* terdiri dari beberapa fase. Praproses pertama berjalan pada kueri untuk mencocokkannya dengan basis data tertimbang menggunakan teknik TF-IDF (*Term Frequency-Inverse Document Frequency*). Vektor kueri dan dokumen kemudian diskalakan dan dikalikan satu sama lain. Selain itu, mengkuadratkan bobot masing-masing menghitung panjang vektor kueri dan dokumen dengan menggabungkan hasil kuadrat dan kemudian

memperolehakarnya. Dengan membagi hasil perkalian skalar dengan panjang vektor, diperoleh nilai kesamaan antara kueri dan dokumen. Biasanya digunakan dalam banyak aplikasi seperti kategorisasi teks dan penambangan teks, pendekatan *Cosine similarity* memiliki kelebihan dalam kesederhanaan, efisiensi, dan kejelasan pemahaman. Dalam aplikasi praktis, nilai bobot kata dari setiap konten dapat memengaruhi hasil klasifikasi; di mana materi yang tidak dikategorikan akan memiliki bobot kata dan kesamaan yang dihitung dengan konten yang kategorinya diketahui.

Rumus perhitungan *cosine similarity* (Asriyani Arsad et al., 2024):

$$\cos \alpha = \frac{A \cdot B}{|A||B|} = \frac{\sum_{i=1}^{n} A_i \, x \, B_i}{\sqrt{\sum_{i=1}^{n} (Ai)^2 x} \sqrt{\sum_{i=1}^{n} (Bi)^2}} \qquad (2.14)$$

Keterangan:

A = Vektor A, yang akan dibandingkan kemiripannya

B = Vektor B, yang akan dibandingkan kemiripannya

A•B = dot product antara vektor A dan vektor B

$|A|$ = Panjang Vektor A

$|B|$ = Panjang Vektor B

$|A||B|$ = cross product antara $|A|$ dan $|B|$

2.2.9 *K-Nearest Neighbor* (KNN)

K-Nearest Neighbor (KNN) merupakan salah satu metode pengambilan keputusan dengan menggunakan *Supervisored Learning* dimana hasil data masukan baru diklasifikasikan berdasarkan nilai terdekatnya Uddin et al., (2022). Algoritma KNN merupakan metode yang menggunakan algoritma terbimbing yang termasuk dalam kelompok pembelajaran berbasis *instance* dan juga teknik pembelajaran malas dengan mencari kelompok k objek pada data latih yang terdekat (serupa) dengan objek pada data baru atau pengujian data. KNN bertujuan untuk mengklasifikasikan objek berdasarkan atribut dan sampel pelatihan yang hanya berdasarkan memori dengan memberikan titik kueri, maka akan ditemukan sejumlah k objek atau yang paling dekat dengan titik kueri (Xing & Bei, 2020).

1. Skor set.
2. Menghitung jarak antara data evaluasi dengan seluruh data.

Pada konteks data teks, jenis metode yang digunakan untuk menghitung kesamaan data adalah *cosine similarity* sesuai pada persamaan 2.14, semakin besar skor *similarity* data semakin dekat (mirip). Jadi tidak perlu menghitung jarak secara eksplisit, tapi cukup menggunakan skor *similarity* sebagai pembalik jarak**.**

3. Mengurutkan hasil dari skor similarity yang telah terbentuk.
   Mengurutkan skor *similarity* tersebut dari yang paling tinggi ke paling rendah, karena kita ingin mencari tetangga paling mirip**.**
4. Menentukan k data terdekat.
5. Mengumpulkan kelas yang sama atau sesuai.
6. Menetapkan jumlah kelas terbanyak kelas data yang akan dievaluasi.

Terdapat banyak cara untuk mengukur jarak antara kedekatan data baru dengan data lama, namun yang digunakan dalam penelitian ini adalah *cosine-similarity* yaitu suatu metode untuk menghitung kemiripan antara dua objek yang dinyatakan dalam dua vektor dengan menggunakan kata kunci suatu dokumen sebagai ukurannya.

2.2.10 Evaluasi

Evaluasi klasifikasi didasarkan pada jumlah *record* uji yang diprediksi dengan benar dan tidak benar oleh model. Jumlah *record* uji tersebut dapat diorganisir dalam bentuk *Confusion Matrix*, yang memungkinkan untuk melihat secara rinci hasil prediksi yang benar (*True Positives* dan *True Negatives*) serta hasil prediksi yang salah (*False Positives dan False Negatives*). *Confusion Matrix* adalah tabel yang digunakan untuk mencatat kinerja model klasifikasi Hasnain dkk., (2020) yang disajikan pada Tabel 2.1

Tabel 2.1 *Confusion Matrix* Multi Kelas

		Actual Classification				
Predicted classification	***Classess***	C_1	C_2	C_3	…	C_n
	C_1	T_1	F_{12}	F_{13}	…	F_{1n}
	C_2	T_{21}	T_2	F_{23}	…	F_{2n}
	C_3	T_{31}	F_{32}	T_{33}	…	F_{3n}
	…	…	…	…	…	...
	C_n	F_{n1}	F_{n2}	F_{n3}	…	T_n

Terdapat empat istilah yang merupakan representasi dari *Confusion Matrix* sebagai berikut Markoulidakis dkk., (2021) :

a. *True Positive* (TP) : Jumlah data yang bernilai positif dan diprediksi benar sebagai positif, dengan formulasi (2.20)

$$TP_{c_n} = |\{x_i | \hat{y} = y_i = c_n\}| \quad (2.1)$$

b. *True Negative* (TN): Jumlah data yang bernilai negatif dan diprediksi benar sebagai negatif, dengan formulasi (2.21)

$$TN_{c_n} = |\{x_i | \hat{y} \neq c_n \; dan \; y_i \neq c_n\}| \quad (2.2)$$

c. *False Positive* (FP): Jumlah data yang bernilai negatif tetapi diprediksi sebagai positif, dengan formulasi (2.22)

$$FP_{c_n} = |\{x_i | \hat{y} = c_n \; dan \; y_i \neq c_n\}| \quad (2.3)$$

d. *False Negative* (FN): Jumlah data yang bernilai positif tetapi diprediksi sebagai negatif, dengan formulasi (2.23)

$$FN_{c_n} = |\{x_i | \hat{y} \neq c_n \; dan \; y_i = c_n\}| \quad (2.4)$$

Keterangan Variabel:

1. x_i: Data ke-i.
2. y_i: Label aktual dari data ke-i.
3. $\hat{y}$: Prediksi label dari data ke-i.
4. cn: Kelas positif yang diinginkan.

e. |·|: Jumlah elemen dalam himpunan.

Pada Tabel 2.1 menampilkan kelas yang diprediksi dan kelas actual yang digunakan untuk memvisualisasikan kinerja masing-masing kelas. Berikut ini metrik kinerja yang digunakan dalam *Confusion Matrix* Multi Kelas *:*

a. *Averaged Accuracy* menggambarkan seberapa akurat model dalam mengklasifikasikan dengan benar (Salauddin Khan dkk., 2023). Dengan kata lain, *Averaged accuracy* merupakan tingkat kedekatan nilai prediksi dengan nilai aktual (sebenarnya). Nilai *Averaged Accuracy* dapat diperoleh dengan

$$Averaged\ Accuracy = \frac{\Sigma_{i=1}^{k} \frac{tp_i + tn_i}{tp_i + tn_i + fp_i + tn_i}}{k} \quad (2.5)$$

b. Averaged Error Rate menggambarkan sebarapa sering model membuat kesalahan dalam klasifikasi (Salauddin Khan dkk., 2023). Dengan kata lain,

Averaged Error Rate adalah tingkat kesalahan rata-rata yang dibuat oleh model ketika membandingkannilai prediksi dan nilai actual.

$$Averaged\ Error\ Rate = \frac{\Sigma_{i=1}^{k} \frac{fp_i + fn_i}{tp_i + fp_i + tn_i + fn_i}}{k} \quad (2.6)$$

c. *Averaged Precision* (pM) menggambarkan akurasi antara data yang diminta dengan hasil prediksi yang diberikan oleh model (Salauddin Khan dkk., 2023).

$$Averaged\ Precision\ (pM) = \frac{\Sigma_{i=1}^{k} \frac{fp_i + fn_i}{tp_i + tn_i + fp_i + tn_i}}{k} \quad (2.7)$$

d. *Averaged Recall* (rM) menggambarkan keberhasilan model dalam menemukan kembali sebuah informasi (Salauddin Khan dkk., 2023).

$$Averaged\ Recall\ (rM) = \frac{\Sigma_{i=1}^{k} tp_i}{\Sigma_{i=1}^{k} (tp_i + fn_i)} \quad (2.8)$$

e. *F-1 Score (Averaged F-Measure)* secara definisi adalah *harmonic mean* dari *Averaged Precision* dan *Averaged Recall* yang menggambarkan perbandingan rata-rata *Averaged Precision* dan *Averaged Recall* yang dibobotkan (Riyono dkk., 2022).

$$F\text{-}1\ Score = \frac{(2 * pM * rM)}{(pM + rM)} \quad (2.9)$$

Keterangan variable:

1. k: Jumlah kelas atau kelompok yang dievaluasi.
2. tp_i : *True Positive* untuk kelas ke-i.
3. tn_i : *True Negative* untuk kelas ke-i.
4. fp_i : *False Positive* untuk kelas ke-i.
5. fn_i : *False Negative* untuk kelas ke-i.
6. pM: *Averaged Precision*.
7. rM: *Averaged Recall.*