

Klasifikasi Multi-Label Hadits Shahih Muslim Menggunakan Metode Support Vector Machine (SVM)

Dinda Lutfiah, Nazruddin Safaat Harahap*, Febi Yanto, Eka Pandu Cynthia

Fakultas Sains dan Teknologi, Teknik Informatika, Universitas Islam Negeri Sultan Syarif Kasim Riau, Pekanbaru, Indonesia

Email: ¹12250125077@students.uin-suska.ac.id, ^{2,*}nazruddin.safaat@uin-suska.ac.id, ³febiyanto@uin-suska.ac.id,

⁴eka.pandu.cynthia@uin-suska.ac.id

Email Penulis Korespondensi: nazruddin.safaat@uin-suska.ac.id

Abstrak—Dalam kajian Islam, hadits merupakan salah satu bentuk teks yang digunakan sebagai rujukan dalam memahami ajaran dan praktik keagamaan. Permasalahan penelitian ini adalah kompleksitas klasifikasi hadits karena satu teks dapat memuat beberapa kategori makna sekaligus, seperti anjuran, larangan, dan informasi, sehingga pendekatan satu label belum cukup merepresentasikan isi hadits secara menyeluruh. Penelitian ini bertujuan menerapkan Support Vector Machine (SVM) untuk klasifikasi multi-label Hadits Shahih Muslim terjemahan Bahasa Indonesia. Dataset terdiri dari 5.362 teks matan hadits, yaitu 4.557 data latih dan 805 data uji. Tahapan penelitian meliputi preprocessing teks, ekstraksi fitur TF-IDF, klasifikasi One-Vs-Rest, dan evaluasi menggunakan precision, recall, F1-score, hamming loss, serta subset accuracy. Baseline SVM memperoleh micro precision 0,8990, micro recall 0,8080, micro F1-score 0,8511, macro precision 0,6404, macro recall 0,4446, macro F1-score 0,4798, hamming loss 0,1226, dan subset accuracy 0,6696. Setelah hyperparameter tuning menggunakan GridSearchCV, model memperoleh micro precision 0,7919, micro recall 0,8395, micro F1-score 0,8150, macro precision 0,5482, macro recall 0,6341, macro F1-score 0,5809, hamming loss 0,1652, dan subset accuracy 0,5764. Hasil menunjukkan bahwa tuning meningkatkan macro recall dan macro F1-score, tetapi menurunkan micro F1-score, hamming loss, dan subset accuracy. Kontribusi utama terletak pada evaluasi performa terhadap ketidakseimbangan label menggunakan metrik macro dan micro serta analisis fitur dominan pada setiap kategori makna hadits. Dengan demikian, SVM dapat digunakan untuk klasifikasi multi-label Hadits Shahih Muslim, meskipun ketidakseimbangan label masih memengaruhi performa model.

Kata Kunci: Klasifikasi; Hadits; *Multi-Label*; *Machine Learning*; *Support Vector Machine*

Abstract—In Islamic studies, hadith is a form of text used as a reference in understanding religious teachings and practices. The problem of this research is the complexity of hadith classification because a single text can contain several categories of meaning at once, such as recommendations, prohibitions, and information, so that a single-label approach is not sufficient to represent the contents of the hadith as a whole. This study aims to apply Support Vector Machine (SVM) for multi-label classification of the Indonesian translation of Sahih Muslim Hadith. The dataset consists of 5,362 hadith texts, namely 4,557 training data and 805 test data. The research stages include text preprocessing, TF-IDF feature extraction, One-Vs-Rest classification, and evaluation using precision, recall, F1-score, hamming loss, and subset accuracy. The baseline SVM obtained micro precision of 0.8990, micro recall of 0.8080, micro F1-score of 0.8511, macro precision of 0.6404, macro recall of 0.4446, macro F1-score of 0.4798, hamming loss of 0.1226, and subset accuracy of 0.6696. After hyperparameter tuning using GridSearchCV, the model obtained micro precision of 0.7919, micro recall of 0.8395, micro F1-score of 0.8150, macro precision of 0.5482, macro recall of 0.6341, macro F1-score of 0.5809, hamming loss of 0.1652, and subset accuracy of 0.5764. The results showed that tuning increased macro recall and macro F1-score, but decreased micro F1-score, hamming loss, and subset accuracy. The main contribution lies in the performance evaluation of label imbalance using macro and micro metrics as well as the analysis of dominant features in each category of hadith meaning. Thus, SVM can be used for multi-label classification of Sahih Muslim Hadith, although label imbalance still affects the model performance.

Keywords: Classification; Hadith; *Multi-Label*; *Machine Learning*; *Support Vector Machine*

1. PENDAHULUAN

Dalam ajaran Islam, hadits menjadi salah satu sumber rujukan yang digunakan untuk memperjelas kandungan Al-Qur'an, khususnya pada aturan maupun praktik yang memerlukan penjelasan lebih rinci. Secara terminologis, hadits mencakup segala sesuatu yang disandarkan kepada Nabi Muhammad SAW, baik berupa perkataan, perbuatan, maupun ketetapan (*taqrir*) beliau. Sebagian isi Al-Qur'an memerlukan penjabaran tambahan sehingga hadits berperan membantu memberikan pemahaman yang lebih jelas terhadap maksud suatu ayat. Oleh karena itu, hadits tidak hanya berfungsi sebagai pelengkap, tetapi juga sebagai pedoman dalam penetapan hukum serta dalam membimbing praktik keagamaan umat Islam secara komprehensif [1],[2].

Salah satu kitab yang memiliki kedudukan penting adalah Hadits Shahih Muslim, yang disusun melalui proses pengumpulan dan penyaringan hadits secara sistematis oleh Imam Muslim. Dalam penyusunannya, beliau menyeleksi sanad dan matan dengan cukup ketat sehingga hadits-hadits yang dimuat memiliki tingkat keotentikan yang tinggi. Oleh karena itu, Hadits Shahih Muslim sering dijadikan sebagai salah satu rujukan dalam kajian hadits maupun penelitian keislaman. Seiring dengan perkembangan teknologi, teks dalam kitab ini juga mulai dimanfaatkan sebagai sumber data dalam penelitian berbasis pengolahan teks dan *Natural Language Processing* (NLP), khususnya dalam analisis dan klasifikasi hadits [3],[4].

Pada praktiknya, satu teks hadits tidak selalu hanya memuat satu pesan tertentu, tetapi dapat mengandung beberapa makna sekaligus dalam satu isi pembahasan, seperti anjuran, larangan, dan informasi dalam satu teks. Hal ini menunjukkan bahwa pemaknaan terhadap hadits tidak selalu sederhana dan sering kali bergantung pada konteks yang melatarbelakanginya. Kondisi tersebut membuat pendekatan satu label menjadi kurang memadai karena belum dapat



menggambarkan seluruh konteks yang muncul pada sebuah hadits yang terkandung dalam hadits. Karena itu, penelitian ini menggunakan pendekatan *multi-label* agar satu data hadits dapat masuk ke beberapa kategori sekaligus.

Perkembangan teknologi dalam bidang NLP memungkinkan teks hadits dianalisis secara otomatis melalui beberapa tahapan, seperti *preprocessing*, representasi teks, dan klasifikasi. Proses *preprocessing* dilakukan untuk membersihkan dan menyiapkan teks agar dapat diolah dengan lebih baik oleh model *machine learning*. Selanjutnya, teks diubah ke dalam bentuk numerik menggunakan metode *Term Frequency-Inverse Document Frequency* (TF-IDF) agar dapat diproses oleh algoritma klasifikasi. Pemilihan metode representasi teks yang tepat menjadi salah satu faktor yang berpengaruh terhadap hasil klasifikasi yang diperoleh.

Support Vector Machine (SVM) merupakan salah satu metode *machine learning* yang cukup banyak digunakan dalam berbagai tugas klasifikasi karena kemampuannya dalam menangani data berdimensi tinggi. Metode ini bekerja dengan cara mencari *hyperplane* terbaik yang dapat memisahkan data ke dalam kelas yang berbeda dengan margin maksimum. Selain itu, SVM juga dikenal memiliki kemampuan generalisasi yang baik sehingga hasil klasifikasinya cenderung tetap stabil ketika diterapkan pada data baru. Berdasarkan karakteristik tersebut, SVM dinilai sesuai untuk digunakan dalam klasifikasi teks hadits yang memiliki kompleksitas makna [5].

Beberapa penelitian sebelumnya menunjukkan bahwa pendekatan klasifikasi *multi-label* telah digunakan dalam pengolahan teks hadits dengan berbagai metode *machine learning*. Pendekatan ini dipilih karena satu hadits dapat memiliki lebih dari satu kategori, sehingga tidak cukup jika hanya menggunakan satu label. Selain itu, metode SVM juga menunjukkan hasil yang cukup baik dalam klasifikasi *multi-label* pada teks hadits. Untuk menilai kinerja model, klasifikasi *multi-label* umumnya menggunakan metrik seperti *precision*, *recall*, dan *F1-score* agar hasil evaluasi dapat menggambarkan performa model secara lebih menyeluruh [6].

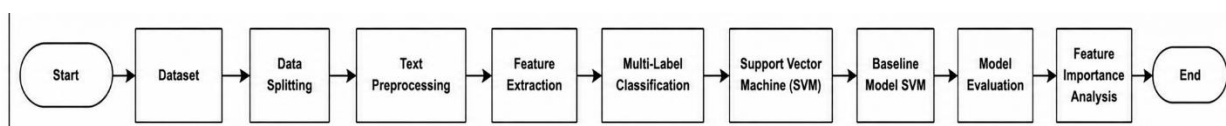
Meskipun berbagai penelitian sebelumnya telah membahas klasifikasi *multi-label* pada teks hadits menggunakan berbagai metode *machine learning* seperti *k-Nearest Neighbor*, *CART*, *Bagging*, *SVM*, dan *LSTM*, sebagian besar penelitian tersebut masih berfokus pada peningkatan performa model secara umum tanpa memberikan analisis yang mendalam terhadap distribusi label dan dampaknya pada ketidakseimbangan data. Selain itu, penelitian terdahulu cenderung menggunakan dataset hadits dengan cakupan yang lebih umum, seperti *Hadits Bukhari* atau dataset yang tidak difokuskan pada struktur makna tertentu. Penelitian yang secara khusus mengkaji *Hadits Shahih Muslim* dengan pendekatan *multi-label* masih terbatas, terutama pada pengelompokan makna seperti anjuran, larangan, dan informasi secara bersamaan. Selain itu, masih sedikit penelitian yang mengevaluasi kinerja *Support Vector Machine* (SVM) pada kondisi ketidakseimbangan label serta analisis performa pada setiap kategori secara terpisah.

Penelitian ini memberikan kontribusi dalam bidang klasifikasi teks keagamaan berbasis NLP melalui penerapan SVM pada klasifikasi *multi-label* *Hadits Shahih Muslim*. Kontribusi utama penelitian ini terletak pada analisis performa model dalam kondisi ketidakseimbangan label menggunakan evaluasi *macro* dan *micro* secara bersamaan, serta identifikasi fitur dominan yang membantu menjelaskan karakteristik kata pada setiap kategori makna hadits.

Berdasarkan permasalahan tersebut, penelitian ini bertujuan untuk menerapkan metode SVM dalam melakukan klasifikasi *multi-label* pada teks *Hadits Shahih Muslim* ke dalam kategori anjuran, larangan, dan informasi. Penelitian ini dilakukan melalui tahapan pengolahan teks meliputi *preprocessing*, representasi fitur menggunakan TF-IDF, serta penerapan pendekatan *One-Vs-Rest* untuk klasifikasi *multi-label*. Selanjutnya, performa model dievaluasi menggunakan *precision*, *recall*, *F1-score*, *hamming loss*, dan subset *accuracy* untuk mengukur kemampuan model dalam mengklasifikasikan setiap kategori makna hadits. Hasil penelitian ini diharapkan dapat mendukung pengembangan pemanfaatan *machine learning* dalam klasifikasi teks keagamaan secara otomatis dan lebih terstruktur, serta memberikan gambaran mengenai kemampuan SVM dalam menangani data *multi-label* pada teks hadits.

2. METODOLOGI PENELITIAN

Proses penelitian dalam studi ini digambarkan melalui *flowchart* pada Gambar 1.



Gambar 1. Alur Metode Penelitian

Gambar 1 menunjukkan tahapan penelitian yang dilakukan, mulai dari pengumpulan data, pelabelan data, pembagian data, *preprocessing* teks, ekstraksi fitur menggunakan *Term Frequency-Inverse Document Frequency* (TF-IDF), pembentukan baseline model *Support Vector Machine* (SVM), *hyperparameter tuning* menggunakan *GridSearchCV*, klasifikasi *multi-label* dengan pendekatan *One-Vs-Rest*, evaluasi model, hingga analisis fitur yang berpengaruh. Alur tersebut menggambarkan penerapan metode SVM dalam klasifikasi *multi-label* *Hadits Shahih Muslim* secara sistematis.

2.1 Dataset

Dataset dalam penelitian ini berupa teks matan Hadits Shahih Muslim terjemahan Bahasa Indonesia. Pada tahap ini, data dikumpulkan dari sumber digital, kemudian dilakukan pemeriksaan terhadap kesesuaian nomor hadits dan isi matan agar data yang digunakan sesuai dengan kebutuhan penelitian. Data yang telah dikumpulkan selanjutnya diberi label berdasarkan tiga kategori utama, yaitu anjuran, larangan, dan informasi. Pelabelan dilakukan dengan pendekatan *multi-label* karena satu teks hadits dapat memuat lebih dari satu kategori makna dalam satu pembahasan. Setiap label direpresentasikan dalam bentuk biner, yaitu nilai 1 menunjukkan bahwa suatu hadits termasuk ke dalam kategori tertentu, sedangkan nilai 0 menunjukkan bahwa hadits tidak termasuk ke dalam kategori tersebut.

2.2 Data Splitting

Data splitting dilakukan untuk membagi *dataset* menjadi data latih dan data uji. Data latih digunakan untuk membangun model klasifikasi, sedangkan data uji digunakan untuk mengukur kemampuan model dalam mengklasifikasikan data yang belum pernah diproses pada tahap pelatihan. Pada penelitian ini, data dibagi menggunakan perbandingan 85:15. Pembagian data tersebut digunakan agar model dapat mempelajari pola dari data latih dan diuji performanya pada data uji secara terpisah [7],[8].

2.2.1 Text Preprocessing

Sebelum dilakukan proses klasifikasi, data teks terlebih dahulu melalui tahap *preprocessing*. Tahap ini bertujuan untuk membersihkan teks, menyeragamkan bentuk penulisan, serta menghilangkan unsur yang kurang relevan sehingga data lebih siap digunakan dalam proses ekstraksi fitur dan klasifikasi [9],[10]. Tahapan *preprocessing* yang digunakan dalam penelitian ini meliputi *case folding*, *cleaning*, *tokenization*, dan *stopword removal*. *Case folding* dilakukan dengan mengubah seluruh huruf menjadi huruf kecil agar penulisan kata menjadi seragam. *Cleaning* dilakukan untuk menghapus tanda baca, angka, dan karakter khusus yang tidak diperlukan. *Tokenization* digunakan untuk memisahkan kalimat menjadi satuan kata. Selanjutnya, *stopword removal* dilakukan untuk menghapus kata-kata umum yang dianggap kurang berpengaruh terhadap proses klasifikasi [11],[12].

Setelah itu dilakukan *tokenization* untuk memisahkan kalimat menjadi beberapa kata. Tahap berikutnya adalah *stopword removal*, yaitu menghapus kata-kata umum yang dianggap kurang berpengaruh terhadap proses klasifikasi. Pada penelitian ini, *stemming* tidak digunakan agar bentuk kata asli seperti melarang, memerintahkan, pakailah, dan jangan tetap dipertahankan [13]. Hal ini dilakukan agar fitur yang muncul dari hasil pembobotan lebih mudah dijelaskan secara makna [14].

2.3 Feature Extraction

Feature extraction dilakukan untuk mengubah data teks hasil *preprocessing* menjadi bentuk numerik agar dapat diproses oleh algoritma klasifikasi. Pada penelitian ini, metode yang digunakan adalah *Term Frequency-Inverse Document Frequency* (TF-IDF) [15]. TF-IDF merupakan metode pembobotan kata yang memberikan nilai pada suatu kata berdasarkan frekuensi kemunculan kata dalam dokumen dan tingkat keunikan kata tersebut dalam keseluruhan data.

Pada penelitian ini, TF-IDF digunakan pada proses ekstraksi fitur sebelum dilakukan klasifikasi menggunakan metode *Support Vector Machine* (SVM). Hasil *preprocessing* diubah menjadi representasi vektor menggunakan TF-IDF dengan parameter *max_features=5000*, *ngram_range=(1,2)*, *min_df=1*, *max_df=0.4*, *sublinear_tf=True*, dan *smooth_idf=True*. Perhitungan TF-IDF dapat dituliskan sebagai berikut:

$$TF - IDF(t, d) = TF(t, d) \times IDF(t) \quad (1)$$

TF-IDF terdiri dari dua komponen utama, yaitu *Term Frequency* (TF) dan *Inverse Document Frequency* (IDF). $TF(t, d)$ menunjukkan jumlah kemunculan kata t pada dokumen d , sedangkan $IDF(t)$ merupakan nilai yang digunakan untuk mengukur tingkat kepentingan suatu kata dalam keseluruhan dokumen. Kedua komponen ini digunakan untuk memberikan bobot pada kata berdasarkan frekuensi kemunculan dan tingkat keunikannya dalam dataset.

2.4 Multi-Label Classification

Penelitian ini menggunakan pendekatan klasifikasi *multi-label* karena satu teks hadits dapat memiliki lebih dari satu kategori dalam satu data, seperti anjuran, larangan, dan informasi. Pendekatan ini berbeda dengan klasifikasi *single-label* karena satu data tidak hanya diklasifikasikan ke dalam satu kelas, tetapi dapat masuk ke beberapa label sekaligus sesuai dengan kandungan maknanya.

Proses klasifikasi *multi-label* pada penelitian ini menggunakan pendekatan *One-Vs-Rest* (OvR). Metode OvR bekerja dengan cara membangun model klasifikasi secara terpisah untuk setiap label. Dengan pendekatan ini, setiap label dipelajari secara independen sehingga satu teks hadits tetap dapat memperoleh lebih dari satu label apabila pola teksnya memenuhi beberapa kategori secara bersamaan [16],[17].

2.5 Support Vector Machine (SVM)



Support Vector Machine (SVM) merupakan salah satu algoritma *machine learning* yang banyak digunakan dalam tugas klasifikasi teks karena mampu bekerja dengan baik pada data berdimensi tinggi. Data teks yang telah direpresentasikan menggunakan TF-IDF umumnya menghasilkan jumlah fitur yang besar, sehingga diperlukan metode klasifikasi yang stabil dalam memisahkan pola antar kategori [18],[19].

SVM bekerja dengan mencari *hyperplane* terbaik yang dapat memisahkan data ke dalam kelas yang berbeda dengan margin maksimum. Data yang berada paling dekat dengan *hyperplane* disebut sebagai *support vector* dan digunakan dalam pembentukan batas pemisah antar kelas. Secara umum, *hyperplane* pada metode SVM dapat dituliskan sebagai berikut:

$$w \cdot x + b = 0 \quad (2)$$

Pada metode SVM, vektor bobot (w) merupakan parameter yang menentukan arah dan posisi *hyperplane* dalam memisahkan data antar kelas, sedangkan data masukan (x) merepresentasikan fitur teks yang telah diubah ke dalam bentuk numerik menggunakan TF-IDF. Nilai bias (b) digunakan untuk menggeser posisi *hyperplane* agar diperoleh pemisahan kelas yang optimal. Ketiga komponen tersebut bekerja bersama dalam membentuk fungsi keputusan pada proses klasifikasi.

Pada penelitian ini, SVM diterapkan untuk melakukan klasifikasi *multi-label* terhadap teks Hadits Shahih Muslim yang telah direpresentasikan menggunakan TF-IDF. Model SVM digunakan dengan pendekatan *One-Vs-Rest* agar setiap label dapat diklasifikasikan secara terpisah.

SVM dipilih karena memiliki performa yang cukup baik dalam klasifikasi teks serta mampu menghasilkan model yang stabil pada data hasil representasi numerik.

2.6 Baseline Model SVM

Baseline model SVM digunakan sebagai pengujian awal untuk mengetahui kemampuan dasar model dalam melakukan klasifikasi *multi-label*. Pada tahap ini, model dijalankan menggunakan parameter awal untuk memperoleh hasil evaluasi awal. Hasil *baseline* digunakan sebagai pembandingan terhadap model yang telah melalui proses optimasi. Penggunaan SVM sebagai model awal masih relevan dalam klasifikasi teks karena metode ini mampu bekerja dengan baik pada data berdimensi tinggi dan banyak digunakan dalam penelitian klasifikasi berbasis.

Setelah *baseline model* diperoleh, penelitian ini melakukan *hyperparameter tuning* menggunakan *GridSearchCV*. Proses ini dilakukan untuk mencari kombinasi parameter terbaik yang dapat meningkatkan performa model. Pemilihan parameter dalam SVM menjadi bagian penting karena nilai parameter yang digunakan dapat memengaruhi kemampuan model dalam membentuk batas pemisah dan mengenali pola data.

Selain itu, optimasi *hyperparameter* juga menjadi bagian penting dalam proyek *machine learning* karena dapat digunakan untuk memperoleh konfigurasi model yang lebih sesuai terhadap data yang digunakan. Parameter yang diuji dalam proses *tuning* meliputi parameter pada TF-IDF dan SVM, seperti jumlah fitur maksimum, *n-gram*, nilai C , dan *class weight*. Dengan demikian, performa model sebelum dan sesudah *tuning* dapat dibandingkan secara lebih jelas.

2.7 Model Evaluation

Model evaluation dilakukan untuk mengetahui kemampuan SVM dalam melakukan klasifikasi *multi-label* pada teks Hadits Shahih Muslim. Evaluasi dilakukan menggunakan data uji yang tidak digunakan dalam proses pelatihan model. Metrik evaluasi yang digunakan dalam penelitian ini meliputi *precision*, *recall*, *F1-score*, *hamming loss*, dan *subset accuracy*. *Precision* digunakan untuk melihat tingkat ketepatan model dalam memprediksi suatu label, sedangkan *recall* digunakan untuk melihat kemampuan model dalam menemukan seluruh data yang termasuk ke dalam label tertentu. *F1-score* digunakan untuk menyeimbangkan nilai *precision* dan *recall*. *Hamming loss* digunakan untuk mengukur tingkat kesalahan prediksi label pada klasifikasi *multi-label*, sedangkan *subset accuracy* digunakan untuk melihat kesesuaian seluruh kombinasi label antara hasil prediksi dan label sebenarnya.

Penelitian ini juga menggunakan *micro average* dan *macro average*. *Micro average* digunakan untuk melihat performa model secara keseluruhan berdasarkan seluruh prediksi label, sedangkan *macro average* digunakan untuk melihat rata-rata performa pada setiap label tanpa mempertimbangkan jumlah data pada masing-masing label.

2.8 Feature Importance Analysis

Feature importance analysis dilakukan untuk mengetahui kata atau fitur yang berpengaruh terhadap hasil klasifikasi pada setiap kategori. Fitur yang dianalisis berasal dari hasil pembobotan TF-IDF dan model SVM. Analisis ini membantu menjelaskan kata-kata yang berkontribusi dalam mengenali label anjuran, larangan, dan informasi.

Dengan adanya *feature importance analysis*, hasil penelitian tidak hanya menampilkan nilai evaluasi model, tetapi juga memberikan gambaran mengenai fitur teks yang memengaruhi proses klasifikasi. Analisis ini digunakan untuk memperjelas bagaimana model mengenali pola kata pada masing-masing kategori.

3. HASIL DAN PEMBAHASAN

3.1 Deskripsi Dataset Penelitian

Dataset yang digunakan dalam penelitian ini berupa kumpulan teks matan Hadits Shahih Muslim dalam terjemahan Bahasa



Indonesia. Data diperoleh dari sumber digital, kemudian diperiksa kembali berdasarkan nomor hadits dan kesesuaian isi matan agar data yang digunakan tetap relevan dengan kebutuhan penelitian. Pemeriksaan data dilakukan untuk memastikan bahwa teks yang digunakan benar-benar berupa matan hadits dan tidak mengandung data yang tidak sesuai dengan kategori penelitian.

Jumlah data yang digunakan dalam penelitian ini sebanyak 5.362 teks matan hadits. Setiap teks hadits dikategorikan ke dalam tiga label utama, yaitu anjuran, larangan, dan informasi. Pemilihan tiga label tersebut didasarkan pada karakteristik isi hadits yang dapat memuat lebih dari satu makna dalam satu teks. Satu hadits dapat berisi anjuran untuk melakukan suatu perbuatan, larangan terhadap suatu tindakan, serta informasi atau penjelasan dalam satu pembahasan. Oleh karena itu, pendekatan klasifikasi yang digunakan dalam penelitian ini adalah klasifikasi *multi-label*.

Pada klasifikasi *multi-label*, satu data dapat memiliki lebih dari satu label secara bersamaan. Hal ini berbeda dengan klasifikasi *single-label* yang hanya menempatkan satu data ke dalam satu kelas. Dalam konteks penelitian ini, pendekatan *multi-label* lebih sesuai karena kandungan teks hadits tidak selalu dapat direpresentasikan dengan satu kategori saja. Setiap label kemudian direpresentasikan dalam bentuk biner, yaitu nilai 1 menunjukkan bahwa suatu hadits termasuk ke dalam kategori tertentu, sedangkan nilai 0 menunjukkan bahwa hadits tidak termasuk dalam kategori tersebut. Contoh representasi *dataset* ditunjukkan pada Tabel 1.

Tabel 1. Representasi Dataset

No. Hadits	Hadits	Kelas		
		Anjuran	Larangan	Informasi
436	Rasulullah shallallahu ‘alaihi wasallam mencuci bekas mani pada pakaian sebelum digunakan kembali untuk melaksanakan shalat serta menjelaskan pentingnya menjaga kebersihan pakaian.	1	0	1
1967	Rasulullah shallallahu ‘alaihi wasallam menganjurkan puasa Dawud sebagai puasa yang paling dicintai Allah dan melarang ibadah secara berlebihan hingga menyebabkan tubuh menjadi lemah.	1	1	1
2490	Rasulullah shallallahu ‘alaihi wasallam melarang hidup membujang secara terus-menerus serta menjelaskan pentingnya menjaga kehidupan rumah tangga dalam Islam.	0	1	1

Berdasarkan Tabel 1, hadits nomor 436 memiliki label anjuran dan informasi, sedangkan hadits nomor 1967 memiliki tiga label sekaligus, yaitu anjuran, larangan, dan informasi. Hal ini menunjukkan bahwa satu teks hadits dapat mengandung beberapa kategori makna secara bersamaan. Dengan demikian, penggunaan pendekatan *multi-label* dapat merepresentasikan kandungan teks hadits secara lebih tepat dibandingkan pendekatan satu label.

3.2 Hasil Text Preprocessing

Sebelum data diproses oleh model klasifikasi, teks hadits terlebih dahulu melalui tahap *text preprocessing*. Tahap ini dilakukan untuk membersihkan data, menyeragamkan bentuk penulisan, dan menghilangkan unsur yang kurang relevan terhadap proses klasifikasi. Data teks mentah umumnya masih mengandung huruf kapital, tanda baca, angka, karakter khusus, serta kata umum yang tidak terlalu berpengaruh terhadap pembentukan pola klasifikasi. Jika data langsung digunakan tanpa tahap *preprocessing*, model dapat menghasilkan banyak fitur yang kurang penting sehingga proses klasifikasi menjadi kurang optimal.

Tahapan *preprocessing* yang digunakan dalam penelitian ini meliputi *case folding*, *cleaning*, *tokenization*, dan *stopword removal*. Tahap *case folding* dilakukan dengan mengubah seluruh huruf menjadi huruf kecil. Tahapan ini diperlukan agar kata yang sama tidak dianggap berbeda hanya karena perbedaan bentuk huruf kapital. Selanjutnya, tahap *cleaning* dilakukan untuk menghapus tanda baca, angka, dan karakter khusus. Tahap *tokenization* digunakan untuk memecah kalimat menjadi satuan kata. Setelah itu, tahap *stopword removal* dilakukan untuk menghapus kata-kata umum yang dianggap kurang berpengaruh terhadap proses klasifikasi.

Pada penelitian ini, tahap *stemming* tidak digunakan. Keputusan ini dilakukan agar bentuk kata asli seperti melarang, memerintahkan, pakailah, hendaklah, dan jangan tetap dipertahankan. Kata-kata tersebut memiliki makna penting dalam membedakan kategori anjuran dan larangan. Jika dilakukan *stemming*, beberapa bentuk kata dapat berubah dan berpotensi mengurangi kejelasan makna fitur yang dihasilkan. Contoh hasil tahapan *text preprocessing* ditunjukkan pada Tabel 2.

Tabel 2. Tahapan Text Preprocessing

Tahapan	Sebelum	Sesudah
<i>Case Folding</i> (Perubahan huruf menjadi huruf kecil)	Telah menceritakan kepada kami [Abu Bakr bin Abi Syaibah] bahwa Rasulullah shallallahu ‘alaihi wasallam memerintahkan	telah menceritakan kepada kami [abu bakr bin abi syaibah] bahwa rasulullah shallallahu ‘alaihi wasallam memerintahkan menjaga

Tahapan	Sebelum	Sesudah
	menjaga shalat tepat waktu dan melarang meninggalkan shalat.	shalat tepat waktu dan melarang meninggalkan shalat
<i>Cleaning</i> (Pembersihan tanda baca dan karakter khusus)	telah menceritakan kepada kami [abu bakr bin abi syaibah] bahwa rasulullah shallallahu ‘alaihi wasallam memerintahkan menjaga shalat tepat waktu dan melarang meninggalkan shalat.	telah menceritakan kepada kami abu bakr bin abi syaibah bahwa rasulullah shallallahu alaihi wasallam memerintahkan menjaga shalat tepat waktu dan melarang meninggalkan shalat
<i>Tokenization</i> (Pemisahan kalimat menjadi kata)	telah menceritakan kepada kami abu bakr bin abi syaibah bahwa rasulullah shallallahu alaihi wasallam memerintahkan menjaga shalat tepat waktu dan melarang meninggalkan shalat	['telah', 'menceritakan', 'kepada', 'kami', 'abu', 'bakr', 'bin', 'abi', 'syaibah', 'bahwa', 'rasulullah', 'shallallahu', 'alaihi', 'wasallam', 'memerintahkan', 'menjaga', 'shalat', 'tepat', 'waktu', 'dan', 'melarang', 'meninggalkan', 'shalat']
<i>Stopword Removal</i> (Penghapusan kata umum)	['telah', 'menceritakan', 'kepada', 'kami', 'abu', 'bakr', 'bin', 'abi', 'syaibah', 'bahwa', 'rasulullah', 'shallallahu', 'alaihi', 'wasallam', 'memerintahkan', 'menjaga', 'shalat', 'tepat', 'waktu', 'dan', 'melarang', 'meninggalkan', 'shalat']	abu bakr bin abi syaibah rasulullah shallallahu alaihi wasallam memerintahkan menjaga shalat tepat waktu melarang meninggalkan shalat

Berdasarkan Tabel 2, setiap tahapan *preprocessing* memberikan perubahan terhadap teks awal. Tahap *case folding* menyeragamkan bentuk huruf, tahap *cleaning* menghapus karakter yang tidak diperlukan, tahap *tokenization* memisahkan kalimat menjadi satuan kata, dan tahap *stopword removal* menghapus kata umum yang kurang berpengaruh terhadap klasifikasi. Hasil akhir dari proses ini berupa teks yang lebih bersih dan terstruktur. Kata-kata seperti memerintahkan, menjaga, shalat, melarang, dan meninggalkan tetap dipertahankan karena dapat membantu model mengenali pola pada label anjuran dan larangan.

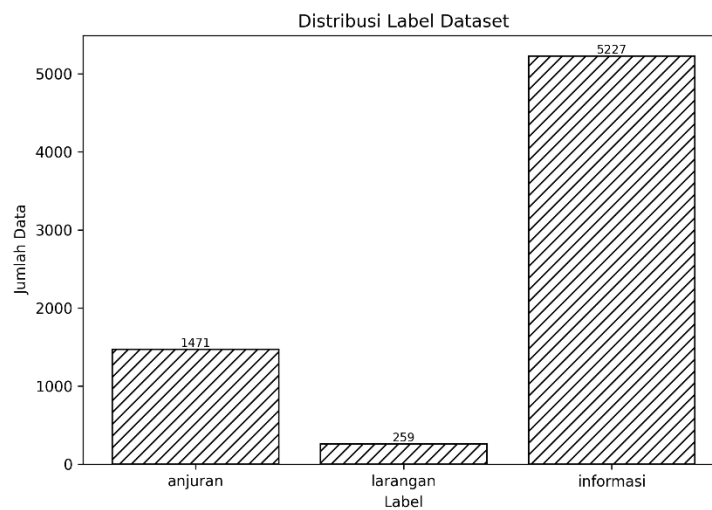
3.3 Distribusi Label Dataset

Distribusi label digunakan untuk melihat jumlah data pada setiap kategori. Karena penelitian ini menggunakan pendekatan *multi-label*, satu data hadits dapat dihitung pada lebih dari satu label apabila teks tersebut memuat beberapa kategori makna secara bersamaan. Distribusi label pada *dataset* ditunjukkan pada Tabel 3.

Tabel 3. Distribusi Label

Label	Jumlah Data
Anjuran	1.471
Larangan	259
Informasi	5.227

Berdasarkan Tabel 3, label informasi memiliki jumlah data paling banyak, yaitu 5.227 data. Label anjuran memiliki 1.471 data, sedangkan label larangan memiliki 259 data. Jumlah tersebut menunjukkan bahwa distribusi label pada *dataset* tidak seimbang. Label informasi menjadi label yang paling dominan, sedangkan label larangan menjadi label dengan jumlah data paling sedikit.



Gambar 2. Distribusi Label

Berdasarkan Gambar 2, dapat dilihat bahwa label informasi mendominasi keseluruhan data. Kondisi ini menunjukkan bahwa sebagian besar teks Hadits Shahih Muslim dalam *dataset* memuat unsur informasi atau narasi. Sebaliknya, label larangan memiliki jumlah data yang jauh lebih sedikit. Ketidakseimbangan ini dapat memengaruhi performa model karena model cenderung lebih mudah mengenali label yang memiliki jumlah data besar. Label dengan jumlah data sedikit memiliki pola yang lebih terbatas untuk dipelajari, sehingga berpotensi menghasilkan nilai *recall* dan *F1-score* yang lebih rendah.

3.4 Proses Ekstraksi Fitur Menggunakan TF-IDF

Setelah melalui tahap *preprocessing*, data teks diubah menjadi bentuk numerik menggunakan metode *Term Frequency-Inverse Document Frequency (TF-IDF)*. Tahap ini diperlukan karena algoritma *SVM* tidak dapat memproses teks secara langsung. Model membutuhkan data dalam bentuk angka atau vektor agar dapat melakukan proses pembelajaran.

TF-IDF bekerja dengan memberikan bobot pada setiap kata berdasarkan frekuensi kemunculan kata dalam suatu dokumen dan tingkat keunikan kata tersebut dalam keseluruhan dokumen. Kata yang sering muncul dalam satu dokumen, tetapi tidak terlalu sering muncul pada seluruh dokumen, akan memperoleh bobot yang lebih tinggi. Sebaliknya, kata yang terlalu sering muncul pada banyak dokumen akan memperoleh bobot lebih rendah karena dianggap kurang spesifik dalam membedakan kategori.

Dalam penelitian ini, hasil *preprocessing* diubah menjadi vektor *TF-IDF*. Vektor tersebut kemudian digunakan sebagai data masukan pada model *SVM*. Melalui proses ini, kata-kata seperti memerintahkan, hendaklah, melarang, dilarang, membunuh, meriwayatkan, dan kata lainnya dapat memperoleh bobot tertentu sesuai dengan kontribusinya terhadap teks hadits. Bobot tersebut membantu model mengenali pola kata yang berkaitan dengan label anjuran, larangan, dan informasi [6].

3.5 Baseline Model SVM

Tahap *baseline* model dilakukan untuk mengetahui kemampuan awal model *SVM* sebelum dilakukan optimasi parameter. Pada tahap ini, model dijalankan menggunakan parameter awal. Tujuan dari pengujian *baseline* adalah memperoleh nilai awal yang dapat digunakan sebagai pembandingan terhadap hasil model setelah dilakukan *hyperparameter tuning*.

Proses klasifikasi dilakukan menggunakan pendekatan *One-Vs-Rest*. Pada pendekatan ini, setiap label dipelajari secara terpisah. Model membangun pemisah untuk label anjuran, label larangan, dan label informasi. Untuk setiap label, model akan mempelajari apakah suatu data termasuk atau tidak termasuk ke dalam label tersebut. Dengan pendekatan ini, satu teks hadits tetap dapat memperoleh lebih dari satu label apabila pola kata dalam teks memenuhi beberapa kategori sekaligus [20].

Setelah data teks diubah menjadi vektor *TF-IDF*, model *SVM* membentuk batas pemisah atau *hyperplane* untuk membedakan data yang termasuk dan tidak termasuk pada suatu label. Pada klasifikasi label anjuran, model mempelajari kata-kata yang sering berkaitan dengan perintah atau anjuran. Pada klasifikasi label larangan, model mempelajari pola kata yang berkaitan dengan pelarangan. Pada klasifikasi label informasi, model mempelajari pola teks yang bersifat naratif atau informatif. Hasil evaluasi *baseline* model *SVM* ditunjukkan pada Tabel 4.

Tabel 4. Hasil Baseline SVM

Label	Precision	Recall	F1-Score	Support
Anjuran	0.4462	0.2673	0.3343	217

Larangan	0.5000	0.0667	0.1176	45
Informasi	0.9752	1.0000	0.9874	785
Macro Avg	0.6404	0.4446	0.4798	1047
Micro Avg	0.8990	0.8080	0.8511	1047

Berdasarkan Tabel 4, label informasi memperoleh performa terbaik dengan nilai *F1-score* sebesar 0,9874. Nilai *recall* pada label informasi mencapai 1,0000, yang menunjukkan bahwa seluruh data informasi pada data uji berhasil dikenali oleh model. Hal ini dipengaruhi oleh jumlah data informasi yang sangat dominan dibandingkan label lainnya.

Label anjuran memperoleh nilai *F1-score* sebesar 0,3343 dengan *recall* sebesar 0,2673. Nilai ini menunjukkan bahwa model masih belum optimal dalam mengenali data anjuran. Sementara itu, label larangan memperoleh nilai *F1-score* paling rendah, yaitu 0,1176, dengan *recall* sebesar 0,0667. Artinya, dari seluruh data yang seharusnya termasuk label larangan, hanya sebagian kecil yang berhasil dikenali oleh model.

Nilai *micro F1-score* sebesar 0,8511 menunjukkan bahwa performa model secara keseluruhan cukup baik. Namun, nilai *macro F1-score* sebesar 0,4798 menunjukkan bahwa performa antarlabel belum merata. Perbedaan tersebut terjadi karena nilai *micro* lebih dipengaruhi oleh label dengan jumlah data besar, sedangkan nilai *macro* menghitung rata-rata performa setiap label secara seimbang.

3.6 Hasil Hyperparameter Tuning SVM

Setelah diperoleh hasil *baseline*, dilakukan *hyperparameter tuning* menggunakan *GridSearchCV*. Tahap ini bertujuan untuk mencari kombinasi parameter terbaik yang dapat meningkatkan performa model. Parameter yang diuji meliputi parameter pada *TF-IDF* dan *SVM*, seperti jumlah fitur maksimum, *ngram_range*, *max_df*, nilai *C*, dan *class_weight*.

Pada proses tuning, *GridSearchCV* mencoba beberapa kombinasi parameter secara sistematis. Setiap kombinasi parameter diuji, kemudian dipilih kombinasi yang menghasilkan skor terbaik. Berdasarkan hasil pengujian, parameter terbaik yang diperoleh adalah nilai *C* sebesar 0,1, *class_weight* menggunakan *balanced*, *max_df* sebesar 0,6, *max_features* sebesar 3000, *ngram_range* (1,1), *smooth_idf* bernilai True, dan *sublinear_tf* bernilai True. Nilai best cross-validation score yang diperoleh adalah 0,5778.

Penggunaan *class_weight balanced* membuat model memberikan bobot yang lebih besar pada label dengan jumlah data lebih sedikit. Dalam penelitian ini, label larangan memiliki jumlah data paling sedikit sehingga model *baseline* sulit mengenali pola pada label tersebut. Dengan adanya penyesuaian bobot kelas, model diharapkan mampu mengenali label minoritas dengan lebih baik.

Hasil evaluasi model setelah *hyperparameter tuning* ditunjukkan pada Tabel 5.

Tabel 5. Hasil Evaluasi Setelah Hyperparameter Tuning

Label	Precision	Recall	F1-Score	Support
Anjuran	0.4052	0.5023	0.4486	217
Larangan	0.2564	0.4444	0.3252	45
Informasi	0.9830	0.9554	0.9690	785
Macro Avg	0.5482	0.6341	0.5809	1047
Micro Avg	0.7919	0.8395	0.8150	1047

Berdasarkan Tabel 5, hasil setelah *hyperparameter tuning* menunjukkan peningkatan pada label anjuran dan larangan. Nilai *F1-score* label anjuran meningkat dari 0,3343 menjadi 0,4486. Label larangan juga mengalami peningkatan *F1-score* dari 0,1176 menjadi 0,3252. Peningkatan ini menunjukkan bahwa proses tuning membantu model mengenali label yang jumlah datanya lebih sedikit.

Pada label larangan, nilai *recall* meningkat dari 0,0667 menjadi 0,4444. Peningkatan ini menunjukkan bahwa model setelah tuning lebih mampu menemukan data yang termasuk kategori larangan. Namun, nilai *precision* pada label larangan menurun dari 0,5000 menjadi 0,2564. Hal ini menunjukkan bahwa model menjadi lebih banyak memprediksi data sebagai label larangan, tetapi sebagian prediksi tersebut belum sepenuhnya tepat.

Label informasi mengalami sedikit penurunan *F1-score*, yaitu dari 0,9874 menjadi 0,9690. Penurunan ini terjadi karena model setelah tuning tidak lagi terlalu dominan pada label mayoritas. Model menjadi lebih sensitif terhadap label minoritas, sehingga keseimbangan performa antarlabel menjadi lebih baik meskipun performa label informasi sedikit menurun.

3.7 Perbandingan Evaluasi Baseline dan Setelah Hyperparameter Tuning

Perbandingan hasil evaluasi dilakukan untuk melihat pengaruh *hyperparameter tuning* terhadap performa model. Hasil perbandingan ditunjukkan pada Tabel 6.

Tabel 6. Hasil Evaluasi Model SVM

Metrik	Baseline SVM	SVM Setelah Hyperparameter Tuning
--------	--------------	-----------------------------------

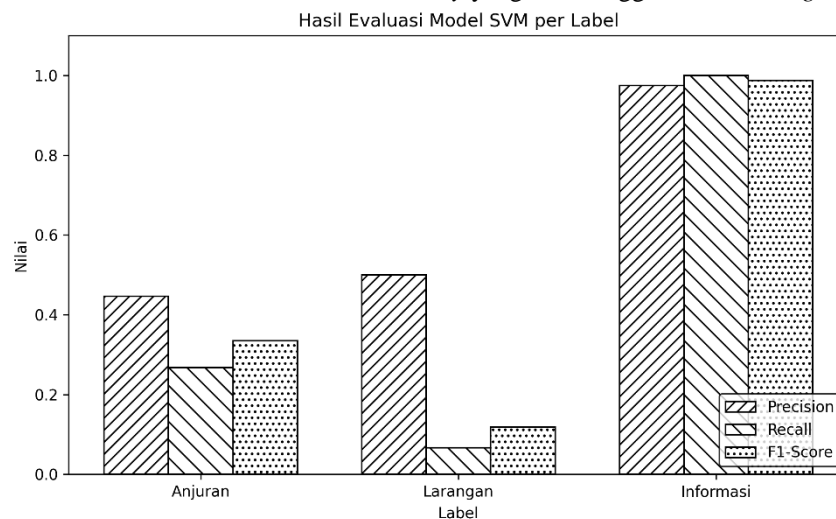


<i>Micro Precision</i>	0.8990	0.7919
<i>Micro Recall</i>	0.8080	0.8395
<i>Micro F1-Score</i>	0.8511	0.8150
<i>Macro Precision</i>	0.6404	0.5482
<i>Macro Recall</i>	0.4446	0.6341
<i>Macro F1-Score</i>	0.4798	0.5809
<i>Hamming Loss</i>	0.1226	0.1652
<i>Subset Accuracy</i>	0.6696	0.5764

Berdasarkan Tabel 6, nilai *macro recall* meningkat dari 0,4446 menjadi 0,6341. Nilai *macro F1-score* juga meningkat dari 0,4798 menjadi 0,5809. Peningkatan kedua metrik tersebut menunjukkan bahwa model setelah tuning memiliki kemampuan yang lebih baik dalam mengenali label secara lebih merata. Hal ini terutama terlihat pada peningkatan performa label anjuran dan larangan.

Namun, nilai *micro F1-score* menurun dari 0,8511 menjadi 0,8150. Nilai *hamming loss* juga meningkat dari 0,1226 menjadi 0,1652, sedangkan *subset accuracy* menurun dari 0,6696 menjadi 0,5764. Kondisi tersebut menunjukkan bahwa setelah tuning, model menjadi lebih sensitif dalam mengenali label minoritas, tetapi menghasilkan lebih banyak kesalahan pada kombinasi label secara keseluruhan.

Dengan demikian, hasil *hyperparameter tuning* memberikan pengaruh yang berbeda terhadap setiap metrik. Jika dilihat dari keseimbangan performa antarlabel, model setelah tuning lebih baik karena nilai *macro recall* dan *macro F1-score* meningkat. Namun, jika dilihat dari performa keseluruhan dan ketepatan kombinasi label, model *baseline* masih lebih stabil karena memiliki nilai *micro F1-score* dan *subset accuracy* yang lebih tinggi serta *hamming loss* yang lebih rendah.



Gambar 3. Grafik Hasil Evaluasi SVM

Berdasarkan Gambar 3, terlihat bahwa hasil evaluasi model mengalami perubahan setelah dilakukan *hyperparameter tuning*. Nilai *macro recall* dan *macro F1-score* mengalami peningkatan, sedangkan nilai *micro F1-score* dan *subset accuracy* mengalami penurunan. Pola ini menunjukkan bahwa proses tuning berhasil meningkatkan kemampuan model dalam mengenali label minoritas, tetapi belum menghasilkan peningkatan pada seluruh metrik evaluasi.

3.8 Fitur yang Berpengaruh pada Klasifikasi SVM

Analisis fitur dilakukan untuk mengetahui kata-kata yang berpengaruh terhadap proses klasifikasi pada setiap label. Fitur diperoleh dari hasil pembobotan *TF-IDF* dan koefisien model *SVM*. Kata dengan bobot tinggi menunjukkan bahwa kata tersebut memiliki kontribusi besar dalam membantu model mengenali label tertentu. Hasil fitur dominan pada setiap label ditunjukkan pada Tabel 7.

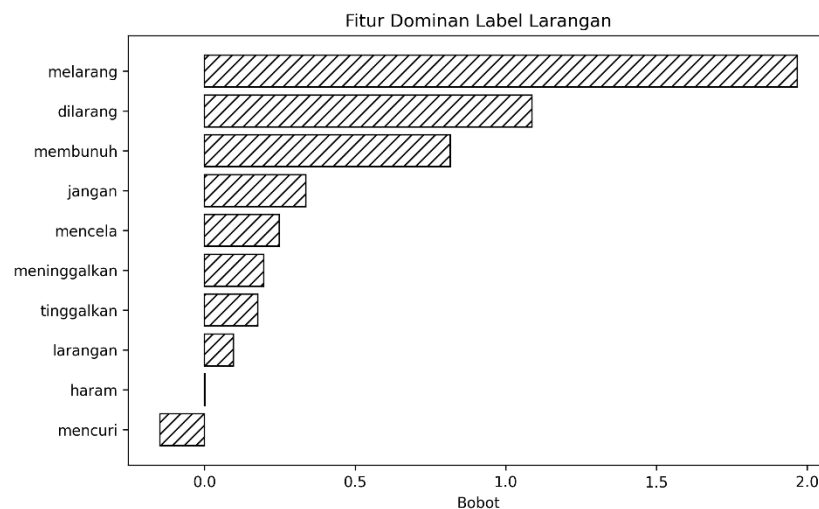
Tabel 7. Fitur Dominan Setiap Label

Label	Fitur	Bobot
Anjuran	memerintah	0.4674
Anjuran	hendaklah	0.4345
Anjuran	shalat	0.2263
Larangan	melarang	1.9673
Larangan	dilarang	1.0864
Larangan	membunuh	0.8153

Informasi	bakar	0.4902
Informasi	malik	0.1088
Informasi	meriwayatkan	0.0350

Berdasarkan Tabel 7, setiap label memiliki fitur dominan yang berkaitan dengan karakteristik maknanya. Pada label anjuran, fitur memerintahkan dan hendaklah berkaitan dengan bentuk perintah atau anjuran. Kata shalat juga muncul sebagai fitur yang berpengaruh karena banyak teks hadits yang berisi anjuran atau perintah terkait ibadah.

Pada label larangan, fitur melarang, dilarang, dan membunuh memiliki hubungan makna yang kuat dengan kategori larangan. Fitur-fitur tersebut membantu model mengenali teks yang mengandung unsur pelarangan. Pada label informasi, fitur bakar, malik, dan meriwayatkan lebih berkaitan dengan teks yang bersifat naratif atau informatif. Hal ini menunjukkan bahwa model tidak hanya menghasilkan nilai evaluasi, tetapi juga mampu menunjukkan fitur kata yang berpengaruh pada setiap kategori.



Gambar 4. Fitur Dominan Label Larangan

Berdasarkan Gambar 4, fitur melarang memiliki bobot paling tinggi pada label larangan, diikuti oleh fitur dilarang dan membunuh. Hal ini menunjukkan bahwa model mampu menangkap pola kata yang secara langsung berkaitan dengan kategori larangan. Selain itu, fitur seperti jangan, meninggalkan, dan tinggalkan juga berkontribusi terhadap klasifikasi label larangan. Fitur-fitur tersebut tetap berada dalam bentuk kata asli karena tahap *stemming* tidak digunakan, sehingga hubungan antara fitur dan label menjadi lebih mudah dijelaskan secara makna.

3.9 Pembahasan

Hasil penelitian menunjukkan bahwa proses klasifikasi *multi-label* pada teks Hadits Shahih Muslim dapat dilakukan menggunakan metode *SVM* dengan pendekatan *One-Vs-Rest*. Tahapan penelitian dimulai dari pengolahan *dataset*, *preprocessing* teks, transformasi fitur menggunakan *TF-IDF*, hingga proses klasifikasi. Model yang dibangun mampu mengenali pola teks hadits dalam tiga kategori utama, yaitu anjuran, larangan, dan informasi, dengan hasil evaluasi menunjukkan *micro F1-score* sebesar 0,8722, *macro F1-score* sebesar 0,5547, *hamming loss* sebesar 0,1048, dan *subset accuracy* sebesar 0,7031. Hasil ini menunjukkan bahwa model memiliki performa yang cukup baik dalam klasifikasi *multi-label*, meskipun masih terdapat perbedaan kemampuan dalam mengenali setiap kategori label.

Jika dibandingkan dengan penelitian sebelumnya, hasil ini sejalan dengan studi yang dilakukan oleh Hanafi et al. [7] dan Kustiawan et al. [8] yang menunjukkan bahwa pendekatan *multi-label* efektif digunakan untuk teks hadits karena satu teks dapat mengandung lebih dari satu makna. Namun, penelitian tersebut menggunakan metode lain seperti *k-NN*, *CART*, dan *Bagging*, sedangkan penelitian ini menunjukkan bahwa *SVM* tetap mampu memberikan performa yang kompetitif dalam skenario *multi-label* dengan data teks berdimensi tinggi.

Selain itu, hasil penelitian ini juga sejalan dengan temuan Ramadhani et al. [9] yang membandingkan *SVM* dengan metode *deep learning* (*LSTM*) pada klasifikasi hadits. Dalam penelitian tersebut, *SVM* menunjukkan performa yang stabil pada data teks, meskipun metode *deep learning* memiliki keunggulan pada representasi fitur yang lebih kompleks. Hal ini juga terlihat pada penelitian ini, di mana *SVM* mampu menghasilkan performa yang baik terutama pada label dengan distribusi data besar seperti label informasi.

Hasil baseline pada penelitian ini menunjukkan bahwa model sangat baik dalam mengenali label informasi, tetapi lemah pada label anjuran dan larangan. Pola ini juga konsisten dengan penelitian Ningsih et al. [16] yang menyatakan bahwa ketidakseimbangan data pada klasifikasi teks menyebabkan model lebih dominan mempelajari kelas mayoritas dan mengabaikan kelas minoritas. Dengan demikian, permasalahan *imbalance* bukan hanya terjadi pada penelitian ini, tetapi

juga umum ditemukan pada studi klasifikasi teks lainnya.

Setelah dilakukan *hyperparameter tuning*, terjadi peningkatan pada kemampuan model dalam mengenali label minoritas, terutama pada nilai *recall* dan *macro F1-score*. Hasil ini sejalan dengan penelitian Lundel [25] dan Turner et al. [26] yang menunjukkan bahwa optimasi *hyperparameter* dapat meningkatkan keseimbangan performa model, meskipun tidak selalu meningkatkan seluruh metrik secara bersamaan. Dalam penelitian ini, peningkatan *macro recall* menunjukkan bahwa tuning membantu model lebih sensitif terhadap label minoritas, meskipun terjadi penurunan pada *micro F1-score* dan *subset accuracy*.

Jika dibandingkan dengan penelitian sebelumnya yang cenderung hanya mengevaluasi performa model secara umum, penelitian ini memberikan tambahan kontribusi berupa analisis ketidakseimbangan label secara lebih mendalam melalui perbandingan *macro* dan *micro metric*. Selain itu, penelitian ini juga memperkuat temuan pada studi Efrizoni et al. [10] bahwa representasi fitur dan distribusi label sangat memengaruhi hasil klasifikasi *multi-label*.

Secara keseluruhan, hasil penelitian ini menunjukkan bahwa SVM masih relevan dan efektif digunakan untuk klasifikasi teks hadits *multi-label*, terutama pada data berdimensi tinggi berbasis TF-IDF. Namun, seperti yang juga ditemukan pada penelitian-penelitian sebelumnya performa model tetap sangat dipengaruhi oleh ketidakseimbangan distribusi label. Oleh karena itu, pengembangan selanjutnya perlu mempertimbangkan teknik *balancing data* atau pendekatan model lain untuk meningkatkan performa pada label minoritas.

4. KESIMPULAN

Penelitian ini memberikan kontribusi dalam pengembangan klasifikasi teks keagamaan berbasis Natural Language Processing (NLP) dengan menerapkan Support Vector Machine (SVM) pada data Hadits Shahih Muslim. Kontribusi utama penelitian ini terletak pada analisis performa model dalam kondisi ketidakseimbangan label menggunakan evaluasi *macro* dan *micro*, serta identifikasi fitur dominan yang memberikan interpretasi terhadap pola kata pada setiap kategori makna hadits. Penelitian ini berhasil membuktikan bahwa metode Support Vector Machine (SVM) efektif digunakan untuk klasifikasi *multi-label* pada teks Hadits Shahih Muslim terjemahan Bahasa Indonesia ke dalam kategori anjuran, larangan, dan informasi. Seluruh rangkaian penelitian, mulai dari pengolahan dataset, tahap text preprocessing, transformasi fitur menggunakan TF-IDF, hingga penerapan klasifikasi *multi-label* dengan pendekatan One-Vs-Rest dan evaluasi model, menunjukkan bahwa SVM mampu mengenali pola teks dengan baik. Hasil evaluasi model memperlihatkan *micro F1-score* sebesar 0,8722, *macro F1-score* sebesar 0,5547, *hamming loss* sebesar 0,1048, dan *subset accuracy* sebesar 0,7031, menandakan performa keseluruhan yang cukup tinggi serta kemampuan model dalam memprediksi kombinasi label dengan akurasi yang memadai. Analisis fitur mengungkapkan bahwa kata-kata dominan seperti memerintahkan, melarang, dan bakar berkontribusi signifikan dalam membedakan kategori label, menunjukkan bahwa model tidak hanya akurat tetapi juga mampu mengekstrak pola linguistik yang relevan. Meski demikian, performa pada label minoritas, terutama larangan, masih lebih rendah karena ketidakseimbangan distribusi data. Hasil tuning *hyperparameter* dengan GridSearchCV meningkatkan *macro F1-score* dan *macro recall*, memperbaiki kemampuan model dalam mengenali label minoritas, meskipun beberapa metrik global seperti *micro F1-score* dan *subset accuracy* menurun. Secara keseluruhan, penelitian ini menegaskan bahwa SVM dapat diterapkan secara efektif untuk klasifikasi *multi-label* teks hadits, dan pengembangan lebih lanjut melalui penyeimbangan data, optimasi parameter, serta perluasan dataset dapat meningkatkan keseimbangan performa antar label, sehingga model mampu mengenali kategori minoritas secara lebih optimal tanpa mengorbankan akurasi keseluruhan.

REFERENCES

- [1] M. R. Athallah and K. M. Lhaksana, "Hadisth Text Classification Based on Topic Using Convolutional Neural Network (CNN) and TF-IDF," *J. Renew. Energy, Electr. Comput. Eng.*, vol. 5, no. November, pp. 30–36, 2025.
- [2] M. Yuslan and A. Bakar, "Klasifikasi Teks Hadis Bukhari Terjemahan Indonesia Menggunakan Recurrent Convolutional Neural Network," *J. JTIK*, vol. 8, no. 5, pp. 907–918, 2021, doi: 10.25126/jtiik.202183750.
- [3] C. H. Saputra, "Analisis Bert-Cnn Untuk Klasifikasi Multi-Label Diskusi Keagamaan Dan Asosiasi Dengan Al-Qur'an Dan Hadits," *JTIK (Jurnal Tek. Inform. Kaputama)*, vol. 12, no. 6, pp. 1415–1424, 2025.
- [4] R. A. Munir, A. Voutama, U. S. Karawang, T. Timur, and J. Barat, "Analisis Sentimen Cuitan Di Media Sosial X Tentang Program Makan Bergizi Gratis Dengan Metode NLP," *J. Inform. dan Elektro Terap.*, vol. 13, no. 3, pp. 589–595, 2025.
- [5] R. Aria, K. M. Lhaksana, and R. Forest, "Klasifikasi Multi-Label Ayat-Ayat Al-Qur'an Menggunakan Random Forest dan Word Centrality," *J. Log.*, vol. 2, no. 2, pp. 9–14, 2024.
- [6] R. Kustiawan and M. D. Purbolaksono, "A Multi-label Classification on Topic of Hadith Verses in Indonesian Translation using CART and Bagging," vol. 6, no. April, pp. 868–875, 2022, doi: 10.30865/mib.v6i2.3787.
- [7] I. Akbar et al., "Penerapan Long Short-Term Memory untuk Klasifikasi Multi-Label," *J. JOINTECH*, vol. 7, no. 1, pp. 41–54, 2026.
- [8] M. T. Hidayat, M. Arifin, and S. Muzid, "Prediction Sentiment Analysis Grab Reviews using SVM Linear Based Streamlit," *Indones. J. Comput. Cybern. Syst.*, vol. 19, no. 2, pp. 1–12, 2025, doi: 10.22146/ijccs.104924.
- [9] L. Pertiwi, "Penerapan Algoritma Text Mining, Steaming Dan Texrank Dalam Peringkasan Bahasa Inggris," *Bull. Multi-Disciplinary Sci. Appl. Technol.*, vol. 1, no. 3, pp. 100–104, 2022.



- [10] Y. T. Handika, S. Defit, and G. W. Nurcahyo, “Text Mining Dalam Membandingkan Metode Naive Bayes Dengan C.45 Dalam Mengidentifikasi Berita Hoax Pada Media Sosial,” *Rang Tek. J.*, vol. 5, no. 1, pp. 116–123, 2022, doi: 10.31869/rtj.v5i1.2855.
- [11] H. Farman, M. A. Hussain, S. Hassan, S. Shaikh, and K. Ali, “Activation function impact on rainfall prediction: comparative insights across ML and DL architectures,” *Model. Earth Syst. Environ.*, vol. 11, no. 6, 2025, doi: 10.1007/s40808-025-02630-6.
- [12] R. Meenal, K. Kailash, P. A. Michael, J. J. Joseph, F. T. Josh, and E. Rajasekaran, “Machine learning based smart weather prediction,” *Indones. J. Electr. Eng. Comput. Sci.*, vol. 28, no. 1, pp. 508–515, 2022, doi: 10.11591/ijeecs.v28.i1.pp508-515.
- [13] Sopiatal Ulum, R. F. Alifa, P. Rizkika, and C. Rozikin, “Perbandingan Performa Algoritma KNN dan SVM dalam Klasifikasi Kelayakan Air Minum,” *Gener. J.*, vol. 7, no. 2, pp. 141–146, 2023, doi: 10.29407/gj.v7i2.20270.
- [14] N. Rini, “Sistem Informasi Manajemen Pengklasifikasian Hadist Shahih Bukhari dan Muslim Menggunakan Algoritma Neural Network,” *J. NCBMA*, vol. 2, no. june, pp. 1011–1023, 2024.
- [15] D. F. Surlianto, M. F. B, and M. R. Mulia, “Comparative Analysis of the Performance of Hadith Text Classification Methods : A Case Study with ANN and SVM,” *J. JESSI*, vol. 05, no. 01, pp. 89–98, 2024.
- [16] E. Damayanti, A. V. Vitianingsih, S. Kacung, and D. Cahyono, “Sentiment Analysis of Alfagift Application User Reviews Using Long Short-Term Memory (LSTM) and Support Vector Machine (SVM) Methods,” *J. Decod.*, vol. 4, no. 2, pp. 509–521, 2024.
- [17] R. Z. N. Ahmad, “Perbandingan Performa Random Forest dan Long Short-Term Memory Dalam Klasifikasi Teks Multi Abel Terjemahan Hadist Bukhari,” *J. Malcom*, vol. 5, no. 3 juli, pp. 862–874, 2025.
- [18] H. Mustakim and S. Priyanta, “Aspect-Based Sentiment Analysis of KAI Access Reviews Using NBC and SVM,” *IJCCS (Indonesian J. Comput. Cybern. Syst.*, vol. 16, no. 2, p. 113, 2022, doi: 10.22146/ijccs.68903.
- [19] C. A. Bahri and L. H. Suadaa, “Aspect-Based Sentiment Analysis in Bromo Tengger Semeru National Park Indonesia Based on Google Maps User Reviews,” *IJCCS (Indonesian J. Comput. Cybern. Syst.*, vol. 17, no. 1, p. 79, 2023, doi: 10.22146/ijccs.77354.
- [20] A. S. Aiman and K. M. Lhaksmana, “Topic Classification of Quranic Verses in English Translation,” *J. REST*, vol. 5, no. 158, pp. 803–809, 2026.