Maliki Interdisciplinary Journal (MIJ) eISSN: 3024-8140
Volume 2, Issue June, 2024 pp. 2052-2065 http://urj.uin-malang.ac.id/index.php/mij/index

Klasifikasian abstrak paper menggunakan algoritma KKN

Farid Ardianto[1]*, Khadijah Fahmi Hayati Holle, S.Kom., M.Kom[2]
[1,] Program Studi Teknik Informatika, Universitas Islam Negeri Maulana Malik Ibrahim Malang
*e-mail: *200605110101@student.uin-malang.ac.id*

Kata Kunci:
K-Nearest Neighbor, klasifikasi abstrak, metrik jarak, evaluasi akurasi.

Keywords:
K-Nearest Neighbor, abstract classification, distance metrics, accuracy evaluation.

A B S T R A K

Algoritma K-Nearest Neighbors (KNN) digunakan dalam penelitian ini untuk mengategorikan abstrak penelitian menurut topiknya. Penelitian ini, dibuat untuk membantu penulis dan peneliti dalam memilih tempat publikasi jurnal yang sesuai dengan topik dalam abstrak. Dataset dikumpulkan dengan cara manual melalui pembatasan dengan hanya mencari jurnal terindeks Sinta 2 untuk abstrak terkait dalam uji coba. Topik yang relevan kemudian diekstraksi dari abstrak menggunakan teknik pemrosesan bahasa alami (NLP). Abstrak dikategorikan ke dalam kelompok mata pelajaran tertentu menggunakan algoritma KNN. Penelitian ini, dilakukan dengan mengatur variabel dari parameter yang digunakan. Melalui penetapan nilai random_state, penentuan nilai n-neighbors, serta menentukan matriks jarak terbaik, yang akan menghasilkan presisi tinggi, recall, dan peringkat F1-score yang bagus. Presisi tinggi, recall, dan peringkat F1-score dalam evaluasi kinerja sistem digunakan sebagai penunjuk dari seberapa baik mengategorikan topik penelitian dan pengukur tingkat ke efisiennya.

A B S T R A C T

The K-Nearest Neighbors (KNN) algorithm was used in this study to categorize research abstracts according to their topic. This research was created to assist writers and researchers in choosing a place for journal publication that is appropriate to the topic in the abstract. The dataset was collected manually through restrictions by only searching Sinta 2-indexed journals for related abstracts in the trial. The relevant topics are then extracted from the abstract using natural language processing (NLP) techniques. Abstracts are categorized into certain subject groups using the KNN algorithm. This research was conducted by adjusting the variables of the parameters used. By setting random_state values, determining n-neighbors values, and determining the best distance matrix, which will produce high precision, recall, and a good F1-score rating. High precision, recall, and F1-score ratings in system performance evaluation are used as indicators of how well research topics are categorized and as a measure of efficiency.

Pendahuluan

Publikasi ilmiah memiliki dampak yang signifikan terhadap bagaimana ide, penemuan, dan penelitian dikomunikasikan di berbagai tempat ilmiah (Björk, 2017). Langkah krusial yang perlu mendapat perhatian khusus dalam penyusunan artikel ilmiah adalah pemilihan jurnal publish tepat yang memerlukan perhatian khusus (Gasparyan et al., 2013). Pemilihan abstrak berbasis topik, yang melibatkan evaluasi abstrak penelitian

untuk memilih jurnal terbaik serta sesuai, merupakan komponen penting dari pemilihan jurnal (Bornmann & Daniel, 2010).

Namun, tugas pemilihan abstrak berbasis topik dapat menjadi tantangan dan memakan waktu lama karena semakin banyaknya tempat publikasi yang beragam dan kemajuan pesat dalam banyak kategori domain ilmiah (Gusenbauer et al., 2020). Untuk menentukan tempat jurnal yang relevan sebagai sarana publikasikan temuan penelitian mereka, diperlukan sistem tambahan yang dapat membantu peneliti dalam proses seleksi abstrak berbasis topik (Mongeon & Paul-Hus, 2016).

Melalui seleksi abstrak berbasis topik, diharapkan mempunyai tujuan dari untuk menciptakan sistem bantu kepada pihak peneliti. Yang dapat berguna untuk menentukan tempat jurnal dipublikasikan setelah membuat artikel ilmiah. Sistem ini dapat diintegrasikan dengan database jurnal, sehingga menampilkan berbagai saran untuk jurnal berbahasa Indonesia yang terindeks di Sinta 2 dengan menganalisis abstrak penelitian, mengidentifikasi subjek yang paling relevan, dan menggunakan kemampuan Natural Language Processing (NLP) (Arora, 2020).

Tingkat presisi pemetaan subjek dari literatur ilmiah akan menjadi pertimbangan utama dalam mengembangkan sistem bantu ini (Björk, 2017; Blei et al., 2003). Sistem bantu ini diharapkan dapat merekomendasikan jurnal yang bersangkutan secara akurat sehingga penulis atau peneliti dapat memilih publikasi terbaik untuk menerbitkan karyanya. Ketersediaan jurnal berbahasa Indonesia dan adanya indeksasi di Sinta 2 menjadi faktor krusial yang harus diperhatikan dalam situasi ini.

Sistem bantu yang akan dikembangkan akan memiliki kompetensi khusus dalam penggalian informasi, analisis konten, dan pemetaan topik dengan akurasi tinggi dari teks ilmiah berbahasa Indonesia, dengan mempertimbangkan fokus penelitian Anda pada bidang Natural Language Processing (NLP) dan preferensi Anda untuk jurnal berbahasa Indonesia yang terindeks di Sinta 2. Berdasarkan analisis topik yang menyeluruh, sistem pendukung ini akan membantu penulis atau peneliti memilih jurnal terbaik untuk menerbitkan karyanya.

Untuk mencapai tingkat akurasi yang memadai dalam pemetaan topik, saya akan memeriksa lebih lanjut perencanaan dan implementasi sistem pelengkap ini dalam pekerjaan ini dan melakukan evaluasi kinerja (Blei et al., 2003). Diharapkan bahwa hasil temuan penelitian ini dapat membantu penulis atau peneliti dalam memilih jurnal yang sesuai untuk menerbitkan karya mereka berdasarkan topik yang bersangkutan.

Metode Penelitian

Pada bab ini, saya akan menjelaskan metode penelitian yang digunakan dalam penelitian ini secara lebih rinci, termasuk desain penelitian, sumber data, teknik analisis data, dan prosedur pengumpulan data.

A. Pengumpulan Data

Pada tahap pengumpulan data, pengumpulan data secara manual dilakukan dengan mencari secara manual setiap abstrak penelitian yang relevan dengan penelitian ini. Informasi akan dikumpulkan dari jurnal yang termasuk dalam indeks Sinta 2. Saat

mengumpulkan data, akan dipastikan bahwa abstrak dalam bahasa Indonesia dan berkaitan dengan lima class mata pelajaran yang berbeda, antara lain politik, pendidikan anak, kesehatan, dan olahraga, serta teknik informatika.

B. Data Pre - processing

Data pre-processing mengikuti pengumpulan data abstrak penelitian. Pra-pemrosesan data berupaya menyiapkan data untuk analisis lebih lanjut dengan membersihkan dan mengaturnya. Langkah-langkah yang terlibat dalam pra-pemrosesan data adalah sebagai berikut:

1) Pembersihan teks

Menghilangkan tanda baca yang tidak perlu, karakter khusus, dan fitur lain yang tidak berpengaruh dalam klasifikasi dari abstrak penelitian.

2) Normalisasi teks

Normalisasikan teks dengan mengubah semua huruf besar menjadi huruf kecil, hilangkan variasi kata seperti lemmatisasi dan stemming, dan hilangkan stopwords (kata-kata umum yang tidak berkontribusi signifikan pada topik pemetaan) (Pradana & Hayaty, 2019).

3) Representasi vektor

Teknik klasifikasi KNN dapat digunakan untuk mengubah bahasa abstrak menjadi representasi vektor numerik. Term Frequency - Inverse Document Frequency, atau biasa disebut TF-IDF, merupakan salah satu pendekatan yang sering digunakan yang akan diimplementasikan pada penelitian ini.

C. Pemodelan Klasifikasi Algoritma KNN:

Langkah selanjutnya pada tahap ini adalah membuat model untuk klasifikasi dengan teknik K-Nearest Neighbors (KNN) setelah pra-pemrosesan informasi selesai. Abstrak penelitian akan dikategorikan ke dalam kategori topik yang relevan menggunakan algoritma KNN, antara lain teknik informatika, politik, pendidikan anak, kesehatan, dan olahraga (Purnomo & Hartanto, 2017).

Saat menggunakan algoritma KNN untuk pemodelan klasifikasi, prosedur berikut akan diikuti:

1. Pengelompokan data

Data yang diproses sebelumnya akan dibagi menjadi dua kategori: data pelatihan dan data pengujian. Model klasifikasi dilatih menggunakan data pelatihan, dan keefektifannya dievaluasi menggunakan data uji (Hastuti, 2016).

2. Pelatihan model:

Model KNN kemudian dilatih menggunakan data pelatihan yang dibuat sebelumnya setelah parameter yang sesuai ditetapkan. Abstrak penelitian dipetakan ke dalam kelompok topik tertentu sebagai bagian dari prosedur pelatihan ini (Kusumawardani & Mariana, 2019).

3. Menguji model:

Menggunakan data uji yang telah disediakan sebelumnya, model KNN akan diuji setelah dilatih. Menggunakan model yang telah dikembangkan sebelumnya, kategori topik dalam abstrak penelitian dipetakan selama pengujian. Keakuratan pemetaan topik yang dilakukan oleh model kemudian akan dinilai menggunakan metrik evaluasi seperti presisi, recall, dan F1-score (Hidayat & Nurhayati, 2019).

4. Menentukan ukuran pengacakan data

Pada tahapan ini, akan dilakukan pencarian nilai pengacakan dari dataset pelatihan sekaligus dataset pengujian. Nilai pengacakannya sendiri akan berlaku seterusnya, yang membuat data teracak, akan tetapi dalam setiap pengujiannya nanti jika memiliki besaran nilai yang sama, maka akan terjadi pengacakan yang sama pula. Hal ini tentunya berdampak pada persentase hasil yang cenderung stabil, dikarenakan data akan diacak dengan sama jika menggunakan nilai yang sama. Pengacakan tertentu juga mempengaruhi perolehan persentase klasifikasi yang maksimal juga. Oleh karena itu diperukkan nilai yang paling tepat untuk melanjutkan tepat selanjutnya. Agar nanti bisa diperoleh nilai yang sama dengan tetap mempertahankan faktor pengacakannya. Hal ini juga membuat berlakunya pengeluaran hasil persentase yang adil pada setiap percobaan proses selanjutnya. Dengan nilai random_state yang sama dapat digunakan untuk merepilkasikan hasil tersebut serta dapat digunakan untuk memastikan pembagian data yang sama untuk proses yang lainnya dengan nilai yang sama (Pedregosa et al., 2011).

5. Penentuan Parameter Optimal untuk Algoritma KNN

Jumlah tetangga terdekat (k), ukuran jarak, dan mekanisme voting hanyalah beberapa faktor yang harus ditetapkan untuk algoritma KNN. Pemilihan parameter ideal algoritma KNN akan dilakukan pada titik ini (Hanani et al., 2018). Percobaan akan dilakukan berulang kali agar bisa mendapatkan nilai maksimumnya.

6. Penentuan Metrik Jarak

Sangat penting untuk memilih metrik jarak yang tepat dalam algoritma k-Nearest Neighbors (k-NN) untuk pengklasifikasian data. Metrik jarak digunakan untuk menghitung jarak antara titik-titik data dalam ruang fitur. Dalam penelitian ini, menggunakan pertimbangan beberapa metrik jarak yang umum digunakan, yaitu:

a. Jarak Euclidean

Metrik jarak Euclidean mengukur jarak geometris antara dua titik di ruang Euclidean. Berikut ini adalah rumus menghitung jarak Euclidean:

$$d(A, B) = \text{sqrt}((x_2 - x_1)^2 + (y_2 - y_1)^2 + \ldots + (n_2 - n_1)^2)$$

b. Jarak Manhattan

Metrik jarak Manhattan menguantifikasi varians relatif total di antara setiap pasangan koordinat dan juga disebut sebagai jarak blok kota. Berikut rumus jarak Manhattan:

Rumus untuk d(A, B) adalah:

|x2 - x1| + |y2 - y1| +... + |n2 - n1|

c. Kesamaan cosinus

Dalam ruang fitur, kemiripan antara dua vektor dinilai menggunakan metrik jarak kosinus. Rumus berikut digunakan untuk menentukan jarak kosinus, yang merupakan kosinus sudut yang dibentuk oleh dua vektor, yaitu: Kemiripan(A, B) sama dengan (A B) / (A * B).

d. Jarak Minkowski

Metrik jarak Manhattan dan Euclidean digeneralisasikan dalam metrik jarak Minkowski. Sensitivitas terhadap variasi pada setiap dimensi dikontrol oleh parameter p. Berikut ini adalah rumus perhitungan jarak Minkowski:

Rumus untuk d(A, B)	(|x2 - x1|p + |y2 - y1|p +... + |n2 - n1|p)(1/p)

e. Jarak Chebyshev

Metrik ini menghitung kemungkinan pemisahan terbesar antara dua koordinat mana pun. Berikut rumus menghitung jarak Chebyshev:

maks(|x2 - x1|, |y2 - y1|,..., |n2 - n1|)

f. Jarak Hamming

Untuk membandingkan dua string dengan panjang yang sama, seseorang menggunakan metrik jarak Hamming. Jumlah tempat yang berbeda antara dua string dihitung menggunakan metrik ini.

g. Kemiripan dengan Jaccard

Saat membandingkan dua set data, metrik jarak Jaccard digunakan. Jarak Jaccard ditentukan dengan membagi ukuran gabungan dari dua set dengan rasio ukuran irisan dari dua set.

Berdasarkan temuan eksperimental dan penilaian kinerja klasifikasi, pada percobaan ini bertujuan untuk memilih ukuran jarak terbaik dalam penelitian ini.

D. Evaluasi Kinerja Sistem Bantu:

Evaluasi kinerja sistem bantu yang dirancang merupakan tahap terakhir. Data tes yang disiapkan sebelumnya digunakan untuk evaluasi. Berdasarkan temuan klasifikasi sistem, metrik evaluasi termasuk presisi, daya ingat, dan skor F1 akan dihitung. Peneliti juga dapat melakukan pengujian manual sebagai bagian dari proses evaluasi untuk mengukur seberapa baik sistem melakukan pemetaan topik yang benar dari abstrak penelitian (Rachmawati & Prasetio, 2019). Metodologi ini digunakan untuk menentukan seberapa baik suatu sistem dapat melakukan pemetaan topografi yang akurat dari penelitian abstrak. Evaluasi dilakukan dengan menggunakan data yang dikumpulkan sebelumnya, dengan hasil klasifikasi sistem dibandingkan dengan label yang jelas pada data.

Presisi (precision) mengukur beberapa sistem yang andal saat mengklasifikasikan penelitian abstrak ke dalam kategori tingkat atas yang sesuai. Asumsikan bahwa sistem mampu mengenali tulisan yang benar-benar abstrak, termasuk kategori yang relevan secara topikal. F1-score adalah rasio antara presisi dan perolehan yang memberikan indikasi efisiensi sistem secara keseluruhan dalam melakukan topik pemecatan secara bersamaan. Selain itu, evaluasi juga dapat mengarah pada pengujian manual oleh peneliti di mana peneliti secara sistematis menilai hasil sistem klasifikasi terhadap berbagai survei sampel penelitian. Hal ini dimaksudkan untuk memperoleh pengetahuan yang lebih mendalam tentang ketahanan sistem saat melakukan pemetakan topik yang akurat.

Tabel 1. Rumus

Akurasi = (Jumlah Prediksi Benar) / (Jumlah Total Prediksi)
Presisi = (Jumlah True Positive) / (Jumlah True Positive + Jumlah False Positive)
Recall = (Jumlah True Positive) / (Jumlah True Positive + Jumlah False Negative)
*F1-Score = 2 * (Presisi * Recall) / (Presisi + Recall)*

1. Akurasi

Mengevaluasi seberapa baik suatu sistem dapat membuat klasifikasi yang akurat. Saat menggunakan metode KNN untuk klasifikasi, akurasi dihitung dengan membagi jumlah total prediksi dengan jumlah prediksi yang benar.

2. Presisi

Mengukur seberapa sukses sistem mengidentifikasi sampel yang bersangkutan. Dengan membagi jumlah total positif sejati (sampel yang dikategorikan positif dengan benar) dengan jumlah total positif sejati ditambah positif palsu (sampel yang salah diklasifikasikan sebagai positif), akurasi dihitung dalam konteks klasifikasi menggunakan KNN.

3. Recall

Sering disebut sebagai sensitivitas, menilai seberapa baik sistem mengidentifikasi semua sampel yang benar-benar termasuk dalam kelas target. Jumlah true positive ditambah jumlah true positive ditambah false negative (sampel yang salah dikategorikan sebagai negatif) digunakan untuk menghitung perolehan dalam klasifikasi KNN.

4. Skor-F1(F1-Score)

Rata-rata harmonik antara daya ingat dan presisi dikenal sebagai skor-F1. Ini memberikan gambaran luas tentang kinerja sistem dalam mengklasifikasikan dengan keseimbangan antara daya ingat dan presisi. Dengan menggunakan rumus tersebut, skor F1 ditentukan.

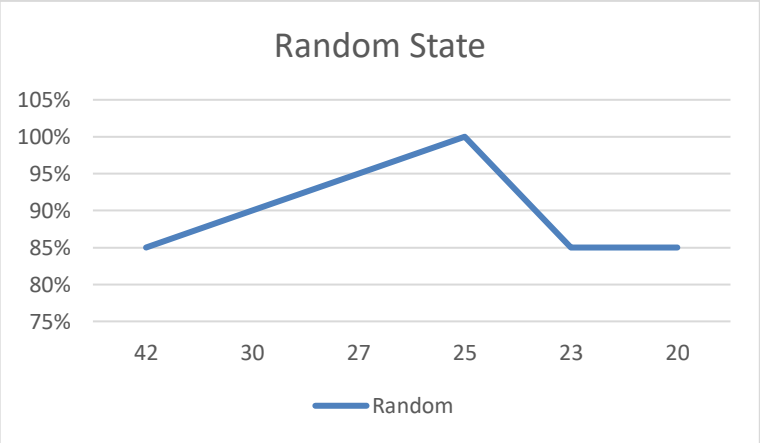
Hasil dan Pembahasan

Hasil dan diskusi dari penerapan metode yang telah dijelaskan sebelumnya akan diuraikan pada bagian ini. Untuk menilai kemampuan algoritma K-Nearest Neighbor (KNN) dalam klasifikasi dan pemetaan data, tahapan pengelompokan data, penentuan ukuran pengacakan data, penentuan parameter, pelatihan model, dan pengujian model telah dilakukan. Dalam pengujian model, saya menggunakan metrik evaluasi seperti skor F1, presisi, dan recall untuk mengevaluasi keakuratan pemetaan topik pada abstraksi penelitian. Dalam pembahasan berikutnya, saya akan menjelaskan hasil eksperimen, memberikan penjelasan yang relevan, dan membahas hasil yang menarik.

A. Parameter (random_state) untuk Konsistensi Pengacakan Data

Subbagian ini membahas penggunaan parameter random_state untuk mencapai konsistensi pengacakan data. Nilai random_state digunakan untuk memisahkan data menjadi data pelatihan dan pengujian, dan untuk mengacak kembali data yang sama untuk setiap percobaan. Saya mencoba berbagai nilai random_state dalam eksperimen ini untuk melihat bagaimana hal itu memengaruhi hasil klasifikasi. Hasil percobaan yang dilakukan adalah sebagai berikut:

Diagram 1.1 Random State



1. Random_state = 40: Pada percobaan ini, diperoleh akurasi sebesar 85%.
2. Random_state = 30: Percobaan ini menghasilkan akurasi sebesar 90%.
3. Random_state = 27: Dengan menggunakan nilai random_state 27, akurasi meningkat menjadi 95%.
4. Random_state = 25: Percobaan ini menghasilkan akurasi mencapai 100%.
5. Random_state = 23: Pada percobaan ini, akurasi sebesar 85% diperoleh.
6. Random_state = 20: Dalam percobaan terakhir, akurasi juga sebesar 85%.

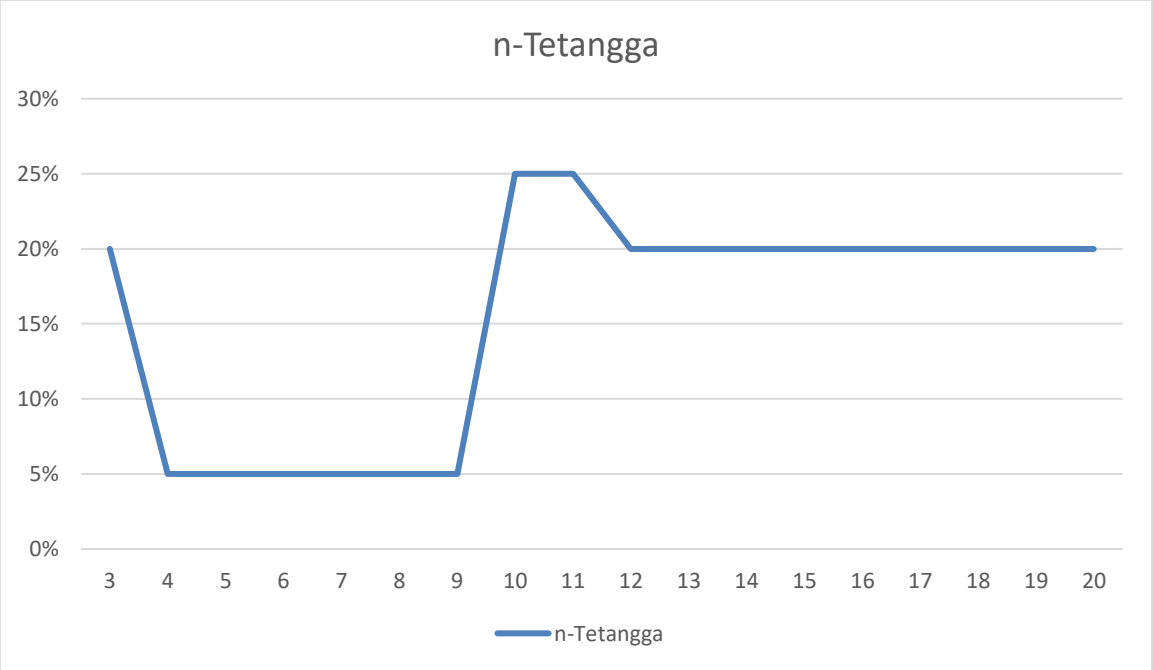
Dari hasil percoba ketika menggunakan nilai random_state yang berbeda, hasil klasifikasi berbeda. Misalnya, dalam percobaan dengan random_state = 25, akurasi mencapai 100%, sementara dalam percobaan dengan random_state = 23 dan 20, akurasi hanya 85%. Ini menunjukkan bahwa memilih nilai random_state yang tepat dapat membantu menghasilkan hasil klasifikasi yang lebih baik.

Dalam penelitian selanjutnya, saya akan menggunakan nilai random_state = 30, yang menghasilkan akurasi sebesar 90%. Dengan menggunakan akurasi yang tidak maksimal 100%, saya dapat menentukan apakah ada metode yang lebih baik dari metode yang telah digunakan sebelumnya. Serta penggunaan akurasi yang tidak terlalu rendah membuat sistem tidak menampilkan hasil yang salah. Hal ini memungkinkan saya untuk menampilkan hasil klasifikasi yang optimal dan menemukan peningkatan performa dari metode yang digunakan dengan mempertahankan konsistensi dengan menggunakan nilai random_state = 30.

B. Penentuan Parameter Optimal untuk Algoritma KNN

Subbagian ini akan memberikan penjelasan tentang penentuan parameter yang paling ideal untuk algoritma K-Nearest Neighbor (KNN). Jumlah tetangga terdekat yang ideal untuk klasifikasi (n_neighbors) akan dipilih sebagai parameter. Pada penelitian ini telah melakukan beberapa percobaan dengan variasi nilai n_neighbors dan mencatat persentase akurasi. Ini adalah hasil penelitian yang telah dilakukan:

Diagram 1.2 n-Tetangga



Penelitian ini melakukan eksperimen dengan berbagai nilai n_neighbors dari 3 hingga 20. Data dibagi menjadi data pelatihan dan pengujian dengan menggunakan metode pengacakan yang konsisten dengan random_state=30. Data pelatihan digunakan untuk melatih model KNN, dan data pengujian digunakan untuk mengujinya untuk mendapatkan persentase akurasi.

Pada percobaan pertama, dengan n_neighbors = 3, diperoleh akurasi sebesar 20%, menunjukkan bahwa model KNN memiliki performa yang terbatas dalam melakukan klasifikasi dengan hanya mempertimbangkan tiga tetangga terdekat. Pada percobaan berikutnya, dengan n_neighbors = 4, 5, dan 6, diperoleh akurasi sebesar 5%, menunjukkan bahwa dengan jumlah tetangga terdekat yang lebih sedikit, model KNN mengalami kesulitan dalam melakukannya. Penelitian ini juga terus melakukan

percobaan dengan mulai n_neighbors = 7 hingga n_neighbors = 9, dan percobaan tersebut mendapatkan hasil dengan akurasi 5% untuk setiap nilai n_neighbors tersebut. Hal ini menunjukkan bahwa kinerja model KNN cenderung stabil dan tidak berubah secara signifikan dalam rentang n_neighbors tersebut.

Namun, pada percobaan selanjutnya menunjukkan peningkatan yang signifikan dalam akurasi pada percobaan dengan n_neighbors = 10 dan n_neighbors = 11. Kedua percobaan menunjukkan akurasi sebesar 25%. Hal ini menunjukkan bahwa model KNN dapat mengklasifikasikan data dengan lebih akurat dengan mempertimbangkan sepuluh atau sebelas tetangga terdekat.

Selain itu, pada percobaan dengan n_neighbors = 12 hingga n_neighbors = 20 telah mengalami penurunan, yang mana memperoleh hasil akurasi sebesar 20%. Meskipun ini tidak mencapai tingkat akurasi yang sama dengan percobaan dengan n_neighbors = 10 dan n_neighbors = 11, namun tetap berhasil melakukan klasifikasi dengan baik.

Berdasarkan hasil penelitian di atas, saya sebagai peneliti dapat menyimpulkan bahwa nilai n_neighbors mempengaruhi akurasi klasifikasi algoritma KNN. Dalam penelitian ini, menemukan bahwa nilai n_neighbors = 10 dan n_neighbors = 11 memberikan tingkat akurasi tertinggi, yaitu 25%. Oleh karena itu, jika berpacuan pada percobaan sebelumnya menyarankan untuk menggunakan nilai n_neighbors = 10 atau n_neighbors = 11 sebagai parameter yang paling cocok untuk algoritma KNN yang menunjukkan hasil yang optimal.

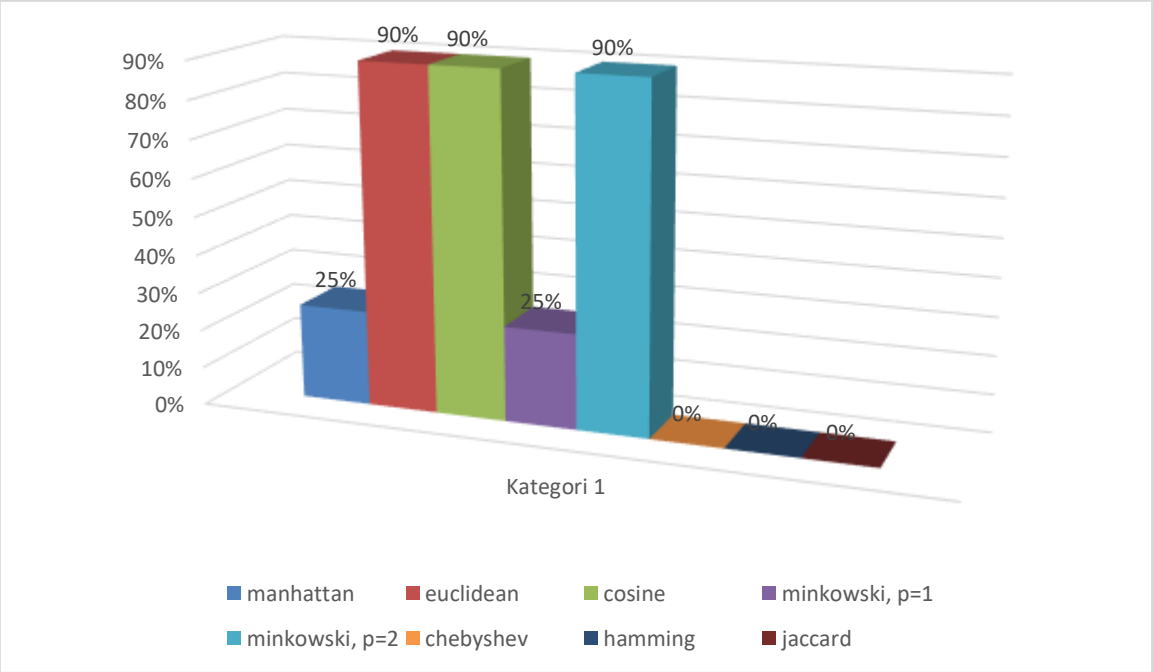
Dalam algoritma KNN, penentuan parameter ideal sangat penting karena dapat mempengaruhi kinerja dan akurasi model. Dengan memilih parameter yang tepat, kita dapat meningkatkan kualitas klasifikasi dan mendapatkan hasil yang lebih baik. Namun, perlu diingat bahwa parameter ideal dapat berbeda untuk setiap dataset, sehingga penelitian lebih lanjut diperlukan untuk menemukan parameter yang paling sesuai untuk dataset lain (Heryadi & Wahyono, 2020).

C. Matriks yang Sesuai

Beberapa metrik jarak yang sering digunakan oleh teknik k-Nearest Neighbors (k-NN) untuk klasifikasi data diperhitungkan dalam penelitian ini. Jarak Euclidean, jarak Manhattan, kesamaan cosinus, jarak Minkowski, jarak Chebyshev, jarak Hamming, dan kemiripan Jaccard adalah metrik jarak yang saya gunakan untuk mengevaluasi kinerja model k-NN.

Hasil penelitian ini menunjukkan bahwa metrik jarak yang digunakan memengaruhi seberapa baik kinerja model k-NN. Sebagai konsekuensi dari pengujian dan evaluasi kinerja klasifikasi yang telah dilakukan, menunjukkan metrik jarak ditemukan sebagai berikut:

Diagram 1.3 Perbandingan Metode



1. Jarak Manhattan

Percobaan menghasilkan pencapaian tingkat akurasi 25% saat menggunakan metrik jarak Manhattan. Metrik ini sesuai ketika ada hipotesis tentang hubungan linier antara elemen-elemen dalam kumpulan data atau ketika karakteristik dalam data memiliki skala yang bervariasi.

2. Jarak Euclidean

Tingkat akurasi 90% disediakan dengan menggunakan metrik ini. Metrik rentang paling populer di k-NN adalah metrik Euclidean ini. Metrik Euclidean menghitung pemisahan geometris dalam ruang Euclidean antara dua titik.

3. Cosine similarity

Metrik ini memiliki tingkat akurasi 90% dan digunakan untuk mengukur jarak. Ketika karakteristik dalam kumpulan data memiliki arah atau orientasi yang signifikan, metrik ini sangat membantu. Kosinus sudut antara setiap vektor dalam ruang fitur dihitung menggunakan ukuran ini.

4. Jarak Minkowski dengan p=1

Percobaan menghasilkan pencapaian tingkat akurasi 25% menggunakan ukuran jarak Minkowski dengan p=1. Metrik ini merupakan perluasan dari metrik Manhattan dan Euclidean, dengan tingkat kepekaan terhadap variasi tiap dimensi dikontrol oleh parameter p.

5. Jarak Minkowski dengan p=2

Percobaan menghasilkan pencapaian tingkat akurasi 90% dengan pengukuran jarak Minkowski dengan p=2. Dengan tingkat sensitivitas yang bervariasi, metrik ini juga merupakan perluasan dari metrik jarak Manhattan dan Euclidean.

6. Pengukuran jarak Chebyshev, Hamming, dan Jaccard tidak valid untuk masukkan renggang

Karena akurasinya yang tidak signifikan, yaitu 0%, Metrik jarak Hamming digunakan untuk membandingkan dua string dengan panjang yang sama, sedangkan metrik nilai Chebyshev mengukur jarak maksimum antara setiap pasangan koordinat. Kedua set data dikontraskan menggunakan metrik kesamaan Jaccard. Namun dalam keseluruhan setting investigasi ini, ketiga kriteria tersebut tidak memberikan hasil klasifikasi data yang akurat.

Temuan penelitian membawa pada kesimpulan bahwa, mengingat kumpulan data yang digunakan, jarak antara metrik korelasi Euclid dan Cosine berkinerja terbaik. Namun, berdasarkan sifat informasi dan masalah yang dihadapi, ukuran jarak terbaik dapat dipilih secara berbeda.

D. Hasil Evaluasi Kinerja Sistem Bantu

Pada penelitian ini, saya mencoba menggunakan metrik Euclidean dan Cosine menggunakan parameter random_state=30 dan n_neighbors=10, guna mengevaluasi kinerja sistem tambahan. Tujuan dari tes ini adalah untuk mengukur seberapa efektif sistem tambahan dapat mengategorikan subjek menggunakan abstrak penelitian. Temuan evaluasi yang terkumpul ditampilkan dalam bentuk tabel bersama dengan ukuran penilaian lainnya, termasuk presisi, daya ingat, skor f1, dan akurasi.

Berapa banyak dari semua hasil baik yang diproyeksikan benar-benar terjadi diukur dengan presisi (presisi). Presisi tinggi berarti bahwa sebagian besar perkiraan sukses sistem tambahan akurat. Ingat menghitung jumlah kesuksesan sejati di antara semua kesuksesan sejati. Ingatan yang tinggi berarti sebagian besar topik dapat diidentifikasi dengan tepat oleh sistem asistif. Skor F1, yang memberikan penilaian menyeluruh atas kinerja sistem tambahan, adalah pengukuran rata-rata harmonik antara presisi dan daya ingat.

Tabel 2. Temuan penilaian kinerja sistem bantu adalah sebagai berikut:

	precision	recall	f1-score	support
Kesehatan	1.00	1.00	1.00	1
Olahraga	0.75	0.75	0.75	4
Pendidikan Anak	0.86	1.00	0.92	6
Politik	1.00	1.00	1.00	4
Teknik Informatika	1.00	0.80	0.89	5
accuracy			0.90	20
macro avg	0.92	0.91	0.91	20
weighted avg	0.91	0.90	0.90	20

Pada hasil dari temuan evaluasi di atas menyatakan dengan jelas, bahwa sistem tambahan mencapai tingkat akurasi 90%. Ini menunjukkan bahwa hingga 90% dari abstrak penelitian diidentifikasi secara efektif dan akurat. Saat membandingkan presisi, daya ingat, dan skor f1 dari setiap kelompok topik, kategori "Kesehatan" memiliki skor 1, menunjukkan bahwa sistem bantuan dapat mengategorikan topik ini dengan sempurna. Kategori "Olahraga" tampil cukup baik, sebagaimana dibuktikan dengan presisi, daya ingat, dan skor f1 0,75, meskipun ada beberapa kasus di mana sistem bantuan gagal mengklasifikasikan secara akurat. Kelompok "Pendidikan Anak" tampil baik dalam mengklasifikasikan subjek ini, dengan presisi 0,86, daya ingat 1,00, dan skor f1 0,92. Dengan presisi, daya ingat, dan skor f1 1, kategori "Politik" juga mencapai kinerja sempurna. Kelas "Teknik Informatika" berperforma baik, dengan presisi 1,00, daya ingat 0,80, dan skor f1 0,89, meskipun terkadang sistem bantuan dapat memberikan prediksi yang tidak akurat.

Presisi rata-rata (rata-rata makro), rata-rata perolehan kembali (rata-rata makro), dan skor rata-rata f1 (rata-rata makro) untuk evaluasi total masing-masing adalah 0,92, 0,91, dan 0,91. Evaluasi rata-rata tertimbang menghasilkan nilai presisi, daya ingat, dan skor f1 masing-masing sebesar 0,91, 0,90, dan 0,90. Ini menunjukkan bahwa sistem bantu secara umum baik dalam mengategorikan mata pelajaran menurut abstrak penelitian.

Temuan evaluasi memberikan gambaran umum tentang seberapa baik sistem bantu mengategorikan topik. Namun perlu diingat bahwa penilaian ini didasarkan pada kondisi tertentu yang digunakan dalam penelitian ini. Hasil yang berbeda dapat diperoleh dari analisis lebih lanjut menggunakan metrik atau parameter yang berbeda. Oleh karena itu, pengujian lebih lanjut diperlukan untuk menjamin keandalan sistem bantu dalam berbagai skenario dan lingkungan.

Kesimpulan dan Saran

Penelitian ini memperlihatkan bagaimana algoritma K-Nearest Neighbor (KNN) dapat digunakan untuk memetakan dan mengklasifikasikan data. Performa algoritme dievaluasi melalui sejumlah proses, termasuk agregasi data, pengacakan data acak, penentuan parameter, pelatihan model, dan pengujian model. Keakuratan pemetaan topik dalam abstrak penelitian dinilai dengan menggunakan kriteria evaluasi seperti skor F1, presisi, dan recall. Pada hasil klasifikasi bagaimana parameter random_state digunakan untuk pengacakan data. Menguji beberapa nilai random_state mengungkapkan bahwa pemilihan random_state berdampak besar pada akurasi kategorisasi. Hasil terbaik diperoleh dengan menggunakan nilai random_state 25 yang menghasilkan akurasi 100%, dan 30 yang menghasilkan akurasi 90%. Studi ini menemukan bahwa nilai random_state 30 akan menghasilkan hasil klasifikasi yang andal.

Tidak ketinggalan andilnya memilih jumlah tetangga (n_neighbours), yang merupakan parameter metode KNN yang paling penting. Keakuratan nilai n_neighbours yang diuji, yang berkisar antara 3 hingga 20, telah dilakukan pencatatan. Uji coba menunjukkan bahwa akurasi maksimum 25% dicapai dengan n_neighbours = 10 dan

n_neighbours = 11. Oleh karena itu, angka-angka ini ditentukan sebagai parameter yang memungkinkan algoritma KNN menghasilkan hasil terbaik.

Efektivitas model KNN diperiksa dalam kaitannya dengan berbagai metrik jarak, termasuk jarak Euclidean, jarak Manhattan, kesamaan Cosinus, jarak Minkowski, jarak Chebyshev, jarak Hamming, dan Jaccard. Ditemukan bahwa akurasi model sangat dipengaruhi oleh pemilihan metrik jarak. Jarak Euclidean dan kesamaan Cosinus ditemukan sebagai metrik jarak yang menunjukkan kinerja terbaik untuk dataset yang disediakan dalam penelitian ini. Dengan parameter seperti random_state=30 dan n_neighbors=10, metrik jarak Euclidean dan kesamaan Cosinus digunakan untuk menilai kinerja sistem tambahan. Algoritma mengklasifikasikan abstrak penelitian dengan akurasi rata-rata 90%, ditemukannya masalah yang berbeda memiliki tingkat presisi, daya ingat, dan skor f1 yang berbeda, dengan kategori "Kesehatan" menerima nilai tertinggi. Kategori "Olahraga" berhasil, tetapi kategori "Pendidikan Anak", "Politik", dan "Teknologi Informasi" juga berhasil, tetapi kadang-kadang dengan sedikit tidak tepatan.

Daftar Pustaka

Arora, G. (2020). *iNLTK: Natural Language Toolkit for Indic Languages*. 66–71. https://doi.org/10.18653/v1/2020.nlposs-1.10

Björk, B. C. (2017). Scholarly journal publishing in transition- from restricted to open access. *Electronic Markets*, *27*(2), 101–109. https://doi.org/10.1007/S12525-017-0249-2/FIGURES/1

Blei, D., Ng, A., research, M. J.-J. of machine L., & 2003, undefined. (2003). Latent dirichlet allocation. *Jmlr.Org*, *3*, 993–1022. https://www.jmlr.org/papers/volume3/blei03a/blei03a.pdf?ref=https://githubhelp.com

Bornmann, L., & Daniel, H. D. (2010). The Usefulness of Peer Review for Selecting Manuscripts for Publication: A Utility Analysis Taking as an Example a High-Impact Journal. *PLOS ONE*, *5*(6), e11344. https://doi.org/10.1371/JOURNAL.PONE.0011344

Gasparyan, A., … L. A.-J. of K., & 2013, undefined. (2013). Multidisciplinary bibliographic databases. *Synapse.Koreamed.Org*. https://synapse.koreamed.org/articles/1022051

Gusenbauer, M., methods, N. H.-R. synthesis, & 2020, undefined. (2020). Which academic search systems are suitable for systematic reviews or meta-analyses? Evaluating retrieval qualities of Google Scholar, PubMed, and 26 other. *Wiley Online Library*, *11*(2), 181–217. https://doi.org/10.1002/jrsm.1378

Hanani, L., Nugrahaeni, E., & Mustofa, K. (2018). Analisis Perbandingan Algoritma K-Nearest Neighbor (K-NN) dan Naïve Bayes pada Klasifikasi Data Siswa. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, *2*(6), 2449–2456.

Hastuti, S. D. (2016). Penerapan Metode k-Nearest Neighbor (k-NN) dalam Klasifikasi Dini Pneumonia pada Anak Menggunakan Fitur Gejala. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, *1*(2), 315–322.

Heryadi, Y., & Wahyono, T. (2020). *Machine learning : (konsep dan implemetasi) / penulis, C.* https://www.researchgate.net/publication/344419764_Machine_Learning_Konsep_dan_Implementasi

Hidayat, R., & Nurhayati, R. (2019). Penerapan Metode k-Nearest Neighbor (k-NN) dalam Klasifikasi Tingkat Kecanduan Gadget pada Remaja. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, *3*(11), 4612–4619.

Kusumawardani, R. A., & Mariana, M. (2019). Klasifikasi Topik Bahasa Indonesia dengan Algoritma k-Nearest Neighbor dan TF-IDF. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, *3*(6), 2582–2589.

Mongeon, P., & Paul-Hus, A. (2016). The journal coverage of Web of Science and Scopus: a comparative analysis. *Scientometrics*, *106*(1), 213–228. https://doi.org/10.1007/S11192-015-1765-5

Pedregosa et al. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*.

Pradana, A. W., & Hayaty, M. (2019). The Effect of Stemming and Removal of Stopwords on the Accuracy of Sentiment Analysis on Indonesian-language Texts. *Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control*, *4*(4), 375–380. https://doi.org/10.22219/KINETIK.V4I4.912

Purnomo, H., & Hartanto, R. (2017). *Penerapan Metode k-Nearest Neighbor dalam Klasifikasi Berita Kriminal* (2nd ed., Vol. 1). Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer.

Rachmawati, D., & Prasetio, A. P. (2019). Klasifikasi Teks Bahasa Indonesia Menggunakan Metode k-Nearest Neighbor dan Seleksi Fitur Chi-Square. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, *3*(12), 5112–5119.