

KLASIFIKASI KOLEKSI PERPUSTAKAAN BERBASIS DEWEY DECIMAL CLASSIFICATION MENGGUNAKAN ALGORITMA TF-RF DAN K-NEAREST NEIGHBOR

Anis Sulala¹, Lutfi Hakim^{2*}, Sepyan Purnama Kristanto³, Ruth Ema Febrita⁴, Khoirul Umam⁵

^{1,2,3,4,5}Jurusan Bisnis dan Informatika, Politeknik Negeri Banyuwangi

¹anissulala343@gmail.com, ²lutfi@poliwangi.ac.id, ³sepyan@poliwangi.ac.id, ⁴ruthemafebrita@poliwangi.ac.id,

⁵khoirulumam@poliwangi.ac.id

*Corresponding Author

Abstrak

Perpustakaan Politeknik Negeri Banyuwangi memiliki banyak koleksi buku dari berbagai bidang keilmuan yang terus bertambah setiap tahunnya. Pertambahan koleksi ini menimbulkan tantangan dalam proses pengklasifikasian buku yang masih dilakukan secara manual, sehingga memakan waktu dan tenaga yang besar. Oleh karena itu, penelitian ini bertujuan untuk mengembangkan sistem klasifikasi koleksi perpustakaan berbasis *Dewey Decimal Classification* (DDC) menggunakan algoritma *Term Frequency-Relevance Frequency* (TF-RF) dan *K-Nearest Neighbor* (kNN). Data yang digunakan berupa judul buku berbahasa Indonesia dengan jumlah 4056 data pada 73 label. Prosesnya dimulai dari pre-processing, ekstraksi fitur menggunakan TF-RF untuk mengukur relevansi kata kunci dan pengklasifikasian model dengan kNN, serta evaluasi performa model dengan K-Fold Cross validation 10. Hasil penelitian ini menunjukkan bahwa sistem klasifikasi yang dikembangkan mampu menghasilkan akurasi terbaik sebesar 74,38% dan presisi 78,91% dengan *tunning parameter* menggunakan pendekatan *GridSearch* pada parameter K=1 yang diperoleh pada skenario ketiga dengan 7 kelas dan 2831 data. Penelitian ini membuktikan bahwa kombinasi TF-RF dan kNN efektif diterapkan untuk klasifikasi teks koleksi buku.

Kata kunci: pengelompokan koleksi perpustakaan, sistem identifikasi, *term frequency-relevance frequency*, digitalisasi perpustakaan, koleksi e-library Poliwangi

1. Pendahuluan

Perpustakaan merupakan salah satu sumber informasi yang mencakup beragam jenis koleksi buku. Untuk mengelola jenis koleksi buku yang beragam dan melimpah pustakawan dapat memanfaatkan fitur klasifikasi buku dengan menggunakan teknologi sistem klasifikasi (Harahap and Husna 2019). Perpustakaan Politeknik Negeri Banyuwangi (Poliwangi) sudah menggunakan klasifikasi *Dewey Decimal Classification* (DDC) yang dibagi ke dalam 10 kelas utama. Klasifikasi DDC sering digunakan di berbagai perpustakaan di seluruh dunia karena dapat membantu menganalisis subjek dan penentuan nomor kelas pada data buku yang tersedia (Fadilla 2021). Namun, Perpustakaan Poliwangi menghadapi persoalan serius dalam proses pengklasifikasian buku menggunakan sistem DDC. Proses ini dinilai memakan waktu yang cukup lama, mengingat jumlah koleksi buku yang terus bertambah setiap tahunnya sekitar 500 eksemplar buku. Saat ini, untuk pengklasifikasian buku masih dilakukan secara manual, dimana pustakawan harus mengidentifikasi kata kunci dari judul buku terlebih dahulu, kemudian menentukan klasifikasi berdasarkan DDC. Proses ini tidak hanya memakan waktu, tetapi juga memerlukan tenaga yang cukup besar, terutama dalam menentukan nomor klasifikasi. Untuk mengklasifikasikan 500 eksemplar buku pustakawan menghabiskan waktu hingga 3 bulan. Masalah ini teridentifikasi berdasarkan hasil wawancara dengan salah satu pustakawan. Dari

wawancara tersebut, terungkap bahwa proses pengklasifikasian buku memerlukan waktu dan tenaga yang cukup banyak. Mengingat perpustakaan memiliki lebih dari 6.000 koleksi buku dari berbagai rumpun keilmuan, proses ini semakin tidak efisien.

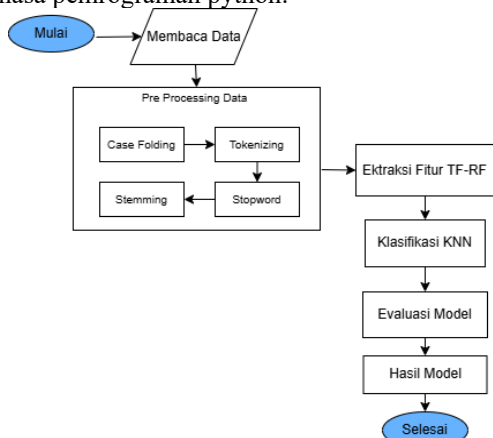
Penelitian sebelumnya menunjukkan bahwa penggunaan *Multi-Objective Optimization on the basis of Ratio Analysis* (MOORA) dapat diterapkan dengan baik untuk memberikan rekomendasi pada buku di perpustakaan (Firmansyah, Hakim, and Kristanto 2024). Untuk domain klasifikasi koleksi dengan DDC dilakukan oleh (Muhammad Alwi dkk, 2021) menunjukkan bahwa algoritma klasifikasi koleksi perpustakaan dilakukan dengan *Support Vector machine* (SVM) dan mendapatkan nilai akurasi tertinggi (sebesar 72%) dibandingkan dengan *Naïve Bayes* sebesar 59%, Random Forest sebesar 69%, dan algoritma *regression* sebesar 69% dengan menggunakan 8 kelas dan 1015 data uji. Penelitian lain menunjukkan bahwa penggunaan *k-Nearest Neighbors* (kNN) menunjukkan hasil lebih tinggi dibandingkan dengan SVM yaitu dengan akurasi sebesar 85,71% dari 7 sampel (Saputra et al. 2023). Selain itu, penelitian yang dilakukan oleh (Yunitarini, Gultom, and Stefany 2024) membuktikan bahwa tingkat akurasi model kNN dengan pembobotan *Term-Frequency-Inverse Document Frequency* (TF-IDF) dalam sistem klasifikasi jamu tradisional dapat mencapai nilai akurasi 85,71%. kNN merupakan algoritma yang sering digunakan dalam proses klasifikasi berdasarkan data training yang jaraknya paling dekat dengan objek tersebut dan dianggap

sebagai metode populer untuk mengenali pola karena kinerjanya yang efektif, hasil yang efisien dan kesederhanaannya (Hakim et al. 2021; Saadatfar et al. 2020; Setiawan, Muhammad, and Firizkiansah 2024). Selain itu, penelitian ini memanfaatkan algoritma *Term Frequency-relevance Frequency* (TF-RF) sebagai pengukur kemunculan kata kunci yang paling relevan pada semua kategori dalam suatu dokumen. Menurut (Sari et al. 2022) yang membahas tentang perbandingan pembobotan TF-IDF dan TF-RF menunjukkan hasil perbandingan yang didapat yaitu TF-IDF 96% dan TF-RF 98%.

Penelitian ini muncul sebagai respons terhadap dinamika pengelolaan koleksi perpustakaan yang semakin kompleks akibat pertumbuhan jumlah buku dan beban kerja perpustakaan. Sementara berbagai penelitian sebelumnya telah menguji algoritma klasifikasi seperti SVM dan kNN, serta pembobotan TF-IDF pada berbagai domain, belum banyak yang mengintegrasikan pembobotan TF-RF dengan kNN secara khusus untuk klasifikasi koleksi buku berbasis DDC di perpustakaan. Dengan memanfaatkan algoritma TF-RF yang menunjukkan akurasi lebih tinggi dibanding TD-IDF dan metode kNN yang dikenal sederhana namun efektif dalam klasifikasi berbasis kemiripan, diharapkan dapat terbentuk sistem klasifikasi koleksi perputakaan yang lebih cepat dan akurat sehingga mendukung peningkatan kualitas layanan perpustakaan.

2. Metode

Metodologi penelitian ditunjukkan pada Gambar 1 dimana terdapat beberapa tahapan yang dilakukan yaitu dimulai dari pengumpulan dan akuisisi data, *pre-processing* data, ekstraksi fitur, klasifikasi model dan evaluasi performa sistem. Setiap tahapan diimplementasikan menggunakan Bahasa pemrograman python.



Gambar 1. Alur Metode Penelitian

Langkah pertama adalah melakukan pengumpulan data, di mana data diambil dari database perpustakaan dan disimpan dalam format csv. Kemudian dilakukan *pre-processing* data yang terdiri dari beberapa tahapan seperti, *case folding*, *tokenizing*, *stopword* dan *Stemming*. Proses ini bertujuan untuk membersihkan teks agar siap untuk

diproses dalam tahap selanjutnya. Pada tahap ini menggunakan library NLTK dan Sastrawi agar memudahkan dalam mengimplementasikan proses *pre-processing*. NLTK merupakan platform untuk membantu pemrosesan bahasa yang berhubungan dengan Bahasa manusia yang digunakan untuk proses pengolahan teks yaitu, *case folding*, *tokenizing* dan *stopword*. Sastrawi merupakan library yang dirancang untuk memproses teks dalam Bahasa Indonesia, fungsi sastrawi yaitu untuk pengolahan teks pada *stemming* dikarenakan pada NLTK dukungan Bahasa Indonesia yang masih optimal (Albab, P., and Fawaiq 2023). Tahap berikutnya adalah pembobotan setiap kata pada judul dokumen menggunakan algoritma TF-RF. Setelah mendapatkan nilai pembobotan akan dilakukan klasifikasi dengan kNN untuk mengetahui nilai jarak terdekat dengan dataset yang sudah ada dengan rumus jarak Euclidean. Pada tahap terakhir akan dilakukan evaluasi performa sistem dengan mencari nilai akurasi dan presisi.

2.1 Pengumpulan dan Akuisisi Data

Pengumpulan data ini diperoleh dari database sistem *E-Libray* Poliwangi (<https://e-library.poliwangi.ac.id/>) dengan bentuk file csv yang berisi berbagai jenis koleksi, seperti buku, laporan proyek akhir, laporan MKI, serta laporan penelitian dan pengabdian dosen. Dari keseluruhan data tersebut, dilakukan penyaringan untuk memperoleh data yang sesuai dengan keutuhan klasifikasi. Penyaringan awal dilakukan dengan mengambil data yang berbahasa Indonesia, berisi judul buku dan kategori buku yang telah terklasifikasi. Hasil data yang telah didapatkan, selanjutnya akan dilakukan 3 kali eksperimen atau skenario uji coba dan setiap eksperimen yang dilakukan setiap data memiliki karakteristik tertentu yaitu, 1) Skenario uji coba 1 dengan menggunakan seluruh data yang didapatkan dari hasil pengumpulan data, 2) Skenario uji coba 2 dilakukan dengan menguji coba label yang rumpun ilmunya ada di Politeknik Negeri Banyuwangi, 3) Skenario uji coba 3 mengambil data diatas 100 pada setiap kelas.

2.2 Pre-Processing Data

Pre-processing merupakan tahap awal dalam pengolahan data yang diperoleh dari pengumpulan data untuk memperbaiki data agar lebih terstruktur di proses pada setiap dokumen yang digunakan (Assidyk, Setiawan, and Kurniawan 2020). Tahap ini bertujuan untuk merapikan data mentah yang tidak terstruktur agar menjadi data yang terstruktur.

A. Case Folding

Tahap *case folding* adalah sebuah proses untuk mengubah semua kalimat *Uppercase* (Huruf Kapital) menjadi *Lowercase* (Huruf Kecil). Pada tahap ini dilakukan pengolahan data judul buku yang didapatkan dari proses akuisisi data. Pada proses *case folding* ini melakukan penghapusan simbol dan karakter khusus dalam bentuk angka, tanda baca dan

whitespace agar data yang diolah lebih bersih (Asiyah and Fithriasari 2016). Hasil dari proses case folding akan berupa teks dengan semua huruf dalam bentuk kecil dan tanpa simbol atau karakter khusus.

B. Tokenizing

Pada tahap ini bertujuan untuk pemotongan kata atau setiap kalimat dataset akan dipecah berdasarkan spasi untuk menghasilkan daftar kata (token). Pada tahap ini, setiap kalimat akan dipecah menjadi token-token yang terpisah, yang nantinya dapat dianalisis dan diproses lebih lanjut.

C. Stopword

Tahap *stopword* adalah tahap untuk menghapus kata-kata umum yang tidak penting atau tidak terlalu berpengaruh pada proses klasifikasi seperti kata penghubung “salah satu”, kata depan “di”, “atau”, “dari” (Apriansyah, Hermawan, and Avianto 2024). Tujuan dari tahap ini adalah untuk mengurangi jumlah kata dalam dokumen dan menghindari redundansi.

D. Stemming

Stemming adalah langkah terakhir pada tahap pre-processing, yang bertujuan untuk menemukan akar kata dengan mengubah kata menjadi bentuk dasarnya setelah melewati proses *stopword*. Pada tahap ini dilakukan usaha untuk mengembalikan berbagai bentuk kata dengan menghilangkan semua imbuhan, seperti awalan, sisipan atau akhiran dari kata-kata tersebut (Rahmadhani et al. 2024).

2.3 Ekstraksi Fitur

Setelah melakukan tahap pre-processing data, langkah selanjutnya adalah proses ekstraksi fitur. Ekstraksi fitur ini merupakan sebuah proses teks diubah menjadi format numerik, karena komputer hanya dapat memproses data numerik. Salah satu teknik yang digunakan ekstraksi fitur penelitian adalah pembobotan TF-RF. TF-RF merupakan metode pembobotan yang digunakan untuk menghitung kata yang sering muncul dalam setiap dokumen atau objek. Metode ini termasuk metode baru yang bertujuan memperbaiki metode pembobotan yang telah ada sebelumnya (Assidyk, Setiawan, and Kurniawan 2020; Rahmadhani et al. 2024). TF-RF ini menggabungkan Term-Frequency (TF) dan Relevance Frequency (RF). Term-Frequency (TF) menghitung seberapa sering suatu kata muncul dalam dokumen, sedangkan RF mengukur seberapa relevan suatu kata muncul dalam konteks seluruh dokumen yang sedang dipertimbangkan. Formula perhitungan dari metode TF-RF dapat dilihat pada Persamaan (1). Dengan mengimplementasikan metode TF-RF dalam sistem klasifikasi memungkinkan pengukuran yang lebih cermat terhadap kemunculan kata penting yang relevan dalam setiap dokumen.

$$tf_{td}rf = tf_{td} * rf \quad (1)$$

atau

$$\log \left(2 + \frac{b}{(\max(1, c))} \right)$$

Keterangan:

TF_{rf} : Pembobotan dokumen ke dalam model ruang TF-RF

tf_{td} : Frekuensi kemunculan kata t dalam dokumen

t : Istilah atau kata kunci yang sedang dipertimbangkan

b : Jumlah judul dokumen yang mengandung kata (*term*)

c : Jumlah judul dokumen yang tidak mengandung kata (*term*)

2.4 Klasifikasi dengan kNN

Algoritma kNN merupakan sebuah metode untuk mengklasifikasikan objek berdasarkan data training yang terdekat dengan objek tersebut berdasarkan data yang sudah terklasifikasi sebelumnya. Prinsip kerja KNN adalah mencari jarak terdekat antara data yang akan dievaluasi dengan K tetangga (*neighbor*) terdekatnya dalam data pelatihan. Metode ini salah satu klasifikasi yang sederhana untuk mengukur tingkat akurasi dengan mengklasifikasikan berdasarkan jarak terdekat dengan *neighbor* (tetangga) terdekatnya (Sutianah, Sokibi, and Kartika 2024). Kemudian untuk mengukur jarak atau kemiripan antara dua titik yaitu data training x_i dan data testing y_i dapat menggunakan rumus seperti *Euclidean Distance*, *Manhattan Distance* dan *Minkowski Distance* (Alvian et al. 2021), dikarenakan rumus *Euclidean Distance* digunakan sebagai rumus jarak yang umumnya digunakan pada algoritma KNN dalam mengukur jarak terdekat antar data, perhitungan yang mudah diimplementasikan dan memiliki akurasi yang besar seperti pada persamaan (2). Penggunaan rumus jarak *Euclidean* menunjukkan performa yang baik dalam mengelompokkan data bibit varietas unggul padi, karena menghasilkan nilai *Davies Bouldin Index* yang kecil (Pangestu and Fitriani 2022). Rumus dari metode kNN untuk menghitung kedekatan antara data training dan data testing ditunjukkan pada Persamaan (2).

$$dist(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (2)$$

dimana:

$dist_{(x,y)}$: Jarak antar data atau objek x dan y

N : Banyaknya variabel bebas atau dimensi data

x_i : Data training

y_i : Data uji

Skenario klasifikasi juga dilakukan *tunning parameter* K pada metode kNN untuk mencari K optimal menggunakan pendekatan *Grid Search*.

2.5 Evaluasi Performa Sistem

Evaluasi performa pada penelitian ini dilakukan dengan menghitung jumlah *True Positive (TP)*, *True Negative (TN)*, *False Positive (FP)*, dan *False Negative (FN)* dari hasil pengujian setiap iterasi *K-Fold Cross Validation*. Keempat nilai ini kemudian digunakan untuk menghitung dua metrik utama, yaitu akurasi dan presisi seperti yang ditunjukkan pada persamaan 3 dan 4. Akurasi diperoleh dengan membagi jumlah prediksi benar

(TP+TN) dengan seluruh jumlah data pengujian (TP+TN+FP+FN). Nilai akurasi menunjukkan seberapa besar proporsi prediksi tepat dari keseluruhan prediksi yang dibuat oleh sistem. Semakin tinggi nilai akurasi, semakin baik sistem dalam membedakan data positif dan negatif secara umum. Sementara, Presisi dihitung dengan membagi jumlah TP dengan jumlah seluruh prediksi positif (TP+FP). Metrik ini menunjukkan seberapa akurat sistem dalam menemukan dokumen atau data positif tanpa memperbanyak kesalahan memilih kelas negatif sebagai positif. Semakin tinggi nilai presisi, semakin sedikit kesalahan sistem dalam memilih data yang seharusnya tidak dimasukkan sebagai kelas positif. Sementara itu, pendekatan *K-Fold Cross Validation* paling sering digunakan karena umumnya menghasilkan model yang bias atau meminimalisir terjadinya kondisi *overfitting* (Widyaningsih, Arum, and Prawira 2021). Metode k-fold memiliki parameter K, dimana pemilihan nilai k disarankan adalah 5 atau 10 (Dananjaya and Indradewi 2023). Proses ini diulangi sebanyak k kali hingga setiap kelompok diperlakukan sebagai data validasi dan tersisa sebagai data latih. Dengan demikian, melalui proses *K-Fold Cross Validation*, nilai TP, TN, FP dan FN diakumulasikan untuk setiap *fold*, lalu digunakan dalam persamaan akurasi dan presisi sehingga diperoleh gambaran kinerja sistem yang stabil dan tidak bias.

$$\text{Akurasi} = \frac{TP+TN}{TP+TN+FP+FN} \tag{3}$$

$$\text{Presisi} = \frac{TP}{TP+FP} \tag{4}$$

Keterangan:

TP : data judul buku relevan yang terklasifikasi benar oleh sistem

FP : total data buku tidak relevan tetapi terklasifikasi relevan oleh sistem

TN: total data tidak relevan terklasifikasi dengan tidak relevan oleh sistem

FN: total data relevan yang terklasifikasi dengan tidak relevan oleh sistem

3. Hasil dan Pembahasan

3.1 Hasil

Data yang didapatkan dari proses akuisisi data adalah sejumlah 4056 *instance* dan 73 label atau kategori DDC. Dari masing-masing 73 kategori tersebut menunjukkan proporsi data yang tidak seimbang dan mayoritas jumlahnya didominasi oleh Karya Umum (1017), Ilmu teknik (528), manajemen (393), pertanian dan teknologi yang berkaitan (270), kesustraan Indonesia (283), Ilmu ekonomi (203), dan psikologi (137). Sedangkan kategori DCC lainnya, data yang didapatkan kurang dari 100 dan bahkan jumlah dalam rentang satuan. Dari hasil pengumpulan dan pemfilteran data tersebut, selanjutnya dibagi menjadi 3 kali skenario eksperimen yaitu, 1) Skenario 1 dengan jumlah data 4056 dan 73 kelas, 2) Skenario 2 dengan jumlah 2631 data dan 16 kelas, 3) Skenario

3 mengambil data diatas 100 dengan jumlah 2831 data dan 7 kelas. Setelah pengumpulan data selesai dilakukan, pada penelitian ini dilakukan tahap *pre-processing* yang meliputi tahapan *case-folding*, *tokenizing*, *stopword*, dan *stemming*. Pada tahap ini melakukan pengolahan data menggunakan bahasa pemrograman python dan memanfaatkan library NLTK dan Sastrawi. Hasil dari setiap langkah pada tahapan *pre-processing* ditunjukkan pada Gambar 2-5. Secara keseluruhan, pada tahapan *pre-processing* ini didapatkan hasil akhir berupa setiap kata yang dikembalikan ke dalam bentuk dasar, menghilangkan kata imbuhan dan istilah yang tidak penting sehingga meminimalisasi variasi kata yang memiliki arti serupa.

	title	case_folding_title
	MEMBERIKAN PETUNJUK TEKNIS KEPADA PELANGGAN	memberikan petunjuk teknis kepada pelanggan
	MENERAPKAN METODA DAN PRAKTIK PENGGUNAAN KEMBA...	menerapkan metoda dan praktik penggunaan kempa...
	MENGIMPLEMTASIKAN RANCANGAN ENTITAS DAN KETER...	mengimplemtasikan rancangan entitas dan keter...
	KNOWLEDGE MANAGEMENT A GUIDE BOOK EFEKTIF BERIL...	knowledge management a guide book efektif beri...
	ANALISA DAN PERANCANGAN SISTEM DENGAN CODEIGNI...	analisa dan perancangan sistem dengan codeigni...

Gambar 2. Hasil case-folding

	case_folding_title	tokenized_title
0	memberikan petunjuk teknis kepada pelanggan	[memberikan, petunjuk, teknis, kepada, pelanggan]
1	menerapkan metoda dan praktik penggunaan kempa...	[menerapkan, metoda, dan, praktik, penggunaan,...
2	mengimplemtasikan rancangan entitas dan keter...	[mengimplemtasikan, rancangan, entitas, dan, k...
3	knowledge management a guide book efektif beri...	[knowledge, management, a, guide, book, efekti...
4	analisa dan perancangan sistem dengan codeigni...	[analisa, dan, perancangan, sistem, dengan, co...

Gambar 3. Hasil Tokenizing

Gambar 2 menunjukkan hasil dari proses *case folding* dimana pada tahap ini, seluruh karakter pada judul buku yang semula terdidi dari campuran huruf besar dan kecil telah diubah menjadi huruf kecil semua. Proses ini membantu menyamakan format penulisan teks, mengurangi perbedaan penulisan antar data, serta mempermudah tahap pengolahan data selanjutnya karena tidak ada lagi perbedaan antara huruf kapital dan huruf kecil dalam analisis. Selanjutnya, dilakukan proses *tokenizing* dimana sampel hasilnya ditunjukkan pada Gambar 3. Pada tahap *tokenizing*, setiap judul buku yang telah melewati tahap *case folding* yang sebelumnya berupa kalimat utuh dipecah menjadi daftar kata secara terpisah berdasarkan spasi. Dengan ini, setiap kata dapat dikenali sebagai satuan fitur yang berdiri sendiri, sehingga proses analisis data menjadi lebih terfokus dan akurat.

Tahap selanjutnya adalah proses *stopword* yang bertujuan untuk menghapus kata-kata yang dianggap tidak memiliki makna penting dalam analisis, seperti kata sambung atau kata umum lainnya. Dengan penghapusan ini, kata-kata yang tersisa adalah daftar kata penting dari judul buku seperti yang ditunjukkan pada Gambar 4. Langkah selanjutnya adalah proses *stemming*, yang mana setiap kata dari proses *stopword* diubah ke dalam bentuk dasarnya sehingga dapat meminimalisir variasi kata yang memiliki arti serupa seperti pada sampel hasil yang ditunjukkan pada Gambar 5.

tokenized_title	stopword_removal_title
[memberikan, petunjuk, teknis, kepada, pelanggan]	[petunjuk, teknis, pelanggan]
[menerapkan, metoda, dan, praktik, penggunaan, kemb...]	[menerapkan, metoda, praktik, penggunaan, kemb...]
[mengimplementasikan, rancangan, entitas, dan, k...]	[mengimplementasikan, rancangan, entitas, keterk...]
[knowledge, management, a, guide, book, efektif...]	[knowledge, management, a, guide, book, efektif...]
[analisa, dan, perancangan, sistem, dengan, co...]	[analisa, perancangan, sistem, codeigniter, la...]

Gambar 4. Hasil Stopword

stopword_removal_title	stemmed_title
[petunjuk, teknis, pelanggan]	tunjuk teknis langgan
[menerapkan, metoda, praktik, penggunaan, kemb...]	terap metoda praktik guna kembali reusable sub...
[mengimplementasikan, rancangan, entitas, keterk...]	mengimplementasikan rancang entitas kait entitas
[knowledge, management, a, guide, book, efektif...]	knowledge management a guide book efektif inov...
[analisa, perancangan, sistem, codeigniter, la...]	analisa ancang sistem codeigniter laravel

Gambar 5. Hasil Stemming

Pada proses ini dapat membantu mengurangi kompleksitas data dan meningkatkan konsistensi pada proses klasifikasi data teks. Selanjutnya, pada Tabel 1 merangkum contoh hasil dari tahapan *pre-processing*.

Tabel 1. Hasil *Pre-Processing*

Judul Buku	Hasil Pre-Processing
A1 Explore Indonesia	explore, indonesia
A2 Panduan Psikotes	pandu, psikotes
A3 Etika Dalam Islam	etika, islam
A4 Pengantar Ilmu Politik	antar, ilmu, politik
A5 Jago Kuasai Bahasa Mandarin	jago, kuasa, bahasa, mandarin
A6 Pengantar Statistika	antar, statistika
A7 Dasar Teknik Listrik	dasar, teknik, listrik
A8 Dari Hobi Jadi Profesional	hobi, profesional
A9 Janji Sahabat	janji, sahabat
A10 Pemasaran Pariwisata	pasar, pariwisata
A11 Teknik Menggunakan Alat Ukur	teknik, menggunakan, alat, ukur

Setelah tahap *pre-processing*, selanjutnya yaitu menghitung kemiripan antar teks menggunakan algoritma TF-RF. Tabel 3 dan 4 pada lampiran menunjukkan gambaran hasil perhitungan skor RF dan pembobotan TF-RF. Pada tahapan ini bertujuan untuk mengekstraksi fitur atau ciri dari data teks judul yang dilakukan untuk mengubah representasi teks menjadi format numerik dengan menggunakan rumus pada persamaan (1). Output dari tahapan ini adalah setiap kata dalam dokumen memiliki nilai bobot yang selanjutnya akan digunakan dalam tahapan klasifikasi dengan kNN. Tahap selanjutnya dalam penelitian ini adalah melatih model algoritma kNN dengan tahapannya sebagai berikut:

A. Split Data

Tahap awal pada klasifikasi KNN adalah pembagian dataset. Dataset dibagi berdasarkan k buah *fold validation* dalam proses *training* dan *testing*. Penelitian ini juga dilakukan uji coba sebanyak 3 kali eksperimen dengan jumlah data latih yang bervariasi seperti yang ditunjukkan pada Tabel 2.

Tabel 2. Hasil Split Data

Nama Dataset	Jumlah data	
	Latih	Uji
Skenario uji coba 1	2945	737
Skenario Uji coba 2	2104	527
Skenario Uji coba 3	2261	566

B. Implementasi Model kNN dengan K-Fold

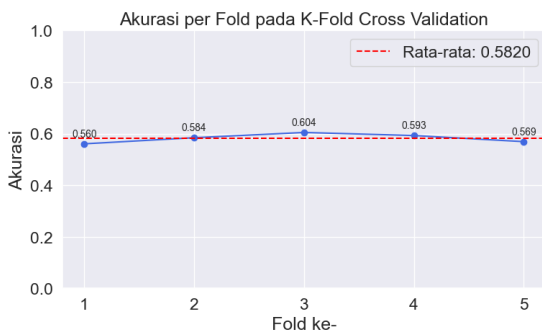
Pada tahap ini dilakukan klasifikasi menggunakan KNN dengan pendekatan *Grid-Search* untuk *tunning parameter k* pada metode kNN. Tahap ini dilakukan pada tiga dataset yaitu data 1 yang terdiri dari 73 kelas, data 2 yang terdiri dari 16 kelas dan data 3 yang terdiri dari 7 kelas. Setiap dataset diperlakukan dengan tahapan yang sama. Tahap ini bertujuan untuk menemukan nilai *n_neighbors* (jumlah tetangga terdekat) yang optimal. Parameter yang diuji pada penelitian ini meliputi *n_neighbors*, nilai *p2* sebagai rumus Minkowski yang berarti jarak euclidean dengan menggunakan rumus persamaan (4), serta jenis metric yang digunakan untuk menghitung jarak antar data. Jumlah tetangga yang diuji pada penelitian ini yaitu rentang nilai $k = 1$ hingga $k = \sqrt{n}$ pada pelatihan metode kNN dari jumlah data pelatihan. Selain itu, pemilihan nilai *k* menggunakan angka ganjil yang bertujuan untuk menghindari hasil klasifikasi seri. Setelah mendapatkan nilai *k* yang optimal, kemudian akan dilakukan evaluasi dengan teknik validasi dengan *k-fold cross validation* dengan nilai $k=5$. Tahap akhir untuk implementasi KNN adalah menguji performa model dengan data uji yang telah dipisah. Evaluasi ini dilakukan dengan menghitung metrik akurasi menggunakan rumus (2) dan presisi menggunakan rumus (3).

C. Evaluasi Model

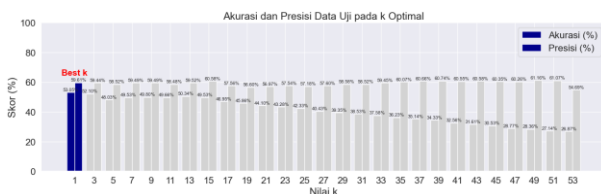
Evaluasi model pada penelitian ini dilakukan dengan 3 skenario dan 3 dataset yang berbeda seperti yang ditunjukkan pada Tabel 2. Pada skenario ini juga dilakukan *tunning parameter K* menggunakan pendekatan *Grid Search* (Nugraha and Sasongko 2022; Ogunsanya, Isichei, and Desai 2023; Suryadi et al. 2024) untuk mencari *K* optimal pada kNN (Taufiq et al. 2024). Optimasi nilai *K* pada algoritma kNN perlu dilakukan dengan mencari *hyperparameter* terbaik agar dapat mengoptimalkan performa klasifikasi (Manurung et al. 2025; Saranya and Pravin 2023; Sukamto, Hadiyanto, and Kurnianingsih 2023). Pada skenario uji coba 1 dari total 73 kelas menghasilkan nilai *k* yang paling optimal yaitu 1 yang dihasilkan oleh *Grid Search*, kemudian diuji menggunakan *k-fold cross validation* dengan $k=5$. Gambar 6 menunjukkan nilai rerata akurasi dari tahap pengujian dengan menggunakan *k-fold cross validation* dari $k=1$ sampai dengan $k=5$. Gambar 7 menunjukkan performa hasil klasifikasi dari $K=1$ sampai dengan $k=53$ pada kNN dengan hasil terbaik menunjukkan nilai akurasi 53.05% dan presisi 59.61% pada $k=1$.

Pada skenario uji coba 2 melibatkan 16 kelas dengan jumlah data yang bervariasi yaitu sebesar 2104 data latih dan 527 data uji. Skenario ini dilakukan karena ada kondisi jumlah dataset yang tidak berimbang (*imbalance dataset*). Setelah melakukan proses yang sama dengan skenario uji

coba 1, dilakukan pengujian pada skenario uji 2. Pada eksperimen dengan k-fold cross validation menunjukkan nilai akurasi tertinggi dengan skor 73.1% pada k=3.

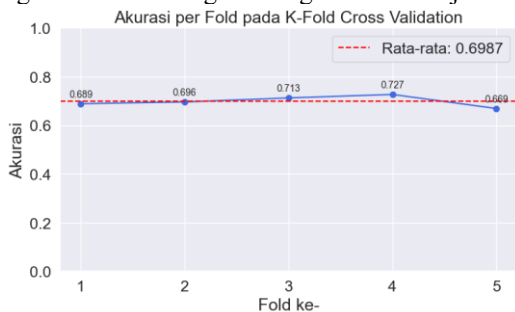


Gambar 6. Nilai Akurasi Pada Skenario Uji Coba 1

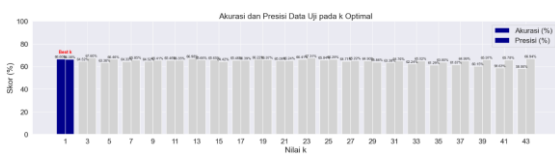


Gambar 7. Hasil Evaluasi Model pada Skenario Uji Coba 1

Selanjutnya, pada Gambar 9 menunjukkan nilai akurasi dan presisi yang diujicoba dari k=1 sampai dengan k=43 pada kNN dengan mengambil k bernilai ganjil menghasilkan nilai akurasi sebesar 66.60% dan presisi sebesar 66.38%. Hal tersebut menunjukkan bahwa terdapat peningkatan yang signifikan dibandingkan dengan skenario uji coba 1.



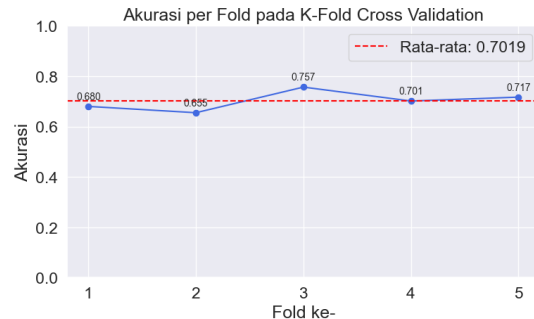
Gambar 8. Skor Akurasi pada Setiap fold dengan k-fold cross validation pada skenario Uji Coba 2



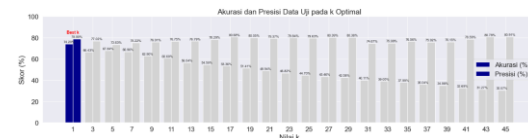
Gambar 9. Hasil Evaluasi Model pada Skenario 2

Sementara itu, pada skenario uji coba 3 dilakukan dengan menggunakan 7 kelas dengan jumlah data yang bervariasi pada setiap kelasnya. Penggunaan 7 kelas ini didasarkan pada penggunaan kelas dengan melibatkan label dengan jumlah instance lebih dari 100 pada setiap kelas. Gambar 10 dan 11 menunjukkan nilai performa sistem yang didapatkan pada skenario uji coba 3 ini, dimana performa hasil akurasi paling optimal adalah k=1 pada kNN, dengan akurasi sebesar

74.20% dan presisi sebesar 78.99%. Skor ini merupakan yang tertinggi dibandingkan dengan nilai k lainnya yang diuji dan juga dibandingkan dengan hasil pengujian pada skenario uji coba 1 dan 2.



Gambar 10. Akurasi per Fold pada Skenario Uji Coba 3 dengan K-Fold Cross Validation



Gambar 11. Hasil Uji Coba Skenario 3

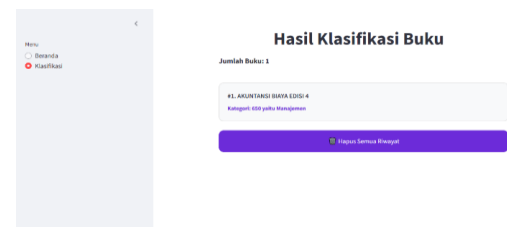
Kemudian untuk memastikan stabilitas model pada k=1 akan dilakukan evaluasi ulang menggunakan jumlah k=10 pada setiap skenario. Pada uji coba 1 menunjukkan sedikit peningkatan dibandingkan dengan k=5. Hasil dari uji coba 1 yang sebelumnya akurasi 53.05% menjadi 53.87% dan presisi meningkat 59.61% menjadi 60.17%. Meskipun hasil menunjukkan sesial kecil, peningkatan ini menunjukkan bahwa validasi model dengan k=10 menghasilkan nilai performa lebih stabil. Pada uji coba 2 juga menunjukkan peningkatan dengan akurasi dari 66.60% menjadi 67.36% dan presisi dari 66.38% menjadi 57.38%. Pada uji coba 3 juga menunjukkan hasil yang stabil dengan peningkatan akurasi dari 74.20% menjadi 74.38%, meskipun presisi mengalami penurunan tipis dari 78.00% menjadi 78.91%. Perubahan kecil ini menunjukkan bahwa model tetap konsisten, dan penggunaan *fold* lebih banyak tetap memberikan hasil yang lebih baik.

D. Deployment Model

Setelah melakukan evaluasi performa model, selanjutnya dilakukan *deployment model* dalam sistem berbasis web seperti yang ditunjukkan pada Gambar 12 dan 13. Pada halaman pertama pengguna dapat memasukkan judul buku yang akan diklasifikasikan ke dalam sistem. Pengguna menekan button kirim dan sistem secara otomatis akan menampilkan hasil klasifikasi di bawah button kirim, kemudian di halaman kedua sistem akan menampilkan seperti, data seluruh buku yang telah diinputkan oleh pengguna, jumlah buku, judul buku beserta hasil klasifikasi yang telah melalui proses dari pre-processing, perhitungan TF-RF dan model KNN.



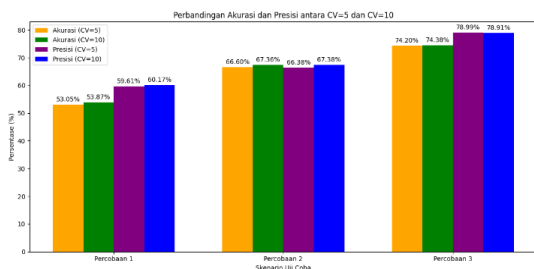
Gambar 12. Halaman Pertama Sistem Klasifikasi Buku



Gambar 13. Halaman Hasil Klasifikasi Buku

3.2 Pembahasan

Pada penelitian ini menggunakan data pada perpustakaan Politeknik Negeri Banyuwangi dengan jumlah data 4056 dengan 73 kelas. Pada penelitian ini dilakukan 3 kali eksperimen. Ketiga dataset tersebut kemudian dilakukan proses tahapan yang sama, mulai dari preprocessing, ekstraksi fitur menggunakan TF-RF, klasifikasi menggunakan algoritma KNN dengan Grid-Search dan K-Fold sampai evaluasi model dengan data uji.



Gambar 14. Hasil Perbandingan Pada Setiap Uji Coba

Berdasarkan grafik yang ditunjukkan pada Gambar 14 dapat dilihat bahwa hasil evaluasi menggunakan K-fold cross validation 10 menunjukkan adanya peningkatan akurasi dan presisi dibanding dengan dengan k-fold = 5. Peningkatan ini menunjukkan bahwa validasi dengan menggunakan k-fold=10 memberikan hasil yang lebih stabil. Hal ini dikarenakan dengan proporsi pelatihan yang lebih besar, model memperoleh lebih banyak informasi untuk belajar sehingga mampu menangkap pola secara lebih baik. Hasil evaluasi yang diperoleh pada penelitian ini menunjukkan bahwa akurasi dan presisi model berbanding terbalik dengan jumlah kelas pada dataset. Semakin sedikit jumlah kelas maka semakin meningkat akurasi dan presisinya. Pada Skenario uji coba 1 memperoleh akurasi 53.87% dan presisi 60.17%, pada percobaan 2 mendapatkan nilai akurasi 67.36% dan nilai presisi 67.38%. sedangkan pada skenario uji coba 3 mencapai hasil terbaik dengan akurasi dari 74.38% dan nilai presisi 78.91%.

Peningkatan ini disebabkan oleh kompleksitas klasifikasi saat jumlah kelas berkurang. Selain itu distribusi jumlah data antar kelas yang lebih seimbang pada percobaan 3 membuat percobaan ini menjadi percobaan terbaik. Ketidakseimbangan data dan tingginya fitur pada skenario uji coba 1 dan 2 menyebabkan model bias terhadap kelas mayoritas karena tetangga terdekat banyak mendominasi oleh kelas yang memiliki jumlah data lebih besar dan menurunkan performa model. Pada uji coba pertama, fitur judul buku menghasilkan 4.423 kata unik dengan dengan jumlah *instance* yang tidak seimbang dimana beberapa label kategori seperti terbitas berseri umum, jurnalisme, alkitab, filsafat kuno, logika, seni perkotaan dan pertamanan, fotografi, seni rekreasi dan pertunjukkan dan sebagainya jumlah masih satuan. Jumlah ini jauh signifikan dengan jumlah *instance* pada beberapa label seperti karya umum (jumlahnya 1017), ilmu ekonomi (203), ilmu teknik (528), pertanian dan teknologi yang berkaitan (270), manajemen (393), dan kesustraan Indonesia (283). Perbedaan yang signifikan ini juga mempengaruhi akurasi hasil klasifikasi. Oleh karena itu, pada skenario uji coba 2 mencoba untuk mengurangi jumlah label yang digunakan dengan kriteria label-label yang berkaitan dengan bidang ilmu program studi di Poliwangi yaitu menjadi 16 label dan data sejumlah 2631. Begitupula pada skenario uji coba 3 mengambil label dengan jumlah *instance* tiap kelas atau label sebanyak 100 data, sehingga jumlah label yang digunakan dalam proses klasifikasi menjadi 7 label dengan jumlah data sebesar 2831.

Selain itu, data yang digunakan dalam ekstraksi fitur pada penelitian inihanya melibatkan judul buku. Judul buku yang cenderung ringkas dan spesifik dapat membantu metode TF-RF dalam menonjolkan kata-kata yang relevan. Namun, rendahnya akurasi pada uji coba 1 menunjukkan bahwa fitur judul saja belum cukup untuk membedakan antar kelas, terutama dalam skenario dengan jumlah kelas yang besar. Karena judul yang pendek tidak selalu mencerminkan isi buku secara mendalam. Lebih lanjut, skenario uji coba ke 2 dan 3 juga menunjukkan performa akurasi yang belum menunjukkan performa yang maksimal (mendekati 100%), sehingga penggunaan judul buku saja sebagai fitur terbukti kurang efektif untuk klasifikasi dengan banyak kelas. Menambahkan fitur seperti ringkasan dan kata kunci (*keyword*) mungkin dapat meningkatkan performa, karena kata kunci dan ringkasan memberikan konteks yang lebih rinci dan memperkaya representasi dokumen dalam TF-RF dan membantu model KNN mengklasifikasikan dokumen dengan lebih akurat. Hasil penelitian ini menunjukkan bahwa kombinasi TF-RF dan KNN cukup efektif untuk klasifikasi dokumen teks dalam konteks buku perpustakaan, terutama pada percobaan ketiga yang memiliki jumlah kelas tidak terlalu banyak dan distribusi data antar kelas yang lebih seimbang, sehingga menghasilkan akurasi dan presisi terbaik. Percobaan ketiga menunjukkan akurasi dan

presisi tertinggi, meskipun akurasi dan presisi tertinggi sebesar 74.38% dan 78.91%. Sementara itu, penggunaan kNN untuk klasifikasi multilabel belum menunjukkan hasil yang optimal. Hal ini ditunjukkan bahwa hasil akurasi tertinggi pada $k=1$ pada semua eksperimen karena sistem penentuan keputusan pada kNN berdasarkan hasil voting, sehingga dibutuhkan pengembangan selanjutnya yang memanfaatkan data yang lebih seimbang dengan tujuan untuk peningkatan kinerja klasifikasi.

4. Kesimpulan

Berdasarkan analisis hasil dan pembahasan menunjukkan bahwa sistem klasifikasi koleksi buku perpustakaan berbasis DDC berhasil dikembangkan. Berdasarkan 3 eksperimen yang dilakukan dengan *tunning parameter k* pada kNN dengan menggunakan $fold=10$ menunjukkan hasil akurasi dan presisi yang tertinggi dengan skor akurasi 74.38% dan presisi 78.91%. Hasil ini didapatkan pada skenario percobaan ketiga yaitu dengan jumlah dataset 2831 dan 7 label. Disusul pada skenario 2 dengan nilai akurasi 67.36% dan presisi 67.38% dengan jumlah dataset 2631 dan 16 label, serta skenario 1 dengan nilai akurasi 53.87% dan presisi 60.17% dengan jumlah dataset 4056 dan 73 label. Dari ketiga nilai akurasi tertinggi untuk setiap skenario tersebut didapatkan pada $k=1$ pada kNN. Hasil ini dipengaruhi oleh jumlah kelas data yang semakin banyak akan meningkatkan kompleksitas fitur, serta distribusi data antar kelas yang tidak seimbang menyebabkan pembobotan TF-RF menjadi kurang optimal. Kedua faktor tersebut menyebabkan model menjadi bias terhadap kelas mayoritas. Namun pada percobaan ketiga, model sudah cukup efektif karena jumlah kelas lebih sedikit (7 kelas) dan distribusi data lebih seimbang.

Beberapa temuan masih perlu pengembangan lebih lanjut seperti jumlah data yang tidak seimbang, sehingga perlu menambahkan jumlah *instance* yang digunakan pada setiap kategori/label dalam klasifikasi terutama pada bidang ilmu yang ada di Poliwangi. Selain itu, penggunaan fitur pada judul buku dengan TF-RF menunjukkan hasil kurang optimal, sehingga dalam penelitian selanjutnya dapat ditambahkan kata kunci (*keyword*) dan/atau ringkasan. Kombinasi TF-RF dan TF-IDF untuk ekstraksi data judul, kata kunci dan ringkasan perlu dipertimbangkan untuk penelitian selanjutnya. Sementara itu, penggunaan KNN untuk klasifikasi juga perlu dipertimbangkan dan mencoba metode klasifikasi lain yang lebih sesuai seperti metode *Support Vector Machine (SVM)*, *Neural Network (NN)*, atau metode *deep learning* yang mempunyai performa lebih optimal dengan jumlah dataset yang relatif lebih banyak.

Daftar Pustaka:

Albab, M. Ulil, Yohana Karuniawati P., and Mohammad Nur Fawaiq. 2023. "Optimization of the Stemming Technique on Text

Preprocessing President 3 Periods Topic." *Jurnal Transformatika* 20(2): 1–12. <https://journals.usm.ac.id/index.php/transformatika/article/view/5374>.

Alvian, Verdi et al. 2021. "Klasifikasi Siswa Berprestasi Menggunakan Metode K-Nearest Neighbor (KNN) Pada SMA Negeri 2 Situbondo Classification Of Achieving Students Using K-Nearest Neighbor (KNN) Method At SMA Negeri 2 Situbondo." *Jurnal Smart Teknologi* 1(1): 2774–1702.

Apriansyah, Ferryma Arba, Arief Hermawan, and Donny Avianto. 2024. "Optimization of K Value in KNN Algorithm for Spam and HAM Classification in SMS Texts." *International Journal Software Engineering and Computer Science (IJSECS)* 4(2): 767–79. <https://journal.lembagakita.org/ijsecs/article/view/2681>.

Asiyah, Siti Nur, and Kartika Fithriasari. 2016. "Klasifikasi Berita Online Menggunakan Metode Support Vector Machine Dan K-Nearest Neighbor." *Jurnal Sains dan Seni ITS* 5(2): 317–22.

Assidyk, Agung N, Erwin Budi Setiawan, and Isman Kurniawan. 2020. "Analisis Perbandingan Pembobotan TF-IDF Dan TF-RF Pada Trending Topic Di Twitter Dengan Menggunakan Klasifikasi K-Nearest Neighbor." *e-proceeding of Engineering* 7(2): 9.

Dananjaya, I Kadek Wahyu, and I Gusti Ayu Agung Diatri Indradewi. 2023. "Perbandingan Metode Pembobotan TF-RF Dan TF-ABS Pada Kategorisasi Berita Di BDI Denpasar." *SINTECH (Science and Information Technology) Journal* 6(1): 16–25. <https://ejournal.instiki.ac.id/index.php/sintechjournal/article/view/1252>.

Fadilla, Nurul. 2021. "Komparasi Pemikiran Berwick Sayers Dan Mary Mortimer Tentang Sistem Klasifikasi Perpustakaan." *Jurnal Pustaka Ilmiah* 6(2): 1065. <https://jurnal.uns.ac.id/jurnalpustakailmiah/article/view/46421>.

Firmansyah, Ferdian, Lutfi Hakim, and Sepyan Purnama Kristanto. 2024. "Implementasi Metode MOORA Pada Sistem Pendukung Keputusan Pengadaan Buku Perpustakaan Implementation of the MOORA Method in the Library Book Procurement Decision Support System." *Sistemasi: Jurnal Sistem Informasi* 13(5): 1876–92.

Hadiansyah, Zikri, Zaenur Rozikin, and Muhamad Fatchan. 2024. "Implementasi Algoritma K-Nearest Neighbor Dalam Klasifikasi Penyakit Kanker Paru Paru." *Journal of Computer System and Informatics (JoSYC)* 6(1): 96–106. <https://ejournal.seminar-id.com/index.php/josyc/article/view/6195>.

- Hakim, Lutfi et al. 2021. "Disease Detection of Dragon Fruit Stem Based on The Combined Features of Color and Texture." *INTENSIF: Jurnal Ilmiah Penelitian dan Penerapan Teknologi Sistem Informasi* 5(2): 161–75.
- Harahap, Bella Lisda Hasunita, and Jazzimatul Husna. 2019. "Penerapan Sistem Klasifikasi Mandala Di Perpustakaan Gelaran Indonesia Buku Yogyakarta." *Jurnal Ilmu Perpustakaan* 7(1): 181–90. <https://ejournal3.undip.ac.id/index.php/jip/article/view/22829>.
- Manurung, Jonson, Hondor Saragih, Muhammad Azhar Prabukusumo, and Eryan Ahmad Firdaus. 2025. "Optimizing the Performance of the K-Nearest Neighbors Algorithm Using Grid Search and Feature Scaling to Improve Data Classification Accuracy." *Jurnal Mandiri IT* 14(2): 260–68. www.ejournal.isha.or.id/index.php/Mandiri.
- Muhammad Alwi, Oddy Virgantara Putra, and Dihin Muriyatmoko. 2021. "Classification of Book Collections Based on DDC 23 Using Text Mining Algorithm at UNIDA Gontor Library." *Procedia of Engineering and Life Science 2*. <https://pels.umsida.ac.id/index.php/PELS/article/view/1164>.
- Nugraha, Wahyu, and Agung Sasongko. 2022. "Hyperparameter Tuning on Classification Algorithm with Grid Search." *SISTEMASI* 11(2): 391. <http://sistemasi.ftik.unisi.ac.id/index.php/stmsi/article/view/1750>.
- Ogunsanya, Michael, Joan Isichei, and Salil Desai. 2023. "Grid Search Hyperparameter Tuning in Additive Manufacturing Processes." *Manufacturing Letters* 35: 1031–42. <https://linkinghub.elsevier.com/retrieve/pii/S21384632300113X>.
- Pangestu, Mohamad Sugeng, and Maulida Ayu Fitriani. 2022. "Perbandingan Perhitungan Jarak Euclidean Distance, Manhattan Distance, Dan Cosine Similarity Dalam Pengelompokan Data Bibit Padi Menggunakan Algoritma K-Means." *Sainteks* 19(2): 141. <http://jurnalnasional.ump.ac.id/index.php/SAI-NTEKS/article/view/14495>.
- Rahmadhani, Sadesty et al. 2024. "Sistem Rekomendasi Penelusuran Buku Berbasis Content-Based Filtering Dengan Pembobotan TF-RF." *JIP (Jurnal Informatika Polinema)* 10(4): 491–500. <https://jurnal.polinema.ac.id/index.php/jip/article/view/5565>.
- Saadatfar, Hamid et al. 2020. "A New K-Nearest Neighbors Classifier for Big Data Based on Efficient Data Pruning." *Mathematics*. <https://www.semanticscholar.org/paper/632669af604dc2418f635b041573d1bb204378a3>.
- Saputra, Joni, Yuan Sa'adati, Valian Yoga Pudya Ardhana, and M Afriansyah. 2023. "Klasifikasi Kematangan Buah Alpukat Mentega Menggunakan Metode K-Nearest Neighbor Berdasarkan Warna Kulit Buah." *RESOLUSI: Rekayasa Teknik Informatika dan Informasi* 3(5): 214–21. <https://djournals.com/resolusi>.
- Saranya, G., and A. Pravin. 2023. "Grid Search Based Optimum Feature Selection by Tuning Hyperparameters for Heart Disease Diagnosis in Machine Learning." *The Open Biomedical Engineering Journal* 17(1). <https://openbiomedicalengineeringjournal.com/VOLUME/17/ELOCATOR/e187412072304061/>.
- Sari, Yuslena, Andreyan Rizky Baskara, Puguh Budi Prakoso, and Noorhanida Royani. 2022. "Perbandingan Metode Pembobotan Tf-Rf Dan Tf-Idf Dikombinasikan Dengan Weighted Tree Similarity Untuk Sistem Rekomendasi Buku." *Jurnal Teknologi Informasi dan Ilmu Komputer* 9(6): 1323–32. <http://localhost/jtiik.ub.ac.id/index.php/jtiik/article/view/5709>.
- Setiawan, Dita, Ali Muhammad, and Angge Firizkiansah. 2024. "Pengklasifikasian Dokumen Teks Bahasa Indonesia Berbasis Vektor Space Model Dengan Menggunakan Metode K-Nearest Neighbor (k-NN) Dan Euclidean Distance." *JIKOMTI: Jurnal Ilmiah Ilmu Komputer dan Teknologi Informasi* 1(1): 30–37. <https://ojs.sains.ac.id/index.php/Jikomti/article/view/37>.
- Sukamto, Hadiyanto, and Kurnianingsih. 2023. "KNN Optimization Using Grid Search Algorithm for Preeclampsia Imbalance Class" eds. R. Isnanto, Hadiyanto, and B. Warsito. *E3S Web of Conferences* 448: 02057. <https://www.e3s-conferences.org/10.1051/e3sconf/202344802057>.
- Suryadi, Mulia Kevin et al. 2024. "Comparative Study of Various Hyperparameter Tuning on Random Forest Classification With SMOTE and Feature Selection Using Genetic Algorithm in Software Defect Prediction." *Journal of Electronics, Electromedical Engineering, and Medical Informatics*. <https://www.semanticscholar.org/paper/cf156c99e43cc894fb6d95103400a920af5bb9c1>.
- Sutianah, Tia, Petrus Sokibi, and Viar Dwi Kartika. 2024. "Sistem Klasifikasi Penentuan Penerima Bantuan Langsung Tunai Menggunakan Metode K-Nearest Neighbor." *Jurnal Eksplorasi Informatika* 13(2): 145–55. <https://eksplorasi.stikom-bali.ac.id/index.php/eksplorasi/article/view/907>.
- Taufiq, Akmal et al. 2024. "Comparison of Hyperparameter Tuning Techniques on KNN Algorithm to Find the Best K Value Using Grid

