Institut Riset dan Publikasi Indonesia (IRPI)
MALCOM: Indonesian Journal of Machine Learning and Computer Science
Journal Homepage: https://journal.irpi.or.id/index.php/malcom
Vol. 5 Iss. 3 July 2025, pp: 896-907
ISSN(P): 2797-2313 | ISSN(E): 2775-8575

Performance Comparison of Multilabel Text Classification Methods on Translated Hadiths of Bukhari Using Support Vector Machine and Long Short Term Memory

Perbandingan Performa Metode Klasifikasi Teks *Multilabel* Hadis Terjemahan Bukhari Menggunakan Support Vector Machine dan Long Short Term Memory

Aulia Ramadhani[1], Nazruddin Safaat Harahap[2*], Surya Agustian[3] , Iwan Iskandar[4], Suwanto Sanjaya[5]

[1,2,3,4,5]Program Studi Teknik Informatika, Fakultas Sains dan Teknologi,
Universitas Islam Negeri Sultan Syarif Kasim Riau, Indonesia

E-Mail: [1]12150124752@students.uin-suska.ac.id, [2]nazruddin.safaat@uin-suska.ac.id,
[3]surya.agustian@uin-suska.ac.id, [4]iwan.iskandar@uin-suska.ac.id, [5]suwantosanjaya@uin-suska.ac.id

Received Apr 16th 2025; Revised Jun 12th 2025; Accepted Jun 18th 2025; Available Online Jun 24th 2025, Published Jun 24th 2025
Corresponding Author: Nazruddin Safaat Harahap

Abstract

Hadith is the second source of law in Islam, and one of the most well-known hadith collections is Sahih al-Bukhari. To support accurate understanding and practice, hadith needs to be classified precisely. Considering that a single hadith can contain more than one type of information, a multilabel classification approach becomes highly relevant. This study aims to contribute to the field of text classification by exploring optimal combinations of methods and parameters for multilabel classification of hadith. The results show that Support Vector Machine (SVM) achieved the best performance on the Prohibition label with a Macro F1-score of 82.57%, using a combination of SVM + TF-IDF with a linear kernel and regularization parameter C = 1, without stopword removal or class balancing. Meanwhile, LSTM also performed well on the Prohibition label with a Macro F1-score of 82.66%, using parameters of 20 epochs, 0.5 dropout rate, 128 dense units, and a batch size of 64, also without stopword removal or class balancing. This configuration also resulted in the lowest Hamming Loss of 10.452%, which outperformed previous studies and demonstrated that LSTM is overall more effective when properly tuned. This study also contributes to data quality improvement by completing the matan (text) of the hadith used, thereby achieving better classification performance..

Keyword: Classification, Hadith Bukhari, Long Short Term Memory, Multilabel, Support Vector Machine

Abstrak

Hadis merupakan sumber hukum kedua dalam Islam, dan salah satu kitab hadis yang paling dikenal adalah Shahih al-Bukhari. Untuk mendukung pemahaman dan pengamalan yang tepat, hadis perlu diklasifikasikan secara akurat. Mengingat satu hadis dapat mengandung lebih dari satu informasi, pendekatan klasifikasi *multilabel* menjadi sangat relevan. Penelitian ini bertujuan untuk memberikan kontribusi dalam bidang klasifikasi teks dengan mengeksplorasi kombinasi metode dan parameter yang optimal untuk klasifikasi *multilabel* hadis. Hasil penelitian menunjukkan bahwa Support Vector Machine (SVM) memberikan performa terbaik pada label Larangan dengan Macro F1-score sebesar 82,57%, melalui kombinasi SVM + TF-IDF menggunakan kernel = linear, parameter C (*regularization parameter*) = 1 tanpa stopword removal dan tanpa balancing. Sementara itu, Long Short Term Memory (LSTM) juga unggul pada label Larangan dengan Macro F1-score 82,66% pada kombinasi parameter Epoch = 20, Dropout = 0.5, Dense = 128 dan Batch Size = 64 tanpa stopword removal dan tanpa balancing kombinasi ini juga menghasilkan nilai Hamming Loss terendah sebesar 10,452%, yang lebih baik dibandingkan dengan penelitian sebelumnya serta menunjukkan bahwa LSTM terbukti lebih efektif secara keseluruhan dengan penyetelan parameter yang tepat. Penelitian ini juga berkontribusi dalam peningkatan kualitas data dengan melengkapi matan hadis yang digunakan, sehingga menghasilkan performa klasifikasi yang lebih baik.

Kata Kunci: Hadis Bukhari, Klasifikasi, Long Short Term Memory, *Multilabel,* Support Vector Machine

1. PENDAHULUAN

Mayoritas penduduk Indonesia beragama Islam, dengan lebih dari 207 juta muslim atau 87,2% dari total populasi [1]. Sebagai Umat Islam terdapat dua pedoman hidup yaitu Al-Qur'an dan Hadis. Hadis yang berisi ucapan dan tindakan Nabi Muhammad SAW, berfungsi melengkapi Al-Qur'an dan memberikan penjelasan lebih lanjut tentang ajaran Islam. Salah satu kitab hadis yang populer adalah Shahih al-Bukhari, yang berisi ribuan hadis sahih yang dikumpulkan oleh Imam Bukhari. Kitab ini telah diterjemahkan ke dalam bahasa Indonesia dan menjadi rujukan utama bagi umat Islam [2].

Salah satu tantangan utama dalam mempelajari hadis adalah mengidentifikasi jenis ajaran yang terkandung di dalamnya. Matan hadis dapat memuat anjuran, larangan, atau informasi lainnya secara bersamaan, sehingga ketidakjelasan ini seringkali menyulitkan umat Islam dalam memahami dan mengamalkan hadis secara tepat. Klasifikasi teks konvensional yang hanya mengasumsikan satu label per dokumen kurang efektif digunakan karena satu hadis dapat mengandung beberapa kategori sekaligus. Oleh karena itu, diperlukan metode klasifikasi *multilabel* yang lebih canggih dan sistematis agar hadis dapat dikategorikan secara tepat dan komprehensif [3]. Klasifikasi *multilabel* telah menjadi perhatian besar dalam komunitas pembelajaran mesin, dengan berbagai kajian yang menyusun survei komprehensif mengenai algoritma *multilabel* dan *dataset* terkait [4].

Seiring dengan perkembangan pembelajaran mesin, algoritma *Support Vector Machine* (SVM) dinilai sebagai metode paling tepat dan akurat untuk *multilabel* dari algoritma *machine learning* lainnya [5]. SVM juga merupakan algoritma yang menjadi tren pada lima tahun terakhir untuk topik klasifikasi [6]. Sementara itu, algoritma *Long Short Term Memory* (LSTM) juga dinilai unggul dalam menangani klasifikasi teks berdimensi tinggi karena kemampuannya mengelola informasi berurutan tanpa kehilangan detail penting [7]. Selain itu, LSTM efektif dalam klasifikasi *multilabel*, karena dapat mengenali bahwa satu teks bisa masuk ke lebih dari satu kategori [8].

Beberapa penelitian klasifikasi *multilabel* telah dilakukan seperti yang dilakukan oleh Bakar dkk. (2018) yang mengklasifikasikan hadis Bukhari terjemahan menggunakan *Information Gain* sebagai metode seleksi fitur dan *Backpropagation Neural Network* (BPNN) sebagai metode klasifikasi, mendapat akurasi 88,42% pada kasus *multilabel* dan 65,28% pada *single-label*. Studi tersebut juga menemukan bahwa *stemming* dapat menghilangkan informasi diskriminatif yang berpotensi menurunkan performa model *multilabel* [9]. Hal ini sejalan dengan penelitian yang dilakukan Hanafi dkk. (2020) yang menunjukkan bahwa *stemming* tidak meningkatkan kinerja karena menghilangkan informasi morfologi. Penelitian tersebut menggunakan *Mutual Information* dan *K-Nearest Neighbor* (K-NN) dan memperoleh akurasi 91,14% dalam waktu 595 detik [3].

Algoritma *machine learning* juga digunakan pada klasifikasi *multilabel* pada hadis Bukhari terjemahan dengan *Classification And Regression Trees* (CART) dan metode *ensemble learning* Bagging oleh Kustiawan dkk. (2022), yang mencapai akurasi 80,86% menggunakan pra-pemrosesan sederhana, yaitu penghapusan tanda baca dan konversi huruf menjadi kecil tanpa *stemming* [10]. Sementara itu, penelitian juga dilakukan dengan algoritma SVM dan metode Chi-Square yang dilakukan oleh Taufiqurrahman dkk. (2021) mencatat Macro F1-score sebesar 75,32%, dengan hasil yang menunjukkan bahwa *stopword removal* meningkatkan performa dan *kernel* "linear" merupakan pilihan terbaik untuk SVM [11].

Selain hadis, penelitian pada klasifikasi *multilabel* terjemahan Al-Qur'an juga dilakukan. Salah satunya menggunakan *Bidirectional* LSTM (BiLSTM) dan Word2Vec menggunakan dengan *Continuous Bag of Words* (CBOW) dengan hasil akurasi 70,21%, *precision* 64,31%, *recall* 61,13%, dan *hamming loss* 36,52% , yang menyoroti tantangan kompleksitas bahasa Al-Qur'an [8]. Studi lain mengombinasikan *Convolutional Neural Network* (CNN), BiLSTM, dan *FastText* untuk terjemahan Al-Qur'an Indonesia, dengan model CNN+BiLSTM tanpa *FastText* mencapai akurasi 68,70% (pengujian 80:20), dan model dengan *FastText* mencapai 73,30% pada *embedding* 200 dan *epoch* 100 (pengujian 90:10) [12]. Klasifikasi multilabel Al-Qur'an juga dilakukan oleh Fatiara dkk (2024) membandingkan metode K-NN dan LSTM pada data terjemahan Al-Qur'an, dan menunjukkan bahwa LSTM memberikan performa terbaik dengan rata-rata F1-score 65% dan akurasi 96%, sedangkan K-NN hanya mencapai F1-score 55% dan akurasi 93% [13].

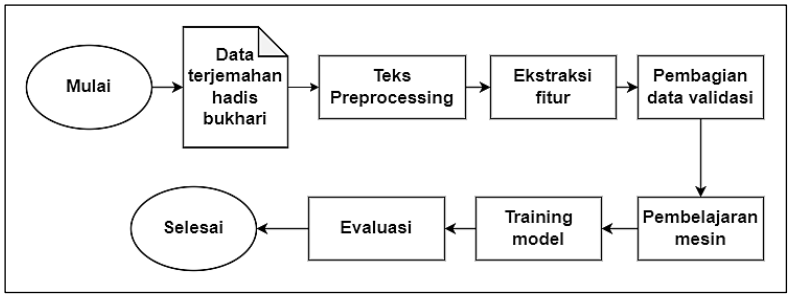
Penelitian oleh Sari dkk. (2020) dengan menggunakan LSTM dan kombinasi embedding *GloVe* mendapatkan kinerja yang baik dalam tugas klasifikasi teks. Penelitian menunjukkan bahwa pemilihan hyperparameter yang tepat dan preprocessing data sangat penting untuk mencapai performa tinggi. Model keenam dalam penelitian ini mencapai akurasi tertinggi sebesar 95,17%, dengan rata-rata *presision*, *recall*, dan F1-score juga sekitar 95. Algoritma LSTM dengan *embedding GloVe* menunjukkan performa tinggi dalam klasifikasi teks, dengan model terbaik mencapai akurasi 95,17%, serta rata-rata *Precision*, *Recall*, dan *Macro* F1-score sekitar 95, menegaskan efektivitas LSTM dalam pemrosesan bahasa alami [14].

Perbandingan performa kedua metode diperlukan untuk mengevaluasi efektivitas masing-masing algoritma klasifikasi, baik dari segi akurasi maupun efisiensi waktu komputasi. Algoritma klasifikasi berbasis *machine learning* cenderung memiliki waktu komputasi yang lebih cepat, sementara metode *deep learning* menawarkan performa klasifikasi yang lebih unggul [15]. Dengan mempertimbangkan keunggulan SVM dan LSTM, diharapkan penelitian ini dapat menghasilkan nilai F1-score yang optimal dalam tugas klasifikasi *multilabel* pada teks hadis.

Dalam upaya memajukan kajian Islam melalui pendekatan teknologi, penelitian ini mengevaluasi dan membandingkan performa algoritma SVM dan LSTM dalam klasifikasi *multilabel* pada teks terjemahan hadis Shahih al-Bukhari. Pendekatan perbandingan antara *machine learning* dan *deep learning* yang masih jarang dikaji dalam studi sebelumnya, bertujuan untuk memberikan wawasan baru mengenai metode, fitur, dan parameter yang paling efektif dalam klasifikasi *multilabel* teks hadis. Hasil evaluasi ini diharapkan dapat menjadi acuan dalam pengembangan sistem klasifikasi otomatis untuk teks keislaman bagi masyarakat khususnya Umat Islam.

2. METODOLOGI PENELITIAN

Penelitian ini dibangun dengan Gambaran *flowchart* yang ditunjukkan pada Gambar 1.



Gambar 1. Alur metode penelitian

Gambar 1, memvisualisasikan alur dari penelitian yang dilakukan. Berikut penjelasan lebih rinci mengenai setiap poin dari *flowchart* tersebut.

2.1. *Dataset*

Dataset yang digunakan dalam penelitian ini berupa kumpulan hadis Bukhari yang telah diterjemahkan ke dalam bahasa Indonesia. Data ini diperoleh dari penelitian terdahulu [9] yang telah melalui proses pelabelan oleh para ahli. *Dataset* terbagi menjadi dua komposisi: data *training* yang digunakan untuk melatih model agar dapat memahami pola dalam data dan data *testing* digunakan untuk mengevaluasi performa model setelah dilatih. *Dataset* terdiri dari 7.000 data hadis yang telah dilabeli dalam tiga kategori utama: Anjuran, Informasi, Larangan. Ketiga kategori ini diambil dari penelitian sebelumnya yang menjadi sumber data penelitian ini [9]. Tabel 1 merepresentasikan struktur *dataset* dengan format *multilabel.*

Tabel 1. Representasi dataset [9]

No.	Hadis	Kelas		
		Anjuran	Larangan	Informasi
25	Aku diperintahkan untuk memerangi manusia hingga mereka bersaksi; tidak ada ilah kecuali Allah dan bahwa sesungguhnya Muhammad adalah utusan Allah, menegakkan shalat, menunaikan zakat. Jika mereka lakukan yang demikian maka mereka telah memelihara darah dan harta mereka dariku kecuali dengan haq Islam dan perhitungan mereka ada pada Allah	1	0	1
36	tidaklah kamu menafkahkan suatu nafkah yang dimaksudkan mengharap wajah Allah kecuali kamu akan diberi pahala termasuk sesuatu yang kamu suapkan ke mulut istrimu.	0	1	1
167	sebab itu hanyalah semisal keringat dan bukan darah haid. Jika datang haidmu maka tinggalkan shalat, dan jika telah terhenti maka bersihkanlah sisa darahnya lalu shalat. Hisyam berkata, Bapakku (Urwah) menyebutkan, Berwudlulah kamu setiap akan shalat hingga waktu itu tiba.	1	1	1

Dari Tabel 1, *dataset* merupakan klasifikasi *multilabel* yang artinya memberi banyak label pada satu data [4]. Nilai 0 berarti data tidak termasuk dalam label dan nilai 1 data merupakan bagian dari label.

2.2. Teknik Pembagian *Dataset*

Pembagian *dataset* dilakukan untuk memastikan model memiliki data yang cukup untuk belajar sekaligus diuji performanya. Skema pembagian yang digunakan adalah sebagai berikut: 80% Data *training* (5.600 hadis) dan 20% Data *Testing* (1.400 hadis) [9].

2.3. Text Preprocessing dan Pelengkapan Data

Pada tahap ini dilakukan peninjauan ulang terhadap isi matan hadis dari *dataset* penelitian sebelumnya [9]. Ditemukan bahwa beberapa matan hadis tidak utuh, seperti salah satu contoh pada Tabel 2, matan hadis hanya terdiri dari frasa pendek seperti *"dia pun dirajam."*, hal ini dapat menyebabkan kehilangan informasi penting pada proses pelatihan dan pelabelan. Oleh karena itu, dilakukan pelengkapan matan hadis untuk memastikan bahwa informasi yang digunakan oleh model benar-benar representatif dan memiliki konteks yang utuh. Langkah ini diambil untuk meminimalkan potensi kesalahan pemahaman model terhadap makna suatu hadis, serta untuk meningkatkan akurasi klasifikasi. Representasi *dataset* dari penelitian sebelumnya [9].

Tabel 2. Representasi data frasa pendek [9]

No.	Isi Hadis
117	bila berbicara diulangnya tiga kali hingga dapat dipahami dan bila mendatangi kaum, Beliau memberi salam tiga kali.
6964	Berpuasalah kalian pada hari itu.
6999	dia pun dirajam.

Pada Tabel 2, terlihat matan hadis tidak utuh maka dilakukan proses pelengkapan matan hadis dengan mencocokkan antara data lama dengan sumber hadis terjemahan yang lebih lengkap [16]. Untuk mempercepat dan mengefisienkan proses ini, digunakan pendekatan berbasis *regular expression* (regex) guna mencari kesamaan atau potongan frasa yang cocok antara dua sumber. Agar fungsi regex dapat maksimal dalam mencocokkan data, beberapa kata seperti "rasulullah", "nabi", "shallallahu", "alaihi", "wasallam" dikecualikan dalam pencarian. Dengan teknik ini, sistem dapat mengidentifikasi bagian hadis dengan mengecualikan kata-kata tersebut, kemudian mencocokkannya dengan versi hadis asli secara otomatis dan semi-otomatis. Sebelum data diproses, kedua dataset akan dilakukan beberapa tahapan *teks preprocessing* terlebih dahulu seperti, cleaning (membersihkan data dari tanda baca) dan case folding (membuat semua huruf menjadi *lowercase*). Tujuan dilakukan teks preprocessing tersebut agar memaksimalkan fungsi regex dalam memproses data. Setelah data berhasil ditemukan, hasil temuan akan dikembalikan dalam versi aslinya tanpa *teks preprocessing.* Representasi *dataset* yang telah dilengkapkan ditunjukkan pada Tabel 3.

Tabel 3. Representasi data baru

No. Hadis	Isi Hadis
93	Telah menceritakan kepada kami ['Abdah bin Abdullah Ash Shafar] Telah menceritakan kepada kami [Abdushshamad] berkata, Telah menceritakan kepada kami [Abdullah bin Al Mutsanna] berkata; [Tsumamah bin Abdullah] telah menceritakan kepada kami dari [Anas] dari Nabi shallallahu 'alaihi wasallam, bahwa Nabi shallallahu 'alaihi wasallam bila berbicara diulangnya tiga kali hingga dapat dipahami dan bila mendatangi kaum, Beliau memberi salam tiga kali.
1866	Telah menceritakan kepada kami ['Ali bin 'Abdullah] telah menceritakan kepada kami [Abu Usamah] dari [Abu 'Umais] dari [Qais bin Muslim] dari [Thoriq bin Sihab] dari [Abu Musa radliallahu 'anhu] berkata: "Hari 'Asyura' telah dijadikan oleh orang-orang Yahudi sebagai hari raya mereka, maka Nabi shallallahu 'alaihi wasallam bersabda: "Berpuasalah kalian pada hari itu".
6324	Telah menceritakan kepada kami [Abdullah bin Muhammad Al Ju'fi] Telah menceritakan kepada kami [Wahb bin Jarir] telah menceritakan kepada kami [Ayahku] ia mengatakan; aku mendengar [Ya'la bin Hakim] dari ['Ikrimah] dari [Ibnu 'Abbas] radliallahu 'anhuma mengatakan; 'Ketika Ma'iz bin Malik menemui Nabi shallallahu 'alaihi wasallam, Nabi bertanya: "bisa jadi kamu hanya sekedar mencium, meremas, atau memandang!" Ma'iz menjawab; 'Tidak ya Rasulullah! ' -beliau bertanya lagi; "apakah kamu benar-benar menyetubuhinya?" -beliau tidak menggunakan bahasa kiasan.- maka pada saat itu dia pun dirajam.

Setelah data hadis dilengkapi seperti pada Tabel 3, dataset tersebut yang akan digunakan selama penelitian. Tahapan *text preprocessing* penting untuk dilakukan guna meringkas dan mengelompokkan dokumen [17]. Sebelum dilakukan teks preprocessing, dilakukan pengecekan untuk tanda '] ' terakhir pada setiap dokumen guna memisahkan sanad dengan kandungan matan. Tahapan *text preprocessing* yang dilakukan seperti pada penelitian [2] adalah:

1. *Case folding*, proses mengubah teks yang ada pada data menjadi huruf kecil semua.
2. *Cleaning*, proses menghapus tanda baca.
3. *Tokenization*, pemenggalan setiap kata untuk diolah.
4. *Stopword removal,* proses menghapus kata yang bersifat umum.

Pada penelitian ini, proses *stemming* tidak dilakukan. Hal ini disebabkan oleh dua alasan utama: (1) *stemming* membutuhkan waktu komputasi yang cukup lama [9], dan (2) hasil *stemming* cenderung menurunkan akurasi klasifikasi karena dapat menghilangkan struktur asli dari kata yang memiliki makna khusus dalam konteks hadis [3], [10].

2.4. Ekstraksi Fitur

Metode *Term Frequency-Inverse Document Frequency* (TF-IDF) adalah suatu metode ekstraksi fitur yang digunakan untuk memberikan bobot pada kata-kata dalam teks. Bobot ini didasarkan pada frekuensi kemunculan kata dalam dokumen dan seberapa unik kata tersebut di seluruh kumpulan dokumen [3]. Pada penelitian ini, TF-IDF digunakan untuk ekstraksi fitur pada metode SVM dengan nilai max_features=5000, ngram_range=(1,2), min_df=1, max_df=0.4, sublinear_tf=True, smooth_idf=True.

2.5. Data Validasi

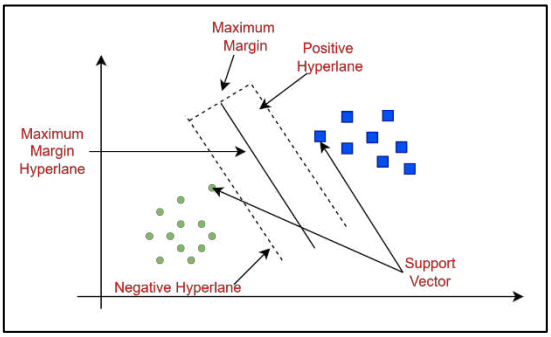
Pada tahapan pelatihan model, *Data training* kembali dibagi menjadi: 90% untuk Data *training* (5.040 hadis) 10% untuk Data validasi (560 hadis). Data validasi digunakan untuk mengevaluasi performa model sebelum diterapkan ke Data *Testing* [13], [17], [18].

2.6. *Training* Model

Pada tahapan ini, dilakukan dua model latih terhadap dua metode, *baseline* dan optimasi. Pada model *baseline*, kedua metode tidak dilakukan penyetelan parameter sedangkan untuk optimasi beberapa hal dilakukan seperti penyetelan parameter untuk SVM dan LSTM, melakukan *balancing hybridsampling* serta tambahan *preprocessing* menggunakan *stopword removal.*

2.7. *Support Vector Machine* (SVM)

SVM sebagai metode yang klasifikasi seperti yang dikutip dalam penelitian [19] adalah metode yang mencari garis batas terbaik untuk memisahkan data menjadi kelompok-kelompok yang berbeda. Garis batas ini dipilih sedemikian rupa sehingga jarak antara data dari kelompok yang berbeda sejauh mungkin. SVM awalnya didesain untuk masalah klasifikasi *linear*. Akan tetapi, melalui penerapan kernel, SVM mampu menangani masalah *non-linear* dengan memetakan data ke ruang fitur berdimensi tinggi. Dalam ruang ini, SVM mencari *hyperplane optimal* yang memaksimalkan margin antara kelas-kelas data [20]. Gambar 2 merepresentasikan arsitektur metode SVM.



Gambar 2. Arsitektur SVM [21]

Secara arsitektural seperti pada Gambar 2, proses kerja SVM dimulai dari data input yang telah direpresentasikan sebagai vektor fitur. SVM kemudian mengidentifikasi *hyperplane* optimal dalam ruang fitur tersebut menggunakan subset data yang disebut support vectors—yaitu titik-titik data yang berada paling dekat dengan *hyperplane* dan paling berpengaruh terhadap pembentukannya. Formula rumus dari SVM [22] dalam mencari hyperparameter optimal ditunjuukan pada persamaan 1.

$$W \cdot X + b = 0 \tag{1}$$

W merupakan sebuah vektor bobot yang terdiri dari elemen-elemen $\{W_1, W_2, ..., W_n\}$, di mana n adalah jumlah atribut, dan b adalah sebuah skalar yang disebut *bias*. Misalkan terdapat dua atribut A_1 dan A_2 dengan contoh data pelatihan X = (x1, x2), di mana x1 dan x2 adalah nilai dari atribut A1 dan A2. Jika bias b dianggap sebagai bobot tambahan w0, maka persamaan hyperplane pemisah dapat dinyatakan kembali pada persamaan 2.

$$w_0 + w_1 w_1 + w_2 w_2 = 0 \tag{2}$$

Setelah persamaan didefinisikan, nilai x1 dan x2 dapat disubstitusikan ke dalam persamaan tersebut untuk menentukan nilai bobot w1, w2, dan w0 (atau b). SVM bertujuan menemukan *hyperplane* pemisah maksimum, yaitu *hyperplane* yang memiliki jarak terjauh terhadap titik data pelatihan terdekat. Titik-titik terdekat ini disebut *support vector* dan ditunjukkan dengan garis batas yang tebal. Oleh karena itu, setiap titik yang berada di sisi atas *hyperplane* pemisah akan memenuhi persamaan 3

$$w_0 + w_1 w_1 + w_2 w_2 > 0 \quad (3)$$

Sedangkan titik yang terletak dibawah *hyperplane* pemisah memenuhi rumus persamaan 4.

$$w_0 + w_1 w_1 + w_2 w_2 < 0 \quad (4)$$

Dari persamaan diatas, maka terbentuk dua persamaan *hyperplane* yaitu persamaan 5 dan 6.

$$H_1 : w_o + w_1 w_1 + w_2 w_2 \geq 1 \quad (5)$$

$$\text{untuk } y_1 = +1$$

$$H_1 : w_o + w_1 w_1 + w_2 w_2 \leq -1 \quad (6)$$

$$\text{untuk } y_1 = -1$$

Pemodelan SVM diformulasikan dengan pendekatan matematis melalui *Lagrangian formulation*. Berdasarkan pendekatan ini, *Maximum Margin Hyperplane* (MMH) dapat direpresentasikan ulang sebagai batas keputusan (*decision boundary*) pada persamaan 7.

$$d(X^T) = \Sigma_i^1 y_i a_i X_i X^T + b_0 \quad (7)$$

Nilai yi merepresentasikan label kelas dari *support vector* Xi.XT dalah tupel uji (*test tuple*). Parameter ai dan b0 merupakan nilai numerik yang ditentukan secara otomatis melalui proses optimasi algoritma SVM dan 1 menyatakan jumlah *support vector* yang digunakan.

Pada penelitian ini proses pelatihan ini juga melibatkan pemilihan *kernel* berupa Radian Basis Function (RBF) pada *baseline* dan linear pada optimasi untuk menentukan bentuk transformasi data ke ruang fitur baru. Hasil akhir dari proses ini adalah model klasifikasi yang mampu memetakan input baru ke kelas yang sesuai berdasarkan sisi *hyperplane* tempat data tersebut berada. Dalam penelitian ini, *parameter tuning* pada SVM dilakukan menggunakan teknik *grid search*, dengan percobaan pada *kernel* = "linear", variasi nilai parameter regulasi C = [0.1, 1, 10, 100] dan *class_weight* ="balanced", guna menemukan kombinasi parameter yang menghasilkan performa klasifikasi terbaik.

2.8. Long Short Term Memory (LSTM)

LSTM menurut [13] merupakan salah satu metode dalam *deep learning* yang dikembangkan dari arsitektur *Recurrent Neural Network* (RNN). Perbedaan utamanya terletak pada keberadaan *memory cell* yang memungkinkan penyimpanan informasi dalam jangka waktu lebih panjang, sehingga LSTM mampu mengatasi permasalahan *long-term dependency* yang sering terjadi pada RNN dan menghasilkan kinerja yang lebih akurat [23]. LSTM mampu mengolah data secara berurutan dengan mempertahankan informasi dari langkah-langkah sebelumnya dalam urutan tersebut, sehingga memungkinkan model untuk melakukan prediksi pada langkah selanjutnya dengan lebih akurat. Kemampuan ini menjadikannya sangat efektif untuk menangani permasalahan yang berkaitan dengan ketergantungan jangka panjang dalam data [24].

Pada arsitektur LSTM, terdapat tiga gerbang utama yaitu gerbang *input*, gerbang *output*, dan gerbang lupa, serta sebuah sel memori. Sel memori ini menyimpan nilai pada setiap interval waktu, sementara ketiga gerbang tersebut mengatur aliran informasi yang masuk dan keluar dari sel. Pada setiap langkah waktu, LSTM menerima *input* dari waktu saat ini dan *output* dari waktu sebelumnya, lalu menghasilkan *output* yang akan diteruskan ke langkah waktu berikutnya. Lapisan tersembunyi pada langkah terakhir digunakan sebagai representasi untuk proses klasifikasi [25].

Pada penelitian ini LSTM dibagi menjadi dua tahapan, pada tahap *baseline*, model LSTM dibangun menggunakan *embedding layer* dengan dimensi vektor sebesar 100. Model ini terdiri dari lapisan *SpatialDropout*1D dengan *dropout rate* 0.2, diikuti oleh satu lapisan LSTM dengan 50 unit, *dropout* 0.2, dan *recurrent dropout* 0.2. Setelah itu, ditambahkan satu lapisan *dense* berjumlah 128 unit dengan aktivasi ReLU, dan dilanjutkan dengan lapisan *output* sebanyak dua unit dengan aktivasi *sigmoid* untuk kebutuhan klasifikasi *multilabel*. Model ini dikompilasi menggunakan *loss function binary_crossentropy* dan *optimizer Adam*, serta hanya dilatih selama 10 *epoch* dengan ukuran *batch* sebesar 32.

Pada model yang dioptimasi, arsitektur yang digunakan memiliki kompleksitas lebih tinggi. *Embedding layer* tetap digunakan dengan dimensi vektor yang sama, namun *dropout* pada *SpatialDropout*1D ditingkatkan menjadi 0.5 untuk mengurangi *overfitting* [26]. Jumlah unit pada lapisan LSTM ditingkatkan menjadi 128, dengan *dropout* 0.5 dan *recurrent dropout* 0.2. Lapisan *dense* tetap dipertahankan dengan 128 unit dan aktivasi ReLU. *Output layer* masih menggunakan dua unit dengan aktivasi *sigmoid*. Model ini dilatih menggunakan skema yang sama namun dengan jumlah *epoch* yang ditingkatkan menjadi 20 dan *batch size* sebesar 64. Proses

pelatihan juga dilengkapi dengan model *checkpoint* untuk menyimpan model terbaik berdasarkan nilai *validation accuracy*. Perbedaan utama antara *baseline* dan model yang telah dioptimasi terletak pada peningkatan jumlah unit LSTM, nilai dropout yang lebih besar, serta jumlah *epoch* dan *batch size* yang disesuaikan untuk mendukung proses pembelajaran yang lebih dalam dan stabil.

2.9. Evaluasi Model

Pada tahap ini, dilakukan evaluasi terhadap model klasifikasi yang telah dibangun oleh algoritma SVM dan LSTM. Data uji yang terdiri dari 1400 hadis (20% dari total data) akan digunakan untuk mengukur kemampuan model yang telah diperoleh selama proses pelatihan. Evaluasi hasil yang ditinjau dalam penelitian ini mencakup: (1) performa *baseline* model SVM dan LSTM yang diujikan pada *dataset* baru yang telah dilengkapi dengan F1-score, (2) perbandingan F1-score antara model SVM dan LSTM setelah dilakukan optimasi, serta (3) perbandingan nilai *hamming loss* dari model terbaik SVM dan LSTM terhadap nilai *hamming loss* pada penelitian sebelumnya. Evaluasi ini bertujuan untuk memperoleh kesimpulan mengenai peningkatan performa model klasifikasi setelah dilakukan pelengkapan konteks matan hadis.

3. HASIL DAN PEMBAHASAN

3.1. Metode Penelitian Terdahulu

Penelitian [9] menggunakan data hadis yang belum terlengkapi menggunakan metode *Backpropagation Neural Network* (BPNN) dengan *Information Gain* sebagai seleksi fitur dan TF-IDF untuk ekstraksi fitur. Evaluasi menggunakan 5-fold *cross validation* menghasilkan nilai *hamming loss* terbaik sebesar 0.1158 pada *threshold Information Gain* 0.75, yang setara dengan 88.42% akurasi klasifikasi. Penelitian ini merupakan pengembangan dari penelitian terdahulu dengan memberikan kontribusi utama berupa pelengkapan matan hadis pada *dataset* terjemahan Hadis Shahih Bukhari.

3.2. *Baseline*

Hasil *baseline* ditampilkan pada Tabel 4 sebagai berikut dengan menggunakan evaluasi performa F1-score.

Tabel 4. *Macro* F1-score hasil Baseline (dalam persen)

Label	Data validasi		Data *testing*	
	SVM	LSTM	SVM	LSTM
Anjuran	63,89	67,01	58,73	69,67
Informasi	49,09	57,83	49,55	56,53
Larangan	69,68	70,72	80,42	79,10

Berdasarkan Tabel 4 *Macro* F1-score hasil *baseline*, model LSTM menunjukkan performa terbaik pada keseluruhan label. Sementara itu pada data testing performa LSTM hanya meningkat pada label Anjuran dengan F1-score tertinggi sebesar 69,67% dan tetap unggul tipis di label informasi dengan F1-score 56,53%. Sedangkan SVM menunjukkan performa terbaik pada data testing khususnya pada label larangan dengan nilai F1-score tertinggi yaitu 80,42% sedangkan untuk performa label lainnya SVM masih dibawah nilai LSTM. Hal ini mengGambarkan baik SVM maupun LSTM masih memiliki potensi peningkatan performa melalui penerapan teknik optimasi seperti penggunaan stopword removal, *balancing*, dan penyetelan parameter.

3.3. Optimasi SVM dan LSTM

Setelah melakukan proses *baseline* dilakukan proses optimasi dengan menerapkan penyetelan parameter pada kedua metode guna mencapai nilai model yang terbaik dari masing-masing metode. Berikut Tabel hasil optimasi dari metode SVM.

Tabel 5. *Macro* F1-score hasil optimasi metode SVM (dalam persen)

No	Optimasi		Label					
	STP	*Balancing*	Anjuran		Informasi		Larangan	
			Validasi	*Testing*	Validasi	*Testing*	Validasi	*Testing*
1	No	No	72,23	69,41	65,56	66,51	78,51	82,57
2	Yes	No	71,53	67,21	63,00	67,40	70,29	73,32
3	No	Yes	67,72	66,41	59,75	63,52	70,71	78,46
4	Yes	Yes	67,48	65,10	65,37	62,33	67,79	65,81

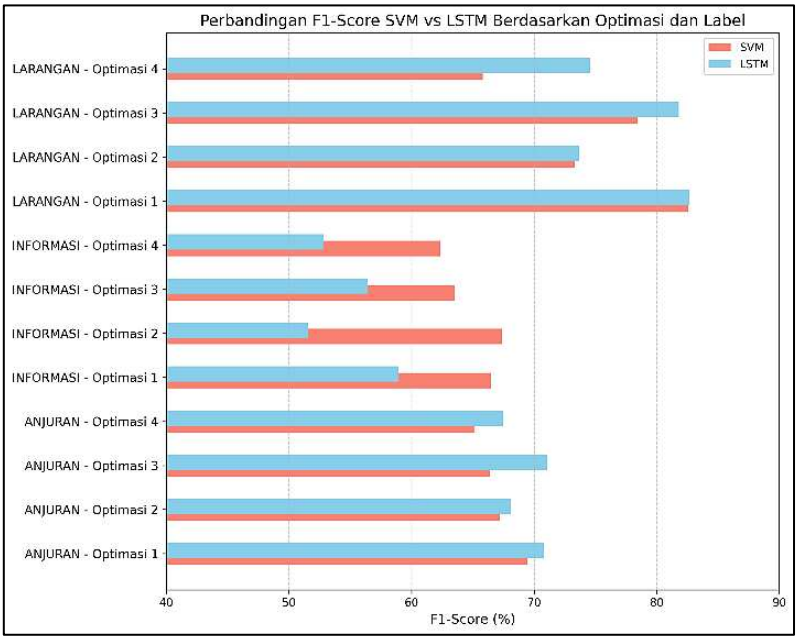
Tabel 5 Macro F1-score hasil optimasi metode SVM, menyajikan hasil klasifikasi setelah dilakukan penyetelan parameter untuk setiap konfigurasinya. Hasil optimasi terbaik metode SVM diperoleh tanpa menerapkan *stopword removal* (STP) dan *balancing*, dengan F1-score tertinggi baik pada data validasi maupun *testing*, khususnya pada label Larangan (78,51% dan 82,57%). Teknik optimasi ini berlaku juga pada hasil

validasi dan *testing* label Anjuran (72,23% dan 69,41%). Sementara itu, pada label Informasi teknik optimasi tersebut hanya berlaku pada hasil validasi (65,56%) sedangkan hasil *testing* tertinggi didapatkan dengan menerapkan STP dan tanpa *balancing* (67,40%). Penerapan STP tanpa *balancing* cenderung menurunkan performa validasi, meskipun sedikit meningkatkan hasil *testing* pada label Informasi. Penggunaan *balancing*, baik sendiri maupun dikombinasikan dengan STP, justru menurunkan performa validasi dan *testing* di hampir semua label. Temuan ini menunjukkan bahwa optimasi seperti STP dan *balancing* tidak selalu memberikan dampak positif terhadap performa model, terutama pada data tidak seimbang dan teks yang bergantung pada konteks kata tertentu. Selanjutnya, dilakukan proses optimasi terhadap metode LSTM hasil yang didapat dari melakukan berbagai optimasi ditampilkan pada Tabel 6.

Tabel 6. *Macro* F1-score hasil optimasi metode LSTM (dalam persen)

No	Optimasi		Label					
	STP	Balancing	Anjuran		Informasi		Larangan	
			Validasi	Testing	Validasi	Testing	Validasi	Testing
1	No	No	70,36	70,82	64,56	58,94	77,04	82,66
2	Yes	No	64,95	68,09	58,27	51,54	61,17	73,69
3	No	Yes	70,44	71,04	65,22	56,45	76,94	81,80
4	Yes	Yes	66,60	67,52	57,83	52,81	66,82	74,57

Tabel 6 Macro F1-score *h*asil optimasi metode LSTM, menunjukkan bahwa konfigurasi terbaik metode LSTM setelah dilakukan penyetelan parameter diperoleh tanpa penerapan STP dan tanpa *balancing*, ditandai dengan F1-score tertinggi pada label Larangan dengan nilai data *testing* (82,66%) dan validasi (77,04%). Metode optimasi tersebut berlaku untuk label Informasi khususnya pada data *testing* dengan nilai (58,94%). Pada label Anjuran, dengan melakukan *balancing* dan tanpa STP menghasilkan nilai tertinggi untuk data validasi dan *testing* (70,44% dan 71,04%). Hal ini juga berlaku pada data validasi label Informasi (65,22%). Jika dibandingkan, performa tanpa *balancing* menghasilkan skor tertinggi pada label Larangan namun lebih rendah pada label lain, khususnya Informasi. Sedangkan dengan *balancing*, hasil meningkat pada label Anjuran baik pada *testing* dan validasi serta data validasi pada label Informasi namun menurun pada performa label Larangan. Penerapan STP justru menurunkan performa pada seluruh label, baik pada data validasi maupun *testing*. Hal ini menunjukkan bahwa LSTM sensitif terhadap penghilangan kata umum (stopword) yang mungkin berperan dalam membentuk konteks urutan kata. Dari Tabel 5 dan 6, didapatkan grafik hasil F1-score sebagai penunjang visualisasi untuk menganalisa hasil performa kedua algoritma yang ditunjukkan pada Gambar 3.



Gambar 3. Grafik F1-score data *testing* SVM dan LSTM

Berdasarkan Gambar 3, terlihat bahwa metode SVM dan LSTM menunjukkan pola performa yang berbeda terhadap masing-masing label pada data *testing* terhadap tiga label klasifikasi Anjuran, Informasi dan Larangan pada empat skenario optimasi. Untuk label Anjuran, model LSTM secara konsisten menunjukkan performa lebih baik dibandingkan SVM di seluruh tahapan optimasi. F1-score tertinggi untuk label ini dicapai

pada optimasi ke-3 LSTM dengan konfigurasi *balancing* dan tanpa STP (71.04%). Hal ini mengindikasikan bahwa LSTM mampu menangkap pola sekuensial dalam teks anjuran dengan lebih efektif dibandingkan SVM setelah dilakukan balancing tanpa menghilangkan kata umum (STP).

Pada label Informasi, model SVM menunjukkan kinerja yang unggul secara signifikan dibandingkan LSTM di keempat skenario optimasi. Skor tertinggi untuk label ini tercatat pada optimasi ke-2 dengan konfigurasi penerapan STP tanpa *balancing* (67.40%). Hal ini menunjukkan bahwa penerapan STP tidak memberikan pengaruh yang signifikan terhadap struktur kalimat pada label Informasi, sehingga tidak secara substansial memengaruhi performa model klasifikasi.

Sementara itu, pada label Larangan, LSTM kembali menunjukkan keunggulan performa dengan skor tertinggi dicapai pada optimasi ke-1 tanpa menerapkan konfigurasi STP maupun *balancing* (82.66%), yang juga mengungguli performa SVM untuk label ini. Hal ini menegaskan kekuatan LSTM dalam memproses struktur kalimat yang mungkin lebih kompleks dan bergantung pada konteks sekuensial. Selain itu, hal ini juga membuktikan bahwa label Larangan sensitif terhadap penghilangan kata umum (STP) karena dapat menghilangkan informasi penting saat proses klasifikasi.

Secara keseluruhan, model LSTM cenderung lebih unggul dalam menangani label-label yang membutuhkan pemahaman konteks urutan kata, seperti Anjuran dan Larangan, sedangkan SVM lebih efektif pada label dengan distribusi fitur yang lebih sederhana seperti Informasi. Di antara keempat skenario optimasi yang dilakukan, konfigurasi tanpa penerapan STP menghasilkan performa yang lebih baik. Hal ini menunjukkan bahwa STP dapat mengubah struktur kalimat dan berpotensi menghilangkan informasi penting, terutama dalam konteks teks keagamaan seperti hadis, yang memiliki makna kontekstual kuat pada setiap kata.

Hasil menunjukkan bahwa LSTM unggul pada sebagian besar label karena kemampuannya menangkap informasi dalam teks hadis yang kompleks dan panjang. LSTM menunjukkan performa klasifikasi yang lebih baik berdasarkan nilai F1-score pada klasifikasi *multilabel* data hadis dibandingkan SVM, sedangkan SVM lebih efisien dari segi waktu pelatihan, dengan durasi tercepat sekitar 60 detik, jauh lebih cepat dibandingkan LSTM yang membutuhkan waktu pelatihan hingga 1.560 detik dengan *epoch* 20 pada konfigurasi terbaik.

3.4. Fitur yang berpengaruh untuk klasifkasi dengan SVM

Dalam dataset, teks-teks dengan muatan larangan memiliki ciri linguistik yang lebih eksplisit dan mudah dikenali oleh model. Kata-kata seperti *"janganlah"*, *"jangan"*, dan *"melarang"* diprediksi memiliki kontribusi paling besar dalam membentuk prediksi klasifikasi untuk label “Larangan”. Sementara itu, label Anjuran juga menunjukkan satu fitur yang cukup menonjol, dengan kata-kata seperti *"hendaklah"* sebagai kata kunci yang relevan. Namun kata-kata yang tidak spesifik muncul untuk kelas “Informasi”. Label Informasi memiliki kontribusi fitur yang relatif lebih rendah, dengan kata seperti *"mereka"*, *"memberi"*, dan *"di atas"* yang kurang eksplisit dalam konteks kategori, sehingga berdampak pada performa klasifikasi yang lebih lemah pada label ini.

Tabel 7. Tiga fitur paling dominan untuk setiap label berdasarkan model SVM

No	Label	Fitur (kata)	Koefisien
1	Larangan	Janganlah	7,78019
		Jangan	5,28189
		Melarang	5,07315
2	Anjuran	Hendaklah	4,24944
		Kalian	3,50222
		Memerintahkan	3,34316
3	Informasi	Mereka	1,70123
		Memberi	1,57045
		Di Atas	1,51816

Dari hasil penghitungan bobot koefisien pada fitur SVM, nilai koefisien tertinggi diperoleh kata-kata yang paling berpengaruh untuk masing-masing kelas. Tabel 7 menyajikan tiga fitur paling berpengaruh dari masing-masing label berdasarkan model SVM. Kata-kata eksplisit seperti "janganlah", "hendaklah", dan "lakukanlah" terbukti memberikan kontribusi besar dalam proses klasifikasi, sedangkan label Informasi cenderung lebih sulit dikenali karena minimnya fitur linguistik yang eksplisit.

3.5. Evaluasi *Hamming Loss*

Setelah optimasi pada kedua metode dilakukan, maka evaluasi terakhir perbandingan nilai *hamming loss* antara penelitian ini dan penelitian [9]. Nilai *hamming loss* terbaik diambil dari masing-masing metode untuk perbandingan dapat dilihat pada Tabel 8.

Tabel 8. Nilai *hamming loss*

Metode	Dataset yang digunakan	*Hamming loss*
BPNN + *Information Gain* [9]	Dataset[9] matan dieliminasi	11,580%
SVM + TF-IDF (optimized)	Dataset [9] matan dilengkapi	11,167%
LSTM (optimized)	Dataset [9] matan dilengkapi	10,452%

Berdasarkan Tabel 7 nilai *hamming loss* pada data yang telah dilengkapi dari sumber matan hadis yang lengkap [16]. Metode LSTM yang telah dilakukan optimasi dengan penyetalan parameter dan konfigurasi tanpa *stopword removal* serta *balancing* menghasilkan nilai *hamming loss* terendah sebesar 10,452%, yang menunjukkan bahwa LSTM memiliki kemampuan lebih baik dalam meminimalkan kesalahan prediksi pada tugas klasifikasi *multilabel* secara keseluruhan. Meskipun pada beberapa label tertentu F1-score LSTM lebih rendah dibandingkan SVM, dari sisi akurasi per label per *instance*.

Sebagai perbandingan, metode SVM dengan ekstraksi fitur TF-IDF yang telah dioptimasi dengan penyetelan parameter dan konfigurasi tanpa n*stopword removal* serta tanpa *balancing* mencatat nilai *hamming loss* sebesar 11,167%, yang masih lebih baik dibandingkan metode BPNN dengan *Information Gain* dari penelitian sebelumnya dengan menggunakan evaluasi 5-fold *cross validation* menghasilkan nilai *hamming loss* terbaik sebesar 11,580% pada *threshold Information Gain* 0.75 [9]. Ini menandakan bahwa pendekatan SVM dalam penelitian ini memiliki peningkatan performa meskipun tanpa penerapan metode *balancing* dan *stopword removal*. Secara khusus, kontribusi penting dari penelitian ini terletak pada penyempurnaan kualitas data melalui pelengkapan matan hadis, yang terbukti mampu meningkatkan performa model, baik pada algoritma LSTM maupun SVM.

Penelitian ini menemukan bahawa model LSTM terbukti lebih efektif dalam mengenali label yang benar dari rendahnya nilai *hamming loss*. Hal ini konsisten dengan temuan dalam penelitian [13] dan [14], yang juga menunjukkan bahwa LSTM unggul dalam mempelajari hubungan dalam data sekuensial. Penelitian ini mengisi kesenjangan pengetahuan dari studi sebelumnya yang belum memanfaatkan pendekatan *deep learning* dan *machine learning* untuk klasifikasi *multilabel* pada data teks keagamaan serta melanjutkan pembahasan penelitian sebelumnya dengan memberikan kontribusi berupa melengkapi dataset yang digunakan. Dengan membandingkan dua pendekatan berbeda yaitu metode konvensional berbasis pembobotan fitur (SVM + TF-IDF) dan metode berbasis representasi sekuensial (LSTM) terhadap data yang sudah dilengkapi matan hadisnya dari penelitian sebelumnya [9], memberikan wawasan empiris terkait efektivitas masing-masing model terhadap data *multilabel* Hadis Shahih al-Bukhari.

4. KESIMPULAN

Berdasarkan hasil penelitian, evaluasi menggunakan metrik *Macro* F1-score menunjukkan bahwa metode LSTM mengungguli SVM pada label Larangan dan Anjuran dengan nilai F1-score masing-masing sebesar 82,66% dan 71,04%. Sebaliknya, SVM menunjukkan performa lebih baik pada label Informasi dengan F1-score sebesar 67,40%. Penyetelan parameter yang tepat terbukti dapat meningkatkan performa metode dalam melakukan klasifikasi. Namun, penerapan *stopword removal* pada penelitian ini tidak memberikan hasil yang optimal, baik pada algoritma SVM maupun LSTM. Sementara itu, penerapan teknik *hybridsampling* pada klasifikasi data hadis dinilai kurang tepat, karena metode ini mengurangi data mayoritas melalui *undersampling*, sehingga berisiko menghilangkan informasi penting yang terkandung di dalamnya. Walaupun teknik ini menunjukkan peningkatan performa pada label tertentu, namun tidak memberikan dampak positif yang konsisten terhadap keseluruhan label klasifikasi. Dari sisi evaluasi menggunakan *hamming loss*, metode LSTM dengan penyetelan parameter dan tanpa penerapan *stopword removal* dan tanpa *balancing* menunjukkan performa terbaik dengan nilai terendah sebesar 10,452%, yang menunjukkan tingkat kesalahan prediksi label yang lebih rendah dibandingkan metode SVM maupun BPNN. Selain itu, pelengkapan matan hadis juga terbukti berkontribusi terhadap peningkatan nilai klasifikasi. Untuk penelitian selanjutnya, disarankan mengeksplorasi lebih lanjut teknik balancing lain yang mampu menyesuaikan dengan karakteristik data, khususnya data seperti hadis, guna meningkatkan performa model secara keseluruhan.

REFERENSI

[1] "Agama di Indonesia, 2024," Badan Pusat Statistik Kota Samarinda. Accessed: May 15, 2025. [Online]. Available: https://samarindakota.bps.go.id/id/statistics-table/1/MzI0IzE=/agama-di-indonesia-2024.html

[2] M. Y. A. Bakar and Adiwijaya, "Klasifikasi Teks Hadis Bukhari Terjemahan Indonesia Menggunakan Recurrent Convolutional Neural Network (CRNN)," *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK)*, vol. 8, no. 5, pp. 907–918, 2021, doi: 10.25126/jtiik.202183750.

[3] A. Hanafi, A. Adiwijaya, and 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, pp. 357–364, Sep. 2020, doi: 10.32736/sisfokom.v9i3.980.

[4] J. Bogatinovski, L. Todorovski, S. Džeroski, and D. Kocev, "Comprehensive comparative study of multi-label classification methods," *Expert Syst Appl*, vol. 203, Oct. 2022, doi: 10.1016/j.eswa.2022.117215.

[5] N. Endut, W. M. A. F. W. Hamzah, I. Ismail, M. K. Yusof, Y. A. Baker, and H. Yusoff, "A Systematic Literature Review on Multi-Label Classification based on Machine Learning Algorithms," *TEM Journal*, vol. 11, no. 2, pp. 658–666, May 2022, doi: 10.18421/TEM112-20.

[6] P. Pangestu, R. Novita, and Mustakim, "Systematic Literature Review: Perbandingan Algoritma Klasifikasi," *Jurnal Inovtek Polbeng*, vol. 8, no. 2, pp. 431–440, 2023, doi: http://dx.doi.org/10.35314/isi.v8i2.3698.

[7] S. P. Afrisia, F. M. Hana, and W. C. Wahyudin, "Implementasi Metode Long Short Term Memory (LSTM) pada Chatbot Kesehatan Mental Mahasiswa," *Sainteks*, vol. 21, no. 2, pp. 107–116, Oct. 2024, doi: 10.30595/sainteks.v21i2.23869.

[8] I. Akbar, M. Faisal, and T. Chamidy, "Penerapan Long Short-Term Memory untuk Klasifikasi Multi-Label Terjemahan Al-Qur'an dalam Bahasa Indonesia," *JOINTECS(Journal of Information Technology and Computer Science)*, vol. 8, no. 1, pp. 41–54, 2024, doi: https://doi.org/10.31328/jointecs.v8i1.5291.

[9] M. Y. A. Bakar, Adiwijaya, and A. F. Faraby, "Multi-Label Topic Classification of Hadith of Bukhari (Indonesian Languagetranslation) using Information Gain and Backpropagation Neural Network," in *2018 International Conference on Asian Language Processing : 15-17 November 2018, Bandung, Indonesia*, Institute of Electrical and Electronics Engineers, 2018, pp. 344–350. doi: 10.1109/IALP.2018.8629263.

[10] R. Kustiawan, A. Adiwijaya, and M. D. Purbolaksono, "A Multi-label Classification on Topic of Hadith Verses in Indonesian Translation using CART and Bagging," *Jurnal Media Informatika Budidarma*, vol. 6, no. 2, p. 868, Apr. 2022, doi: 10.30865/mib.v6i2.3787.

[11] F. Taufiqurrahman, S. Al Faraby, and M. D. Purbolaksono, "Klasifikasi Teks Multi Label pada Hadis Terjemahan Bahasa Indonesia Menggunakan Chi-Square dan SVM," *e-Proceeding of Engineering*, vol. 8, no. 5, pp. 1–10, 2021.

[12] A. R. Muslikh, I. Akbar, D. R. I. M. Setiadi, and H. M. M. Islam, "Multi-label Classification of Indonesian Al-Quran Translation based CNN, BiLSTM, and FastText," *Techno.COM*, vol. 23, no. 1, pp. 37–50, 2024, doi: https://doi.org/10.62411/tc.v23i1.9925.

[13] N. Fatiara, N. H. Safaat, S. Agustian, Yusra, and I. Afrianty, "Komparasi Metode K-Nearest Neighbors Dan Long Short Term Memory," *ZONAsi: Jurnal Sistem Informasi*, vol. 6, no. 2, pp. 332–345, 2024, doi: https://doi.org/10.31849/zn.v6i2.19863.

[14] W. K. Sari, D. P. Rini, and R. F. Malik, "Text Classification Using Long Short-Term Memory With GloVe Features," *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika*, vol. 5, no. 2, p. 85, Feb. 2020, doi: 10.26555/jiteki.v5i2.15021.

[15] M. F. Naufal and S. F. Kusuma, "Analisis Perbandingan Algoritma Machine Learning Dan Deep Learning Untuk Klasifikasi Citra Sistem Isyarat Bahasa Indonesia (SIBI)," *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK)*, vol. 10, no. 4, pp. 873–882, 2023, doi: 10.25126/jtiik.2023106828.

[16] R. Saputra, N. S. Harahap, Novriyanto, and M. Affandes, "Penerapan Merger Retriever Pada Question Answering System Hadits Tugas Akhir," *SATIN - Sains dan Teknologi Informasi*, vol. 10, no. 1, pp. 24–35, 2024, doi: 10.33372/stn.v9i2.1000.

[17] S. Ningsih, N. H. Safaat, S. Agustian, Yusra, and E. P. Cynthia, "Pengaruh Penyeimbangan Data Pada Klasifikasi Terjemahan Al-Quran Dengan Metode Naïve Bayes dan Long Short Term Memory," *Journal of Computer System and Informatics (JoSYC)* , vol. 5, no. 3, pp. 626–635, 2024, doi: 10.47065/josyc.v5i3.5181.

[18] D. P. Aftari, N. H. Safaat, S. Agustian, Yusra, and I. Afrianty, "Perbandingan Performa Klasifikasi Terjemahan Al-Qur'an Menggunakan Metode Random Forest dan Long Short Term Memory," *Journal of Computer System and Informatics (JoSYC)*, vol. 5, no. 3, pp. 567–577, 2024, doi: 10.47065/josyc.v5i3.5156.

[19] P. A. Octaviani, Y. Wilandari, and D. Ispriyanti, "Penerapan Metode Klasifikasi Support Vector Machine (SVM) Pada Data Akreditasi Sekolah Dasar (SD) Di Kabupaten Magelang," *Jurnal Gaussian*, vol. 3, no. 4, pp. 811–820, 2014, doi: https://doi.org/10.14710/j.gauss.3.4.811-820.

[20] N. H. Ovirianti, M. Zarlis, and H. Mawengkang, "Support Vector Machine Using A Classification Algorithm," *Sinkron*, vol. 6, no. 3, pp. 2103–2107, 2022, doi: 10.33395/sinkron.v7i3.

[21] N. Khilari, P. Hadawale, H. Shaikh, and S. Kolase, "Analysis Of Machine Learning Algorithm To Predict Wine Quality," *International Journal Of Creative Reseacrh Thoughts (IJCRT)*, vol. 10, no. 4, pp. 2320–2882, 2022, doi: https://doi.org/10.32628/IJSRSET229235.

[22] M. P. D. Cahyo, Widodo, and B. P. Adhi, "Kinerja Algoritma Support Vector Machine dalam MenentukanKebenaran Informasi Banjir di Twitter," *PINTER : Jurnal Pendidikan Teknik Informatika dan Komputer*, vol. 3, no. 2, pp. 116–121, Dec. 2019, doi: 10.21009/pinter.3.2.5.

[23] S. S. Nurashila, F. Hamami, and T. F. Kusumasari, "Perbandingan Kinerja Algoritma Recurrent Neural Network (RNN) Dan Long Short-Term Memory (LSTM): Studi Kasus prediksi Kemacetan Lalu Lintas Jaringan PT XYZ," *JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, vol. 8, no. 3, pp. 864–877, Aug. 2023, doi: 10.29100/jipi.v8i3.3961.

[24] S. M. Al-Selwi *et al.*, "RNN-LSTM: From applications to modeling techniques and beyond—Systematic review," *Journal of King Saud University - Computer and Information Sciences*, vol. 36, no. 5, pp. 1–34, Jun. 2024, doi: 10.1016/J.JKSUCI.2024.102068.

[25] E. A. Az Zahra, Y. Sibaroni, and S. S. Prasetyowati, "Classification of Multi-Label of Hate Speech on Twitter Indonesia using LSTM and BiLSTM Method," *JINAV: Journal of Information and Visualization*, vol. 4, no. 2, pp. 170–178, Jul. 2023, doi: 10.35877/454RI.jinav1864.

[26] P. Alkhairi, A. P. Windarto, and M. M. Efendi, "Optimasi LSTM Mengurangi Overfitting untuk Klasifikasi Teks Menggunakan Kumpulan Data Ulasan Film Kaggle IMDB," *Building of Informatics, Technology and Science (BITS)*, vol. 6, no. 2, pp. 1142–1150, Sep. 2024, doi: 10.47065/bits.v6i2.5850.