Klasifikasi Ayat Al-Quran Terjemahan Bahasa Inggris Menggunakan Long Short Term Memory dan Bidirectional Long Short Term Memory

1st Rafisa Arif Irfan
Fakultas Informatika
Universitas Telkom
Bandung, Indonesia
rafisairfan@student.telkomuniversity.ac.id

2nd Kemas Muslim Lhaksmana
Fakultas Informatika
Universitas Telkom
Bandung, Indonesia
kemasmuslim@telkomuniversity.ac.id

***Abstrak*— Di dalam Al-Quran terdapat kandungan ayat yang berbeda-beda, maka sangatlah penting untuk memahami ayat Al-Quran. Al-Quran terdiri atas 30 juz, 144 surat, 6236 ayat, dan 77845 kata. Banyak ayat dan kata yang terdapat pada Al-Quran, untuk mempermudah umat muslim dalam mempelajari ayat maka perlu dilakukan pengklasifikasian terhadap ayat Al-Quran. Pada penelitian ini dilakukan klasifikasi multi label ayat Al-Quran berdasarkan topik-topik yang ada. Perancangan sistem dilakukan dengan menggunakan dua metode yaitu *convolutional long short term memory* (C-LSTM) dan *bidirectional long short term memory* (Bi-LSTM) yang mampu mengklasifikasikan ayat kedalam kelompoknya masing-masing. C-LSTM mampu mengungguli Bi-LSTM pada hampir setiap skenario. Nilai *hamming loss* terbaik yang diberikan C-LSTM sebesar 0.09985, dan Bi-LSTM 0.10122 pada skenario 90% data latih dan dropout.**

***Kata Kunci*— Klasifikasi, Multi Label, Recurrent Neural Network, Long Short Term Memory, Convolutional Neural Network, Bidirectional Long Short Term Memory, Hamming loss.**

I. PENDAHULUAN

A. Latar Belakang

Terdapat dua sumber pedoman hidup yang dimiliki oleh umat muslim yaitu Al-Quran dan hadist. Al-Quran adalah kitab suci umat islam sebagai petunjuk hidup dan sumber hukum bagi umat muslim. Mempelajari kandungan Al-Quran menjadi kewajiban bagi seluruh umat muslim. Al-Quran terdiri atas 30 juz, 144 surat, 6236 ayat, dan 77845 kata [1]. Di dalam Al-Quran terdapat kandungan ayat dengan topik yang berbeda-beda, maka sangatlah penting bagi umat muslim untuk memahami ayat Al-Quran.

Salah satu cara yang dapat dilakukan untuk mempermudah umat musli dalam mempelajari Al-Quran adalah mengelompokan ayat-ayat tersebut berdasarkan topik yang ada. Pengelompokan ayat dilakukan dengan melakukan klasifikasi terhadap ayat-ayat tersebut. Klasifikasi ayat Al-Quran dapat dikategorikan dalam klasifikasi teks multi-label.

Pengklasifikasian dilakukan dengan menggunakan metode convolutional long short term memory(C-LSTM) dan dibandingkan dengan metode bi-directional long short term memory(Bi-LSTM). LSTM merupakan metode yang dibangun untuk menutupi kekurangan pada metode recurrent neural network (RNN), LSTM memiliki kelebihan dapat mengingat informasi dalam jangka Panjang dan menghapus informasi yang sudah lama tidak terpakai. Sedangkan Bi-LSTM merupakan sebuah metode modifkasi dari LSTM yang memiliki kekurangan hanya dapat memproses kata secara 1 arah saja, sedangkan Bi-LSTM dapat memproses secara 2 arah. Convolutional neural network merupakan model yang dapat mengurangi beban saat melakukan komputasi.

Pada penelitian yang dilakukan oleh Vatana dkk [2] permasalahan klasifikasi multilabel ayat Al-Quran Terjemahan Bahasa Inggris berdasarkan topiknya dilakukan dengan menggunakan LSTM dan GRU. Penelitian ini menghasilkan hasil Hamming loss yang baik sebesar 1.10384615. Sementara itu Pulkit Parikh dkk [3] melakukan penelitian klasifikasi multilabel menggunakan Neural Framework. Pada penelitian tersebut didapatkan hasil akurasi F1 score sebesar 69,7% untuk model Bi-LSTM dan 72,8% saat menggunakan attention. Sedangkan Yudi Widhiyasana melakukan klasifikasi terhadap teks berita menggunakan metode C-LSTM menghasilkan akurasi sebesar 93,27% [4].

Maka dari itu dilakukan penelitian dengan menggunakan metode C-LSTM dan Bi-LSTM pada kasus pengklasifikasian multi label topik ayat Al-Quran dengan memanfaatkan Hamming loss sebagai metode evaluasi.

1. Topik dan Batasannya

Sub Penelitian yang dilakukan menggunakan merupakan pengklasifikasian ayat Al-Quran kedalam topiknya masing-masing. Dataset yang digunakan merupakan dataset Al-Quran yang sudah dilabeli dengan 16 topik berdasarkan tafsir Al-Quran Cordova terbitan Syaamil Quran, Bandung. Metode yang digunakan adalah convolutional long short term memory dan bidirectional long short term memory.

2. Tujuan

Tujuan Penelitian Tugas akhir ini adalah membangun model klasifikasi dengan menggunakan metode hybrid yaitu convolutional neural network dengan long short term

memory (C-LSTM) dan bidirectional long short term memory (Bi-LSTM).

Selain itu tujuan penelitian adalah menganalisis performa dari kedua metode yaitu metode convolutional neural network dengan long short term memory (C-LSTM) dan bidirectional long short term memory (Bi-LSTM) pada pengklasifikasian multi label ayat Al-Quran terjemahan Bahasa Inggris berdasarkan topik.

3. Organisasi Tulisan

Pada bagian selanjutnya yaitu bagian kedua membahas mengenai studi terkait penelitian yang telah diteliti sebelumnya dan teori terkait dengan penelitian yang dilakukan. Setelah itu perancangan sistem dilakukan pada bagian ketiga setelah pembahasan studi dan teori terkait. Bagian keempat membahas tentang hasil penelitian dan evaluasi hasil yang diperoleh dari model yang dibangun. Kesimpulan dan saran dibahas pada bagian terakhir yaitu bagian kelima.

II. KAJIAN TEORI

A. Studi Terkait

Bagian Penelitian terkait dengan klasifikasi ayat Al-Quran berdasarkan topik sebelumnya sudah dilakukan oleh beberapa peneliti dalam metode yang berbeda-beda contohnya k-nearest Neighbor (K-NN), multinomial naïve bayes, decision tree, support vector machine (SVM), naïve bayes (NB), recurrent neural network (RNN) dan convolutional neural network (CNN).

Penelitian yang dilakukan oleh Parikh [3] yaitu penelitian terhadap kasus multilabel menggunakan metode Neural Framework. Pada penelitian tersebut didapatkan hasil akurasi F1 score sebesar 69,7% untuk model Bi-LSTM dan 72,8% saat menggunakan attention.

Metode convolutional long short term memory diteliti oleh Yudi Widhiyasana dengan judul 'Penerapan Convolutional Long Short-Term Memory untuk Klasifikasi Teks Berita Bahasa Indonesia' [4]. Pengklasifikasian terhadap data teks tersebut menghasilkan hasil yang baik yaitu dengan akurasi sebesar 93,27%.

Al-Kabi dkk [5], melakukan penelitian terkait dengan klasifikasi topik Al- Quran dalam Bahasa Arab. Pada penelitian tersebut digunakan 4 macam metode yaitu decision tree, k-nearest Neighbor (K-NN), support vector machine (SVM) dan naïve bayes. Evaluasi Dilakukan dengan menggunakan 3 metode yaitu Accuracy, Recall dan Precision. Dari penelitian tersebut didapatkan kesimpulan bahwa naïve bayes merupakan metode yang menghasilkan nilai akurasi tertinggi dibandingkan dengan metode lainnya.

Al-Mira dkk [6], meneliti tentang klasifikasi multi label pada topik dari ayat Al-Quran terjemahan Bahasa Inggris menggunakan metode multinomial naïve bayes. Sebelum model dibentuk, data diproses terlebih dahulu menggunakan case folding, tokenisasi dan stemming. Setelah itu diterapkan metode Bag of Word pada data. Hasil terbaik dari metode hamming loss yang didapatkan dari penelitian sebesar 0,1208.

Penelitian lain yang menggunakan data Al-Quran adalah penelitian yang dilakukan oleh Vatana, dkk [2] dan Sarah Fauziah, dkk [7]. Kedua peneliti memanfaatkan glove sebagai metode word embedding dan hamming loss sebagai metode evaluasinya. Vatana dkk menggunakan LSTM dan GRU dengan hasil sebesar 0.1056 dan 0.1038. Sedangkan Sarah Fauziah, dkk menggunakan metode CNN, menghasilkan nilai hamming loss sebesar 0.0963.

1. Klasifikasi Multi Label Ayat Al-Quran

Klasifikasi multi label merupakan salah satu kasus yang berbeda dengan klasifikasi pada biasanya. Setiap data dapat memiliki lebih dari satu label [8]. Klasifikasi topik Al-Quran merupakan pengelompokan ayat Al-Quran berdasarkan topik yang ada kedalam kelas yang telah ditentukan. Berdasarkan penelitian yang dilakukan oleh Al-Mira, dkk dengan judul "A Multi-label Classification of Topics of Quranic Verses in English Translation using Tree Augmented Naïve Bayes" bahwa ayat-ayat Al- Quran dikelompokan dalam 15 kelas yaitu arkanul Quran, iman, Al-Quran, ilmu dan cabangnya, amal, dakwah, jihad, manusia dan hubungan masyarakat, akhlak, peraturan negara dan masyarakat, pertanian dan perdagangan, yang berhubungan dengan harta, hal-hal yang berkaitan dengan hukum, sejarah dan kisah, dan agama [6].

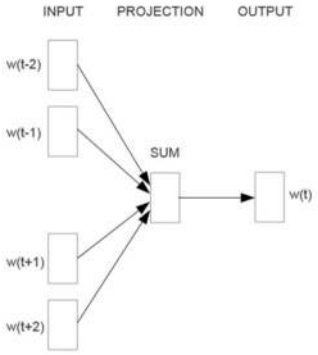
2. Pre-processing

Pre-processing adalah proses pengolahan data untuk membersihan data yang tidak sesuai agar model dapat memproses data dengan baik sehingga menghasilkan performansi yang maksimal. Pada penelitian yang dilakukan terdapat empat proses yaitu remove punctuation merupakan proses penghapusan tanda baca, simbol dan angka pada data, case folding yaitu mengubah semua huruf menjadi huruf kecil, stopword removal merupakan penghapusan kata yang tidak memiliki makna pada teks, tokenizer yaitu memisahkan kalimat menjadi kata kata, dan stemming yang merupakan proses mengubah kata berimbuhan menjadi kata dasar.

3. Word2vec

Word2vec merupakan sebuah transformer yang mengoperasikan dokumen menjadi perhitungan vektor dalam sebuah ruang vector kata. Word2vec bekerja dengan membangun sebuah kosakata dari data teks latih lalu mempelajari representasi vektor dari kumpulan kata [9]. Word2vec bekerja dalam 2 arsitektur yaitu Continuous Bags-of-Words dan Skip-Gram.

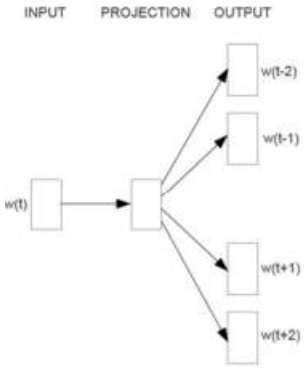
a. *Continuous Bags-of-Words* (CBOW)



GAMBAR 1
Arsitektur CBOW

CBOW merupakan sebuah model arsitektur yang didasarkan pada neural network di mana *non-linear hidden layer* dihilangkan dan projection layer dibagikan ke semua kata [9].

a. Skip-Gram



GAMBAR 2
Arsitektur Skip-Gram

Arsitektur *skip-gram* mirip dengan arsitektur sebelumnya yaitu CBOW. Hasil latih *skip-gram* untuk representasi kata yang berguna dalam memprediksi kata yang ada di dekatnya pada suatu kalimat. [10].

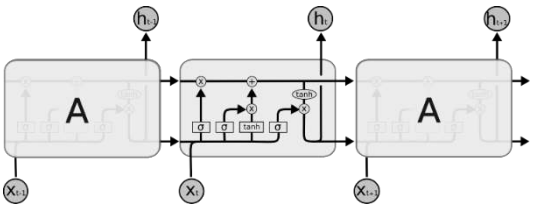
4. Convolutional Nerual Network

Convolutional neural network atau CNN merupakan salah satu metode neural network yang memanfaatkan layer convolutional sebagai layer penyusun dalam model. Convolutional layer pada CNN dapat mengurangi beban komputasi dari neural network [11]. Hal tersebut disebabkan karena layer convolutional merupakan sparse matrix yang memiliki dimensi yang lebih kecil dibandingkan dengan dimensi data yang diolah. Gambar formula pada layer convolutional.

$$S(x) = (I * K)(x) = \sum_n I(n-x)K(n) \quad (1)$$

5. Long Short Term Memory

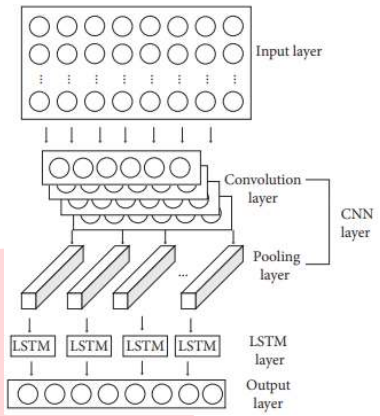
Long short term memory merupakan metode modifikasi dari recurrent neural network, LSTM bekerja dengan cara menyimpan bobot atau weight lebih lama dibanding RNN. Hal ini disebabkan LSTM memiliki sel-sel LSTM, yaitu sebuah node yang memiliki self-recurrent. Hal ini menyebabkan LSTM dapat bekerja lebih baik daripada RNN pada data dengan sekuens yang lebih panjang. LSTM terdiri dari forget gate, input gate, cell state, dan output gate [12]. Gambar 1 menunjukan arsitektur dari LSTM.



GAMBAR 3
Arsitektur LSTM

6. Convolutional Long Short Term Memory

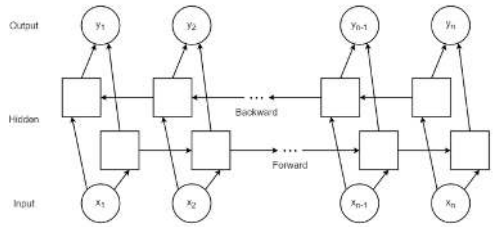
Convolutional long short term memory merupakan metode hybrid dari CNN dan LSTM. Kedua metode tersebut merupakan metode yang popular digunakan untuk klasifikasi teks. Hal tersebut didukung oleh arsitektur model yang memiliki kelebihannya masing-masing. CNN dapat mengurangi beban komputasi [9] dan LSTM dapat memproses kata secara sekuens yang dapat menghasilkan performa yang baik untuk data teks [12]. Berikut arsitektur C-LSTM [13].



GAMBAR 4
Arsitektur Convolutional Long Short Term Memory

7. Bidirectional Long Short Term Memory

Bidirectional long short term memory atau yang dikenal dengan Bi-LSTM merupakan metode modifikasi dari LSTM memiliki konsep yang similar dengan LSTM. Dengan menggunakan LSTM sebagai arsitektur network pada bidirectional RNN, maka dihasilkan bidirectional LSTM yang menyediakan akses terhadap konteks jarak jauh maupun jarak dekat pada kedua arah [14].



GAMBAR 5
Arsitektur BiLSTM

8. *Hamming loss*

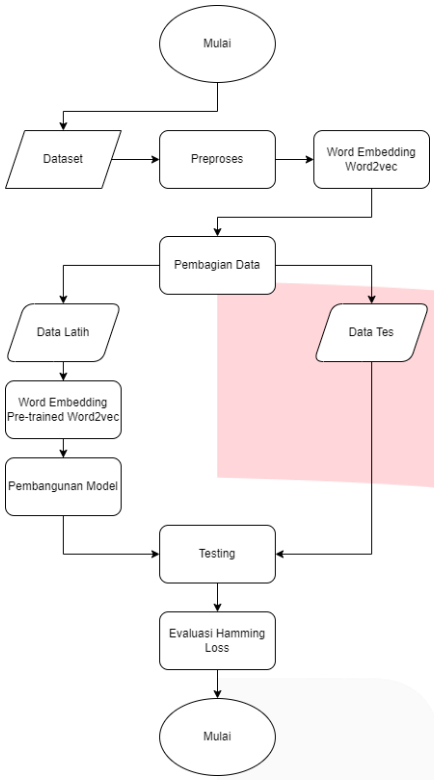
Hamming loss merupakan metode yang digunakan pada tahap evaluasi untuk mendapatkan nilai hasil akurasi. Pada penelitian ini, tahap evaluasi menggunakan *hamming loss* sebagai metodenya. Hal ini dikarenakan *hamming loss* cocok untuk data multi label, dimana data yang digunakan untuk penelitian ini adalah data Al-Quran yang setiap ayatnya dapat memiliki lebih dari satu topik. Semakin kecil nilai *hamming loss* maka semakin baik [6]. Persamaan *hamming loss* mengikuti persamaan 1 dimana N adalah banyak data, L merupakan panjang output multi label, $y^{(i)}$ adalah Target klasifikasi multi label, dan $y^{(i)}$ merupakan Output klasifikasi multi label.

$$Hamming\ loss = \frac{1}{NL}\sum_{i=1}^{N}\sum_{j=1}^{L}[\hat{y}_j^{(i)} \neq \hat{y}_j^i] \quad (2)$$

III. METODE

A. Sistem yang Dibangun

Pada penelitian dibangun dua buah sistem bertujuan untuk mengklasifikasikan ayat Al-Quran secara multi label kedalam topik yang sudah ditentukan. Dilakukan beberapa proses seperti *pre-processing*, melatih model *word2vec*, pembagian data, pembanguan model, dan evaluasi.



GAMBAR 6
Alur Sistem

1. Dataset

Dataset yang digunakan untuk penelitian ini merupakan ayat Al-Quran terjemahan Bahasa Inggris yang sebelumnya digunakan oleh Al-mira dkk untuk penelitian dengan judul "A Multi-label Classification on Topics of Quranic Verses in English Translation Using Tree Augmented Naïve Bayes" [6].

TABEL 1
Dataset Al-Quran

No	Topik	Jumlah Ayat
1	Arkanul Islam	3425
2	Iman	2288
3	Al-Qur'an	533
4	Ilmu dan Cabangnya	523
5	Amal	718
6	Dakwah	197
7	Jihad	356
8	Manusia dan Hubungan Masyarakat	624
9	Akhlak	769
10	Peraturan yang Berhubungan dengan Harta	297
11	Hal-hal yang Berkaitan dengan Hukum	203
12	Negara dan Masyarakat	41
13	Pertanian dan Perdagangan	37
14	Sejarah dan Kisah	676
15	Agama	298
16	Lainnya	754

Dataset terdiri dari 6236 ayat Al-Quran dengan labelnya, tabel 2 menampilkan contoh 5 dataset dengan labelnya.

Ayat	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
In the name of Allah, the Beneficent, the Merciful.	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
All praise is due to Allah, the Lord of the Worlds.	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
The Beneficent, the Merciful.	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Master of the Day of Judgment.	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Thee do we serve and Thee do we beseech for help.	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

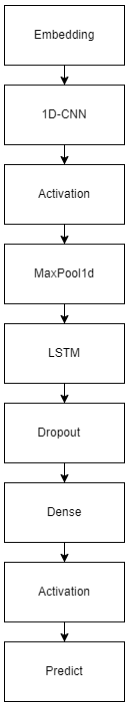
2. Pre-procesing

GAMBAR 7
Tahap Pre-Procesing

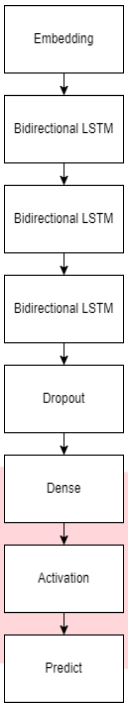
Tahap *pre-processing* digambarkan pada gambar 4. Setelah mendapatkan data ayat Al-Quran beserta labelnya maka tahap selanjutnya adalah melakukan pembersihan data dalam beberapa tahapan seperti *remove punctuation*, *case folding*, *stopword removal* menggunakan library nltk, *tokenizer*, dan *stemming* menggunakan library *porter stemmer.*

3. Arsitektur Convolutional Long Short Term Memory dan Bidirectional Long Short Term Memory

Pada penelitian yang dilakukan terdapat skenario pengujian yang dilakukan seperti pengujian menggunakan perbandingan data latih dan data tes lalu dengan *dropout* dan tanpa *dropout*. Hal ini dilakukan untuk menentukan *baseline* yang akan digunakan pada skenario selanjutnya. Skenario selanjutnya yaitu pengujian menggunakan *pre-trained word embedding* berdasarkan banyak dimensi yaitu 100, 300, dan 500.



GAMBAR 7
Arsitektur C-LSTM



GAMBAR 9
Arsitektur Bi-LSTM

IV. HASIL DAN PEMBAHASAN

A. Evaluasi

1. Hasil Pengujian Skenario 1

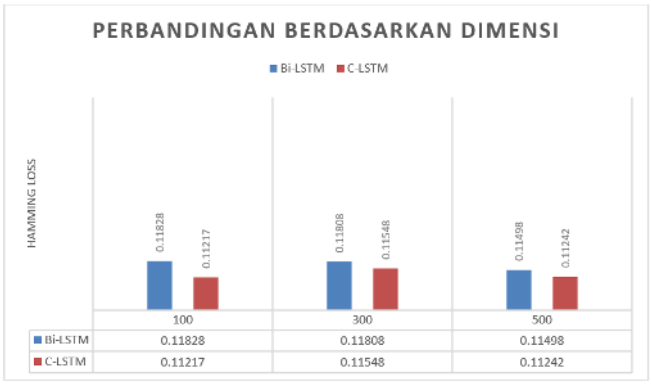
Pengujian telah dilakukan berdasar pada skenario yang sudah ditetapkan. Pada pengujian pertama model Bi-LSTM dan C-LSTM dibandingkan berdasarkan jumlah data latih dan penggunaan *dropoutnya*. Mengingat bahwa dapat terjadinya *overfitting* pada kedua model dan dapat mempengaruhi performa model maka perlu dibandingkan dengan penambahan *dropout*. Jumlah data latih merupakan salah satu faktor yang berpengaruh pada performa model yang dibangun. Data dibagi menjadi 3 komposisi data latih dan data tes yaitu 70:30, 80:20, 90:10. Penambahan *dropout* diterapkan pada setiap komposisi data. Dikarenakan keterbatasan data maka perlu ditentukan komposisi yang paling sesuai untuk kedua model. Karena setiap pengujian menghasilkan hasil yang berbeda, maka pengujian dilakukan sebanyak 3 kali untuk setiap skenarionya lalu diambil nilai rata-ratanya untuk mendapat hasil yang sebenarnya.

TABEL 2
Hasil Perbandingan berdasarkan Data Latih

Data latih : Data tes	**Bi-LSTM**	**C-LSTM**
70:30	0.10923	0.10819
80:20	0.10571	0.10391
90:10	0.10723	0.10369
70:30 + Dropout	0.10542	0.10221
80:20 + Dropout	0.10126	0.10511
90:10 + Dropout	**0.10122**	**0.09985**

Berdasarkan pengujian didapatkan bahwa perubahan data latih dapat mempengaruhi performa dari kedua model. Semakin besar data latih pada model menghasilkan hasil yang semakin baik. Pada skenario perbandingan 70:30 tanpa *dropout* Bi-LSTM dan C-LSTM menghasilkan performa yang tidak lebih baik dibandingkan skenario lainnya yaitu dengan nilai *hamming loss* masing-masing 0.10923 dan 0.10819. Hal ini disebabkan semakin sedikitnya data yang dipelajari oleh model dan semakin banyaknya data tes. Selain itu dengan data latih sebesar 90% dan dropout Bi-LSTM menghasilkan nilai *hamming loss* sebesar 0.10122 sedangkan C-LSTM menghasilkan nilai *hamming loss* sebesar 0.09985. Kenaikan performa disebabkan karena semakin banyaknya data yang dilatih oleh model dan penggunaan dropout juga mempengaruhi performa dari kedua model. Dengan ini arsitektur dengan data latih sebesar 90% dan *dropout* akan digunakan untuk skenario selanjutnya yaitu perbandingan penggunaan *word2vec* berdasarkan dimensinya.

2. Analisis Hasil Pengujian skenario II



GAMBAR 8
Hasil Perbandingan berdasarkan Dimensi *Word2vec*

Gambar 9 merupakan hasil perbandingan dua model menggunakan *word2vec* sebagai *pre-trained word embedding*. Pada skenario ini dilakukan perbandingan dengan membandingkan kedua model dengan besarnya dimensi yaitu 100, 300, dan 500. Pegujian pada skenario ini menghasilkan hasil yang cukup signifikan antara kedua model, C-LSTM unggul disetiap pengujian. Perubahan pada dimensi *word2vec* memberikan hasil yang cukup signifikan. Dalam grafik khususnya pada model Bi-LSTM bahwa semakin besar dimensinya maka semakin baik nilai *hamming loss* yang dihasilkan, tetapi untuk C-LSTM tidak tergantung dengan banyaknya dimensi. Bi-LSTM dapat menghasilkan nilai *hamming loss* terbaiknya saat menggunakan dimensi sebesar 500, berbalik dengan C-LSTM dapat menghasilkan nilai *hamming loss* terbaiknya pada saat menggunakan dimensi sebesar 100. Saat C-LSTM menggunakan dimensi sebesar 500 memberikan nilai yang tidak begitu berbeda saat menggunakan dimensi 100, memiliki perbedaan sekitar 0.0003.

Pada pengujian tersebut C-LSTM menghasilkan nilai yang lebih baik dibandingkan dengan Bi-LSTM yaitu 0.11217 pada dimensi 100 dan 0,11498 pada dimensi 500. Dibandingkan dengan arsitektur sebelumnya, penggunaan *word2vec* sebagai *pre-trained word embedding* menghasilkan angka yang lebih tinggi.

V. KESIMPULAN

A. Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan, dapat disimpulkan bahwa model C-LSTM dan Bi-LSTM dapat digunakan untuk mengklasifikasikan ayat secara multi label. Dengan hasil yang ada dapat dilihat bahwa C-LSTM mengungguli model Bi-LSTM pada semua skenario. Pada skenario 1 yaitu membandingkan distribusi dataset, C-LSTM menghasilkan nilai *hamming loss* yang lebih baik dibandingkan dengan model Bi-LSTM pada hampir semua perbandingan kecuali pada distribusi 80:20 dengan menggunakan *dropout*. Bi-LSTM menghasilkan performa terbaiknya pada distribusi data 90:10 dengan *dropout* sebesar 0.10122, C-LSTM juga menghasilkan performa terbaiknya pada distribusi yang sama dengan nilai *hamming loss* sebesar 0.09985. Pada Skenario II C-LSTM juga unggul untuk semua perbandingan yaitu pada dimensi sebesar 100, 300, dan 500. Penggunaan *word2vec* tidak membuat kedua model menjadi lebih baik. Nilai hamming loss terbaik yang dihasilkan oleh C-LSTM dengan menggunakaan *dropout* dan *word2vec* sebesar 0.11217, sedangkan Bi-LSTM menghasilkan nilai *hamming loss* sebesar 0.11498. Jika membandingkan dengan baseline pada skenario 1 yaitu pengujian dengan membandingkan distribusi data dan dengan menggunakan dropout menghasilkan hasil yang lebih baik dibandingkan dengan menggunakan *word2vec*. Adapun saran untuk penelitian selanjutnya yaitu melakukan penelitian dengan memanfaatkan metode word embedding lainnya seperti BERT dan metode *deep learning* lainnya yang dapat mengklasifikasikan kata secara multi label.

REFERENSI

[1] Y. Luan and S. Lin, “Research on Text Classification Based on CNN and LSTM.”

[2] V. Rianti Aldefi and S. al Faraby, “Klasifikasi Multi-label pada Terjemahan Al-Quran Berbahasa Inggris Menggunakan Recurrent Neural Network (RNN).”

[3] P. Parikh *et al.*, “Multi-label Categorization of Accounts of Sexism using a Neural Framework,” Oct. 2019, [Online]. Available: http://arxiv.org/abs/1910.04602

[4] Y. Widhiyasana, T. Semiawan, I. Gibran, A. Mudzakir, and M. R. Noor, “Penerapan Convolutional Long Short-Term Memory untuk Klasifikasi Teks Berita Bahasa Indonesia (Convolutional Long Short-Term Memory Implementation for Indonesian News Classification),” 2021.

[5] M. N. Al-Kabi *et al.*, “A Topical Classification of Quranic Arabic Text Arabic Natural Language Processing View project Dependency Graph and Metrics for Defects Prediction View project A Topical Classification of Quranic Arabic Text,” 2013. [Online]. Available: https://www.researchgate.net/publication/258282937

[6] Universitas Telkom, Multimedia University, and Institute of Electrical and Electronics Engineers, *A Multi-label Classification on Topics of Quranic Verses in English Translation Using Tree Augmented Naïve Bayes*.

[7] S. Fauziah Lestari and S. al Faraby, “Klasifikasi Multi-label pada Terjemahan Al-Quran Berbahasa Inggris Menggunakan Convolutional Neural Networks.”

[8] M. Dong, Universitas Telkom, Chinese and Oriental Languages Information Processing Society, Institute of Electrical and Electronics Engineers. Indonesia Section. Computer Society Chapter, and Institute of Electrical and Electronics Engineers, *Multi-Label Topic Classification of Hadith of Bukhari (Indonesian Languagetranslation) using Information Gain and Backpropagation Neural Network*.

[9] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient Estimation of Word Representations in Vector Space,” Jan. 2013, [Online]. Available: http://arxiv.org/abs/1301.3781

[10] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Distributed Representations of Words and Phrases and their Compositionality.”

[11] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[12] J. Brownlee, *Long short-term memory networks with python: develop sequence prediction models with deep learning. Machine Learning Mastery*, V1.5., vol. 1. 2017.

[13] W. Lu, J. Li, Y. Li, A. Sun, and J. Wang, “A CNN-LSTM-based model to forecast stock prices,” *Complexity*, vol. 2020, 2020, doi: 10.1155/2020/6622927.

[14] A. Graves, “Supervised Sequence Labelling with Recurrent Neural Networks.”.