

KLASIFIKASI SOAL MENGGUNAKAN MULTI-LABEL PROBLEM TRANSFORMATION DENGAN METODE RANDOM FOREST DAN K-NEAREST NEIGHBOR

Muhammad Rayhan Kurniawan*¹⁾, Oktariani Nurul Pratiwi²⁾, Faqih Hamami³⁾

1. Program Studi Sistem Informasi, Fakultas Rekayasa Industri, Universitas Telkom, Indonesia
2. Program Studi Sistem Informasi, Fakultas Rekayasa Industri, Universitas Telkom, Indonesia
3. Program Studi Sistem Informasi, Fakultas Rekayasa Industri, Universitas Telkom, Indonesia

Article Info

Kata Kunci: Klasifikasi Multi-Label; KNN; *Machine Learning*; *Problem Transformation*; *Random Forest*

Keywords: *KNN*; *Machine Learning*; *Multi-Label Classification*; *Problem Transformation*; *Random Forest*

Article history:

Received 8 October 2024

Revised 1 November 2024

Accepted 3 December 2025

Available online 1 March 2025

DOI :

<https://doi.org/10.29100/jifi.v10i1.5910>

* Corresponding author.

Muhammad Rayhan Kurniawan

E-mail address:

mrayhankurniawan@student.telkomuniversity.ac.id

ABSTRAK

Pendidikan merupakan komponen utama dalam membangun sumber daya manusia yang berkualitas. Ujian merupakan bagian dari proses evaluasi pendidikan untuk mengukur kemampuan siswa dalam memahami materi yang dipelajari. Proses ujian secara *online* memerlukan fasilitas mengenai pengelolaan soal, sehingga diperlukan klasifikasi untuk mengelompokkan soal sesuai dengan topiknya. Klasifikasi multi-label adalah proses pengelompokan data ke dalam beberapa kelas berdasarkan kesamaan ciri atau karakteristik data, di mana setiap soal dapat memiliki lebih dari satu topik. Penelitian ini berfokus pada pengklasifikasian soal mata pelajaran Bahasa Indonesia tingkat SMP dengan menggunakan metode *Problem Transformation* dan algoritma *Random Forest* serta *K-Nearest Neighbor* (K-NN). Metode *Problem Transformation* yang digunakan yaitu *Binary Relevance*, *Classifier Chain*, dan *Label Powerset*. Metrik evaluasi untuk menentukan kinerja terbaik yaitu berdasarkan *F1-Score* dengan *K-Fold Cross Validation*. Hasil penelitian menunjukkan bahwa algoritma *Random Forest* memberikan kinerja terbaik dibandingkan K-NN dengan nilai *F1-Score* terbaik di semua metode *Problem Transformation*. Nilai *F1-Score* terbaik dengan metode *Label Powerset* pada algoritma *Random Forest* sebesar 69%, dan K-NN sebesar 44%. Berdasarkan hasil tersebut, algoritma *Random Forest* dengan *Label Powerset* lebih efektif dalam mengklasifikasikan soal multi-label. Diharapkan penelitian ini dapat memberikan kontribusi dalam meningkatkan efisiensi pengelolaan soal ujian pada sistem pembelajaran online seperti *Learning Management System* (LMS).

ABSTRACT

Education is the main component in building quality human resources. Exams are part of the education evaluation process to measure students' ability to understand the material learned. The online exam process requires facilities regarding question management, so classification is needed to group questions according to their topics. Multi-label classification is the process of grouping data into several classes based on the similarity of data characteristics, where each question can have more than one topic. This research focuses on classifying Indonesian language subject questions at the junior high school level using the *Problem Transformation* method and *Random Forest* and *K-Nearest Neighbor* (K-NN) algorithms. The *Problem Transformation* methods used are *Binary Relevance*, *Classifier Chain*, and *Label Powerset*. The evaluation metric to determine the best performance is based on *F1-Score* with *K-Fold Cross Validation*. The results show that the *Random Forest* algorithm provides the best performance compared to K-NN with the best *F1-Score* value in all *Problem Transformation* methods. The best *F1-Score* value with *Label Powerset* method on *Random Forest* algorithm is 69%, and K-NN is 44%. Based on these results, the *Random Forest* algorithm with *Label Powerset* is more effective in classifying multi-label questions. It is hoped that this research can contribute to improving the efficiency of managing exam questions on online learning systems such as *Learning Management System* (LMS).

I. PENDAHULUAN

PENDIDIKAN adalah salah satu komponen utama dalam meningkatkan kesejahteraan hidup manusia serta membangun sumber daya manusia yang berkualitas. Pendidikan pada era saat ini telah mengalami perubahan dan inovasi dengan mengembangkan bentuk literasi yang baru, seperti literasi digital, literasi informasi, dan literasi media [1]. Inovasi ini tidak hanya berdampak pada cara penyampaian materi, tetapi juga pada metode evaluasi pembelajaran. Salah satu komponen penting dalam proses pendidikan adalah evaluasi, yang seringkali dilakukan melalui soal-soal ujian.

Menurut Arfandy & Musdar, ujian adalah metode yang digunakan untuk menilai kemampuan siswa dalam memahami dan menerapkan pengetahuan yang mereka peroleh terhadap materi yang telah diajarkan dengan menggunakan instrumen tertentu [2]. Instrumen yang digunakan dapat berupa kumpulan soal yang harus dikerjakan atau tugas untuk menghasilkan produk tertentu. Tujuan dari proses evaluasi ini adalah untuk menentukan seberapa baik siswa telah memahami materi yang disampaikan. Meskipun demikian, pelaksanaan ujian secara konvensional seringkali menimbulkan beberapa kendala, seperti keterbatasan waktu dan ruang, serta kesulitan dalam penilaian dan pengolahan hasil ujian. Oleh karena itu, teknologi informasi hadir sebagai solusi untuk mengatasi berbagai permasalahan tersebut.

Salah satu inovasi dari teknologi informasi yang sangat bermanfaat di bidang Pendidikan adalah *E-Learning* [3]. *E-learning* memiliki peranan yang sangat penting dalam era pendidikan digital saat ini, terutama selama pandemi *Covid-19* dan setelahnya. UNESCO memperkirakan bahwa lebih dari 90% siswa di seluruh dunia tidak dapat bersekolah akibat pandemi *Covid-19*, dengan lebih dari 1,5 miliar siswa terkena dampaknya [4]. Pemerintah juga mengharuskan penerapan sistem Pembelajaran Jarak Jauh (PJJ) selama pandemi, yang menjadikan pandemi *Covid-19* sebagai salah satu pendorong utama peningkatan penggunaan *E-Learning* di Indonesia [5]. *E-learning* merupakan salah satu bentuk model pembelajaran yang memanfaatkan perangkat elektronik sebagai media utamanya, serta merupakan jenis pembelajaran berbasis web yang dapat diakses melalui internet [6]. *E-Learning* memiliki kemampuan untuk otomatisasi proses penilaian dan pengolahan hasil ujian, sehingga dapat mengefesiesikan waktu dan tenaga. Salah satu bentuk model *E-learning* dalam pembelajaran adalah *Pinterran* dan LMS, atau singkatan dari *Learning Management System*. LMS (*Learning Management System*) adalah sistem perangkat lunak pembelajaran yang kuat dengan menggunakan teknologi web untuk meningkatkan pembelajaran siswa dengan cara yang lebih efisien [7].

Proses ujian secara *online* seperti dengan LMS memerlukan fasilitas mengenai pengelolaan siswa, pengajar, dan salah satunya adalah pengelolaan soal [8]. Oleh karena itu, dalam pengelolaan soal ujian diperlukan proses klasifikasi soal menjadi beberapa kategori sesuai dengan topiknya. Tujuannya adalah agar soal-soal tersebut menjadi sebuah soal yang terstruktur sehingga memudahkan siswa ataupun pengajar untuk mengetahui topik soal [3], seperti soal yang diujikan pada LMS. Klasifikasi merupakan salah satu metode yang terdapat pada *machine learning* atau pembelajaran mesin untuk mengelompokkan objek berdasarkan ciri-ciri yang dimiliki oleh objek klasifikasi [9]. Terdapat beberapa metode algoritma klasifikasi pada *machine learning*, diantaranya yaitu *K-Nearest Neighbors*, *Naïve Bayes*, *Support Vector Machine*, *Decision Tree*, *Logistic Regression* dan *Random Forest* [10].

Namun seringkali sebuah soal memiliki lebih dari satu topik, sehingga diperlukan penyelesaian masalah klasifikasi multi-label. Dalam klasifikasi multi-label, setiap soal dapat masuk kedalam satu atau lebih kelompok, dalam kasus ini adalah topik. Terdapat beberapa pendekatan pada implementasi klasifikasi multi-label, salah satunya adalah *Problem Trasformation* [11]. Metode *Problem Transformation* dapat menyelesaikan masalah klasifikasi multi-label dengan mengubahnya menjadi masalah klasifikasi *single label* dan kemudian menggabungkannya kembali hasilnya ke dalam bentuk multi-label, dengan beberapa metode seperti *Binary Relevance*, *Classifier Chain*, dan *Label Powerset* [12].

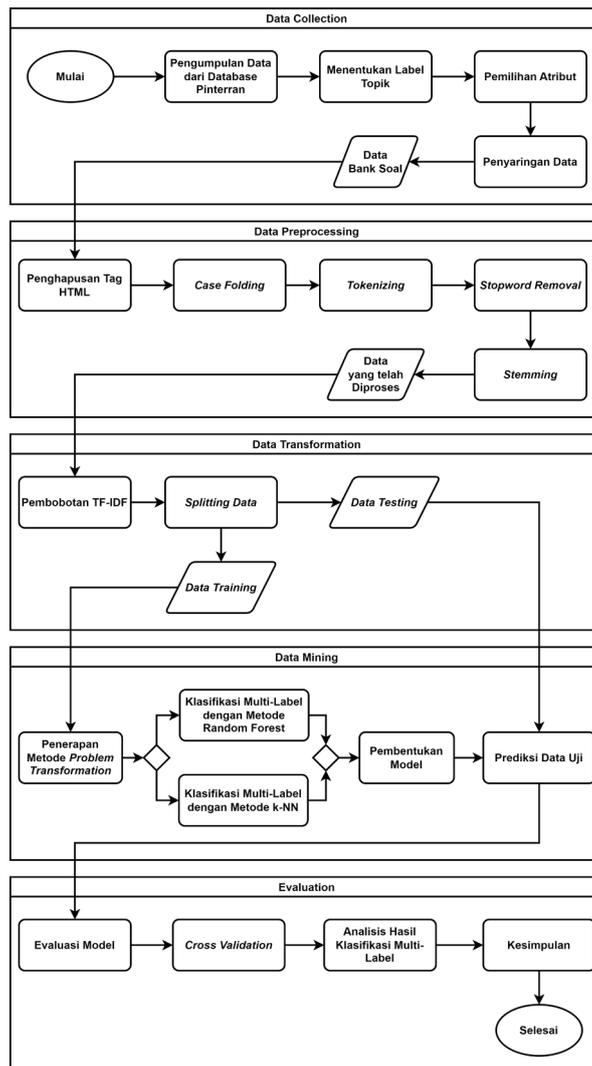
Penelitian mengenai klasifikasi multi-label teks dengan pendekatan *Problem Transformation* sudah pernah dilakukan oleh beberapa peneliti terdahulu, seperti klasifikasi multi-label pada teks hadis dengan algoritma *Random Forest* yang dilakukan oleh Wiraguna [12] yang menemukan *Problem Transformation Binary Relevance* tanpa *stemming* memberikan performa *hamming loss* terbaik sebesar 0,0663. Serta penelitian lain pada klasifikasi multi-label teks berita dengan algoritma K-NN yang dilakukan oleh Ramadhani [13] menemukan bahwa metode *Binary Relevance* memiliki nilai *hamming loss* terbaik sebesar 0,11625, dan penelitian dengan teks pertanyaan Kotakode menggunakan algoritma KNN yang dilakukan oleh Fitriani & Yustanti [11] menemukan bahwa metode *Label Powerset* memberikan nilai *F1-Score* terbaik sebesar 87%. Namun, belum banyak penelitian yang membandingkan algoritma *Random Forest* dan *K-Nearest Neighbors* pada penerapan klasifikasi multi-label dengan pendekatan *Problem Transformation* untuk teks soal.

Oleh karena itu, penelitian ini bertujuan untuk melakukan klasifikasi multi-label pada teks soal mata pelajaran Bahasa Indonesia tingkat SMP dengan pendekatan *Problem Transformation* dan algoritma *Random Forest* serta

K-Nearest Neighbors, dan menentukan metode *Problem Transformation* dan algoritma yang terbaik. Dengan penyesuaian dan optimasi yang tepat, metode dan algoritma ini dapat menjadi solusi yang efektif dan efisien dalam membantu proses pengelolaan soal pada database pembelajaran online terutama pada LMS. Dengan demikian, penelitian ini diharapkan dapat meningkatkan kualitas pendidikan di Indonesia.

II. METODE PENELITIAN

Metode penelitian yang digunakan pada penelitian ini berdasarkan *framework* KDD (*Knowledge Discovery in Database*). KDD adalah proses untuk mengidentifikasi pola yang berguna, serta dapat dipahami dalam data [14]. KDD bertujuan untuk menemukan informasi yang berharga dan pola-pola yang tersembunyi di dalam data, dengan pengembangan metode dan teknik untuk membuat data menjadi bermakna. Terdapat beberapa tahapan dalam KDD, yaitu *data selection*, *data preprocessing*, *data transformation*, *data mining*, dan *evaluation* yang dijabarkan pada Gambar 1.



Gambar. 1. Sistematika Metode Penelitian KDD

A. Data Selection

Data Selection merupakan tahap awal dalam *framework* KDD (*Knowledge Discovery in Database*). *Data selection* adalah proses memilih data dari sekumpulan data operasional sebelum mendapatkan pengetahuan dan menemukan informasi dalam sebuah basis data [15]. Tahap pengumpulan data bertujuan untuk melakukan penelitian dengan cara mengumpulkan data untuk menemukan permasalahan yang terjadi. Data yang dikumpulkan bersumber dari database lms Pinterran, yang merupakan sebuah platform pembelajaran online di Indonesia. Pada database ini terdapat beberapa kumpulan soal dengan beragam mata pelajaran pada tingkat SMP mulai dari Matematika, Bahasa Indonesia, Bahasa Inggris, dan Ilmu Pengetahuan Alam.

Penelitian ini berfokus pada soal mata pelajaran Bahasa Indonesia, dengan jumlah total 700 item soal yang akan dikumpulkan menjadi bank soal. Kumpulan soal Bahasa Indonesia yang digunakan dari database Pinterran ini

sudah sesuai dengan materi pendidikan tingkat SMP pada kurikulum 2013 dan telah diterapkan pada sistem latihan soal di Pinterran. Dataset bank soal ini kemudian diubah menjadi dokumen dalam format *comma separated values* (.csv) dan menentukan atribut sebagai label topik. Selanjutnya, dataset ini akan disaring dengan beberapa kriteria berdasarkan topik yang digunakan, untuk kemudian dilakukan proses klasifikasi berdasarkan kategori topik.

B. Data Preprocessing

Pada tahap praproses data, peneliti melakukan pembersihan teks konten soal dan opsi pilihan ganda menggunakan bahasa pemrograman *python*. *Data preprocessing* adalah serangkaian teknik yang digunakan untuk meningkatkan kualitas data mentah sebelum dilakukan pemrosesan oleh model klasifikasi [16]. *Data preprocessing* pada penelitian ini dikenal sebagai *text preprocessing*, karena data yang digunakan pada penelitian ini berupa teks. Metode *preprocessing text* ini merupakan bagian yang penting pada penerapan dalam *text mining*. Tahap pembersihan teks ini mencakup dari penghapusan tag html, penghapusan karakter selain alfabet dan numerik, dan tahapan umum praproses pada teks seperti *case folding*, *tokenizing*, *stopword removal*, dan *stemming* [17]. Hasil dari *preprocessing text* ini dapat mempengaruhi tingkat akurasi klasifikasi dan waktu komputasi yang dibutuhkan [18].

1) Penghapusan Tag HTML

Tahap penghapusan tag HTML ini bertujuan untuk mengubah format data yang memiliki atribut tag html menjadi teks yang sesuai agar dapat dilakukan tahap *preprocessing* selanjutnya.

2) Case Folding

Case Folding bertujuan untuk mengubah semua huruf dalam dokumen teks soal menjadi huruf kecil secara keseluruhan [18]. Proses ini membantu mengatasi masalah variasi huruf besar dan kecil yang tidak konsisten dalam dokumen teks, sehingga dapat digunakan untuk metode *preprocessing* selanjutnya.

3) Tokenizing

Tokenizing yaitu sebuah proses memisahkan kalimat menjadi kata-kata tunggal [19]. Strategi umum dalam *tokenizing* adalah memotong kata berdasarkan spasi atau *whitespace* serta penghapusan karakter tanda baca [3]. Pada dasarnya, proses *tokenizing* memecah urutan karakter menjadi kalimat dan selanjutnya memecah kalimat tersebut menjadi token.

4) Stopword Removal

Stopword removal atau *filtering* melibatkan pemeriksaan setiap kata yang dihasilkan, seperti kata hubung atau kata-kata lain yang tidak memiliki hubungan dengan klasifikasi teks, maka kata-kata tersebut akan dihapus karena memiliki dimensi ruang yang terlihat berat untuk di analisis [19]. Kata-kata umum dalam teks dokumen seperti kata sambung, kata benda, dan sebagainya yang tidak memberikan makna pada dokumen akan dihapus karena tidak dianggap sebagai kata kunci dalam proses *text mining* [18]. Sehingga setelah melalui tahap ini, dokumen teks hanya menyisakan kata-kata yang penting sebagai kata kunci dalam klasifikasi soal.

5) Stemming

Proses *stemming* yaitu mengubah hasil *stopwords* menjadi bentuk kata dasar [19]. Proses *stemming* ini merupakan tahap paling penting dalam analisis teks, karena *stemming* membantu menghilangkan bentuk kata yang memiliki imbuhan berbeda dengan membuat kata tersebut memiliki makna yang sama [20]. Proses ini dapat berpengaruh terhadap model klasifikasi. Hasil *stemming* yang baik dapat mempengaruhi kesuksesan aplikasi analisis teks [18].

C. Data Transformation

Tahap transformasi data merupakan tahap mengubah data yang telah diproses menjadi format yang sesuai untuk penambangan data. Tahap ini meliputi pembobotan TF-IDF untuk mengubah teks menjadi representasi numerik, serta pembagian data (*data splitting*) untuk membagi data menjadi data training dan data testing untuk memastikan bahwa model dapat dievaluasi dengan benar terhadap data yang belum pernah dilihat sebelumnya.

1) Pembobotan TF-IDF

TF-IDF merupakan singkatan dari *Term Frequency-Inverse Document Frequency*. TF-IDF adalah sebuah teknik yang digunakan untuk menentukan seberapa penting suatu kata dalam sebuah teks atau dokumen [21]. Teknik ini mengubah teks menjadi representasi numerik untuk memungkinkan pemrosesan oleh model. Prinsip kerja TF-IDF melibatkan perhitungan tingkat pentingnya kata dalam dokumen, yang kemudian dikalikan dengan frekuensi kemunculannya (*term frequency*) dan dibagi dengan logaritma dari total dokumen dibagi jumlah dokumen yang

mengandung kata tersebut [22]. Hasil dari perhitungan ini berguna untuk mengenali kata-kata yang memiliki signifikansi tertinggi dan mewakili teks dalam dokumen. Rumus perhitungan TF-IDF ditulis pada (1).

$$TF-IDF_{td} = tf_{td} \times \log\left(\frac{D+1}{df_t+1}\right) + 1 \quad (1)$$

Keterangan:

$TF - IDF_{td}$ = Nilai TF-IDF pada kata ke- t pada dokumen ke- d

tf_{td} = Frekuensi kata ke- t pada dokumen ke- d

D = Total dokumen

df_t = Jumlah dokumen yang mengandung kata ke- t

2) Data Splitting

Data splitting atau pembagian data dilakukan untuk memisahkan data yang akan digunakan untuk melatih model, menguji kinerja model, dan memvalidasi model. Proporsi pembagian data pada model klasifikasi umumnya yaitu 80:20 (data *training* 80%, data *testing* 20%), proporsi ini pernah diterapkan pada penelitian milik Fitriani & Yustanti [11] dan Kautsar [23]. Sehingga pembagian data pada penelitian ini menggunakan perbandingan 80:20 untuk memisahkan *data training* dengan *data testing* dari keseluruhan data yang digunakan.

D. Data Mining

Data mining merupakan proses ekstraksi dan identifikasi informasi atau pengetahuan dari kumpulan data yang besar dan tersebar di berbagai sumber [15]. *Data mining* termasuk pada langkah dalam proses KDD yang melibatkan analisis data untuk menemukan pola dan hubungan yang bermakna [14]. Proses ini dapat membantu dalam memahami data dan membuat keputusan yang tepat berdasarkan temuan dari data tersebut.

1) Problem Transformation

Problem Transformation adalah sebuah metode pendekatan yang mengubah masalah klasifikasi multi-label menjadi satu atau lebih klasifikasi *single label* [22]. Terdapat beberapa jenis metode dalam *problem transformation*, diantaranya adalah *Binary Relevance*, *Classifier Chain*, dan *Label Powerset* [13].

a) Binary Relevance

Binary Relevance adalah salah satu metode dari *problem transformation* pada kasus pengklasifikasian multi label. Metode ini dilakukan dengan menerapkan *binary classifier* pada setiap label dan *output* yang dihasilkan akan digabungkan untuk membangun kumpulan label hasil prediksi [13].

b) Classifier Chain

Classifier Chain adalah salah satu metode *problem transformation* untuk kasus multi-label selain *Binary Relevance* (BR). Metode ini melakukan pelatihan sejumlah k *classifier* sesuai dengan jumlah label yang ada, mirip dengan metode *Binary Relevance*. *Classifier* pertama dilatih menggunakan data input asli. Kemudian hasil prediksi dari label pertama ditambahkan sebagai fitur input baru untuk melatih *classifier* kedua [11]. Proses ini berlanjut secara berurutan hingga *classifier* atau label ke- k . Dengan menggunakan pendekatan *Classifier Chain* ini, diharapkan dependensi antar label dapat diperhitungkan dengan lebih baik [13].

c) Label Powerset

Label Powerset (LP) adalah metode *problem transformation* untuk kasus multi-label yang lebih sederhana dari *Binary Relevance* (BR) karena tidak perlu melatih banyak model dan menggabungkan output yang dihasilkan menjadi sekumpulan label. *Label powerset* langsung mengubah kumpulan data yang multi label menjadi satu kumpulan data yang single label, dengan cara mentransformasikan setiap kombinasi label yang berbeda pada data multi label menjadi kelas yang berbeda pada data single label [11].

2) Random Forest

Random Forest merupakan salah satu algoritma *supervised learning* yang dapat dianalogikan sebagai gabungan dari *decision tree* [24]. Pada buku yang disusun oleh Bedy Purnama dengan judul “Pengantar Machine Learning: Konsep dan Praktikum Dengan Contoh Latihan Berbasis R dan Python” [25] menyebutkan bahwa metode *random forest* ini diperkenalkan oleh Adele Cutler dan Leo Breidman. Menurut Breidman metode *random forest* ini terdiri dari beberapa pohon keputusan (*decision tree*) di mana setiap pohon dibuat menggunakan data acak yang dipilih secara independen dengan distribusi yang sama [26]. Dengan demikian, algoritma *random forest* dapat membantu meningkatkan akurasi dan kestabilan prediksi dengan menggabungkan hasil dari beberapa *decision tree*. *Decision tree* adalah algoritma klasifikasi yang menggunakan struktur seperti pohon atau *flowchart* untuk mencari solusi

dari permasalahan. Cara kerja *decision tree* yaitu mengubah data menjadi pohon keputusan dan diberi aturan-aturan keputusan [25].

Random forest dapat dianggap sebagai turunan dari *decision tree*, sehingga algoritma *random forest* ini lebih canggih dibandingkan *decision tree* [25]. Model *random forest* ini memiliki kelebihan seperti mempunyai tingkat akurasi yang baik, kinerja yang bagus, dan efektif dalam menangani set data yang besar dan merupakan teknik yang cukup baik untuk melakukan klasifikasi terhadap data yang memiliki nilai yang hilang [27].

3) *K-Nearest Neighbors*

K-Nearest Neighbors (K-NN) adalah algoritma *supervised learning* yang menentukan hasil *query instance* baru melalui klasifikasi berdasarkan nilai tetangga terbanyak dari kategori yang paling sering muncul [28]. Algoritma K-NN bekerja dengan mengklasifikasikan objek baru berdasarkan data pelatihan yang memiliki jarak terdekat dengan objek tersebut [29]. Prinsip utama dari K-NN yaitu menemukan jarak terdekat antara data yang dievaluasi dengan k tetangga terdekatnya [30]. Jumlah k tetangga dapat diatur melalui parameter pada K-NN.

Perhitungan *Euclidean Distance* digunakan untuk mengukur jarak antara dua titik variabel pada K-NN. Semakin kecil jarak antara dua titik, semakin mirip dan dekat kedua titik tersebut [31]. *Euclidean Distance* dianggap efektif ketika data baru memiliki jarak minimum dan kemiripan tinggi dengan data pelatihan [28]. Perhitungan jarak pada K-NN dilakukan menggunakan rumus *Euclidean Distance* yang ditunjukkan pada (2).

$$d(P, Q) = \sqrt{\sum_{i=1}^n (p_i - q_i)^2} \quad (2)$$

Keterangan:

- $d(P, Q)$ = jarak
- p_i = data training
- q_i = data testing
- i = variabel data
- n = dimensi data

E. Evaluation

Evaluasi performansi merupakan tahap penting dalam pengembangan model *machine learning*. Tujuannya adalah untuk mengukur seberapa baik model tersebut bekerja pada data baru yang belum pernah dilihat sebelumnya [32]. Dalam penelitian ini, evaluasi performansi dilakukan menggunakan *confusion matrix* dan *k-fold cross validation* dari *library scikit learn*, dan evaluasi yang mencakup evaluasi nilai-nilai seperti *accuracy*, *F1-Score*, *recall*, dan *precision*. Evaluasi pada kelas multi label berbeda dengan evaluasi pada umumnya. Untuk klasifikasi multi-label, *confusion matrix* tidak disajikan dalam bentuk tunggal seperti pada klasifikasi biner atau multi-kelas. Sebaliknya, setiap label memiliki *confusion matrix* tersendiri yang menggambarkan performa prediksi model terhadap label tersebut.

1) *Confusion Matrix*

Confusion Matrix merupakan alat ukur performa model klasifikasi dalam *machine learning*, di mana hasil output dapat terdiri dari dua atau lebih kelas. *Confusion matrix* berbentuk sebuah tabel dengan 4 kombinasi berbeda antara nilai prediksi dan nilai aktual [33]. Tabel ini mencakup empat istilah utama yang mewakili hasil dari proses klasifikasi, yaitu *True Positive*, *True Negative*, *False Positive*, dan *False Negative* [34]. Persamaan (3-5) merupakan rumus *confusion matrix* untuk menghitung *accuracy*, *precision*, dan *recall*.

$$accuracy = \frac{TP + TN}{Total} \quad (3)$$

$$precision = \frac{TP}{TP + FP} \quad (4)$$

$$recall = \frac{TP}{TP + FN} \quad (5)$$

F1-Score adalah sebuah matriks evaluasi yang digunakan untuk menghitung performa model klasifikasi, terutama dalam multilabel. Matriks ini dapat mempertimbangkan banyak label dan frekuensi kemunculan kata kunci [35]. Nilai *F1-Score* merupakan *harmonic mean* dari nilai *precision* dan *recall* [11]. *F1-Score* ditulis pada (6).

$$F1\ score = 2 \times \frac{precision \times recall}{precision + recall} \quad (6)$$

Dalam kasus klasifikasi multi-label, penilaian akurasi tidak sama dengan klasifikasi multi-kelas ataupun *single label*. Penilaian akurasi dalam multi-label menggunakan *subset accuracy*. *Subset accuracy* juga dikenal sebagai 0/1 subset akurasi atau *classification accuracy*, adalah matriks evaluasi yang digunakan dalam klasifikasi multilabel [36]. Matriks ini sangat ketat, karena hanya memberikan nilai 1 (*True*) jika seluruh label prediksi untuk sebuah *instance* sepenuhnya sesuai dengan semua label sebenarnya (aktual). Sebaliknya, Jika ada satu label saja yang tidak sesuai, nilainya adalah 0 (*False*). *Subset accuracy* ditulis pada (7).

$$subset\ accuracy = \frac{\text{Subset prediksi tepat}}{\text{Total dokumen}} \quad (7)$$

2) K-Fold Cross Validation

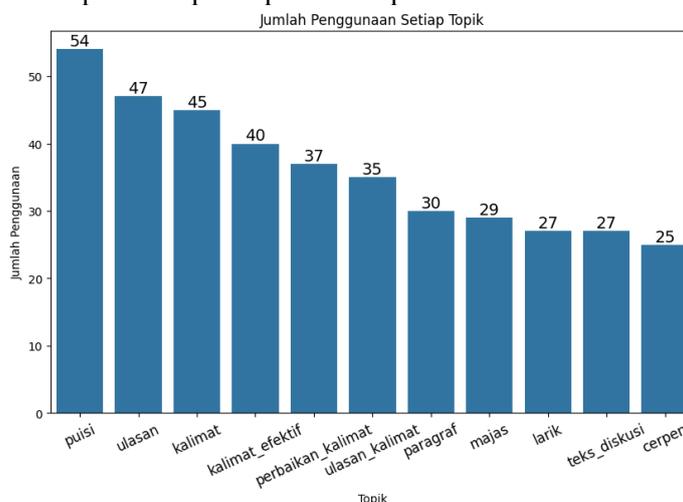
K-Fold Cross validation merupakan metode validasi model untuk mengevaluasi tingkat keakuratan hasil analisis [37]. Pada *K-Fold cross validation*, dataset awal dibagi menjadi sebanyak K sub-sampel secara acak yang berukuran sama, yang sering disebut sebagai "*fold*". *K-Fold cross validation* membantu mengurangi varians estimasi kinerja model dibandingkan dengan penggunaan satu kali pembagian data pelatihan/pengujian, *K-Fold cross validation* memungkinkan semua observasi digunakan untuk pelatihan dan pengujian dengan membagi data menjadi *k* bagian [38], serta memastikan setiap observasi yang digunakan dalam set pengujian tepat satu kali.

III. HASIL DAN PEMBAHASAN

A. Pengumpulan Data

Pada yang digunakan bersumber dari database lms Pinterran dalam bentuk sql. Database ini terdapat 55 tabel yang berbeda, setiap tabelnya memiliki fungsi dan tujuan masing-masing. Penelitian ini menggunakan 3 dari 55 tabel yang digunakan untuk mengumpulkan data soal yang akan digunakan pada topik penelitian, yaitu tabel 'bank_soal', 'soal_has_tags', 'tags'. Fokus soal yang digunakan adalah soal mata pelajaran Bahasa Indonesia, yang terdiri dari 700 item soal dengan 4 opsi pilihan.

Peneliti menentukan kolom 'tag' sebagai label topik di setiap soal. Alasan peneliti menggunakan kolom 'tag' sebagai topik setiap soal dikarenakan kolom ini merupakan representasi yang sesuai dengan isi teks, yang merupakan tagar dari setiap soalnya. Setiap soal yang berbeda memiliki satu atau lebih topik dari kolom 'tag', sehingga penelitian mengangkat masalah klasifikasi multi-label. Kolom 'tag' pada dataset ini memiliki 306 tagar yang berbeda, di mana terdapat 154 tagar yang hanya digunakan pada satu soal saja, serta beberapa tagar lainnya memiliki jumlah soal yang sedikit. Penggunaan tagar dengan soal yang sedikit ini mengakibatkan model sulit untuk memahami pola soal dari setiap topik tagar tersebut. Agar model dapat mengenali pola soal dengan baik dari masing-masing topik tagar, peneliti melakukan penyaringan data dengan hanya menggunakan topik tagar yang terdapat pada 25 soal atau lebih dalam keseluruhan dataset. Setelah dilakukan penyaringan topik tersebut, didapatkan 11 topik unik yang berbeda dengan 191 soal yang akan digunakan pada klasifikasi multi-label ini. Jumlah penggunaan soal pada setiap label topik dapat dilihat pada Gambar 2.



Gambar. 2. Jumlah Penggunaan Setiap Topik

Setelah didapatkan sampel dokumen dari hasil praproses data pada Tabel III, selanjutnya peneliti melakukan perhitungan TF-IDF. Pada Tabel IV merupakan hasil perhitungan TF-IDF pada setiap kata dari sampel dokumen dengan nilai *term frequency*, *document frequency*, *inverse document frequency*, dan nilai akhir TF-IDF setelah dilakukan normalisasi.

TABEL IV
 HASIL PERHITUNGAN TF-IDF

Term	tf			df	D/df	idf	TF-IDF		
	Dok01	Dok02	Dok03				Dok01	Dok02	Dok03
bahasa	1	0	0	1	0.33	1.69314	0.3162	0	0
baku	1	0	0	1	0.33	1.69314	0.3162	0	0
batas	0	1	0	1	0.33	1.69314	0	0.2137	0
belas	0	1	0	1	0.33	1.69314	0	0.2137	0
berani	0	1	0	1	0.33	1.69314	0	0.2137	0
ciri	2	0	0	1	0.33	1.69314	0.6325	0	0
curah	0	1	0	1	0.33	1.69314	0	0.2137	0
derita	0	0	1	1	0.33	1.69314	0	0	0.2227
desak	0	0	1	1	0.33	1.69314	0	0	0.2227
gerak	0	0	1	1	0.33	1.69314	0	0	0.2227
kaku	0	0	2	1	0.33	1.69314	0	0	0.4455
kasih	0	2	0	1	0.33	1.69314	0	0.4275	0
kasihan	0	1	0	1	0.33	1.69314	0	0.2137	0
kecil	0	1	0	1	0.33	1.69314	0	0.2137	0
korban	0	2	0	1	0.33	1.69314	0	0.4275	0
kutip	0	0	1	1	0.33	1.69314	0	0	0.2227
persuasif	1	0	0	1	0.33	1.69314	0.3162	0	0
pohon	0	0	1	1	0.33	1.69314	0	0	0.2227
poster	1	0	0	1	0.33	1.69314	0.3162	0	0
puisi	0	2	1	2	0.66	1.28768	0	0.3251	0.1694
puncak	0	0	1	1	0.33	1.69314	0	0	0.2227
sayang	0	1	0	1	0.33	1.69314	0	0.2137	0
sepi	0	0	2	1	0.33	1.69314	0	0	0.4455
singkat	1	0	0	1	0.33	1.69314	0.3162	0	0
sunyi	0	0	1	1	0.33	1.69314	0	0	0.2227
tarik	1	0	0	1	0.33	1.69314	0.3162	0	0
tekan	0	0	2	1	0.33	1.69314	0	0	0.4455
tema	0	1	1	2	0.66	1.28768	0	0.1625	0.1694
timbang	0	2	0	1	0.33	1.69314	0	0.4275	0

D. Data Splitting

Pembagian data dimulai dari memisahkan data training dengan data testing dengan perbandingan 80:20 (data *training* 80%, data *testing* 20%). Setelah pembagian data, maka terdapat beberapa detail jumlah soal setiap data, serta penggunaan topik label unik pada setiap data. Detail jumlah soal dan label tersebut dijabarkan pada Tabel V.

TABEL V
 DETAIL JUMLAH SOAL DAN LABEL PADA TIAP DATASET

	Data		Total
	Training	Testing	
Jumlah Soal	152	39	191
Cerpen	20	5	25
Kalimat	32	13	45
Kalimat_efektif	30	10	40
Larik	18	9	27
Majas	19	10	29
Paragraf	26	4	30
Perbaikan_kalimat	27	10	35
Puisi	42	12	54
Teks_diskusi	21	6	27
Ulasan	35	12	47
Ulasan_kalimat	25	10	35

E. Hyperparameter Tuning

Hyperparameter tuning adalah proses penting dalam pengembangan model machine learning yang bertujuan untuk mengoptimalkan kinerja model dengan mencari kombinasi parameter terbaik [11]. Dalam penelitian ini, peneliti menggunakan metode *Grid SearchCV* dari *library scikit-learn* untuk melakukan tuning parameter. *Grid SearchCV* adalah teknik yang secara sistematis menjelajahi ruang parameter dengan menguji semua kombinasi yang mungkin dari parameter yang telah ditentukan [35]. Setelah melakukan *hyperparameter tuning* pada beberapa parameter yang dimiliki oleh algoritma *random forest*, didapatkan hasil parameter terbaik pada Tabel VI.

TABEL VI
 NILAI PARAMETER TERBAIK RANDOM FOREST

Parameter	Nilai
n_estimators	100
bootstrap	False
max_features	200
class_weight	'balanced'
oob_score	False
criterion	'gini'

Selain *random forest*, peneliti juga melakukan *hyperparameter tuning* pada algoritma K-NN. Setelah melakukan *hyperparameter tuning* pada beberapa parameter yang dimiliki oleh algoritma K-NN, didapatkan hasil parameter terbaik pada Tabel VII berikut.

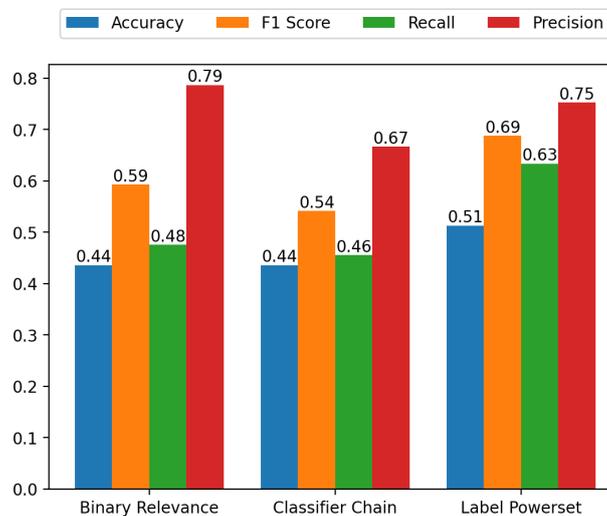
TABEL VII
 NILAI PARAMETER TERBAIK K-NN

Parameter	Nilai
n_neighbors	5
weights	'distance'
algorithm	'auto'
leaf_size	20
p	2
metric	'minkowski'

F. Pengujian Random Forest

Peneliti menggunakan nilai akurasi, *precision*, *recall*, dan *F1-Score* dari setiap pengujian pada data testing untuk dievaluasi. Pengujian dilakukan dengan metode *problem transformation* dan algoritma klasifikasi yang berbeda. Peneliti menggunakan tiga metode *problem transformation*, yaitu *Binary Relevance (BR)*, *Classifier Chain (CC)*, dan *Label Powerset (LP)* pada setiap algoritma *Random Forest* dan K-NN.

Dalam pengujian algoritma *Random Forest*, hasil evaluasi menunjukkan variasi kinerja di antara ketiga metode tersebut. Gambar 3 menunjukkan beberapa nilai evaluasi hasil pengujian data *testing* algoritma *Random Forest* dengan tiga pendekatan *problem transformation* yang berbeda.



Gambar 3. Hasil Pengujian Random Forest

Dari Gambar 3 diketahui metode *Label Powerset* memiliki nilai *F1-Score* terbaik yaitu 69%, kemudian diikuti oleh *Binary Relevance* dan *Classifier Chain* dengan nilai *F1-Score* 59% dan 54%. Metode *Label Powerset* juga menunjukkan kinerja terbaik dengan nilai akurasi yang lebih baik dibandingkan *Binary Relevance* dan *Classifier Chain* dengan nilai 51%. Sedangkan nilai akurasi dari metode *Binary Relevance* dan *Classifier Chain* hanya 44%. Dari hasil evaluasi ini, metode *Label Powerset* lebih efektif dalam menangkap hubungan antar label yang kompleks, yang dapat membuat metode ini lebih mampu mengenali kombinasi label yang benar dalam dataset.

Dari hasil evaluasi yang ditunjukkan pada Gambar 3, dapat dilihat bahwa metode *problem transformation Label Powerset* memberikan nilai *F1-Score* tertinggi dibandingkan metode *Binary Relevance* dan *Classifier Chain*. Untuk mendapatkan pemahaman yang lebih mendalam mengenai kinerja algoritma *Random Forest* dengan pendekatan *Label Powerset*, peneliti menganalisis *confusion matrix* yang disajikan pada Tabel VIII.

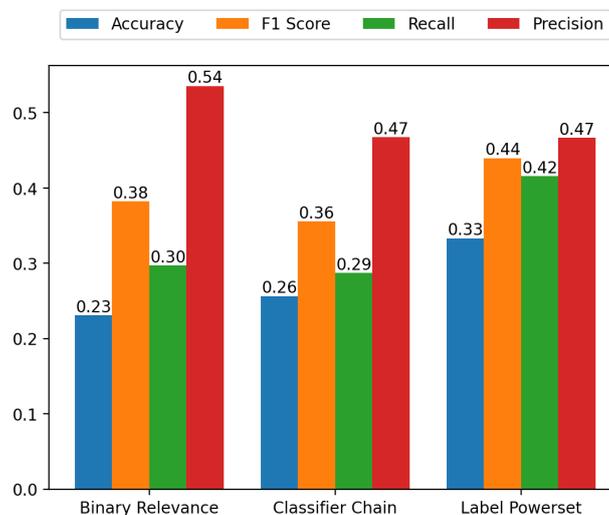
TABEL VIII
 CONFUSION MATRIX RANDOM FOREST DENGAN LABEL POWERSSET

Label	True Positive	False Positive	False Negative	True Negative
Cerpen	5	0	0	34
Kalimat	6	2	7	24
Kalimat_efektif	5	2	5	27
Larik	6	2	3	28
Majas	6	1	4	28
Paragraf	4	2	0	33
Perbaikan_kalimat	5	2	5	27
Puisi	11	2	1	25
Teks_diskusi	5	5	1	28
Ulasan	6	1	6	26
Ulasan_kalimat	5	2	5	27
Total:	64	21	37	307

Dari Tabel VIII dapat dilihat bahwa beberapa label topik seperti topik cerpen dan puisi menunjukkan performa klasifikasi yang baik dengan jumlah *true positive* sebanyak 5 dan 11, serta *false negative* dan *false positive* yang rendah. Misalnya pada topik puisi memiliki nilai 11 *true positive*, 2 *false positive*, dan 1 *false negative*, yang menunjukkan bahwa model sangat akurat dalam mengklasifikasikan teks dengan topik puisi. Sebaliknya, pada topik teks_diskusi, model menunjukkan prediksi kesalahan dengan 5 *false positive*, yang menunjukkan bahwa model sering keliru mengklasifikasikan teks lain sebagai topik teks_diskusi. Selain itu, topik kalimat memiliki 7 *false negative* yang menunjukkan bahwa model sering gagal mendeteksi label ini ketika beberapa soal yang diprediksi *false*, namun seharusnya *true*.

G. Pengujian K-NN

Dalam pengujian algoritma K-NN, hasil evaluasi menunjukkan variasi kinerja di antara ketiga metode problem transformation yang digunakan. Peneliti menggunakan nilai akurasi, *F1-Score*, *recall*, dan *precision* untuk menganalisis hasil evaluasi pada Gambar 4.



Gambar 4. Hasil Pengujian K-NN

Pada Gambar 4 diketahui nilai *F1-Score* tertinggi terdapat pada metode *Label Powerset* dengan nilai 44%. Metode ini lebih baik dari metode *Binary Relevance* dan *Classifier Chain* yang memiliki nilai *F1-Score* 38% dan 36%. Metode *Label Powerset* juga lebih unggul dalam hal akurasi dengan nilai 33%. Hal ini membuktikan bahwa metode *Label Powerset* lebih efektif dalam menangkap kombinasi label secara langsung, sehingga memberikan keseimbangan yang lebih baik antara *precision* dan *recall*. Selain memiliki nilai *F1-Score* tertinggi, metode *Label Powerset* juga lebih unggul dalam hal akurasi. Kemampuan *Label Powerset* untuk mempertimbangkan label sebagai satu kesatuan yang utuh membuatnya lebih unggul dalam konteks klasifikasi multi-label.

Dari hasil evaluasi yang ditunjukkan pada Gambar 4, dapat dilihat bahwa metode *problem transformation Label Powerset* memberikan nilai *F1-Score* dan akurasi tertinggi dibandingkan metode *Binary Relevance* dan *Classifier Chain*. Untuk mendapatkan pemahaman yang lebih mendalam mengenai kinerja algoritma K-NN dengan pendekatan *Label Powerset*, peneliti menganalisis *confusion matrix* yang disajikan pada Tabel IX.

TABEL IX
 CONFUSION MATRIX RANDOM FOREST DENGAN LABEL POWERSSET

Label	True Positive	False Positive	False Negative	True Negative
Cerpen	5	4	0	30
Kalimat	5	5	8	21
Kalimat_efektif	3	7	7	22
Larik	3	1	6	29
Majas	3	0	7	29
Paragraf	2	3	2	32
Perbaikan_kalimat	3	7	7	22
Puisi	10	4	2	23
Teks_diskusi	1	4	5	29
Ulasan	4	6	8	21
Ulasan_kalimat	3	7	7	22
Total:	42	48	59	280

Dari Tabel IX dapat dilihat dari beberapa label topik, topik puisi menunjukkan performa klasifikasi yang baik dengan jumlah *true positive* yang tinggi. Topik puisi memiliki 10 *true positive*, 4 *false negative*, dan 2 *false positive*, menunjukkan bahwa model ini cukup akurat dalam mendeteksi teks dengan topik ini. Sebaliknya, topik ulasan dan kalimat memiliki 8 *false negative* yang menunjukkan bahwa model sering gagal mendeteksi label ini ketika beberapa soal yang diprediksi *false*, namun seharusnya *true*.

H. Evaluasi Perbandingan Algoritma

Pada bagian ini, peneliti mengevaluasi perbandingan antara algoritma *Random Forest* dan *K-Nearest Neighbors* (K-NN) dengan beberapa metrik evaluasi, yaitu akurasi, *F1-Score*, *recall*, dan *precision*. Pada Tabel X berikut adalah beberapa nilai evaluasi dari algoritma dan metode problem transformation yang berbeda.

TABEL X
 PERBANDINGAN NILAI EVALUASI SETIAP ALGORITMA

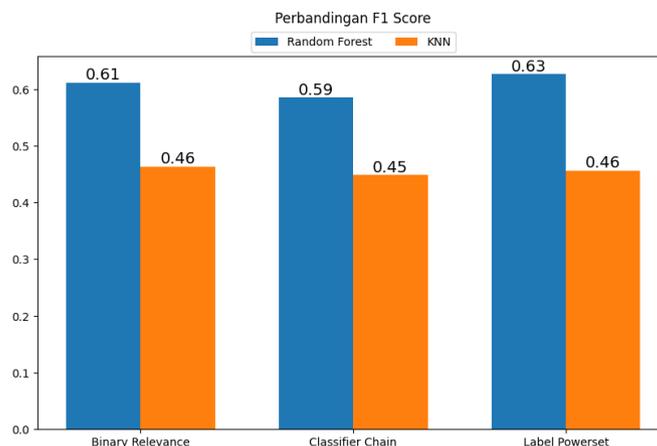
Algoritma	Metode	Akurasi	<i>F1-Score</i>	<i>Recall</i>	<i>Precision</i>
Random Forest	Binary Relevance	0.435897	0.592593	0.475248	0.786885*
	Classifier Chain	0.435897	0.541176	0.455446	0.666667
	Label Powerset	0.512821*	0.688172*	0.633663*	0.752941
K-Nearest Neighbors	Binary Relevance	0.230769	0.382166	0.29703	0.535714
	Classifier Chain	0.25641	0.355828	0.287129	0.467742
	Label Powerset	0.333333	0.439791	0.415842	0.466667

(*) Nilai tertinggi

Dari Tabel X diketahui algoritma *Random Forest* memiliki kinerja yang baik dibandingkan dengan algoritma K-NN pada semua metode *problem transformation* dari seluruh metrik evaluasi seperti akurasi dan *F1-Score*. *Random Forest* cenderung memiliki performa lebih baik karena mampu menangkap kompleksitas dan hubungan antar fitur dengan menggunakan sejumlah besar pohon keputusan yang bekerja secara ansambel. Algoritma *Random Forest* juga lebih baik dalam menangani variasi data dan menghindari *overfitting* yang sangat penting dalam masalah klasifikasi multi-label.

I. Evaluasi K-Fold Cross Validation

Hasil nilai evaluasi pada Gambar 5 merupakan nilai *F1-Score* dari *k-fold cross validation* menggunakan algoritma *Random Forest* dengan *fold* sebanyak 5 kali, dengan seluruh data yang digunakan (data *training*, dan *testing*) yang berjumlah 191 baris data.



Gambar 5. Perbandingan Nilai *F1-Score*

Gambar 5 menunjukkan nilai *F1-Score* terbaik secara keseluruhan yaitu pada algoritma *Random Forest*. Jumlah penggunaan topik soal yang tidak seimbang pada dataset ini merupakan salah satu faktor algoritma K-NN tidak dapat menghasilkan akurasi terbaiknya. Hal ini dikarenakan algoritma K-NN melakukan klasifikasi berdasarkan tetangga terdekat, sehingga rentan terhadap data yang *imbalance*. Sedangkan algoritma *Random Forest* memiliki keunggulan dengan dataset ini, dikarenakan *Random Forest* tahan terhadap data yang *imbalance*. Algoritma *Random Forest* mampu untuk membangun banyak pohon keputusan dari subset data yang berbeda, yang kemudian digabungkan untuk menghasilkan prediksi akhir yang lebih stabil dan akurat.

Pendekatan *problem transformation* terbaik ada pada algoritma *Random Forest* dengan metode *Label Powerset*, dengan hasil nilai *F1-Score* sebesar 63%. Metode ini 1% lebih baik dibandingkan metode *Binary Relevance* dan 4% lebih baik dari metode *Classifier Chain*. Pada algoritma K-NN dengan metode *Binary Relevance* dan *Label Powerset* menunjukkan hasil terbaik dengan *F1-Score* sebesar 46%, ketiga metode *problem transformation* ini menunjukkan nilai *F1-Score* yang seimbang dengan selisih hanya 1%. Perbedaan nilai *F1-Score* yang tidak terlalu jauh mendefinisikan bahwa setiap label pada dataset ini memiliki struktur yang dapat dipahami cukup baik oleh ketiga metode tersebut, meskipun *Label Powerset* menunjukkan sedikit keunggulan karena kemampuannya untuk menangkap kombinasi label yang kompleks secara lebih efektif.

Perbandingan hasil dari penelitian sebelumnya oleh Wiraguna [12] yang membandingkan algoritma *Random Forest* dengan metode *problem transformation* pada teks hadis, menemukan bahwa metode *Binary Relevance* lebih unggul. Serta penelitian oleh Fitriani & Yustanti [11] membandingkan algoritma K-NN dengan metode *problem transformation* pada teks pertanyaan, dan menemukan bahwa metode *Label Powerset* lebih unggul. Penelitian ini menunjukkan bahwa dalam perbandingan antara algoritma *Random Forest* dan K-NN dengan metode *problem transformation* pada teks soal, algoritma *Random Forest* dengan menggunakan *Label Powerset* menunjukkan kinerja yang lebih unggul.

IV. KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan mengenai pengklasifikasian soal multi-label berdasarkan kategori topik menggunakan algoritma *Random Forest* dan K-NN dengan berbagai metode *problem transformation*, dapat disimpulkan bahwa algoritma *Random Forest* menunjukkan kinerja yang lebih baik dari algoritma K-NN pada semua metode *problem transformation* dari seluruh metrik evaluasi seperti akurasi dan *F1-Score*. Algoritma *Random Forest* memiliki nilai *F1-Score* terbaik sebesar 69% dengan metode *Label Powerset*, sedangkan algoritma K-NN mencapai nilai *F1-Score* terbaik sebesar 44% dengan metode yang sama. Metode *problem transformation* terbaik pada algoritma *Random Forest* yaitu metode *Label Powerset*, yang memberikan hasil terbaik dengan *F1-Score* dari pengujian *K-Fold Cross Validation* sebesar 63%, menunjukkan kemampuan model untuk menangkap hubungan antar label yang kompleks. Untuk algoritma K-NN, metode *Binary Relevance* dan *Label Powerset* memberikan hasil terbaik pada pengujian *K-Fold Cross Validation* dengan *F1-Score* yang sama sebesar 46%, menunjukkan kemampuannya dalam memprediksi label tanpa hubungan pada label lain, serta dapat memprediksi dengan label yang disatukan menjadi sebuah kelompok. Metode *Classifier Chain* menunjukkan hasil yang kurang baik dikarenakan metode ini bergantung pada hasil prediksi sebelumnya. Jika hasil prediksi sebelumnya mengalami kekeliruan, maka akan berdampak buruk pada keseluruhan hasil prediksi selanjutnya, begitu juga sebaliknya.

DAFTAR PUSTAKA

- [1] R. H. Mardiyah, S. N. F. Aldriani, F. Chitta, dan M. R. Zulfikar, "Pentingnya keterampilan belajar di abad 21 sebagai tuntutan dalam pengembangan sumber daya manusia," *Lectura: Jurnal Pendidikan*, vol. 12, no. 1, hlm. 29–40, 2021.
- [2] H. Arfandy dan I. A. Musdar, "Rancang Bangun Sistem Cerdas Pemberian Nilai Otomatis Untuk Ujian Essai Menggunakan Algoritma Cosine Similarity," *Inspiration: Jurnal Teknologi Informasi dan Komunikasi*, vol. 10, no. 2, 2020, doi: 10.35585/inspir.v10i2.2580.
- [3] L. A. Muhaimin, O. N. Pratiwi, dan R. Y. Fa'rifah, "Klasifikasi Soal Berdasarkan Kategori Topik Menggunakan Metode Algoritma Naive Bayes dan Algoritma C4.5," *e-Prociding of Engineering*, vol. 10, no. 2, hlm. 1535–1541, 2023.
- [4] UNESCO, "Education: From COVID-19 school closures to recovery." [Daring]. Tersedia pada: <https://www.unesco.org/en/covid-19/education-response>
- [5] L. Roza, N. Aulia, dan Z. Zulherman*, "Analisa Pemanfaatan E-Learning Menggunakan Data Statistik Pengguna Aplikasi Startup Pendidikan Selama Wabah Pandemi Covid-19," *Jurnal Pendidikan Sains Indonesia*, vol. 9, no. 3, 2021, doi: 10.24815/jpsi.v9i3.19731.
- [6] S. Nuryatin, "Adaptasi metode pembelajaran melalui e-learning untuk menghadapi era new normal," 2020.
- [7] I. Maslov, S. Nikou, dan P. Hansen, "Exploring user experience of learning management system," *International Journal of Information and Learning Technology*, vol. 38, no. 4, 2021, doi: 10.1108/IJILT-03-2021-0046.
- [8] A. B. Sidiq dan D. Kurniadi, "Perancangan Sistem Informasi Ujian Online Berbasis Web pada SMK N 1 Solok," *Voteteknika (Vocational Teknik Elektronika dan Informatika)*, vol. 9, no. 2, hlm. 44–53, 2021.
- [9] Y. Pristyanto dan A. A. Zein, "Model Balanced Bagging Berbasis Decision Tree Pada Dataset Imbalanced Class," *Jurnal Sisfokom (Sistem Informasi dan Komputer)*, vol. 12, no. 1, 2023, doi: 10.32736/sisfokom.v12i1.1399.
- [10] G. Sailasya dan G. L. A. Kumari, "Analyzing the Performance of Stroke Prediction using ML Classification Algorithms," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 6, 2021, doi: 10.14569/IJACSA.2021.0120662.

- [11] E. E. Fitriani dan W. Yustanti, "Perbandingan Kinerja Metode Problem Transformation-KNN dan Algorithm Adaptation-KNN pada Klasifikasi Multi-Label Pertanyaan Kotakode," *Journal of Emerging Information Systems and Business Intelligence*, vol. 3, no. 3, 2022.
- [12] A. Wiraguna, S. Al Faraby, dan A. Adiwijaya, "Klasifikasi Topik Multi Label pada Hadis Bukhari dalam Terjemahan Bahasa Indonesia Menggunakan Random Forest," *eProceedings of Engineering*, vol. 6, no. 1, 2019.
- [13] O. Ramadhani, "KLASIFIKASI MULTI-LABEL BERITA ONLINE MENGGUNAKAN PROBLEM TRANSFORMATION DENGAN METODE K-NEAREST NEIGHBOR," Institut Teknologi Sepuluh Nopember, 2020.
- [14] U. Fayyad, G. Piatetsky-Shapiro, dan P. Smyth, "From data mining to knowledge discovery in databases," *AI Mag*, vol. 17, no. 3, 1996.
- [15] S. Alam, M. G. Resmi, dan N. Masripah, "Classification of Covid-19 vaccine data screening with Naive Bayes algorithm using Knowledge Discovery in database method," *Journal of Computer Networks, Architecture and High Performance Computing*, vol. 4, no. 2, 2022, doi: 10.47709/cnahpc.v4i2.1584.
- [16] C. Fan, M. Chen, X. Wang, J. Wang, dan B. Huang, "A review on data preprocessing techniques toward efficient and reliable knowledge discovery from building operational data," *Front Energy Res*, vol. 9, hlm. 652801, 2021.
- [17] I. Afdhal, R. Kurniawan, I. Iskandar, R. Salambue, E. Budianita, dan F. Syafria, "Penerapan Algoritma Random Forest Untuk Analisis Sentimen Komentar Di YouTube Tentang Islamofobia," *Jurnal Nasional Komputasi dan Teknologi Informasi*, vol. 5, no. 1, 2022.
- [18] M. A. Rosid, A. S. Fitriani, I. R. I. Astutik, N. I. Mulloh, dan H. A. Gozali, "Improving text preprocessing for student complaint document classification using sastrawi," dalam *IOP Conference Series: Materials Science and Engineering*, IOP Publishing, 2020, hlm. 12017.
- [19] R. Kosasih dan A. Alberto, "Sentiment analysis of game product on shopee using the TF-IDF method and naive bayes classifier," *ILKOM Jurnal Ilmiah*, vol. 13, no. 2, 2021, doi: 10.33096/ilkom.v13i2.721.101-109.
- [20] K. Shah, H. Patel, D. Sanghvi, dan M. Shah, "A Comparative Analysis of Logistic Regression, Random Forest and KNN Models for the Text Classification," *Augmented Human Research*, vol. 5, no. 1, 2020, doi: 10.1007/s41133-020-00032-0.
- [21] S. Khairunnisa, A. Adiwijaya, dan S. Al Faraby, "Pengaruh Text Preprocessing terhadap Analisis Sentimen Komentar Masyarakat pada Media Sosial Twitter (Studi Kasus Pandemi COVID-19)," *Jurnal Media Informatika Budidarma*, vol. 5, no. 2, hlm. 406-414, 2021.
- [22] S. Kumar, N. Kumar, A. Dev, dan S. Naorem, "Movie genre classification using binary relevance, label powerset, and machine learning classifiers," *Multimed Tools Appl*, vol. 82, no. 1, hlm. 945-968, 2023, doi: 10.1007/s11042-022-13211-5.
- [23] T. Al Kautsar, "Klasifikasi Multi Label Pada Jawaban Esai Menggunakan Algoritma Multi Label Knearest Neighbor (Mlkn)." Fakultas Sains dan Teknologi UIN Syarif Hidayatullah Jakarta, Jakarta, 2023.
- [24] M. Y. Khan, A. Qayoom, M. S. Nizami, M. S. Siddiqui, S. Wasi, dan S. M. K.-R. Raazi, "Automated prediction of Good Dictionary EXamples (GDEX): a comprehensive experiment with distant supervision, machine learning, and word embedding-based deep learning techniques," *Complexity*, vol. 2021, hlm. 1-18, 2021.
- [25] B. Purnama, *Pengantar Machine Learning: Konsep dan Praktikum Dengan Contoh Latihan Berbasis R dan Python*, 1 ed. Bandung: Informatika Bandung, 2019.
- [26] L. Breiman, "Random Forests," *Mach Learn*, vol. 45, no. 1, hlm. 5-32, 2001, doi: 10.1023/A:1010933404324.
- [27] T. Z. Jasman, E. Hasmin, C. Susanto, dan W. Musu, "Perbandingan Logistic Regression, Random Forest, dan Perceptron pada Klasifikasi Pasien Gagal Jantung," *CSRID Journal*, vol. 14, no. 3, 2022.
- [28] A. Yudhana, S. Sunardi, dan A. J. S. Hartanta, "ALGORITMA K-NN DENGAN EUCLIDEAN DISTANCE UNTUK PREDIKSI HASIL PENGGERGAJIAN KAYU SENGON," *Transmisi*, vol. 22, no. 4, 2020, doi: 10.14710/transmisi.22.4.123-129.
- [29] D. Novianto dan T. Sugihartono, "Sistem Deteksi Kualitas Buah Jambu Air Berdasarkan Warna Kulit Menggunakan Algoritma Principal Component Analysis (Pca) dan K-Nearest Neighbor (K-NN)," *Jurnal Ilmiah Informatika Global*, vol. 11, no. 2, 2020.
- [30] I. Setiawati, "Komparasi Algoritma Machine Learning dari Dataset Prediksi Analisis Butir Soal Harian Siswa," *Jurnal Cakrawala Informasi*, vol. 3, no. 1, 2023, doi: 10.54066/jci.v3i1.235.
- [31] I. Riadi, A. Fadlil, dan P. Annisa, "Identifikasi Tulisan Tangan Huruf Katakana Jepang Dengan Metode Euclidean," *J-SAKTI (Jurnal Sains Komputer dan Informatika)*, vol. 4, no. 1, 2020, doi: 10.30645/j-sakti.v4i1.184.
- [32] O.- Pahlevi, A.- Amrin, dan Y.- Handrianto, "Implementasi Algoritma Klasifikasi Random Forest Untuk Penilaian Kelayakan Kredit," *Jurnal Infotech*, vol. 5, no. 1, 2023, doi: 10.31294/infotech.v5i1.15829.
- [33] J. Xu, Y. Zhang, dan D. Miao, "Three-way confusion matrix for classification: A measure driven view," *InfSci (N Y)*, vol. 507, 2020, doi: 10.1016/j.ins.2019.06.064.
- [34] D. Normawati dan S. A. Prayogi, "Implementasi Naïve Bayes Classifier Dan Confusion Matrix Pada Analisis Sentimen Berbasis Teks Pada Twitter," *Jurnal Sains Komputer & Informatika (J-SAKTI)*, vol. 5, no. 2, 2021.
- [35] J. Salminen, V. Yoganathan, J. Corporan, B. J. Jansen, dan S.-G. Jung, "Machine learning approach to auto-tagging online content for content marketing efficiency: A comparative analysis between methods and content type," *J Bus Res*, vol. 101, hlm. 203-217, 2019, doi: <https://doi.org/10.1016/j.jbusres.2019.04.018>.
- [36] D. Polessa Paula dkk., "Comparing machine learning algorithms for multimorbidity prediction: An example from the Elsa-Brasil study," *PLoS One*, vol. 17, no. 10, 2022, doi: 10.1371/journal.pone.0275619.
- [37] D. A. Nasution, H. H. Khotimah, dan N. Chamidah, "Perbandingan Normalisasi Data untuk Klasifikasi Wine Menggunakan Algoritma K-NN," *Computer Engineering, Science and System Journal*, vol. 4, no. 1, 2019, doi: 10.24114/cess.v4i1.11458.
- [38] A. Hanafi, A. Adiwijaya, dan W. Astuti, "Klasifikasi Multi Label pada Hadis Bukhari Terjemahan Bahasa Indonesia Menggunakan Mutual Information dan k-Nearest Neighbor," *Jurnal Sisfokom (Sistem Informasi dan Komputer)*, vol. 9, no. 3, hlm. 357-364, Sep 2020, doi: 10.32736/sisfokom.v9i3.980.