

Terakreditasi SINTA Peringkat 3

Surat Keputusan Direktur Jenderal Pendidikan Tinggi, Riset, dan Teknologi Nomor 225/E/KPT/2022 masa berlaku mulai Vol.7 No. 1 tahun 2022 s.d Vol. 11 No. 2 tahun 2026

Terbit online pada laman web jurnal:
<http://publishing-widyagama.ac.id/ejournal-v2/index.php/jointecs>



Vol. 8 No. 1 (2024) 41 - 54

JOINTECS

(Journal of Information Technology and Computer Science)

e-ISSN:2541-6448

p-ISSN:2541-3619

Penerapan *Long Short-Term Memory* untuk Klasifikasi Multi-Label Terjemahan Al-Qur'an dalam Bahasa Indonesia

Ismail Akbar¹, Muhammad Faisal², Totok Chamidy³

Magister Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang

¹ismaelakbar12@gmail.com, ²mfaisal@ti.uin-malang.ac.id, ³to2k2013@ti.uin-malang.ac.id

Abstract

This research develops a multi-label classification model for translated texts of the Al-Qur'an in Indonesian using Bi-Directional Long Short-Term Memory (Bi-LSTM) and the Word2Vec embedding technique with the Continuous Bag of Words (CBOW) architecture. The data used is the official translation from the Indonesian Ministry of Religion, specifically from Surah An-Nisa', Surah Al-Maidah, and Surah Al-An'am, with a total of 461 verses. The classification of verses considers four main categories: Tauhid, Worship, Morals, and History (Tarikh). This model was tested with various data sharing scenarios and evaluated using accuracy, precision, recall and Hamming Loss metrics. The results show that the Bi-LSTM model with Word2Vec achieves 70.21% accuracy, 64.31% precision, 61.13% recall, and 36.52% Hamming Loss. Although showing significant improvements, this model still faces challenges in achieving higher accuracy due to the complexity of the language of the Qur'an. This research suggests the use of more representative data, other embedding techniques, and more sophisticated model architectures for further improvements. This model is expected to facilitate learning and understanding the Koran more accurately and efficiently.

Keywords: al-qur'an; verse classification; lstm; word embedding; translation.

Abstrak

Mempelajari Al-Qur'an adalah salah satu ibadah yang sangat dianjurkan dalam Islam, dan memahami cara mengklasifikasikan ayat-ayatnya sangat penting. Pengelompokan ayat-ayat ini mempermudah pencarian dan mempelajari ayat-ayat terkait, karena setiap ayat dapat termasuk dalam satu atau lebih kategori yang berbeda. Penelitian ini bertujuan untuk mengembangkan model klasifikasi *multi-label* untuk teks terjemahan Al-Qur'an dalam bahasa Indonesia menggunakan *Bi-Directional Long Short-Term Memory* (Bi-LSTM) dan teknik *word embedding Word2Vec* dengan arsitektur *Continuous Bag of Words* (CBOW). Data yang digunakan berasal dari terjemahan resmi Kementerian Agama Republik Indonesia, khususnya dari Surah An-Nisa', Surah Al-Maidah, dan Surah Al-An'am, dengan total 461 ayat. Proses klasifikasi mempertimbangkan empat kategori utama: Tauhid, Ibadah, Akhlaq, dan Sejarah (Tarikh). Model diuji dengan berbagai skenario pembagian data dan dievaluasi menggunakan metrik akurasi, *precision*, *recall*, dan *Hamming Loss*. Hasil menunjukkan bahwa model Bi-LSTM dengan *Word2Vec* mencapai akurasi 70,21%, *precision* 64,31%, *recall* 61,13%, dan *Hamming Loss* 36,52%. Meskipun menunjukkan peningkatan signifikan, model ini masih menghadapi tantangan dalam mencapai akurasi yang lebih tinggi karena kompleksitas bahasa Al-Qur'an. Penelitian ini menyarankan penggunaan data yang lebih representatif, teknik embedding lain, dan arsitektur model yang lebih canggih untuk perbaikan lebih lanjut. Model ini diharapkan dapat mempermudah pembelajaran dan pemahaman Al-Qur'an secara lebih akurat dan efisien.

Kata kunci: al-qur'an; klasifikasi ayat; lstm; word embedding; terjemahan.



Diterima Redaksi : 21-12-2024 | Selesai Revisi : 13-06-2024 | Diterbitkan Online : 30-06-2024

1. Pendahuluan

Pemahaman dan pembelajaran Al-Qur'an merupakan aspek yang sangat penting bagi umat Islam dan umat manusia secara umum. Al-Qur'an bukan hanya sekedar kitab suci, tetapi juga panduan hidup yang berisi ajaran-ajaran esensial yang harus dipahami dan diaplikasikan dalam kehidupan sehari-hari. Oleh karena itu, mempelajari dan memahami isi Al-Qur'an merupakan salah satu ibadah yang sangat dianjurkan dalam Islam [1]. Dengan memahami ajaran-ajaran yang terkandung dalam Al-Qur'an, setiap individu dapat lebih mendalami prinsip-prinsip Islam secara komprehensif, yang pada akhirnya membantu mereka untuk mematuhi ajaran-ajaran tersebut dengan lebih akurat dan tegas. Namun, memahami dan mengklasifikasikan ayat-ayat Al-Qur'an menjadi sebuah tantangan tersendiri, yang memerlukan pendekatan yang tepat agar pembelajaran dapat dilakukan dengan lebih mudah dan efisien [2]. Dengan mengklasifikasikan ayat-ayat dalam Al-Qur'an, seorang yang mendalami makna Al-Quran dapat lebih mudah menemukan ayat yang diinginkan. Seperti halnya pencarian makna dalam penerjemahan Al-Qur'an [3].

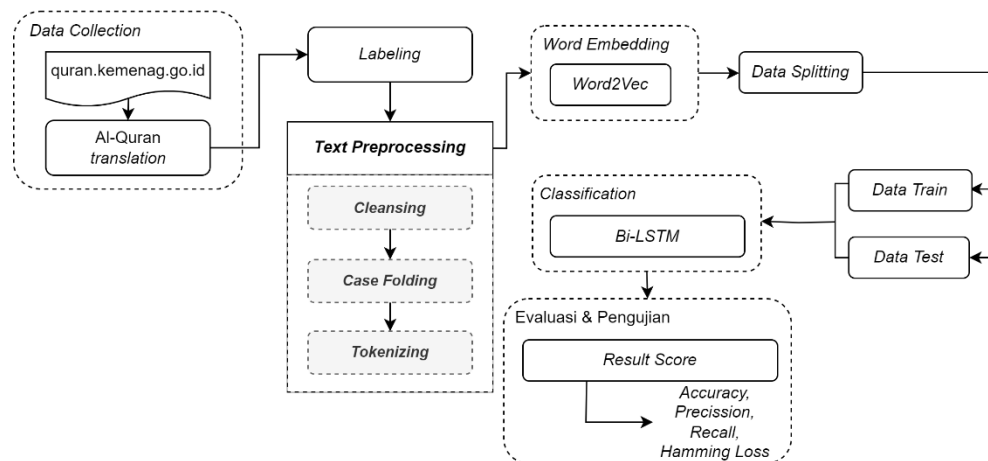
Secara tradisional, ayat-ayat Al-Qur'an diklasifikasikan ke dalam beberapa kategori seperti Tauhid, Ibadah, Muamalah, etika, Tarikh (Sejarah), dan Syariah [4]. Klasifikasi ini mempermudah dalam memahami konteks dan isi dari setiap ayat. Pengklasifikasian yang akurat sangat penting karena satu ayat sering kali dapat masuk ke dalam lebih dari satu kategori, membuat pendekatan *multi-label* menjadi lebih relevan [5]. Dengan pendekatan ini, proses pencarian dan pembelajaran makna ayat-ayat dalam Al-Qur'an dapat menjadi lebih efektif. Misalnya, dalam penerjemahan Al-Qur'an, sering kali ditemukan kata-kata yang memiliki arti yang sama, yang menunjukkan bahwa setiap ayat dapat memiliki banyak konteks yang perlu dipertimbangkan secara simultan.

Berbagai metode telah dikembangkan untuk menangani masalah klasifikasi teks, termasuk klasifikasi ayat-ayat Al-Qur'an. Salah satu metode awal yang digunakan adalah *K-Nearest Neighbor* (K-NN). Penelitian oleh M. Fauzan dan kolega mengaplikasikan K-NN untuk mengklasifikasikan hadis Bukhari terjemahan bahasa Indonesia, namun metode ini menghadapi kesulitan dalam menangani data berdimensi tinggi, yang mengakibatkan waktu komputasi yang lama dan efisiensi klasifikasi yang rendah [6]. Putrisanni dan timnya juga mengevaluasi K-NN untuk klasifikasi Al-Qur'an terjemahan bahasa Inggris, tetapi hasil yang diperoleh masih kurang memuaskan dengan akurasi sekitar 64,10% [7]. Sebagai alternatif, metode lain seperti *Support Vector Machine* (SVM) dan *Naive Bayes* (NB) juga telah digunakan, di mana SVM berhasil mencapai akurasi lebih tinggi hingga 70% dibandingkan metode lainnya [8].

Pendekatan *deep learning* mulai mendapatkan perhatian lebih dalam klasifikasi teks, khususnya LSTM yang merupakan salah satu jenis *Recurrent Neural Network* (RNN) yang efektif untuk data urutan. Kholifatullah dan timnya menggunakan LSTM untuk klasifikasi ujaran kebencian dan mencapai hasil yang optimal dengan akurasi sebesar 86,23% [9]. Selain itu, *Bi-directional Long Short Term Memory* (Bi-LSTM) yang dipadukan dengan *Word2Vec* CBOV berhasil mencapai akurasi sangat tinggi hingga 96,86% dalam penelitian klasifikasi teks [10]. LSTM memiliki keunggulan dalam mengingat informasi jangka panjang dan pendek secara simultan, serta kemampuan untuk memproses data urutan dengan lebih efisien tanpa kehilangan informasi penting [11]. Hal ini membuat LSTM sangat cocok untuk tugas-tugas yang memerlukan pemahaman konteks penuh dari kalimat atau dokumen, yang seringkali diperlukan dalam klasifikasi teks terjemahan Al-Qur'an [12].

Sehingga penelitian ini menggunakan algoritma LSTM sebagai model klasifikasi terjemahan Al-Qur'an bahasa Indonesia, dikarenakan LSTM dirancang untuk mempertahankan informasi jangka panjang dan pendek, sehingga sangat efektif dalam memproses data teks yang membutuhkan pemahaman konteks penuh [13]. Kemampuan LSTM untuk menangani urutan panjang tanpa mengalami masalah yang biasanya dihadapi oleh RNN standar, seperti hilangnya informasi penting, membuatnya ideal untuk tugas klasifikasi teks [14]. Selain itu, LSTM juga unggul dalam menangani tugas klasifikasi *multi-label*, di mana satu teks atau ayat dapat termasuk dalam beberapa kategori berbeda [15]. Dengan memanfaatkan keunggulan ini, diharapkan penelitian ini dapat mengklasifikasikan terjemahan Al-Qur'an bahasa Indonesia ke dalam empat kelompok utama yaitu Tauhid, Ibadah, Akhlaq, dan Sejarah (Tarikh) dengan akurasi yang tinggi.

Berdasarkan penjelasan di atas, penelitian ini bertujuan untuk mengklasifikasikan terjemahan Al-Qur'an dalam bahasa Indonesia menggunakan model Bi-LSTM (*Bi-Directional Long Short-Term Memory*) dengan teknik *word embedding* *Word2Vec* yang menggunakan arsitektur CBOV. Pendekatan ini memungkinkan setiap ayat Al-Qur'an untuk diklasifikasikan ke dalam beberapa label secara simultan, sehingga memudahkan pencarian dan pemahaman ayat-ayat yang relevan dengan topik yang dipelajari. Proses klasifikasi ini meningkatkan efisiensi dalam pembelajaran Al-Qur'an dan membantu menemukan konteks yang lebih mendalam dan akurat. Keunggulan LSTM dalam menangani data urutan dan klasifikasi multi-label membuatnya sangat cocok untuk tugas ini, memastikan bahwa hasil klasifikasi lebih tepat dan bermanfaat bagi umat Muslim. Dengan demikian, Penelitian ini diharapkan dapat mempermudah pembelajaran dan pemahaman Al-Qur'an dengan cara yang lebih terstruktur dan akurat.



Gambar 1. Tahapan Penelitian.

2. Metode Penelitian

Klasifikasi *multi-label* berbeda dengan klasifikasi label tunggal. Klasifikasi label tunggal mengklasifikasikan teks ke dalam kategori label tunggal. Sedangkan klasifikasi *multi-label* dapat mengelompokkan teks ke dalam beberapa kategori atau label [16]. Penelitian ini menggunakan model LSTM secara dua arah atau biasa disebut dengan Bi-LSTM (*Bi-Directional Long Short-Term Memory*) untuk mengklasifikasikan beberapa label terjemahan Al-Qur'an bahasa Indonesia, seperti terlihat pada Gambar 1.

2.1. Pengumpulan Data

Dalam penelitian ini, klasifikasi teks dilakukan menggunakan objek yang berbentuk teks. Data yang digunakan dalam penelitian ini adalah terjemahan Al-Qur'an ke dalam bahasa Indonesia yang diterbitkan pada tahun 2022 oleh Kementerian Agama Republik Indonesia. Terjemahan ini diambil dari situs resmi Kementerian Agama yang tersedia di <https://quran.kemenag.go.id>. Dengan menggunakan data ini, penelitian bertujuan untuk mengembangkan model klasifikasi yang dapat mengelompokkan ayat-ayat Al-Qur'an ke dalam beberapa kategori utama seperti Tauhid, Ibadah, Akhlaq, dan Sejarah (Tarikh).

2.2. Pelabelan Data

Proses pelabelan pada penelitian ini dilakukan dengan cermat oleh para ahli yang melakukan pelabelan secara manual untuk memastikan akurasi dan relevansi dalam konteks studi. Label yang digunakan dalam penelitian ini memuat empat kategori utama yang berkaitan erat dengan tema-tema sentral dalam Al-Qur'an. Pertama, Tauhid merupakan label yang menggambarkan teks-teks yang menekankan keesaan Allah Subhanahu Wa Ta'ala dan keyakinan umat Muslim terhadap ketuhanan yang tunggal dan absolut [17]. Label ini mencakup ayat-ayat yang mengajarkan prinsip dasar dalam Islam yaitu keyakinan terhadap satu Tuhan [18]. Kedua, Ibadah mencakup teks-teks yang mengandung makna ketundukan dan kepatuhan dalam menjalankan serta menghadapi segala larangan Allah Subhanahu Wa

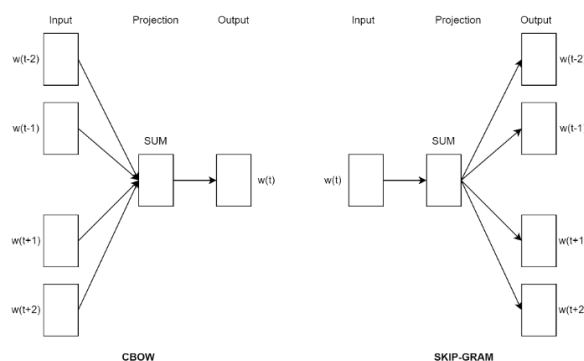
Ta'ala [19]. Ayat-ayat ini memberikan panduan bagi umat Muslim mengenai cara-cara beribadah yang benar dan bagaimana menghindari larangan yang telah ditetapkan dalam agama [20]. Ketiga, Akhlaq adalah label yang menggambarkan teks yang menunjukkan watak dan perilaku seseorang, serta dampaknya terhadap perbuatan baik dan buruk dalam kehidupan sehari-hari [21]. Ayat-ayat ini berfungsi sebagai panduan moral dan etika, membantu individu dalam mengembangkan karakter yang mulia dan menjaga perilaku sesuai dengan ajaran Islam [22]. Keempat, Sejarah (Tarikh) menggambarkan teks yang berisi cerita atau peristiwa dari masa lalu yang memberikan pelajaran dan inspirasi kepada umat Muslim masa kini [23]. Ayat-ayat yang termasuk dalam kategori ini sering kali menceritakan kisah bangsa-bangsa terdahulu, nabi-nabi, dan umat-umat sebelumnya yang bertujuan untuk memberikan pelajaran bagi generasi sekarang [24]. Dengan memahami konteks sejarah dalam Al-Qur'an, pembaca dapat memperoleh wawasan yang lebih dalam tentang peristiwa masa lalu yang membentuk dasar bagi ajaran-ajaran agama yang dipelajari saat ini.

2.3. Text Preprocessing

Tujuan utama dari pemrosesan awal teks atau *text preprocessing* adalah untuk meningkatkan kualitas data dengan membuatnya lebih terstruktur dan siap untuk analisis lanjutan [25]. Langkah *preprocessing* teks diperlukan untuk mengubah kumpulan data mentah menjadi teks yang lebih terstruktur, sehingga data siap untuk diproses pada langkah-langkah berikutnya [26]. Proses ini sangat penting karena data teks yang tidak terstruktur dapat mengandung banyak *noise* atau informasi yang tidak relevan yang dapat mempengaruhi performa dan akurasi model klasifikasi yang digunakan [27]. Dengan menerapkan *preprocessing* yang tepat, kita dapat meningkatkan kinerja model dan memastikan bahwa analisis dilakukan pada data yang relevan dan bersih.

Proses *preprocessing* pada penelitian ini terdiri dari tiga tahapan utama. Tahapan pertama adalah *cleansing*,

yang melibatkan penghapusan atau pembersihan elemen-elemen yang tidak relevan atau mengganggu dalam teks terjemahan, seperti angka, tanda baca yang tidak penting seperti koma (,) dan titik (.), serta delimiter lainnya [28]. Penghapusan elemen ini bertujuan untuk menyederhanakan teks dan mengurangi kebisingan yang dapat mengganggu proses analisis berikutnya. Kedua adalah *case folding*, yaitu proses konversi semua karakter dalam teks menjadi huruf kecil. Langkah ini penting untuk mengurangi keragaman data yang disebabkan oleh perbedaan kapitalisasi huruf, misalnya kata "Allah" dan "allah" dianggap sebagai entitas yang sama dalam konteks analisis teks. Langkah ketiga adalah *tokenizing*, yang melibatkan pemisahan teks menjadi unit-unit yang lebih kecil seperti kalimat, kata, atau frasa. Proses ini memecah teks ke dalam bagian-bagian yang disebut token, yang dapat dianalisis secara individual untuk memahami distribusi dan frekuensi kata atau frasa dalam teks [29]. Misalnya, dalam teks terjemahan Al-Qur'an, *tokenizing* membantu dalam mengidentifikasi dan menghitung kata-kata penting yang mungkin berkaitan dengan kategori tertentu seperti Tauhid atau Ibadah. Selain ketiga langkah utama ini, *preprocessing* juga dapat mencakup teknik tambahan seperti *stopword removal* untuk menghapus kata-kata umum yang tidak memiliki makna signifikan, dan *stemming* atau *lemmatization* untuk mengurangi kata ke bentuk dasarnya.



Gambar 2. Arsitektur CBOW dan Skip-Gram.

2.4. Word2Vec

Word2Vec salah satu teknik *word embedding* yang diusulkan oleh Mikolov pada tahun 2013, dengan mempertimbangkan korpus sebagai masukan dan keluaran sebagai vektor. *Word2Vec* mengubah setiap kata menjadi vektor. Kelebihan *Word2Vec* adalah dapat merepresentasikan kesamaan kontekstual dua kata pada vektor hasil [30]. *Word2Vec* memiliki tiga parameter yang mempengaruhi proses pembelajaran model, yaitu arsitektur, metode evaluasi, dan ukuran. Masing-masing dari ketiga parameter *Word2Vec* memiliki pengaruh terhadap kinerja akurasi *deep learning* [31].

Arsitektur *word embedding* mencakup dua jenis, yaitu *Continuous Bag of Words* (CBOW) dan *Skip-gram* [32] yang digambarkan pada Gambar 2. *Continuous Bag of*

Words (CBOW) dipilih sebagai arsitektur model *word embedding*, karena *Continuous Bag of Words* (CBOW) menghasilkan kinerja yang lebih baik karena memiliki akurasi yang sedikit lebih tinggi pada kata yang sering muncul dan pada dataset pencarian ini terdapat banyak kata yang sering muncul [10]. Skenario model pengujian dengan *word embedding* dilihat pada Tabel 1.

Tabel 1. Skenario Pengujian *Word Embedding*

Word Embedding	Dimensi Size	Epoch
Word2Vec	100	10, 20, 30, 50, 100, 1000
	200	10, 20, 30, 50, 100, 1000
	300	10, 20, 30, 50, 100, 1000

2.6. Pembagian Data

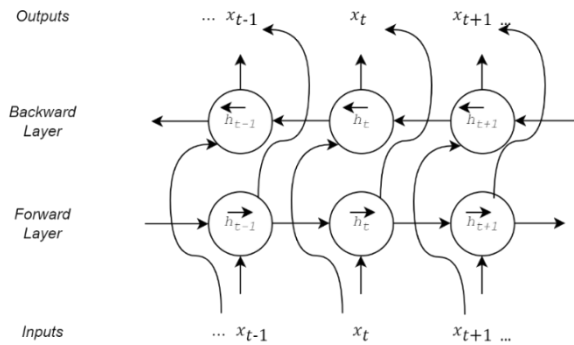
Kumpulan data penelitian ini dibagi menjadi dua bagian utama yaitu data latih dan data uji. Data latih adalah subset dari kumpulan data yang digunakan untuk melatih model agar dapat belajar dan mengenali pola-pola dalam data [33]. Sementara itu, data uji merupakan bagian dari data yang disisihkan dan tidak digunakan selama proses pelatihan; data ini digunakan untuk mengevaluasi kinerja model dan memastikan bahwa model dapat menggeneralisasi dengan baik pada data yang belum pernah dilihat sebelumnya [34]. Pembagian ini penting untuk menghindari *overfitting* dan memberikan estimasi yang lebih akurat tentang bagaimana model akan berkinerja pada data dunia nyata.

Setiap model dalam penelitian diuji dengan empat skenario pembagian data untuk menilai kinerja dan generalisasi model. Skenario-skenario tersebut membagi data latih dan data uji dengan proporsi: 60%-40%, 70%-30%, 80%-20%, dan 90%-10%. Tujuan dari variasi ini adalah untuk melihat bagaimana perubahan proporsi data latih dan data uji mempengaruhi kemampuan model dalam memprediksi data baru. Melalui pengujian berbagai skenario, penelitian ini bertujuan menemukan konfigurasi terbaik untuk klasifikasi *multi-label* teks terjemahan Al-Qur'an.

2.7. Bi-Directional Long Short-Term Memory

Bidirectional Long-Short Term Memory (Bi-LSTM) merupakan arsitektur jaringan saraf yang terdiri dari dua lapisan LSTM, yaitu LSTM maju (*forward LSTM*) untuk memodelkan konteks sebelumnya dan LSTM mundur (*backward LSTM*) untuk memodelkan konteks selanjutnya [35]. *Bidirectional Long-Short Term Memory* (Bi-LSTM) menghubungkan dua lapisan tersebut yang datang dari arah berlawanan dengan keluaran yang sama. Dengan menggunakan bentuk pembelajaran generatif mendalam ini, lapisan *neuron* dapat secara bersamaan memperoleh informasi tentang keadaan masa lalu dan masa depan [36]. Gambar 3

menunjukkan arsitektur dari *Bidirectional Long-Short Term Memory* (Bi-LSTM). *Input* diberikan dari $t - 1$ hingga waktu $t - n$ untuk lapisan sebelumnya dan untuk lapisan berikutnya, input diberikan dalam arah berlawanan dari $t - n$ hingga $t - 1$. *Forward LSTM* operasi dari awal hingga akhir sesuai dengan urutan data. Pada setiap langkah waktu, informasi dari langkah sebelumnya diintegrasikan ke dalam pemrosesan langkah saat ini. Setiap sel LSTM (*Long Short-Term Memory*) memiliki tiga gerbang, yaitu *input gate*, *forget gate*, dan *output gate*, yang memungkinkan pengorganisasian dan penyimpanan informasi jangka panjang. Rumus *Forward LSTM* disajikan pada rumus 1.



Gambar 3. Arsitektur Bi-LSTM.

$$\vec{h}_t = \sigma(W_{x\vec{h}} \cdot x_t + W_{\vec{h}\vec{h}} \cdot \vec{h}_{t-1} + b_{\vec{h}}) \quad (1)$$

Dimana, \vec{h}_t output yang dihasilkan oleh sel LSTM untuk langkah waktu tertentu. σ fungsi aktivasi yang diterapkan pada hasil penjumlahan yang ada di sebelah kanan. $W_{x\vec{h}}$ hasil perkalian matriks antara bobot $W_{x\vec{h}}$ dan vektor *input* x_t . $W_{\vec{h}\vec{h}}$ hasil perkalian matriks antara bobot $W_{\vec{h}\vec{h}}$ dan vektor *output* dari langkah waktu sebelumnya \vec{h}_{t-1} . $b_{\vec{h}}$ adalah bias. Sedangkan *backward LSTM* beroperasi dari awal hingga akhir mengikuti urutan data yang sama. Seperti *forward LSTM*, *backward LSTM* juga memiliki struktur serupa dengan tiga gerbang, sehingga memungkinkan untuk memahami konteks langkah waktu mundur. Rumus 2 merupakan rumus dari *backward LSTM*.

$$\overleftarrow{h}_t = \sigma(W_{x\overleftarrow{h}} \cdot x_t + W_{\overleftarrow{h}\overleftarrow{h}} \cdot \overleftarrow{h}_{t+1} + b_{\overleftarrow{h}}) \quad (2)$$

Dimana, \overleftarrow{h}_t output (*hidden state*) pada langkah waktu t dari LSTM yang bergerak mundur. σ fungsi aktivasi yang diterapkan pada hasil penjumlahan *linier* dari *input* dan *state* sebelumnya. $W_{x\overleftarrow{h}}$ hasil matriks bobot yang menghubungkan *input* x_t ke \overleftarrow{h}_t . $W_{\overleftarrow{h}\overleftarrow{h}}$ matriks bobot yang menghubungkan \overleftarrow{h}_t pada langkah waktu sebelumnya $t + 1$ ke \overleftarrow{h}_t pada langkah waktu i . $b_{\overleftarrow{h}}$ adalah bias. *Output* dari Bi-LSTM, baik *forward* dan *backward*, digabungkan pada setiap langkah waktu. Artinya lapisan Bi-LSTM menghasilkan *vector*

keluaran y_t , dimana setiap elemen dihitung dengan menggunakan rumus 3.

$$y_t = W_{\vec{h}y} \vec{h}_t + W_{\overleftarrow{h}y} \overleftarrow{h}_t + b_y \quad (3)$$

Dimana, y_t *output* pada langkah waktu t . $W_{\vec{h}y}$ bobot yang menghubungkan representasi tersembunyi dari langkah waktu sebelumnya ke *output* y_t . \vec{h}_t representasi tersembunyi pada langkah waktu sebelumnya. $W_{\overleftarrow{h}y}$ bobot yang menghubungkan representasi tersembunyi dari langkah waktu berikutnya ke *output* y_t . \overleftarrow{h}_t Representasi tersembunyi pada langkah waktu berikutnya. b_y bias untuk *output*.

2.8. Evaluasi & Pengujian

Tahapan pengujian merupakan proses melakukan pengujian terhadap model pada penelitian ini. Kinerja model diukur menggunakan metrik *accuracy*, *precision*, *recall* dan *Hamming loss* merupakan salah satu metode yang dapat digunakan untuk mengevaluasi kinerja. Metode ini bertujuan untuk menghitung jumlah kesalahan klasifikasi pada data yang diuji. Semakin kecil nilai yang diperoleh maka kinerja klasifikasi semakin baik dan sebaliknya. Rumus untuk menghitung kerugian *Hamming* dapat dilihat pada Rumus 4.

$$HammingLoss(h) = \frac{1}{p} \sum_{i=1}^p \frac{1}{q} |h(x_i) \Delta y_i| \quad (4)$$

Dimana p adalah jumlah data yang akan digunakan dalam klasifikasi dan q adalah jumlah total label yang ada. Di dalam rumus ini $|h(x) \Delta Y|$ mewakili jumlah kesalahan yang terjadi pada saat klasifikasi, yaitu perbedaan antara label yang diprediksi oleh model dan label sebenarnya untuk setiap data. Metrik ini sangat penting untuk memahami seberapa sering model salah dalam memprediksi label yang tepat dari setiap sampel. Dengan menggunakan *Hamming Loss*, kita dapat mengevaluasi kinerja model klasifikasi dalam memprediksi beberapa label secara simultan dan menentukan seberapa akurat model dalam tugas *multi-label*. Sedangkan *Accuracy* digunakan untuk mengukur sejauh mana model dapat mengklasifikasikan dengan benar seluruh label pada data uji. *Precision* mendeskripsikan dalam mengukur sejauh mana prediksi positif yang dilakukan oleh model dengan benar. *Recall* menentukan kapasitas model untuk mendeteksi atau mengidentifikasi label yang sesuai dalam kumpulan data.

3. Hasil dan Pembahasan

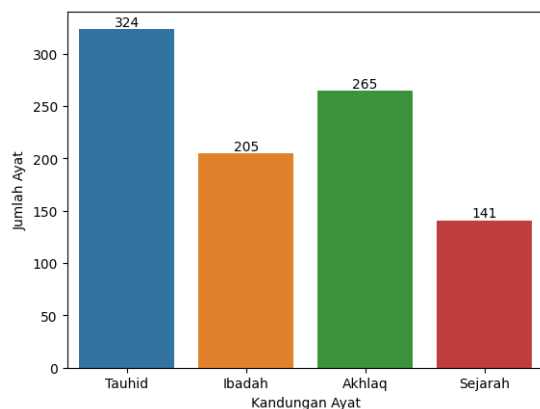
Data yang digunakan dalam penelitian ini diambil dari Al-Qur'an terjemahan yang diterbitkan oleh Kementerian Agama Republik Indonesia. Secara keseluruhan, Al-Qur'an memiliki 6.236 ayat yang tersebar di berbagai surah. Namun, penelitian ini memfokuskan pada teks terjemahan dari tiga surah utama, yaitu Surah An-Nisa', Surah Al-Maidah, dan Surah Al-An'am, yang secara total terdiri dari 461 ayat.

Pemilihan ketiga surah ini dilakukan untuk memberikan variasi dan representasi yang cukup dari teks terjemahan Al-Qur'an. Dengan menggunakan subset data ini, penelitian bertujuan untuk mengembangkan dan menguji model klasifikasi *multi-label* yang efektif dalam memprediksi kategori setiap ayat.

Tabel 2. Pelabelan Terjemahan Al-Qur'an

Terjemahan	Tauhid	Ibadah	Akhlaq	Sejarah
Allah berfirman: "(Jika demikian), maka sesungguhnya negeri itu diharamkan atas mereka selama empat puluh tahun, ... Sungguh kalau kamu menggerakkan tanganmu kepadaku untuk membunuhku, aku sekali-kali tidak akan menggerakkan	0	0	1	1
	1	0	1	1

Kemudian proses pelabelan dilakukan secara manual oleh para ahli dengan memberi label Tauhid, Ibadah, Akhlaq dan Sejarah pada setiap ayat dalam terjemahan Al-Qur'an. Berdasarkan Tabel 2, setiap label mendapat nilai 1 jika terjemahannya termasuk dalam kategori pada label dan nilai 0 jika terjemahannya tidak termasuk dalam label. Dengan demikian, sebaran label ditunjukkan pada Gambar 4 dengan jumlah label Tauhid sebanyak 324 label, Ibadah sebanyak 205 label, Akhlaq sebanyak 265 label, dan Sejarah sebanyak 141 label.



Gambar 4. Distribusi Label Terjemahan Al-Qur'an.

3.1. Text Preprocessing

Data yang digunakan dalam proses ini mewakili 461 terjemahan Al-Qur'an bahasa Indonesia. Langkah ini membersihkan data, mengubah semua huruf menjadi huruf kecil, dan membagi kalimat menjadi beberapa bagian. *Preprocessing* menghasilkan data terjemahan

Al-Qur'an yang bersih, yang kemudian digunakan untuk data pelatihan dan pengujian. Tabel 3 menyajikan hasil *preprocessing* teks.

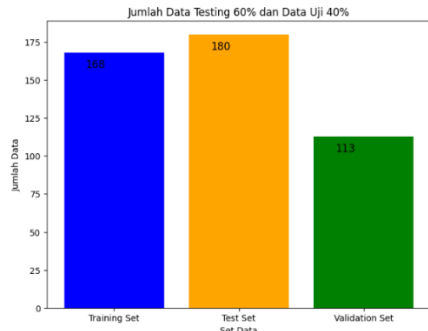
Tabel 3. Hasil *Preprocessing* Terjemahan

Terjemahan Original	Sesudah Preprocessing
Dan mereka (orang-orang munafik) mengatakan: "(Kewajiban kami hanyalah taat". Tetapi apabila mereka telah pergi dari sisimu, sebahagian dari mereka mengatur siasat di malam hari (mengambil keputusan) lain dari yang telah mereka katakan tadi. Allah menulis siasat yang mereka atur di malam hari itu, maka berpalinglah kamu dari mereka dan tawakallah kepada Allah. Cukuplah Allah menjadi Pelindung.	dan mereka orang-orang munafik mengatakan kewajiban kami hanyalah taat tetapi apabila mereka telah pergi dari sisimu sebahagian dari mereka mengatur siasat di malam hari mengambil keputusan lain dari yang telah mereka katakan tadi allah menulis siasat yang mereka atur di malam hari itu maka berpalinglah kamu dari mereka dan tawakallah kepada allah cukuplah allah menjadi pelindung

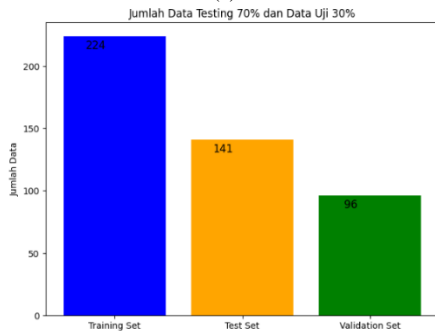
3.2. Pembagian Data Latih dan Data Uji

Memisahkan data pelatihan dan data pengujian untuk mengklasifikasikan terjemahan Al-Qur'an bahasa Indonesia penting dilakukan untuk menguji performa model secara akurat. Biasanya, pembagian data ini memungkinkan model untuk belajar dari data yang ada (data latih), yang kemudian dievaluasi oleh model selama pelatihan dan dilakukan penyesuaian parameter untuk menghindari *overfitting* (data validasi). Setelah itu, data baru (data uji) digunakan untuk mengukur seberapa baik model dapat memprediksi informasi yang belum pernah dilihat sebelumnya. Dalam penelitian ini, kami menguji model dengan beberapa variasi data latih dan data uji pada ukuran 60%:40%, 70%:30%, 80%:20%, dan 90%:10%. Pendekatan ini memastikan bahwa model diuji dalam berbagai kondisi untuk mendapatkan gambaran yang komprehensif tentang kinerjanya.

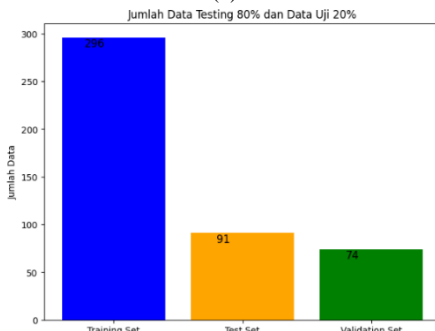
Pembagian data latih dan data uji divisualisasikan pada Gambar 5, menunjukkan empat skenario pengujian model klasifikasi. Pada skenario pertama (a), 60% data digunakan untuk pelatihan dan 40% untuk pengujian. Jumlah data adalah 168 untuk pelatihan, 113 untuk validasi, dan 180 untuk pengujian. Pada skenario kedua (b), 70% data digunakan untuk pelatihan dan 30% untuk pengujian, dengan 224 data untuk pelatihan, 96 data untuk validasi, dan 141 untuk pengujian. Skenario ketiga (c) membagi 80% untuk pelatihan dan 20% untuk pengujian, menghasilkan 296 data untuk pelatihan, 74 untuk validasi, dan 91 untuk pengujian. Terakhir, pada skenario keempat (d), 90% data untuk pelatihan dan 10% untuk pengujian, dengan 372 data untuk pelatihan, 42 untuk validasi, dan 47 untuk pengujian. Melalui berbagai skenario ini, penelitian bertujuan untuk mengevaluasi bagaimana perubahan dalam pembagian data dapat mempengaruhi kinerja model secara keseluruhan.



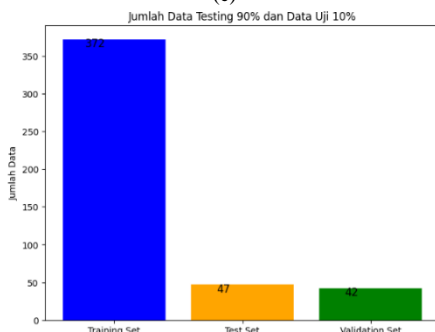
(a)



(b)



(c)



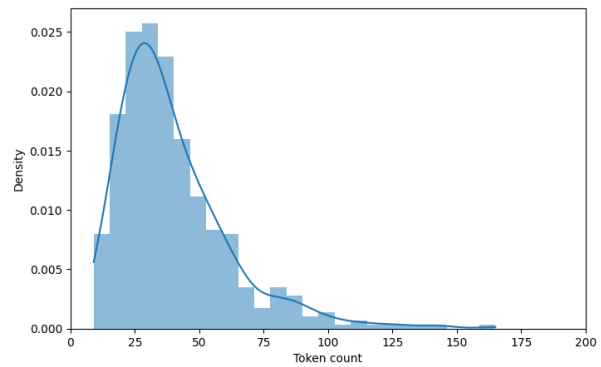
(d)

Gambar 5. Visualisasi Pembagian Data Latih dan Uji.

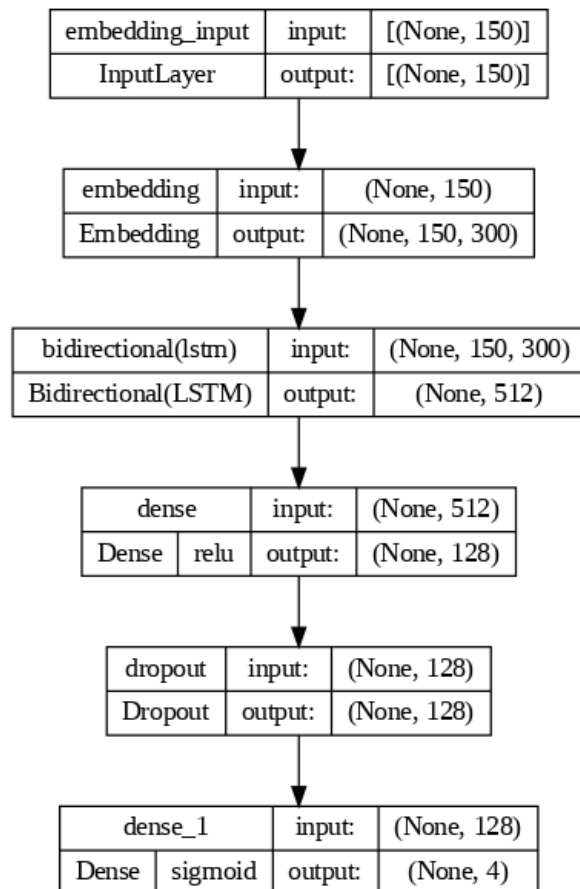
3.3. Pengujian *Bi-directional Long Short-Term Memory*

Pada tahapan ini dengan pengujian Bi-LSTM langkah awal dilakukan dengan *preprocessing* seperti yang dilakukan pada model dasar tahap pertama. Langkah selanjutnya adalah menentukan panjang maksimal rangkaian token kata yang dapat diakomodasi oleh model dengan menghitung distribusi panjang token kata untuk setiap kalimat dalam data. Dari Gambar 6,

panjang token terbesar sekitar 20 hingga 50 token, sedangkan panjang token terkecil sekitar 125 hingga 150 token. Oleh karena itu, nilai *max_seq_length* pada model dasar tahap kedua ditentukan sebesar 150. Selanjutnya model Bi-LSTM didefinisikan pada Gambar 7 yang merupakan struktur model yang menunjukkan bahwa ini adalah model *sequential* dengan beberapa lapisan yang diterapkan untuk klasifikasi teks terjemahan Al-Qur'an Bahasa Indonesia. Pertama, lapisan *embedding* dipakai untuk mengonversi kata-kata ke dalam representasi vektor numerik dengan dimensi 300 di dalam jendela sepanjang 150 kata.



Gambar 6. Distribusi token kata dalam data.



Gambar 7. Struktur model Bi-LSTM.

Ini berguna untuk memahami konteks dan hubungan antar-kata dalam teks Al-Qur'an yang diinterpretasikan. Lapisan *Bidirectional LSTM (Long Short-Term Memory)* digunakan untuk mempelajari pola sekuensial dan hubungan antar kata secara dua arah, memungkinkan pemahaman yang lebih mendalam terhadap urutan kata dalam teks Al-Qur'an. Selanjutnya, lapisan *dense (Dense)* dengan 128 unit diikuti oleh lapisan *dropout* untuk mengurangi *overfitting* dengan mengabaikan sebagian koneksi antar unit. Lapisan *dense* terakhir dengan 4 unit adalah *output* dari model ini, yang mengindikasikan bahwa model ini didesain untuk menghasilkan prediksi pada empat kategori yang ditentukan sebelumnya, yaitu Tauhid, Ibadah, Akhlaq, dan Sejarah (Tarikh). Total parameter yang digunakan dalam model ini adalah sekitar 1,91 juta parameter dengan semua parameter ini dapat disesuaikan (*trainable*), yang artinya seluruh parameter ini akan disesuaikan secara otomatis selama proses pelatihan model.

Berdasarkan hasil evaluasi model menggunakan algoritma Bi-LSTM disampaikan pada Tabel 4 menunjukkan nilai akurasi sebesar 50%, yang mengindikasikan bahwa model berhasil memprediksi dengan benar setengah dari keseluruhan data uji. *Precision* sebesar 62% menunjukkan seberapa tepat model dalam mengidentifikasi setiap kelas secara individual, sedangkan *recall* sebesar 56% mencerminkan sejauh mana model mampu mengambil data yang relevan dari setiap kelas yang diuji. *Hamming loss* sebesar 35% menggambarkan tingkat kesalahan prediksi pada setiap sampel. Meskipun model ini memberikan hasil yang cukup baik dalam beberapa aspek, nilai-nilai evaluasi yang terjadi perlu diperhatikan untuk perbaikan lebih lanjut agar dapat meningkatkan akurasi, presisi, *recall*, dan mengurangi tingkat kesalahan dalam prediksi klasifikasi teks Al-Qur'an Bahasa Indonesia. Sehingga untuk meningkatkan performa pada model ini, dilakukan pengujian dengan menambahkan fitur *word embedding* pada model.

Tabel 4. Hasil evaluasi model Bi-LSTM

Skenario	Accuracy	Precision	Recall	Hamming Loss
60:40	43%	61%	45%	40%
70:30	50%	62%	56%	35%
80:20	48%	60%	54%	37%
90:10	45%	62%	55%	37%

3.4 Word Embedding

Tahapan ini menerapkan representasi kata-kata sebagai vektor dan diharapkan mampu meningkatkan nilai akurasi dari model Bi-LSTM. Skema pengujian pada *word embedding* dipaparkan pada Tabel 1 dengan menghasilkan model *word embedding* dengan 100, 200, dan 300 dimensi menggunakan *word2vec*. Hasil pelatihan *word embeddings* disajikan pada Tabel 5. Tabel 5 tersebut menampilkan hasil evaluasi dari model

menggunakan algoritma *Word2Vec* dalam empat skenario pengujian yang berbeda. Setiap skenario mewakili proporsi pembagian data latih dan data uji yang berbeda. Skenario 1 membagi data latih 60% dan data uji 40%, Skenario 2 membagi 70% latih dan 30% uji, Skenario 3 menggunakan pembagian 80% latih dan 20% uji, sementara Skenario 4 memiliki proporsi 90% latih dan 10% uji.

Dalam setiap skenario, terdapat evaluasi model yang diukur menggunakan dimensi dan *epochs* yang berbeda. Dimensi merujuk pada dimensi representasi vektor kata-kata, sedangkan *epochs* adalah jumlah iterasi selama pelatihan model. Melalui hasil yang dicatat dalam tabel, terlihat performa beragam dari model pada kombinasi dimensi dan *epochs* tertentu dalam masing-masing skenario. Secara umum, terdapat variasi performa yang cukup signifikan antara skenario-skenario. Sebagai contoh, pada skenario 1 untuk *Word2Vec*, peningkatan dimensi dari 100 hingga 300 pada 20 *epochs* menunjukkan peningkatan akurasi secara konsisten dari sekitar 47% menjadi sekitar 53%, dengan variasi kecil dalam pengujian berbeda. Skenario 4 cenderung menunjukkan performa yang lebih tinggi, terutama pada kombinasi dimensi dan *epochs* tertentu. Analisis lebih lanjut terhadap variasi ini dapat membantu menentukan konfigurasi terbaik untuk pembagian data latih dan uji, serta parameter model seperti dimensi dan *epochs* yang dapat meningkatkan akurasi klasifikasi teks Al-Qur'an Bahasa Indonesia.

Tabel 5. Hasil pengujian Bi-LSTM dengan *Word Embedding*

Dimensi	Epoch	Word2Vec			
		60:40	70:30	80:20	90:10
100	10	47.45%	52.54%	52.11%	57.27%
	20	51.20%	51.00%	54.58%	60.64%
	30	50.00%	52.66%	54.12%	56.91%
	50	51.11%	50.47%	56.14%	61.52%
	100	49.91%	48.64%	54.30%	57.09%
	1000	50.42%	51.60%	54.85%	56.03%
200	10	48.94%	54.37%	53.85%	60.82%
	20	51.30%	53.07%	54.85%	60.46%
	30	52.41%	55.85%	55.86%	59.04%
	50	52.92%	52.96%	56.87%	58.16%
	100	52.13%	57.33%	54.85%	58.16%
	1000	48.10%	53.25%	53.11%	57.62%
300	10	50.69%	55.20%	56.96%	57.80%
	20	53.80%	56.21%	57.60%	57.45%
	30	51.85%	56.56%	56.78%	58.87%
	50	51.16%	55.32%	54.21%	56.74%
	100	53.47%	55.56%	58.61%	54.43%
	1000	52.18%	50.83%	56.23%	58.51%

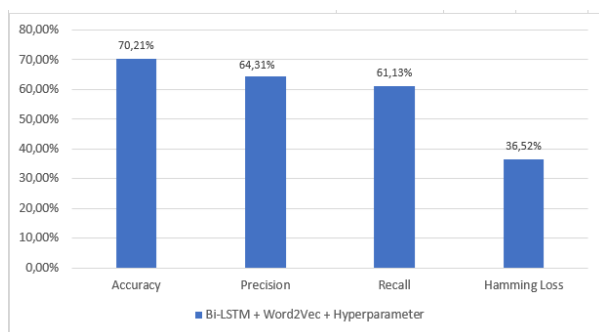
3.5 Hyperparameter Tuning

Untuk melatih model dengan lebih baik dengan mengimplementasikan *word embeddings*, penting untuk menentukan *hyperparameter*. *Hyperparameter* digunakan untuk mencari nilai optimal dalam model dan meningkatkan performa model. *Hyperparameter* yang ditentukan mencakup *epoch*, *batch size*, dan *learning rate*. *Hyperparameter* terbaik akan diterapkan untuk menguji model dasar terbaik yang telah diintegrasikan pada fitur *word embeddings*, sehingga akurasi yang diperoleh akan lebih baik.

Setiap model *word embeddings* baik itu *word2vec* akan pengujian parameter *learning rate* dengan range 0.00001 (1e-5) sampai dengan 0.1 (1e-1). *Batch size* dengan range 16 – 128 dan *epochs* dengan range 10 – 100 pengujian. Hasil yang didapatkan selama proses *hyperparameter* disajikan pada Tabel 6. Model *word2Vec* menghasilkan *learning rate* 0,092857763, *batch size* 107 dan *epoch* 96. Berdasarkan Gambar 8, hasil pengujian *hyperparameter* dengan parameter terbaik sebanyak 96 *epoch*, *batch size* 107, dan *learning rate* sebesar 0,092857763 dari pengujian model *Bi-Directional Long Short-Term Memory* (Bi-LSTM) dengan *Word2Vec* menunjukkan nilai akurasi sebesar 70,21%, *precision* 64,31%, *recall* 61,13% dan *hamming loss* 36,52%.

Tabel 6. Hasil Parameter Terbaik *Hyperparameter Tuning*

Word Embedding	Learning Rate	Batch Size	Epochs
Word2Vec	0,092857763	107	96



Gambar 8. Hasil pengujian model dengan *hyperparameter*.

3.6 Analisa Hasil Pengujian

Berdasarkan pengujian model untuk klasifikasi *multi-label* terjemahan Al-Qur'an berbahasa Indonesia hasil klasifikasi tanpa menggunakan *word embedding* dan penyetulan *hyperparameter* memiliki akurasi yang rendah. Akurasi berangsur naik jika model *Bi-Directional Long Short-Term Memory* (Bi-LSTM) dengan model *word embedding word2Vec* yang dilakukan *tuning hyperparameter* optimal pada model *Bi-directional Long Short-Term Memory* (Bi-LSTM) dengan penyemat kata *Word2Vec* menghasilkan akurasi mencapai 70,21%. Namun, akurasinya belum

mencapai 100% dalam mengklasifikasikan beberapa label terjemahan Al-Qur'an Indonesia. Hal ini mungkin disebabkan oleh beberapa faktor. Pertama, alasannya adalah bahasa yang kompleks dan konteks Al-Qur'an yang sangat spesifik dapat menyebabkan variasi penafsiran yang sulit diwakilkan sepenuhnya oleh model bahasa mesin. Kedua, adanya variasi gaya penulisan dan struktur kalimat yang dapat mempersulit pemodelan bahasa. Selain itu, keberhasilan klasifikasi *multi-label* juga bergantung pada ketersediaan dan kualitas kumpulan data yang digunakan untuk melatih model. Faktor-faktor ini dapat menyebabkan keterbatasan dalam prediksi sempurna, bahkan dengan pengaturan *hyperparameter* terbaik. Dalam konteks kompleks seperti ini, mencapai akurasi 100% bisa jadi sangat sulit karena kompleksitas bahasa dan variasi penafsiran yang mungkin timbul.

3.7 Perbandingan Penelitian Terdahulu

Dalam upaya untuk memahami dan mengklasifikasikan teks terjemahan Al-Qur'an, berbagai pendekatan telah dikembangkan dan diterapkan oleh para peneliti di masa lalu. Penelitian terdahulu ini menjadi fondasi yang penting dalam pengembangan metode klasifikasi teks yang lebih efisien dan akurat. Memahami temuan dan metodologi yang telah digunakan dalam penelitian-penelitian sebelumnya membantu dalam mengevaluasi kontribusi baru dari penelitian ini dan menemukannya dalam konteks pengetahuan yang sudah ada. Penelitian ini tidak hanya melihat bagaimana teknik-teknik yang berbeda telah diterapkan tetapi juga mengeksplorasi bagaimana inovasi dan perbaikan dapat dilakukan untuk menangani tantangan yang ada dalam klasifikasi *multi-label* teks terjemahan Al-Qur'an. Melalui analisis perbandingan ini, dapat diidentifikasi keunggulan, kelemahan, dan kontribusi unik yang ditawarkan oleh pendekatan yang diusulkan dalam penelitian ini, yang pada akhirnya bertujuan untuk memperluas dan meningkatkan pemahaman tentang klasifikasi teks keagamaan yang kompleks. Tabel 7 berikut memberikan perbandingan ringkas antara hasil dari berbagai penelitian sebelumnya dengan hasil yang dicapai dalam studi ini.

Tabel 7. Perbandingan Penelitian Terdahulu

Reference	Algoritma	Accuracy	Precision	Recall
[4]	SVM	70,00%	68,00%	-
[6]	K-NN	64,10%	-	-
[7]	K-NN	65,35%	63,00%	61,00%
[8]	NB	60,09%	-	-
Penulis	Bi-LSTM + Word2Vec CBOW + Tuning Hyperparameter	70,21%	64,31%	61,13%

Dari Tabel 7 diatas menyajikan perbandingan antara berbagai algoritma yang digunakan dalam penelitian sebelumnya dengan pendekatan yang diusulkan dalam studi ini. Penelitian terdahulu yang dirujuk dalam tabel

ini telah menggunakan metode seperti *Support Vector Machine* (SVM), *K-Nearest Neighbor* (K-NN), dan *Naïve Bayes* (NB) untuk klasifikasi teks terjemahan Al-Qur'an. Setiap metode memiliki kekuatan dan kelemahan masing-masing dalam menangani data teks yang kompleks dan *multi-label*. Penelitian ini bertujuan untuk mengkaji bagaimana pendekatan *Bi-Directional Long Short-Term Memory* (Bi-LSTM) yang dipadukan dengan *Word2Vec* CBOW dapat memberikan hasil yang lebih baik dalam konteks klasifikasi *multi-label* terjemahan Al-Qur'an. Pada [4], algoritma SVM menunjukkan akurasi yang cukup tinggi sebesar 70,00%. *Precision* yang dicapai oleh SVM dalam penelitian ini adalah 68,00%, menunjukkan kemampuan yang baik dalam mengidentifikasi label yang benar dalam data teks. Namun, tabel tidak menyertakan nilai *Recall*, yang penting untuk memahami sejauh mana model dapat menangkap semua label yang relevan. SVM, meskipun kuat dalam klasifikasi teks, terkadang menghadapi kesulitan dalam menangani data yang sangat variatif dan kompleks seperti teks Al-Qur'an, di mana beberapa ayat dapat memiliki konteks *multi-label*.

Dua referensi menggunakan K-NN sebagai metode klasifikasi. Pada [6], K-NN mencapai akurasi sebesar 64,10%. Meskipun nilai akurasinya lebih rendah dibandingkan dengan SVM dan metode lain, K-NN tetap relevan dalam studi teks karena kesederhanaan dan kemudahannya dalam implementasi. *Precision* dan *Recall* tidak dilaporkan dalam penelitian ini, yang membuat sulit untuk sepenuhnya mengevaluasi performanya dalam klasifikasi *multi-label*. Pada referensi [7], K-NN mencapai akurasi yang sedikit lebih baik yaitu 65,35%, dengan *Precision* dan *Recall* masing-masing sebesar 63,00% dan 61,00%. Nilai ini menunjukkan bahwa K-NN dapat memberikan hasil yang konsisten tetapi masih di bawah performa metode *deep learning* seperti yang diusulkan dalam penelitian ini. Pada [8] menggunakan *Naïve Bayes* sebagai pendekatan klasifikasi. Algoritma ini mencapai akurasi sebesar 60,09%, yang merupakan nilai terendah di antara metode yang dibandingkan dalam tabel. Meskipun *Naïve Bayes* dikenal dengan kesederhanaannya dan efisiensi komputasinya, hasil ini menunjukkan bahwa NB mungkin kurang optimal dalam menangani klasifikasi *multi-label* pada teks Al-Qur'an yang kompleks. *Precision* dan *Recall* tidak disediakan dalam penelitian ini, sehingga sulit untuk menilai sepenuhnya efektivitasnya dalam memprediksi label dengan benar dalam konteks *multi-label*.

Metode yang diusulkan dalam penelitian ini, yaitu Bi-LSTM yang digabungkan dengan *Word2Vec* CBOW dan *tuning hyperparameter*, menunjukkan akurasi sebesar 70,21%. *Precision* sebesar 64,31% dan *Recall* sebesar 61,13% menunjukkan bahwa metode ini tidak hanya mampu mengidentifikasi label dengan baik tetapi juga cukup efektif dalam menangkap semua label yang relevan. Meskipun akurasi hanya sedikit lebih tinggi

dibandingkan SVM, keunggulan dalam *Precision* dan *Recall* menunjukkan bahwa pendekatan ini lebih efisien dalam menangani kompleksitas dan variabilitas data teks terjemahan Al-Qur'an. Ini menunjukkan potensi besar dari pendekatan *deep learning* dalam meningkatkan klasifikasi teks keagamaan yang kompleks. Melalui analisis ini, bahwa pendekatan yang diusulkan dalam penelitian ini menawarkan keunggulan signifikan dibandingkan dengan metode tradisional seperti SVM, K-NN, dan *Naïve Bayes*. Pendekatan Bi-LSTM + *Word2Vec* CBOW + *tuning hyperparameter* tidak hanya memberikan hasil yang kompetitif dalam hal akurasi tetapi juga unggul dalam aspek *Precision* dan *Recall*, menjadikannya pilihan yang kuat untuk klasifikasi *multi-label* teks terjemahan Al-Qur'an.

4. Kesimpulan

Penelitian ini berhasil mengembangkan model klasifikasi *multi-label* untuk teks terjemahan Al-Qur'an dalam bahasa Indonesia menggunakan *Bi-Directional Long Short-Term Memory* (Bi-LSTM) dengan teknik *word embedding* *Word2Vec*. Fokus penelitian adalah pada tiga surah utama: Surah An-Nisa', Surah Al-Maidah, dan Surah Al-An'am, dengan total 461 ayat terjemahan. Model Bi-LSTM dengan *Word2Vec* terbukti lebih efektif dibandingkan metode klasifikasi tradisional seperti *K-Nearest Neighbor* (K-NN) dan *Support Vector Machine* (SVM), khususnya dalam menangani klasifikasi *multi-label*. Pengujian dilakukan dengan berbagai skenario pembagian data latihan dan data uji (60%-40%, 70%-30%, 80%-20%, dan 90%-10%) untuk mengevaluasi kinerja model. Hasil akhir menunjukkan bahwa model Bi-LSTM dengan *Word2Vec* dan *tuning hyperparameter* optimal mencapai akurasi hingga 70,21%, yang merupakan peningkatan signifikan dari metode lainnya. Meskipun demikian, tantangan tetap ada dalam mencapai akurasi 100%, yang sebagian besar disebabkan oleh kompleksitas bahasa dan konteks dalam teks Al-Qur'an serta variasi dalam gaya penulisan dan struktur kalimat. Faktor-faktor ini menunjukkan bahwa meskipun telah ada perbaikan, masih ada ruang untuk peningkatan lebih lanjut dalam model klasifikasi teks terjemahan Al-Qur'an.

Untuk meningkatkan akurasi dan efisiensi model lebih lanjut, beberapa langkah dapat diambil. Pertama, peningkatan kualitas data dengan mengumpulkan lebih banyak data yang representatif dan melibatkan lebih banyak ahli dalam proses pelabelan manual dapat meningkatkan keakuratan model. Kedua, pengujian teknik *word embedding* lain seperti *GloVe* atau *FastText* mungkin memberikan hasil yang lebih baik dalam representasi konteks kata. Ketiga, eksplorasi arsitektur model lain seperti *Transformer* atau BERT, yang telah menunjukkan hasil yang unggul dalam pemahaman bahasa alami, dapat memberikan alternatif yang lebih efektif. Selain itu, penambahan fitur

kontekstual seperti metadata atau informasi tambahan tentang ayat dan surah, serta integrasi teknik attention dalam model, dapat membantu model memahami konteks dengan lebih baik. Evaluasi lebih lanjut dengan data uji yang lebih bervariasi dan melibatkan pengguna akhir dapat memberikan gambaran yang lebih jelas tentang relevansi hasil klasifikasi dalam penggunaan nyata. Terakhir, melakukan analisis kesalahan dapat membantu mengidentifikasi jenis kesalahan yang sering terjadi, memungkinkan penyesuaian model untuk mengatasi kekurangan tersebut. Dengan penerapan saran-saran ini, model klasifikasi *multi-label* untuk terjemahan Al-Qur'an dapat terus ditingkatkan, memberikan kontribusi yang lebih besar dalam mempermudah pembelajaran dan pemahaman Al-Qur'an secara lebih akurat dan efektif.

Daftar Pustaka

- [1] M. Oktarina, "Faedah Mempelajari Dan Membaca Al-Quran Dengan Tajwid," *Serambi Tarbawi*, vol. 8, no. 2, pp. 147–162, 2020, doi: 10.32672/tarbawi.v8i2.5072.
- [2] S. Qutub, "Sumber-Sumber Ilmu Pengetahuan Dalam Al Qur'an Dan Hadits," *Humaniora*, vol. 2, no. 2, p. 1339, Oct. 2011, doi: 10.21512/humaniora.v2i2.3198.
- [3] M. Robani and A. Widodo, "Algoritma K-Means Clustering Untuk Pengelompokan Ayat Al Quran Pada Terjemahan Bahasa Indonesia," *Jurnal Sistem Informasi Bisnis*, vol. 6, no. 2, p. 164, 2016, doi: 10.21456/vol6iss2pp164-176.
- [4] M. Fauzan, H. Junaedi, and E. Setyati, "Klasifikasi Al – Qur'an Terjemahan Bahasa Indonesia Dengan Menggunakan Algoritma Support Vector Machine (SVM)," *KONVERGENSI*, vol. 18, no. 2, pp. 42–49, Dec. 2022, doi: 10.30996/konv.v18i1.6912.
- [5] M. Irfan, W. Uriawan, N. Lukman, O. Kurahman, and W. Darmalaksana, "The Qur'anic Classification Uses Algorithm C4.5," 2020, doi: 10.4108/eai.2-10-2018.2295558.
- [6] 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, 2020, doi: 10.32736/sisfokom.v9i3.980.
- [7] T. H. Putrisanni, A. Adiwijaya, and S. Al Faraby, "Klasifikasi Ayat Al - Quran Terjemahan Bahasa Inggris Menggunakan K-Nearest Neighbor Dan Information Gain," *KOMIK (Konferensi Nasional Teknologi Informasi dan Komputer)*, vol. 3, no. 1, pp. 362–369, 2019, doi: 10.30865/komik.v3i1.1614.
- [8] A. Abdullahi, N. A. Samsudin, M. H. A. Rahim, S. K. A. Khalid, and R. Efendi, "Multi-Label Classification Approach For Quranic Verses Labeling," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 24, no. 1, pp. 484–490, 2021, doi: 10.11591/ijeecs.v24.i1.pp484-490.
- [9] B. A. H. Kholifatullah and A. Prihanto, "Penerapan Metode Long Short Term Memory Untuk Klasifikasi Pada Hate Speech," *Journal of Informatics and Computer Science (JINACS)*, vol. 04, pp. 292–297, Jan. 2023, doi: 10.26740/jinacs.v4n03.p292-297.
- [10] D. I. Af'idah, D. Dairoh, S. F. Handayani, and R. W. Pratiwi, "Pengaruh Parameter Word2Vec Terhadap Performa Deep Learning Pada Klasifikasi Sentimen," *Jurnal Informatika: Jurnal Pengembangan IT*, vol. 6, no. 3, pp. 156–161, 2021, doi: 10.30591/jpit.v6i3.3016.
- [11] A. Sepas-Moghaddam, A. Etemad, F. Pereira, and P. L. Correia, "Long Short-Term Memory With Gate And State Level Fusion For Light Field-Based Face Recognition," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 1365–1379, 2021, doi: 10.1109/TIFS.2020.3036242.
- [12] B. Jang, M. Kim, G. Harerimana, S. U. Kang, and J. W. Kim, "Bi-LSTM Model To Increase Accuracy In Text Classification: Combining Word2vec CNN And Attention Mechanism," *Applied Sciences (Switzerland)*, vol. 10, no. 17, Sep. 2020, doi: 10.3390/app10175841.
- [13] R. A. Rajagede and R. P. Hastuti, "Al-Quran Recitation Verification For Memorization Test Using Siamese LSTM Network," *Communications in Science and Technology*, vol. 6, no. 1, pp. 35–40, 2021, doi: 10.21924/CST.6.1.2021.344.
- [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, 2020, doi: 10.26555/jiteki.v5i2.15021.
- [15] W. Chen, X. Liu, D. Guo, and M. Lu, "Multi-label Text Classification Based On Sequence Model," in *Communications in Computer and Information Science*, Springer Verlag, 2019,

- pp. 201–210. doi: 10.1007/978-981-32-9563-6_21. [25]
- [16] R. B. Afrianto and L. Y. Kurniawati, “Kategorisasi Dokumen Teks Secara Multi Label Menggunakan Fuzzy C-Means Dan K-Nearest Neighbors Pada Artikel Berbahasa Indonesia,” *JUTI: Jurnal Ilmiah Teknologi Informasi*, vol. 11, no. 1, p. 23, 2013, doi: 10.12962/j24068535.v11i1.a17. [26]
- [17] S. Sayyida, “Ayat-Ayat Tauhid Terhadap Budaya Pemeliharaan Keris Di Jawa (Studi Kasus Buku Mt Arifin),” *Journal of Qur’an and Hadith Studies*, vol. 6, no. 1, pp. 24–52, 2019, doi: 10.15408/quhas.v6i1.13403. [27]
- [18] S. Hadi and L. M. Mujahid Imaduddin, “Tawhid sebagai Prinsip Primordial Peradaban Islam: Studi Pemikiran Isma’il Raji al-Faruqi,” *Tsaqafah - Jurnal Peradaban Islam*, vol. 15, no. 2, pp. 265–290, 2019, [Online]. Available: www.irf.net [28]
- [19] A. Kallang, “Konteks Ibadah Menurut Al-Quran,” *Al-Din: Jurnal Dakwah dan Sosial Keagamaan*, vol. 4, no. 2, pp. 1–13, 2018, doi: 10.35673/ajdsk.v4i2.630. [29]
- [20] R. Abdul Hadi and Y. Suharyat, “Dakwah Dalam Perspektif Al Qur’an Dan Al Hadits,” *Jurnal Agama, Sosial dan Budaya (Religion)*, vol. 1, no. 5, 2022. [30]
- [21] M. Murharyana, I. I. Al Ayyubi, and R. Rohmatulloh, “Pendidikan Akhlak Anak Kepada Orang Tua Dalam Perspektif Al-Quran,” *Piwulang: Jurnal Pendidikan Agama Islam*, vol. 5, no. 2, pp. 175–191, 2023, doi: http://dx.doi.org/10.32478/piwulang.v5i2.1515. [31]
- [22] M. Astuti, R. Febriani, and N. Oktarina, “Pentingnya Pendidikan Islam Dalam Membentuk Karakter Generasi Muda,” *Jurnal Faidatuna*, vol. 4, no. 3, pp. 140–149, 2023, doi: 10.53958/ft.v4i3.302. [32]
- [23] J. Mirdad and S. Rahmat, “Sejarah Dalam Perspektif Islam,” *El -Hekam*, vol. 6, no. 1, p. 9, 2021, doi: 10.31958/jeh.v6i1.3303. [33]
- [24] A. S. Mainiyo and M. M. Sule, “Impact Of Qur’anic Moral Excellence On The Lives Of Muslim Society: An Exposition,” *Jurnal Ilmu-Sosial (Spectra)*, vol. 6, no. 1, 2024, doi: 10.34005/spektra.v6i1.3637. [34]
- L. Hickman, S. Thapa, L. Tay, M. Cao, and P. Srinivasan, “Text Preprocessing For Text Mining In Organizational Research: Review And Recommendations,” *Organ Res Methods*, vol. 25, no. 1, pp. 114–146, Jan. 2022, doi: 10.1177/1094428120971683. [35]
- M. Nesca, A. Katz, C. K. Leung, and L. M. Lix, “A Scoping Review Of Preprocessing Methods For Unstructured Text Data To Assess Data Quality,” *International Journal of Population Data Science Journal Website: www.ijpds.org*, vol. 7, no. 1, 2022, doi: 10.23889/ijpds.v6i1.1757. [36]
- S. Bessou and R. Aberkane, “Subjective Sentiment Analysis For Arabic Newswire Comments,” *Journal of Digital Information Management*, vol. 17, no. 5, p. 289, 2019, doi: 10.6025/jdim/2019/17/5/289-295. [37]
- A. A. Firdaus, A. Yudhana, and I. Riadi, “Public Opinion Analysis Of Presidential Candidate Using Naïve Bayes Method,” *Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control*, vol. 4, no. 2, 2023, doi: 10.22219/kinetik.v8i2.1686. [38]
- E. Y. Sari, A. D. Wierfi, and A. Setyanto, “Sentiment Analysis Of Customer Satisfaction On Transportation Network Company Using Naive Bayes Classifier,” *2019 International Conference on Computer Engineering, Network, and Intelligent Multimedia, CENIM 2019 - Proceeding*, vol. 2019-Novem, 2019, doi: 10.1109/CENIM48368.2019.8973262. [39]
- E. H. Mohamed and W. H. El-Behaidy, “An Ensemble Multi-label Themes-Based Classification For Holy Qur’an Verses Using Word2Vec Embedding,” *Arab J Sci Eng*, vol. 46, no. 4, pp. 3519–3529, 2021, doi: 10.1007/s13369-020-05184-0. [40]
- J. Hermanto, “Klasifikasi Teks Humor Bahasa Indonesia Memanfaatkan SVM,” *Journal of Information System, Graphics, Hospitality and Technology*, vol. 3, no. 01, pp. 39–48, 2021, doi: 10.37823/insight.v3i01.118. [41]
- L. Xiao, G. Wang, and Y. Zuo, “Research On Patent Text Classification Based On Word2Vec And LSTM,” *Proceedings - 2018 11th International Symposium on Computational Intelligence and Design, ISCID 2018*, vol. 1, pp. 71–74, 2018, doi: 10.1109/ISCID.2018.00023. [42]

- [33] A. Saifudin, “Metode Data Mining Untuk Seleksi Calon Mahasiswa,” vol. 10, no. 1, pp. 25–36, 2018, doi: <https://dx.doi.org/10.24853/jurtek.10.1.25-36>.
- [34] C. An, Y. W. Park, S. S. Ahn, K. Han, H. Kim, and S. K. Lee, “Radiomics Machine Learning Study With A Small Sample Size: Single Random Training-Test Set Split May Lead To Unreliable Results,” *PLoS One*, vol. 16, no. 8 August, Aug. 2021, doi: [10.1371/journal.pone.0256152](https://doi.org/10.1371/journal.pone.0256152).
- [35] A. M. Ertugrul and P. Karagoz, “Movie Genre Classification From Plot Summaries Using Bidirectional LSTM,” *Proceedings - 12th IEEE International Conference on Semantic Computing, ICSC 2018*, vol. 2018-Janua, pp. 248–251, 2018, doi: [10.1109/ICSC.2018.00043](https://doi.org/10.1109/ICSC.2018.00043).
- [36] A. R. Isnain, A. Sihabuddin, and Y. Suyanto, “Bidirectional Long Short Term Memory Method And Word2vec Extraction Approach For Hate Speech Detection,” *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, vol. 14, no. 2, p. 169, 2020, doi: [10.22146/ijccs.51743](https://doi.org/10.22146/ijccs.51743).

Halaman ini sengaja dikosongkan