

KOMPARASI METODE *K-NEAREST NEIGHBORS* DAN *LONG SHORT TERM MEMORY* PADA KLASIFIKASI TERJEMAHAN AL-QUR'AN

Nurul Fatiara¹, Nazruddin Safaat H², Surya Agustian³, Yusra⁴, Iis Afrianty⁵

^{1,2,3,4,5} Universitas Islam Negeri Sultan Syarif Kasim Riau

(Program Studi Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Sultan Syarif Kasim Riau)

(Jl. HR. Soebrantas KM. 15 No. 155, Simpang Baru, Kota Pekanbaru, 28293, Indonesia)

Email: ¹12050122431@students.uin-suska.ac.id, ²nazruddin.safaat@uin-suska.ac.id,

³surya.agustian@uin-suska.ac.id, ⁴yusra@uin-suska.ac.id, ⁵iis.afrianty@uin-suska.ac.id

Abstrak

Al-Qur'an merupakan kitab suci yang diturunkan untuk umat islam. Secara harfiah, Al-Qur'an berasal dari kata qara'a yang artinya membaca atau mengumpulkan. Namun untuk memahami terjemahan dari Al-Qur'an tidaklah mudah. Salah satu cara yang dapat dilakukan untuk memahami dan mempelajarinya adalah melakukan klasifikasi terhadap terjemahan ayat Al-Qur'an. Penelitian ini mengklasifikasikan terjemahan Al-Qur'an bahasa Indonesia ke enam kelas yang berbeda. Metode yang digunakan adalah K-Nearest Neighbor (KNN) dan Long Short Term Memory (LSTM) dan membandingkan kedua metode untuk mendapatkan hasil performa klasifikasi yang tertinggi. Hasil klasifikasi menunjukkan model LSTM menghasilkan performa klasifikasi yang lebih tinggi yaitu berupa rata-rata F1-Score sebesar 65% dan rata-rata accuracy 96% dibandingkan model KNN dengan nilai rata-rata F1-Score sebesar 55% dan rata-rata accuracy 93%.

Kata kunci: Accuracy, F1-Score, K-Nearest Neighbors, Long Short Term Memory, Terjemahan Al-Qur'an.

Abstract

The Qur'an is a holy book revealed to Muslims. Literally, the Qur'an comes from the word qara'a, which means to read or to gather. However, understanding the translation of the Qur'an is not easy. One way to comprehend and study it is by classifying the translation of Qur'anic verses. This research aims to classify the Indonesian translation of the Qur'an into six different topics. The methods used are K-Nearest Neighbor (KNN) and Long Short Term Memory (LSTM) classification and comparing both methods to obtain the highest classification performance results. The classification results show that the LSTM model achieved higher classification performance, with an average F1-Score of 65% and an average accuracy of 96%, compared to the KNN model with an average F1-Score of 55% and an average accuracy of 93%.

Keywords: Accuracy, F1-Score, K-Nearest Neighbors, Long Short Term Memory, Qur'an Translation.

1. PENDAHULUAN

Al-Qur'an sebagai kitab suci yang diwahyukan oleh Allah swt untuk umat islam. Al-Qur'an berasal dari kata qara'a yang artinya membaca atau mengumpulkan. Arti membaca atau mengumpulkan dalam konteks ini memiliki konsep yang serupa, membaca juga mengandung arti mengumpulkan, karena seseorang yang membaca akan mengumpulkan ide-ide atau pemikiran berdasarkan yang dibaca [1].

Dengan perkembangan teknologi yang pesat, saat ini banyak organisasi yang bergerak dalam bidang Al-Qur'an menghasilkan terjemahan Al-Qur'an dengan menambahkan klasifikasi ayat di bagian lampiran Al-Qur'an yang dicetak. Salah satu contoh adalah terjemahan Al-Qur'an yang membagi ayat-ayat Al-Qur'an kedalam 15 topik. Adapun pembagian topik-topiknya yaitu Arkanul Islam, Iman, Al-Qur'an, Ilmu dan Cabang-cabangnya, Amal, Dakwah, Jihad, Manusia dan Hubungan Kemasyarakatan, Akhlak, Peraturan yang Berhubungan dengan Harta, Hal-hal yang Berkaitan dengan Hukum, Negara dan Masyarakat, Pertanian dan Perdagangan, Sejarah dan Kisah-kisah, dan Agama-agama [2].

Namun untuk memahami terjemahan dari Al-Qur'an sangat kompleks. Ada beberapa cara yang dapat dilakukan untuk memahami dan mempelajarinya, salah satunya adalah melakukan klasifikasi terhadap terjemahan ayat Al-Qur'an kedalam topik-topik yang ada. Klasifikasi merupakan proses pengelompokan suatu objek kedalam kelas-kelas tertentu [3]. Klasifikasi terjemahan diperlukan untuk mempermudah umat Islam dalam mempelajari Al-Qur'an, baik secara tekstual maupun kontekstual dan juga untuk memberi pemahaman tentang pengetahuan yang diungkapkan pada isi Al-Qur'an kepada umat islam [4]. Dalam Al-Qur'an terdapat terjemahan yang mengutip dari ayat sebelumnya sehingga ayat-ayat tersebut saling berkaitan serta memiliki makna tersirat, yang mana setiap ayat tersebut memiliki makna yang berbeda dan ada juga yang membahas lebih dari satu topik [5]. Oleh karena itu, memberikan label pada terjemahan ayat Al-Qur'an memerlukan analisis yang khusus [6].

Salah satu metode yang diterapkan dalam klasifikasi adalah algoritma KNN. Algoritma ini dianggap efektif digunakan untuk klasifikasi teks dan merupakan metode klasifikasi yang simple, efektif dalam memproses data yang besar dan mudah diimplementasikan [5]. Saat ini, penggunaan *deep learning* juga semakin populer, salah satu alasannya karena dapat digunakan pada dataset yang besar dan kompleks. LSTM termasuk salah satu algoritma *deep learning* yang paling populer dalam klasifikasi teks karena baik digunakan untuk dataset dalam bahasa Inggris dan bahasa Indonesia seperti pada penelitian yang dilakukan oleh Kirana [7], Muhammad Ihsan [16] dan Alvi Hasanah [15].

Dalam sebuah penelitian yang dilakukan oleh Mohamed & El-Behaidy [8] menggunakan pendekatan klasifikasi *multilable ensemble* untuk mengidentifikasi dan mengklasifikasikan secara otomatis ayat-ayat pada Al-Qur'an berdasarkan topik dan tema yang terkandung pada ayat tersebut. Setelah melalui tahapan *preprocessing*, topiknya diberi label secara manual berdasarkan klasifikasi 'Mushaf Al-Tajweed'. Ayat-ayat tersebut kemudian diubah menjadi vektor menggunakan pendekatan TF-IDF dan Word2Vec. Hasilnya menunjukkan nilai *hamming loss* sebesar 0.224, *recall* 81%, *precision* 75%, dan *F1-Score* 77%.

Pada penelitian yang telah dilakukan oleh Abdullahi [9] dengan metode KNN, SVM, NB dan J48 dan penelitian Adeleke Adeleke et al. (2019) dengan metode SVM, NB dan J48 melakukan klasifikasi ayat Al-Qur'an yang membagi datanya kedalam tiga kelas yaitu iman, ibadah dan akhlak dengan menggunakan dataset Quranic Text.

Penelitian Sabri dan Alsayat [11] juga melakukan klasifikasi terhadap 6 surat pertama yang terdapat dalam Al-Qur'an menggunakan metode Support Vector Classifier (SVC) dan GaussianNB. Hasilnya menunjukkan akurasi terbaik secara keseluruhan adalah 80% untuk SVC dan 60% untuk Gaussian Naïve Bayes. Shahriar & Tariq [12] mengklasifikasikan delapan maqam (irama) populer dalam bacaan Al-Qur'an menggunakan algoritma *deep learning*. Berbagai fitur audio, termasuk koefisien cepstral mel-frequency, spektral, energi, dan fitur kroma, diperoleh untuk pelatihan model. Beberapa arsitektur *deep learning* termasuk CNN, LSTM, dan *deep ANN* dilatih untuk mengklasifikasikan pada penelitian ini. Sehingga ANN mendapatkan nilai akurasi tertinggi sebesar 95,7%.

Pada penelitian Naas [13] menggunakan algoritma KNN untuk melakukan klasifikasi pada penerima bantuan pada program rehabilitasi rumah tidak layak huni (RLTH) dengan 13 atribut data telah berhasil dilakukan. Hasil menunjukkan bahwa algoritma KNN dapat diterapkan untuk mengklasifikasikan kelayakan penerima rehabilitasi RTLH dengan menghasilkan akurasi tertinggi sebesar 97,93%. Dalam penelitian Putrisanni [5] dilakukan klasifikasi ayat Al-Qur'an terjemahan bahasa Inggris menggunakan *K-Nearest Neighbor* dan menggunakan *Information Gain* untuk melakukan seleksi fitur. Klasifikasi ayat Al-Qur'an terjemahan diklasifikasikan kedalam tiga kelas yang terdiri dari perintah, larangan dan lainnya. Hasil pengujian yang dilakukan dengan rasio komposisi dataset berupa data testing dan data *training* sebesar 1:9 mendapatkan nilai akurasi sebesar 64.10%, nilai presisi sebesar 63%, dan nilai *recall* sebesar 62.68% dengan menggunakan nilai $k=17$.

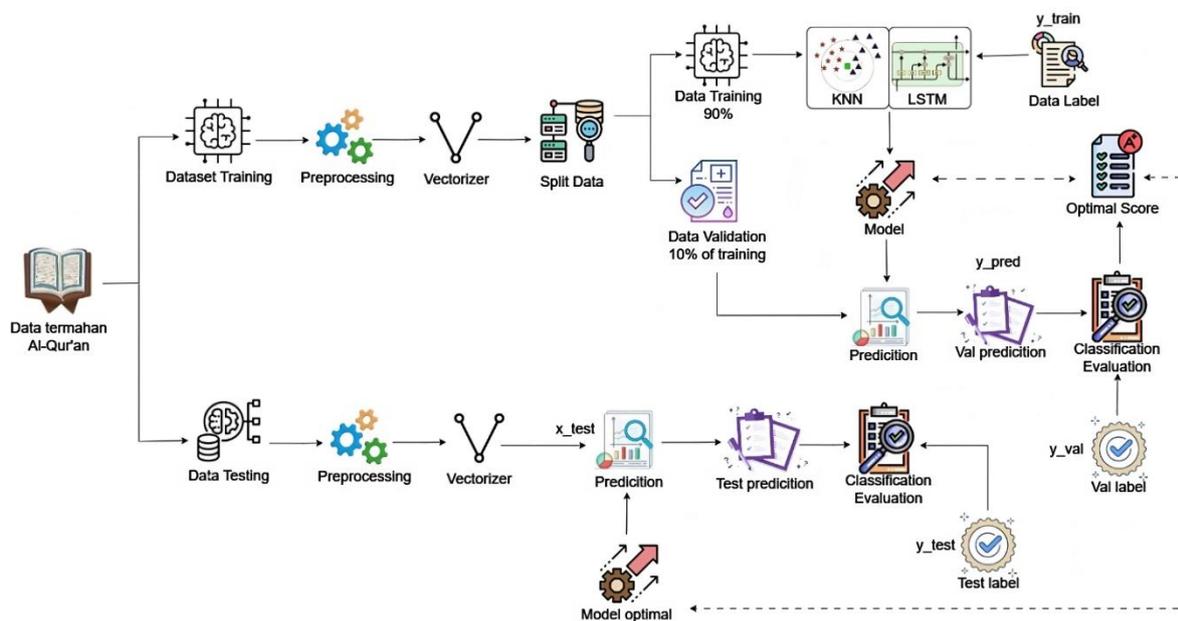
Penelitian Hanafi [14] mengklasifikasikan hadis bukhari terjemahan bahasa Indonesia menggunakan *K-Nearest Neighbor* dan mutual information sebagai metode seleksi fiturnya. Penelitian ini menggunakan pendekatan *Binary Relevance* (BR) untuk melakukan klasifikasi *multilable*. Hasil penelitian menunjukkan bahwa penggunaan mutual information sebagai metode seleksi fitur meningkatkan performansi klasifikasi, dengan nilai *hamming loss* sebesar 0,0886 atau 91,14% data terklasifikasi dengan benar dengan waktu komputasinya selama 595 detik.

Berdasarkan penelitian sebelumnya, klasifikasi teks *multi-class* pada ayat Al-Qur'an sudah dilakukan pada penelitian-penelitian terdahulu, dan juga telah dibuktikan bahwa algoritma KNN dan

LSTM menghasilkan hasil akurasi yang tinggi dalam melakukan klasifikasi dengan berbagai topik. Dari masalah yang telah dijelaskan, penulis akan mengklasifikasikan terjemahan Al-Qur'an bahasa Indonesia kedalam kelas-kelas yang sudah ditentukan. Adapun kelas-kelas yang akan digunakan pada penelitian ini hanya enam kelas yaitu hal-hal yang berkaitan dengan hukum, negara dan masyarakat, pertanian dan perdagangan, sejarah dan kisah-kisah, agama-agama, dan satu kelas untuk ayat yang tidak termasuk kedalam topik manapun yaitu *Undefined class*. Selain itu, penelitian ini akan melakukan optimasi terhadap performa klasifikasi, serta membandingkan hasil performa klasifikasi metode *K-Nearest Neighbor* dan metode *Long Short Term Memory* untuk mendapatkan hasil performa klasifikasi tertinggi diantara kedua metode tersebut dan mengetahui kemampuan metode dalam melakukan klasifikasi terjemahan Al-Qur'an.

2. METODE PENELITIAN

Metode penelitian ini menjelaskan tahapan-tahapan yang akan dilakukan untuk menyelesaikan penelitian seperti pada gambar berikut:



Gambar 1. Tahapan-tahapan penelitian

Pada gambar 1 dapat dilihat bahwa penelitian ini menggunakan terjemahan Al-Qur'an sebagai dataset yang dibagi kedalam data training dan data testing. Masing-masing data dilakukan proses *preprocessing*, proses *vectorizer* dan dilakukan juga proses *split data* pada data *training*. Selanjutnya dilakukan pembuatan model dengan metode *K-Nearest Neighbors* dan *Long Short Term Memory* kemudian dilakukan eksplorasi terhadap model tersebut untuk mendapatkan model optimal yang akan di ujikan pada data testing untuk mendapatkan hasil performa klasifikasi.

2.1. Dataset

Dataset yang digunakan pada penelitian ini adalah terjemahan Al-Quran Yang Mulia [17] berbahasa Indonesia. Setiap ayatnya sudah dilabeli oleh Ananda Pane dan Syahrul Mubarak [2]. Penelitian ini hanya akan menggunakan kelas yaitu hal-hal yang berkaitan dengan hukum, negara dan masyarakat, pertanian dan perdagangan, sejarah dan kisah-kisah, agama-agama, dan satu kelas untuk ayat yang tidak termasuk kedalam topik manapun yaitu *Undefined class*.

1. Data *training* berjumlah 1383 ayat yang terdiri atas terjemahan ayat dari surat ke-2 sampai surat ke-6 dan terjemahan ayat yang termasuk kedalam juz 29 pada Al-Qur'an.
2. Data *testing* yang digunakan untuk pengujian akhir yaitu sebanyak 16 juz dengan pembagian juz berdasarkan profil data ayat-ayat panjang (juz 10 sampai juz 20) dengan jumlah 2003 data, profil data ayat-ayat sedang (juz 25 sampai juz 28) dengan jumlah 977 data dan profil data ayat-ayat

pendek (juz 30) dengan jumlah 571 data dengan total keseluruhannya adalah 3551 data.

3. Data validasi diambil sebanyak 10% dari data *training* sehingga berjumlah 139 data.

2.2. Text Preprocessing

Text Preprocessing adalah proses mengolah dataset sesuai dengan yang diharapkan untuk mengambil informasi dari dataset tersebut. Proses ini digunakan untuk membersihkan dataset dari *noise* sehingga menghasilkan data yang sesuai untuk mempermudah pemrosesan data dan tidak mengganggu hasil [18]. Adapun *text preprocessing* yang akan dilakukan pada penelitian ini adalah sebagai berikut:

1. *Case Folding*, adalah tahapan mengubah semua karakter pada teks yang ada menjadi huruf kecil (*lowercase*).
2. *Tokenizing*, adalah tahapan untuk memecah kalimat menjadi kata atau token.
3. *Punctuation Removal*, adalah tahapan menghapus tanda baca pada suatu teks/dokumen seperti tanda titik (.), koma (,), tanda tanya (?), tanda seru (!) karena tanda baca ini tidak berpengaruh terhadap hasil klasifikasi. Dalam konteks terjemahan ayat Al-Qur'an, penghapusan tanda baca dapat membantu model fokus pada makna kata atau frasa tanpa terpengaruh oleh tanda baca.
4. *Stopword Removal*, adalah tahapan penghapusan fitur-fitur umum suatu dokumen yang tidak memiliki informasi pada kalimat seperti kata penghubung dan kata keterangan seperti sebuah, oleh, pada, di, dan sebagainya.
5. *Stemming*, adalah tahapan yang dilakukan untuk menghapus awalan, akhiran, sisipan dan konfiks untuk mendapatkan kata dasar dari sebuah token. Tujuannya adalah untuk mengurangi jumlah kata yang akan diproses.

2.3. Vectorizer

Selanjutnya dilakukan proses pembentukan fitur atau *feature construction* yaitu proses membuat variabel baru dari dataset [19]. Dalam penelitian ini, *TF.IDF vectorizer* digunakan untuk mengkonversi teks menjadi representasi vektor BoW. *Bag of Words* (BoW) merupakan salah satu metode yang paling sederhana dari *feature extraction*, merupakan model yang merepresentasikan teks berbentuk vektor kata unik (*vocabulary*) pada koleksi dokumen atau dataset [20].

Representasi vektor BoW tersebut menggunakan skema *TF.IDF*. *TF.IDF vectorizer* adalah salah satu *library* dengan *python* yang mengimplementasikan konsep *TF.IDF*. *TF.IDF vectorizer* merupakan kombinasi dari teknik *count vectorizer* dengan teknik *TF.IDF* [21]. *Vectorizer* digunakan untuk menghasilkan representasi vektor BoW dari teks yang telah diproses pada tahapan *preprocessing*. Vektor BoW ini kemudian dapat digunakan sebagai fitur input untuk model klasifikasi KNN atau LSTM.

2.4. Klasifikasi *multi-class*

Klasifikasi adalah proses pengelompokan atau pengategorian suatu data ke dalam kelas atau kategori tertentu berdasarkan karakteristiknya. Klasifikasi *multi-class* atau *multilabel* adalah jenis klasifikasi di mana setiap sampel data dapat diklasifikasikan ke dalam lebih dari satu kategori atau label [8]. Dalam konteks klasifikasi ayat-ayat Al-Qur'an, klasifikasi multilabel digunakan karena sebuah ayat dalam Al-Qur'an sering kali membahas lebih dari satu topik secara bersamaan. Oleh karena itu, klasifikasi *multi class* digunakan untuk secara akurat mengklasifikasikan ayat Al-Qur'an ke dalam topik-topik yang sudah ditentukan [22].

1. KNN

KNN atau *K-Nearest Neighbors* adalah metode klasifikasi yang mengklasifikasikan objek berdasarkan data latih yang memiliki jarak yang paling dekat dengan objek tersebut. Jarak yang paling dekat tersebut ditentukan dengan nilai *k*. Jarak nilai *k* minimal lebih dari satu dan tidak boleh melebihi data latih [23]. Cara kerja algoritma ini yaitu dapat mengklasifikasikan data training berdasarkan nilai *k* dari jarak data terdekat.

2. LSTM

LSTM atau *Long Short Term Memory* adalah salah satu metode dari *deep learning* yang merupakan variasi dari RNN. Penambahan yang dilakukan adalah memori sel untuk menyimpan informasi dalam jangka panjang, yang terbukti bekerja lebih akurat daripada RNN [24]. Algoritma ini terdiri dari unit-unit yang mencakup *cell state*, *gate units*, dan *output gate*. *Cell state* menyimpan informasi

yang akan diteruskan ke langkah berikutnya. *Gate units* memproses informasi yang akan disimpan atau dihapus. Di dalam unit ini, terdapat *input gate* dan *forget gate*. *Input gate* bertanggung jawab untuk menentukan nilai masukan yang akan disalurkan ke *cell state*. Selanjutnya, *input gate* digunakan untuk mengeluarkan informasi dari *cell state*. Pada langkah terakhir, gerbang keluaran menentukan nilai masukan yang dihasilkan [7].

2.5. Training data dan Validasi

Setelah melakukan tahapan *text preprocessing* dan *vectorizer*, maka dilakukanlah tahapan *training* terhadap 1383 ayat yaitu surat ke 2-6 dan juz 29 yang sudah ditentukan sebagai data *training*. Kemudian, melakukan penanganan terhadap data yang tidak seimbang yaitu melakukan pendekatan *over sampling* (menambah data) sebagai bagian dari optimasi. Selanjutnya melakukan validasi untuk menentukan model ML yang paling optimal dari proses *training*. Model dari beberapa variasi *training* diujikan pada data validasi, yang diambil sebanyak 10% dari data *training*. Selanjutnya menemukan/menentukan model yang paling optimal misalnya berdasarkan hasil dari *F1-Score*. Model optimal ditentukan dari hasil *F1-Score* yang paling tinggi.

2.6. Testing dan Evaluasi

Pada tahapan ini, pengujian dilakukan terhadap data *testing* yaitu sebanyak 16 juz dengan pembagian juz berdasarkan profil data ayat-ayat panjang (juz 10 sampai juz 20) dengan jumlah 2003 data, profil data ayat-ayat sedang (juz 25 sampai juz 28) dengan jumlah 977 data dan profil data ayat-ayat pendek (juz 30) dengan jumlah 571 data dengan total keseluruhannya adalah 3551 data.

Setelah pengujian, dilakukan evaluasi performa klasifikasi yang dilihat berdasarkan hasil dari *F1-Score* (1) dan *Accuracy* (2).

$$F1-Score = \frac{2 \times \text{recall} \times \text{precision}}{\text{recall} + \text{precision}} \quad (1)$$

$$Accuracy = \frac{\sum_{i=1}^l \frac{TP_i + TN_i}{TP_i + TN_i + FP_i + FN_i}}{l} \times 100\% \quad (2)$$

Keterangan:

- TP = True Positif
- TN = True Negatif
- FP = False Positif
- FN = False Negatif

3. HASIL DAN PEMBAHASAN

Pada penelitian ini, pengujian dilakukan dengan 2 skenario pada kedua metode untuk mendapatkan perbandingan hasil performa klasifikasi dari *baseline* dan hasil dari proses optimasi (optimal), yaitu:

1. Skenario pertama yaitu pada *baseline* tidak melakukan *balancing* data dan tidak mencari parameter terbaik atau *best parameter* pada KNN dan tidak melakukan penyesuaian *parameter tune* pada LSTM.
2. Skenario kedua yaitu melakukan proses optimasi seperti *balancing* data pada data *training* dengan teknik *oversampling* (menambah data minoritas agar sama dengan data mayoritas) [25], mencari parameter terbaik atau *best parameter* pada KNN dan melakukan penyesuaian *parameter tune* pada LSTM.

Pengujian dilakukan terhadap 3 profil data *testing* yang berbeda yaitu:

1. Profil data ayat-ayat panjang (juz 10 sampai juz 20) dengan jumlah 2003 data.
2. Profil data ayat-ayat sedang (juz 25 sampai juz 28) dengan jumlah 977 data.
3. Profil data ayat-ayat pendek (juz 30) dengan jumlah 571 data.

3.1. Hasil Pengujian Model *Baseline* KNN

Dari pengujian yang telah dilakukan, didapatkan hasil *baseline* pada klasifikasi terjemahan Al-Qur'an bahasa Indonesia dengan menggunakan metode KNN berupa performa klasifikasi yaitu *F1-Score* dan *accuracy* untuk hasil training dan validasi seperti pada Tabel 1.

Tabel 1. Hasil model *baseline* KNN dari proses *training*

Class	Optimization Preprocessing	Balancing	Parameter Tuning	Training		Validation	
				F1-Score	Accuracy	F1-Score	Accuracy
Hal-hal yang berkaitan dengan hukum	Without stemming and stopword	No	K = 5 (Default)	68%	95%	70%	94%
Negara dan Masyarakat	-	No	K = 5 (Default)	60%	99%	50%	98%
Pertanian dan Perdagangan	-	No	K = 5 (Default)	50%	99%	49%	98%
Sejarah dan kisah-kisah	-	No	K = 5 (Default)	65%	96%	66%	97%
Agama-agama	Without stemming and stopword	No	K = 5 (Default)	81%	93%	72%	94%
Undefined class	Without stemming and stopword	No	K = 5 (Default)	64%	96%	56%	93%

Pada penelitian ini, pengujian pada proses *training* dijalankan dengan melakukan rangkaian eksperimen bagian *preprocessing* dan tidak melakukan *balancing* pada data *training*. Untuk nilai K semua kelas hanya menggunakan nilai *default* yang disediakan oleh *library* yaitu K = 5. Hasil *training* dan validasi dari *baseline* algoritma KNN sangat bervariasi antar kelasnya seperti pada Tabel 1. Kemudian, setelah dilakukan proses *training* pada model *baseline* algoritma KKN ini, selanjutnya dilakukan pengujian terhadap data *testing* yaitu juz 10-20, juz 25-28 dan juz 30 dan menghasilkan performa klasifikasi seperti pada tabel 2.

Tabel 2. Hasil pengujian model *baseline* KNN pada data *testing*

Class	Data Test Juz 10-20		Data Test Juz 25-28		Data Test Juz 30	
	F1-Score	Accuracy	F1-Score	Accuracy	F1-Score	Accuracy
Hal-hal yang berkaitan dengan hukum	53%	97%	60%	98%	50%	99%
Negara dan Masyarakat	50%	99%	50%	99%	100%	100%
Pertanian dan Perdagangan	50%	100%	50%	99%	50%	99%
Sejarah dan kisah-kisah	50%	85%	53%	93%	58%	95%
Agama-agama	57%	96%	57%	94%	60%	97%
Undefined class	47%	81%	48%	91%	47%	89%
Rata-rata	51%	93%	53%	96%	61%	97%

Pada tabel 2 dapat dilihat bahwa hasil dari performa klasifikasi pada proses *testing* menggunakan algoritma KNN ini cukup bervariasi terhadap ketiga data *testing* yang berbeda. Secara rata-rata, model ini menunjukkan *F1-Score* 51% dengan akurasi 93% pada Juz 10-20, meningkat menjadi 53% *F1-Score* dengan 96% akurasi pada Juz 25-28, dan mencapai puncak performa dengan 61% *F1-Score* dan 97% akurasi pada Juz 30. Hal ini menandakan bahwa kinerja model KNN memiliki

variasi yang signifikan berdasarkan kategori dan segmentasi juz yang diuji, dengan kecenderungan peningkatan performa pada juz yang lebih tinggi.

3.2. Hasil Pengujian Model Optimal KNN

Setelah melakukan pengujian pada model *baseline*, selanjutnya melakukan proses optimasi pada proses *training* dengan teknik optimasi yang beragam seperti melakukan *balancing* pada data *training* dengan teknik *oversampling*, melakukan beberapa optimasi pada bagian *preprocessing* dan mencari nilai K yang terbaik. Kemudian, menentukan model optimal berdasarkan hasil *F1-Score* yang paling tinggi. Dari pengujian-pengujian yang telah dilakukan, didapatkan model optimal dari proses *training* seperti pada tabel 3.

Tabel 3. Hasil model optimal KNN dari proses *training*

Class	Optimization Preprocessing	Balancing	Parameter Tuning	Training		Validation	
				F1-Score	Accuracy	F1-Score	Accuracy
Hal-hal yang berkaitan dengan hukum	Without stemming and stopword	No	K = 5 (Default)	68%	95%	70%	94%
Negara dan Masyarakat	-	No	K = 5 (Default)	60%	99%	50%	98%
Pertanian dan Perdagangan	-	Yes	K = 20	98%	98%	67%	95%
Sejarah dan kisah-kisah	-	Yes	K = 5	98%	98%	56%	93%
Agama-agama	Without stemming and stopword	No	K = 5 (Default)	81%	93%	72%	94%
Undefined class	-	Yes	K = 5	98%	98%	60%	88%

Pada tabel 3 dapat dilihat bahwa melakukan optimasi pada model cukup memiliki pengaruh yang signifikan terhadap hasil *F1-Score* dan akurasi model KNN pada performa klasifikasi. Optimasi pada bagian *preprocessing*, menggunakan nilai K yang terbaik dan melakukan *balancing* pada data dapat meningkatkan performa klasifikasi untuk mencari model optimal. Namun, pada beberapa kelas proses optimasi tidak dapat meningkatkan hasil performa klasifikasi, sehingga hasil dari model *baseline* memiliki nilai performa klasifikasi yang lebih tinggi dibandingkan dari hasil yang dilakukan proses optimasi pada modelnya. Oleh karena itu, hasil dari proses *testing* model optimal dapat dilihat pada tabel 4.

Tabel 4. Hasil pengujian model optimal KNN pada data *testing*

Class	Data Test Juz 10-20		Data Test Juz 25-28		Data Test Juz 30	
	F1-Score	Accuracy	F1-Score	Accuracy	F1-Score	Accuracy
Hal-hal yang berkaitan dengan hukum	53%	97%	60%	98%	50%	99%
Negara dan Masyarakat	50%	99%	50%	99%	100%	100%
Pertanian dan Perdagangan	51%	97%	55%	95%	56%	96%
Sejarah dan kisah-kisah	56%	80%	60%	86%	60%	89%
Agama-agama	57%	96%	57%	94%	60%	97%
Undefined class	54%	77%	54%	86%	53%	80%
Rata-rata	54%	91%	56%	93%	63%	94%

Pada tabel 4 dapat dilihat bahwa hasil dari performa klasifikasi pada proses *testing* model optimal menggunakan algoritma KNN ini mengalami peningkatan dari model *baseline* dengan rata-rata *F1-Score* dari 51% menjadi 54% pada juz 10-20, dari 53% menjadi 56% pada juz 25-28 dan dari 61% menjadi 6% pada juz 30 yang tidak mengalami peningkatan yang terlalu signifikan antara model *baseline* dan model optimalnya. Oleh karena itu, proses optimasi pada proses *training* dengan teknik optimasi yang beragam seperti melakukan *balancing* pada data *training*, melakukan beberapa optimasi

pada bagian *preprocessing* dan mencari nilai K yang terbaik dapat dikatakan membantu meningkatkan performa model dalam melakukan klasifikasi menggunakan algoritma KNN.

3.3. Hasil Pengujian Model *Baseline* LSTM

Dari pengujian yang telah dilakukan, didapatkan hasil *baseline* pada klasifikasi terjemahan Al-Qur'an bahasa Indonesia dengan menggunakan metode LSTM berupa performa klasifikasi yaitu *F1-Score* dan *accuracy* untuk hasil training dan validasi seperti pada Tabel 5.

Tabel 5. Hasil model *baseline* LSTM dari proses *training*

Class	Optimization Preprocessing	Balancing	Parameter Tuning	Training		Validation	
				F1-Score	Accuracy	F1-Score	Accuracy
Hal-hal yang berkaitan dengan hukum	-	No	Dropout = 0,2, Epoch = 7, Batch Size = 64, Dense Layer = 286	99%	100%	62%	93%
Negara dan Masyarakat	-	No	Dropout = 0,2, Epoch = 6, Batch Size = 64, Dense Layer = 286	99%	100%	100%	100%
Pertanian dan Perdagangan	-	No	Dropout = 0,2, Epoch = 1, Batch Size = 64, Dense Layer = 286	50%	99%	49%	98%
Sejarah dan kisah-kisah	-	No	Dropout = 0,2, Epoch = 1, Batch Size = 64, Dense Layer = 286	49%	95%	49%	97%
Agama-agama	Without stemming and stopword	No	Dropout = 0,2, Epoch = 3, Batch Size = 64, Dense Layer = 286	72%	91%	68%	94%
Undefined class	-	No	Dropout = 0,2, Epoch = 1, Batch Size = 64, Dense Layer = 286	49%	95%	48%	92%

Pada penelitian ini, pengujian pada proses training dijalankan dengan melakukan rangkaian eksperimen bagian preprocessing dan tidak melakukan balancing pada data training. Untuk nilai parameter tune pada semua kelas hanya menggunakan nilai awal pada model LSTM. Hasil training dan validasi dari *baseline* algoritma LSTM ini sangat bervariasi antar kelasnya seperti pada Tabel 5. Kemudian, setelah dilakukan proses training pada model *baseline* algoritma LSTM ini, selanjutnya dilakukan pengujian terhadap data testing yaitu juz 10-20, juz 25-28 dan juz 30 dan menghasilkan performa klasifikasi seperti pada tabel 6.

Tabel 6. Hasil pengujian model *baseline* LSTM pada data *testing*

Class	Data Test Juz 10-20		Data Test Juz 25-28		Data Test Juz 30	
	F1-Score	Accuracy	F1-Score	Accuracy	F1-Score	Accuracy
Hal-hal yang berkaitan dengan hukum	56%	97%	55%	97%	50%	100%
Negara dan Masyarakat	50%	99%	50%	99%	100%	100%
Pertanian dan Perdagangan	50%	100%	50%	99%	50%	99%
Sejarah dan kisah-kisah	46%	85%	48%	92%	49%	95%
Agama-agama	50%	98%	58%	97%	50%	98%
Undefined class	45%	81%	48%	92%	47%	89%
Rata-rata	50%	93%	52%	96%	58%	97%

Pada tabel 6 dapat dilihat bahwa hasil dari performa klasifikasi pada proses *testing* menggunakan algoritma LSTM ini cukup bervariasi terhadap ketiga data *testing* yang berbeda. Secara rata-rata, model ini menunjukkan *F1-Score* 50% dengan akurasi 93% pada Juz 10-20, meningkat menjadi *F1-Score* 52% dengan akurasi 96% pada Juz 25-28, dan mencapai puncak performa dengan *F1-Score* 61 % dan akurasi 97% pada Juz 30. Ini menandakan bahwa model memiliki kapasitas untuk

beradaptasi dan meningkatkan kinerjanya seiring dengan kompleksitas dan variasi data yang bertambah pada juz yang lebih tinggi.

3.4. Hasil Pengujian Model Optimal LSTM

Setelah melakukan pengujian pada model *baseline*, selanjutnya melakukan proses optimasi model LSTM pada proses *training* dengan teknik optimasi yang beragam seperti melakukan *balancing* pada data *training* dengan teknik *oversampling*, melakukan beberapa optimasi pada bagian *preprocessing* dan melakukan penyesuaian terhadap *parameter tune* seperti menambah atau mengurangi nilai *dropout*, *batch size* dan *dense layer*. Kemudian, menentukan model optimal berdasarkan hasil *F1-Score* yang paling tinggi. Dari pengujian-pengujian yang telah dilakukan, didapatkan model optimal dari proses *training* seperti pada tabel 7.

Tabel 7. Hasil model optimal LSTM dari proses *training*

Class	Optimization Preprocessing	Balancing	Parameter Tuning	Training		Validation	
				F1-Score	Accuracy	F1-Score	Accuracy
Hal-hal yang berkaitan dengan hukum	-	Yes	Dropout = 0.5, Epoch = 5, Batch Size = 256, Dense Layer = 350	98%	98%	67%	93%
Negara dan Masyarakat	-	Yes	Dropout = 0.2, Epoch = 3, Batch Size = 64, Dense Layer = 286	100%	100%	100%	100%
Pertanian dan Perdagangan	-	Yes	Dropout = 0.5, Epoch = 1, Batch Size = 32, Dense Layer = 286	100%	100%	49%	98%
Sejarah dan kisah-kisah	-	Yes	Dropout = 0.5, Epoch = 5, Batch Size = 256, Dense Layer = 350	99%	99%	78%	98%
Agama-agama	Without stemming and stopword	Yes	Dropout = 0.5, Epoch = 7, Batch Size = 256, Dense Layer = 350	99%	99%	82%	96%
Undefined class	-	Yes	Dropout = 0.5, Epoch = 8, Batch Size = 256, Dense Layer = 350	99%	99%	59%	91%

Setelah mendapatkan model yang optimal seperti pada Tabel 7 maka dilakukan lah proses *testing* untuk mengevaluasi performanya pada data testing. Pengujian ini melibatkan berbagai segmentasi juz Al-Qur'an yaitu Juz 10-20, Juz 25-28, dan Juz 30. Berikut adalah hasil pengujian model optimal LSTM yang dapat dilihat pada Tabel 8.

Tabel 8. Hasil pengujian model optimal LSTM pada data *testing*

Class	Data Test Juz 10-20		Data Test Juz 25-28		Data Test Juz 30	
	F1-Score	Accuracy	F1-Score	Accuracy	F1-Score	Accuracy
Hal-hal yang berkaitan dengan hukum	57%	96%	58%	96%	50%	99%
Negara dan Masyarakat	50%	99%	50%	99%	100%	100%
Pertanian dan Perdagangan	50%	100%	50%	99%	50%	99%
Sejarah dan kisah-kisah	62%	85%	68%	92%	60%	93%
Agama-agama	56%	98%	63%	96%	73%	98%
Undefined class	54%	81%	54%	89%	54%	87%
Rata-rata	55%	93%	57%	95%	65%	96%

Berdasarkan hasil pada tabel 8 dapat dilihat bahwa hasil dari performa klasifikasi pada proses testing menggunakan algoritma LSTM ini cukup bervariasi saat diujikan pada data testing yang berbeda. Setelah dilakukan optimasi performa klasifikasi mengalami peningkatan dari model *baseline*

dengan rata-rata *F1-Score* dari 50% menjadi 55% pada juz 10-20, dari 52% menjadi 58% pada juz 25-28 dan dari 58% menjadi 65% pada juz 30. Oleh karena itu, proses optimasi pada proses training dengan teknik optimasi terkhusus pada penggunaan parameter tune algoritma LSTM dapat membantu meningkatkan performa model dalam melakukan klasifikasi terjemahan Al-Qur'an menggunakan algoritma LSTM.

3.5. Perbandingan Performa Klasifikasi

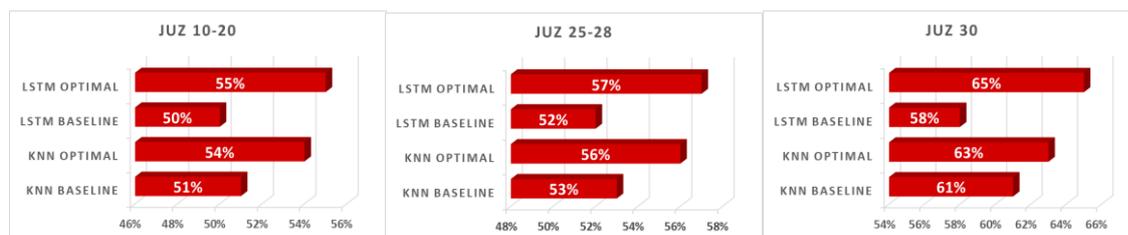
Berdasarkan hasil pengujian yang sudah dilakukan, didapatkan hasil rata-rata dari *F1-Score* pada setiap model *baseline* KNN dan model optimal KNN serta model *baseline* LSTM dan model optimal LSTM yang dapat dibuat perbandingan hasil performa klasifikasi seperti pada tabel 9.

Tabel 9. Perbandingan hasil performa klasifikasi berdasarkan jumlah rata-rata

Model	Data Test Juz 10-20		Data Test Juz 25-28		Data Test Juz 30	
	<i>F1-Score</i>	Accuracy	<i>F1-Score</i>	Accuracy	<i>F1-Score</i>	Accuracy
KNN <i>Baseline</i>	51%	93%	53%	96%	61%	97%
KNN Optimal	54%	91%	56%	93%	63%	94%
LSTM <i>Baseline</i>	50%	93%	52%	96%	58%	97%
LSTM Optimal	55%	93%	57%	95%	65%	96%

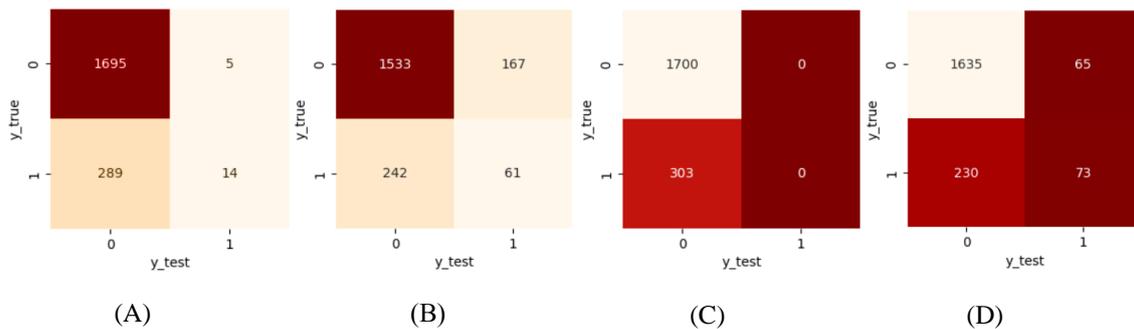
Dalam analisis perbandingan performa klasifikasi antara model KNN dan model LSTM, baik dalam versi *baseline* maupun optimal, terlihat peningkatan signifikan terhadap rata-rata *F1-Score* terhadap pengujian pada data Juz 10-20, Juz 25-28, dan Juz 30. Dari *baseline* ke optimal, model KNN menunjukkan peningkatan total *F1-Score* sebesar 3% pada Juz 10-20, 3% pada Juz 25-28, dan 2% pada Juz 30, menandakan bahwa optimasi berhasil meningkatkan kemampuan klasifikasi model, khususnya pada data Juz 10-20 dan juz 25-28.

Sementara itu, LSTM optimal menunjukkan performa klasifikasi terbaik di antara semua versi model, transisi dari LSTM *baseline* ke LSTM optimal mengalami peningkatan yang lebih signifikan, dengan total peningkatan *F1-Score* sebesar 5% pada Juz 10-20, 5% pada Juz 25-28, dan 7% pada Juz 30. Hal ini menunjukkan bahwa optimasi pada model LSTM sangat efektif dalam meningkatkan sensitivitas dan akurasi klasifikasi, terutama pada Juz 30 yang mencerminkan kompleksitas data yang lebih tinggi. Perbandingan hasil rata-rata *F1-Score* dari model-model ini dapat digambarkan dalam bentuk diagram seperti pada Gambar 2.



Gambar 2. Diagram perbandingan hasil performa klasifikasi berdasarkan rata-rata *F1-Score*

Secara keseluruhan, perbandingan ini menegaskan bahwa proses optimasi model sangat berpengaruh dalam meningkatkan performa klasifikasi teks. Optimasi tidak hanya berkontribusi pada peningkatan kemampuan model untuk mengidentifikasi dan mengklasifikasikan data secara akurat tetapi juga menunjukkan bahwa model KNN dan LSTM setelah dioptimalkan, memiliki kapasitas yang lebih baik dalam menangani variasi dan kompleksitas data. Confusion matrix dapat memberikan visualisasi terhadap kinerja model klasifikasi. Dengan membandingkan confusion matrix antara model *baseline* dan model optimal, dapat dilihat perubahan dalam jumlah *True Positives* (TP), *True Negatives* (TN), *False Positives* (FP), dan *False Negatives* (FN).



Gambar 3. Confusion matrix hasil pengujian juz 10-20 kelas Sejarah dan Kisah-kisah (A)baseline KNN (B)optimal KNN (C)baseline LSTM (D)optimal LSTM

Pada Gambar 3 KNN Baseline (A) memiliki kemampuan yang cukup tinggi dalam mengidentifikasi kasus positif dengan jumlah True Positives (TP) yang signifikan sebesar 1695, namun model ini menunjukkan keterbatasan dalam mengenali kasus negatif, dengan True Negatives (TN) hanya berjumlah 14. Hal ini mengindikasikan bahwa model cenderung mengklasifikasikan kasus sebagai positif, yang juga terlihat dari jumlah False Negatives (FN) yang relatif tinggi 289. KNN Optimal (B) menunjukkan perbaikan dalam mengenali kasus negatif dengan TN meningkat menjadi 61. Selanjutnya, LSTM Baseline (C) menunjukkan kinerja yang seimbang dengan jumlah TP yang tinggi sebesar 1700. LSTM Optimal (D) menampilkan kemajuan lebih lanjut dengan meningkatkan jumlah TN dari 0 menjadi 73 dan mengurangi FN menjadi 230, yang menandakan kemampuan yang lebih baik dalam mengenali kasus negatif.

Tabel 10. Kesalahan klasifikasi pada data test juz 10-20 kelas Sejarah dan kisah-kisah dengan model KNN Optimal

Terjemahan	Data Label	Hasil prediksi
Sungguh, pada yang demikian itu benar-benar terdapat tanda (kekuasaan Allah) bagi orang yang beriman (QS. Al-Hijr : 77)	0	1
Dan sesungguhnya penduduk Aikah itu benar-benar kaum yang zalim, (QS. Al-Hijr : 78)	1	0

Tabel 11. Kesalahan klasifikasi pada data test juz 10-20 kelas Sejarah dan kisah-kisah dengan model LSTM Optimal

Terjemahan	Data Label	Hasil prediksi
Maka ketika kedua golongan itu saling melihat, berkatalah pengikut-pengikut Musa, “Kita benar-benar akan tersusul.” (QS. Asy-Su’ara : 61)	1	0
Dia (Musa) menjawab, “Sekali-kali tidak akan (tersusul); sesungguhnya Tuhanku bersamaku, Dia akan memberi petunjuk kepadaku.” (QS. Asy-Su’ara : 62)	0	1

Dari hasil analisis pada data uji juz 10-20 kelas Sejarah dan kisah-kisah, terlihat bahwa kedua model, baik KNN maupun LSTM, mengalami kesalahan dalam klasifikasi. KNN pada tabel 10 menunjukkan kecenderungan mengklasifikasikan kasus positif dengan baik (True Positives), namun mengalami kesulitan dalam mengenali kasus negatif (False Negatives). Setelah proses optimasi, model KNN Optimal berhasil meningkatkan kemampuan mengenali kasus negatif, meskipun masih terdapat beberapa kesalahan klasifikasi. Sementara itu, LSTM pada tabel 11 menunjukkan performa yang cukup baik dalam mengenali kasus positif dan negatif, namun masih terdapat kesalahan yang signifikan terutama dalam mengenali kasus negatif.

Kesalahan klasifikasi pada tabel tersebut dapat disebabkan oleh beberapa faktor. Salah satunya adalah kompleksitas makna dan konteks teks dalam bahasa yang diterjemahkan, yang kadang-kadang sulit diinterpretasi secara tepat oleh model. Selain itu, penggunaan teknik *preprocessing* dan fitur konstruksi yang kurang optimal juga dapat memengaruhi akurasi klasifikasi. Misalnya, jika proses

tokenisasi tidak dilakukan dengan baik atau jika informasi penting dihilangkan oleh penghapusan stopword yang tidak sesuai, hal ini bisa mengakibatkan representasi teks yang kurang akurat untuk diproses oleh model klasifikasi.

Tabel 12. Hasil pengujian juz 10-20 kelas sejarah dan kisah-kisah

Model	F1-Score	Accuracy
KNN Baseline	50%	85%
KNN Optimal	56%	80%
LSTM Baseline	46%	85%
LSTM Optimal	62%	85%

Perbandingan antara model-model ini menunjukkan dengan dilakukannya proses optimasi *preprocessing* dan pemilihan parameter dapat meningkatkan kinerja model klasifikasi. Optimasi berhasil memperbaiki keterbatasan model *baseline*, terutama dalam mengenali kasus negatif dan mengurangi kesalahan klasifikasi. Seperti pada tabel 12, perbedaan antara KNN *Baseline* 50% dan KNN *Optimal* 56% pada pengujian juz 10-20 kelas Sejarah dan kisah-kisah menunjukkan peningkatan yang signifikan setelah optimasi dilakukan. Namun, dapat dilihat bahwa model LSTM *Optimal* menghasilkan nilai 62%, sehingga LSTM dianggap sebagai model dengan kinerja terbaik dibandingkan dengan model KNN.

4. KESIMPULAN

Berdasarkan penelitian yang sudah dilakukan, dapat disimpulkan:

1. Penelitian ini berhasil mengklasifikasikan terjemahan Al-Qur'an ke dalam enam topik yang ditentukan dengan menggunakan metode KNN dan LSTM.
2. Teknik optimasi seperti *balancing* data, peningkatan *preprocessing*, dan penyesuaian parameter dapat membantu meningkatkan kinerja model baik untuk model KNN maupun model LSTM.
3. Setelah dilakukan optimasi, model LSTM menghasilkan performa klasifikasi yang lebih tinggi dibandingkan model KNN. Oleh karena itu, metode LSTM menunjukkan kemampuan yang baik dalam melakukan klasifikasi teks terjemahan Al-Qur'an dibandingkan dengan metode KNN.

Oleh karena itu, untuk pengembangan penelitian selanjutnya, peneliti memberikan saran atau rekomendasi untuk melakukan eksplorasi atau eksperimen terhadap metode *machine learning* dan *deep learning* yang lainnya yang mungkin menawarkan kinerja yang lebih baik dalam melakukan klasifikasi terjemahan Al-Qur'an. Selain itu, variasi dataset juga dapat dilakukan dengan menggunakan terjemahan Al-Qur'an dengan berbagai bahasa untuk memperluas pengetahuan dalam melakukan penelitian klasifikasi terjemahan ayat Al-Qur'an ini.

DAFTAR PUSTAKA

- [1] Kuswoyo, *Pengantar Studi Ilmu-Ilmu Al-Qur'an*. Bojong Pekalongan: PT. Nasya Expanding Management, 2021.
- [2] R. Ananda Pane and M. Syahrul Mubarak, "Klasifikasi Multi-Label Pada Topik Ayat Al-Quran Terjemahan Bahasa Inggris Menggunakan Multinomial Naive Bayes," *e-Proceeding of Engineering*, vol. 5, no. 1, pp. 1551–1555, 2018.
- [3] M. R. Choirulfikri, K. M. Lhaksamana, and S. Al Faraby, "A Multi-Label Classification of Al-Quran Verses Using Ensemble Method and Naive Bayes," *Building of Informatics, Technology and Science (BITS)*, vol. 3, no. 4, pp. 473–479, Mar. 2022, doi: 10.47065/bits.v3i4.1287.
- [4] B. Arkok and A. M. Zeki, "Classification of Qur'anic topics based on imbalanced classification," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 22, no. 2, pp. 678–687, Apr. 2021, doi: 10.11591/ijeecs.v22.i2.pp678-687.
- [5] T. H. Putrisanni, A. Adiwijaya, and S. Al Faraby, "Klasifikasi Ayat Al-Quran Terjemahan Bahasa Inggris Menggunakan K-Nearest Neighbor (Knn) Dan Information Gain," *KOMIK (Konferensi Nasional Teknologi Informasi dan Komputer)*, vol. 3, no. 1, pp. 362–369, Nov. 2019, doi: 10.30865/komik.v3i1.1614.

- [6] M. Fauzan, H. Junaedi, and E. Setyati, "Klasifikasi Al-Qur'an Terjemahan Bahasa Indonesia Dengan Menggunakan Algoritma Support Vector Machine," *KONVERGENSI*, vol. 18, no. 2, pp. 42–49, 2022, doi: <https://doi.org/10.30996/konv.v18i1.6912>.
- [7] C. P. Kirana, S. P. Fachri, and R. F. S. Nuraini, "Meningkatkan Akurasi Long-Short Term Memory (LSTM) pada Analisis Sentimen Vaksin Covid-19 di Twitter dengan Glove," *Jurnal Telematika*, vol. 16, no. 2, pp. 85–90, 2021, doi: [10.61769/telematika.v16i2.400](https://doi.org/10.61769/telematika.v16i2.400).
- [8] 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, Apr. 2021, doi: [10.1007/s13369-020-05184-0](https://doi.org/10.1007/s13369-020-05184-0).
- [9] 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, Oct. 2021, doi: [10.11591/ijeecs.v24.i1.pp484-490](https://doi.org/10.11591/ijeecs.v24.i1.pp484-490).
- [10] A. Adeleke, N. A. Samsudin, Z. A. Othman, and S. K. Ahmad Khalid, "A two-step feature selection method for quranic text classification," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 16, no. 2, pp. 730–736, 2019, doi: [10.11591/ijeecs.v16.i2.pp730-736](https://doi.org/10.11591/ijeecs.v16.i2.pp730-736).
- [11] N. Sabri Elmitwally and A. Alsayat, "The Multi-Class Classification for the First Six Surats of the Holy Quran," *IJACSA) International Journal of Advanced Computer Science and Applications*, vol. 11, no. 1, pp. 327–332, 2020, doi: [10.14569/IJACSA.2020.0110141](https://doi.org/10.14569/IJACSA.2020.0110141).
- [12] S. Shahriar and U. Tariq, "Classifying Maqams of Quranic Recitations using Deep Learning," *IEEE Access*, vol. 9, pp. 117271–117281, 2021, doi: [10.1109/ACCESS.2021.3098415](https://doi.org/10.1109/ACCESS.2021.3098415).
- [13] A. Naas, S. Na'iema, H. Mulyo, and A. Widiastuti, "Klasifikasi penerima bantuan program rehabilitasi rumah tidak layak huni menggunakan algoritme K-Nearest Neighbor," *Jurnal Teknologi dan Sistem Komputer*, vol. 10, no. 1, pp. 32–37, 2022, doi: [10.14710/jtsiskom.2022.14110](https://doi.org/10.14710/jtsiskom.2022.14110).
- [14] 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](https://doi.org/10.32736/sisfokom.v9i3.980).
- [15] N. Alvi Hasanah, Nanik Suciati, and Diana Purwitasari, "Pemantauan Perhatian Publik terhadap Pandemi COVID-19 melalui Klasifikasi Teks dengan Deep Learning," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 5, no. 1, pp. 193–202, Feb. 2021, doi: [10.29207/resti.v5i1.2927](https://doi.org/10.29207/resti.v5i1.2927).
- [16] M. Ihsan, B. S. Negara, and S. Agustian, "Metode LSTM (Long short term memory) untuk Klasifikasi Sentimen Vaksin Covid-19 pada Twitter," *Digital Zone: Jurnal Teknologi Informasi dan Komunikasi*, vol. 13, no. 1, pp. 79–89, May 2022, doi: [10.31849/digitalzone.v13i1.9950](https://doi.org/10.31849/digitalzone.v13i1.9950).
- [17] "Al-Qur'an yang mulia," Quran.com. Accessed: Feb. 13, 2024. [Online]. Available: <https://quran.com/id/developers>
- [18] A. Nikmatul Kasanah and U. Pujiyanto, "Penerapan Teknik SMOTE untuk Mengatasi Imbalance Class dalam Klasifikasi Objektivitas Berita Online Menggunakan Algoritma KNN," *JURNAL RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 1, no. 3, pp. 196–201, 2019, doi: [10.29207/resti.v3i2.945](https://doi.org/10.29207/resti.v3i2.945).
- [19] R. D. Mendrofa, M. H. Siallagan, J. Amalia, and D. P. Pakpahan, "Credit Risk Analysis With Extreme Gradient Boosting and Adaptive Boosting Algorithm," *Journal of Information System, Graphics, Hospitality and Technology*, vol. 5, no. 1, pp. 1–7, Mar. 2023, doi: [10.37823/insight.v5i1.233](https://doi.org/10.37823/insight.v5i1.233).
- [20] D. Sugiarto, E. Utami, and A. Yaqin, "Perbandingan Kinerja Model TF-IDF dan BOW untuk Klasifikasi Opini Publik Tentang Kebijakan BLT Minyak Goreng," *Jurnal Teknik Industri*, vol. 12, no. 3, pp. 272–277, 2022, doi: [10.25105/jti.v12i3.15669](https://doi.org/10.25105/jti.v12i3.15669).
- [21] E. Mulyani, F. P. B. Muhamad, and K. A. Cahyanto, "Pengaruh N-Gram terhadap Klasifikasi Buku menggunakan Ekstraksi dan Seleksi Fitur pada Multinomial Naïve Bayes," *JURNAL MEDIA INFORMATIKA BUDIDARMA*, vol. 5, no. 1, p. 264, Jan. 2021, doi: [10.30865/mib.v5i1.2672](https://doi.org/10.30865/mib.v5i1.2672).

- [22] A. M. Alashqar, "A Classification of Quran Verses Using Deep Learning," *International Journal of Computing and Digital Systems*, vol. 14, no. 1, pp. 189–201, 2023, doi: 10.12785/ijcds/XXXXXX.
- [23] A. Saputra and I. Z. Yadi, "Klasifikasi Ayat Al-Quran Terjemahan Menggunakan Metode Support Vector Machine Dan K-Nearest Neighbors," *Bina Darma Conference on Computer Science*, vol. 2, no. 4, pp. 449–466, 2020, Accessed: Oct. 30, 2023. [Online]. Available: <https://conference.binadarma.ac.id/index.php/BDCCS/issue/view/19>
- [24] A. Wahdan, S. Hantoobi, S. A. Salloum, and K. Shaalan, "A systematic review of text classification research based on deep learning models in Arabic language," *International Journal of Electrical and Computer Engineering*, vol. 10, no. 6, pp. 6629–6643, Dec. 2020, doi: 10.11591/IJECE.V10I6.PP6629-6643.
- [25] A. Naldi and S. Agustian, "Klasifikasi sentimen Vaksin Covid-19 menggunakan K-Nearest Neighbor Berdasarkan Word Embeddings Fasttext Pada Twitter," *ZONAsi: Jurnal Sistem Informasi*, vol. 5, no. 2, pp. 323–333, 2023, doi: 10.31849/zn.v5i2.12548.



ZONAsi: Jurnal Sistem Informasi

is licensed under a [Creative Commons Attribution International \(CC BY-SA 4.0\)](https://creativecommons.org/licenses/by-sa/4.0/)