

## SISTEM INFORMASI MANAJEMEN PENGKLASIFIKASIAN HADIST SHAHIH BUKHARI DAN MUSLIM MENGGUNAKAN ALGORITMA NEURAL NETWORK

Rasenda<sup>\*)</sup>, Nova Rini<sup>2)</sup>, Supriatiningsih<sup>3)</sup>

<sup>1)</sup>Universitas Teknologi Muhammadiyah Jakarta

<sup>2)</sup> Universitas Teknologi Muhammadiyah Jakarta

<sup>3)</sup> Universitas Teknologi Muhammadiyah Jakarta

e-mail: rasenda@utmj.ac.id

### ABSTRACT

*In the book of Bukhari and Muslim hadith there are 7008 hadith sentences, of the 7008 sentences the Hadith is not yet known a hadith included in the category of prohibitions or orders. By doing the classification, it will be easier for readers to understand the hadith. The classification of hadiths is done in several stages, including: pre-processing text, the use of word vector features, and modeling of neural network architecture with multilayer perceptron. The use of layers in neural networks and feature extraction with word vectors has proven to provide good results for the classification of hadiths. The results showed a fairly high degree of accuracy that is equal to 97.72% by using two layers and 256 neurons, this research can be used to classify hadiths which have the impact of making it easier for people to understand hadiths very well.*

**Keywords:** Classification, Neural Network, System Information Mangement, Hadith

### 1. PENDAHULUAN

Penelitian klasifikasi teks masih cukup jarang diterapkan untuk mengolah data teks tentang hadist. Saat ini selain dibukukan, kumpulan data hadist dapat dengan mudah didapatkan baik dari internet maupun disajikan didalam aplikasi namun tidak disertai dengan informasi makna dari hadist tersebut, sehingga untuk masyarakat awam atau bahkan umat muslim yang sedang mempelajari hadist mungkin mengalami kesulitan dalam memahami makna dari hadist-hadist tersebut mengingat jumlah data hadist yang cukup banyak, maka pengolahan terhadap data hadist ini perlu dilakukan, karena didalam hadist terdapat berbagai pedoman yang bermanfaat untuk masyarakat khususnya umat muslim dalam berperilaku sesuai dengan sunnah Nabi Muhammad Shallallahu ‘Alaihi Wasallam. (Kusumaningrum & Al-faraby, 2017)

Perintah dalam bahasa Arab disebut amar. Menurut mayoritas ulama usul fikih, amar adalah suatu tuntutan (perintah) untuk melakukan sesuatu dari pihak yang kedudukannya lebih tinggi kepada pihak yang tingkatannya lebih rendah. Adapun larangan dalam bahasa Arab disebut nahi. Mayoritas ulama usul fikih mendefinisikan nahi sebagai larangan untuk melakukan suatu perbuatan dari pihak yang kedudukannya lebih tinggi kepada pihak yang tingkatannya. (Abdelaal & Youness, 2019)

Word vector adalah alat yang dikembangkan untuk pemrosesan bahasa alami yang terbukti bermanfaat untuk berbagai tugas pemrosesan bahasa yang berasal dari kemampuan untuk menyandikan hubungan kata dalam ruang vektor. (Park, Byun, Baek, Cho, & Oh, 2018)

Neural Network adalah suatu metode pengelompokan dan pemisahan data yang prinsip kerjanya sama seperti neural network pada manusia. Elemen mendasar dari paradigma tersebut adalah struktur yang baru dari sistem pemrosesan informasi. Neural Network dibentuk untuk memecahkan suatu masalah tertentu seperti pengenalan pola atau klasifikasi karena proses pembelajaran. (Krieger, 1996)

Dalam merancang suatu Neural Network selain memperhatikan struktur hubungan antara simpul masukan dengan simpul keluaran, perlu ditentukan juga cara atau metode pembelajarannya. Belajar bagi Neural Network adalah cara memperbaharui bobot sinapsis disesuaikan dengan isyarat masukan dan keluran yang diharapkan.

Secara umum suatu Neural Network dibentuk atas sejumlah neuron sebagai unit pengolah informasi sebagai dasar operasi untuk menjalankan fungsi atau tugasnya.(Jarvis, 2005)

Evaluasi model klasifikasi didasarkan pada pengujian untuk memperkirakan obyek yang benar dan salah, urutan pengujian ditabulasikan dalam confusion matrix dimana kelas yang diprediksi ditampilkan di bagian atas matriks dan kelas yang diamati di sisi kiri. Setiap sel berisi angka yang menunjukkan beberapa banyak kasus yang sebenarnya dari kelas yang diamati untuk diprediksi (Hastuti, 2012).

Berikut ini tabel matrix yang digunakan dalam data mining:

Tabel 1. Confusion Matrix

	Actual Positive Class	Actual Negative Class
Predictive Positive Class	True Positive (TP)	False Negative (FN)
Predictive Negative Class	False Positive (FP)	True Negative (TN)

Untuk mengukur kinerja dari model klasifikasi dapat dihitung dengan cara menggunakan persamaan berikut ini: (M & M.N, 2015)

$$Accuracy = \frac{(TP+TN)}{(TP+TN+FP+FN)} \times 100 \%$$

Ada banyak cara yang dapat digunakan untuk mengklasifikasi jenis larangan atau perintah yaitu dengan menggunakan metode text mining. *Text mining* terdiri dari berbagai teknik yang dapat digunakan untuk melakukan klasifikasi, dimana teknik ini dapat mengklasifikasikan teks dari suatu kalimat dengan melihat beberapa informasi dan pola data yang ada. Untuk melakukan klasifikasi secara akurat, diperlukan metode yang tepat pula. *Neural network* adalah suatu metode komputasi yang dapat digunakan untuk mengklasifikasi. Dengan sistem pemrosesan informasi yang didesain dengan menirukan cara kerja otak manusia dalam menyelesaikan suatu masalah, dengan melakukan proses belajar melalui perubahan bobot dan mampu melakukan pengenalan kegiatan berbasis data masa lalu yang akan dipelajari oleh *Neural Network* sehingga mempunyai kemampuan untuk memberikan keputusan.

Terdapat beberapa penelitian terkait text mining tentang hadits antara lain: “Penerapan *Particle Swarm Optimization* Pada *Feedforward Neural Network* Untuk Klasifikasi Teks Hadis Bukhari Terjemahan Bahasa Indonesia” yang menggunakan metode *Back propagation – feedforward neural network (BP-FNN)* dan *particle swarm optimization – feedforward neural network (PSO-FNN)* pada dataset Klasifikasi pada hadist dan menghasilkan performance Akurasi BP-FNN 88,57% dan PSO-FNN 89,5%(Ghufran & Faraby, 2018). Kemudian pada penelitian lainnya yang berjudul “Klasifikasi Anjuran , Larangan dan Informasi pada Hadis Sahih Al-Bukhari berdasarkan Model Unigram menggunakan *Artificial Neural Network ( ANN )*” oleh E. Riviera, et.al menggunakan metode *n-gram model* dan *Artificial Neural Network (ANN)* pada dataset kategori anjuran, larangan, dan informasi pada hadist menghasilkan performance f1-score 85%(Riviera, Jasin, & Al-faraby, 2017). Selanjutnya penelitian dengan judul “Klasifikasi Kesahihan Hadits Berdasarkan Perawi Hadits Menggunakan *Principal Component Analysis (PCA)* dan *Backpropagation Neural Network (BPNN)*” oleh U. Nuha, et.al menggunakan metode *Principal Component Analysis (PCA)*, *Back propagation Neural Network (BPNN)* pada dataset Kategori sahah, hasan, dan dhaif pada hadist bukhari dan muslim menghasilkan performance Akurasi 86,53 %(Nuha & Rochmawati, 2019). Kemudian penelitian dengan judul “*Multi-Label Topic Classification of Hadith of Bukhari ( Indonesian Language translation ) using Information Gain and Backpropagation Neural Network*” oleh M. Yuslan, et.al menggunakan metode BPNN pada dataset klasifikasi anjuran, larangan dan perintah pada hadist menghasilkan performance akurasi 88,42 %(Yuslan & Bakar, 2018). Selanjutnya pada penelitian yang berjudul “*Classification of hadith into positive suggestion, negative suggestion, and information*” oleh S. Al Faraby, et.al menggunakan metode TF-IDF, Baseline, SVM, ANN pada dataset kategori perintah, larangan, informasi pada hadist menghasilkan performance F1 score Baseline 69% dan ANN 79%(Faraby, Riviera, & Jasin, 2018).

Berdasarkan hasil penelitian diatas didapatkan algoritma *neural network* cukup baik dalam melakukan pengklasifikasian teks. Adapun alasan *neural network* dianggap cukup baik dalam melakukan pengklasifikasian teks adalah seperti dibawah ini.

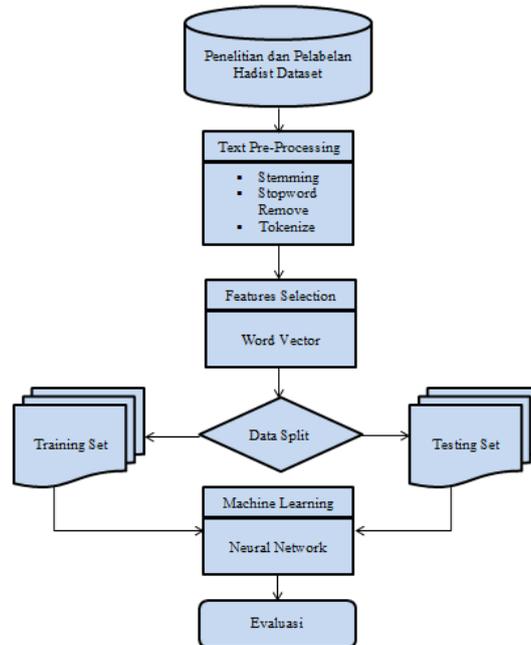
*Neural network* bersifat *non linier*, artinya jaringan cocok digunakan untuk sistem-sistem dengan kompleksitas permasalahan yang tinggi. *Neural network* bersifat *adaptif*, artinya jaringan mampu menghasilkan pemecahan (hubungan pemetaan) antara masukan dan keluarannya. *Neural network* bersifat tahan terhadap kesalahan, artinya jaringan secara keseluruhan masih mampu bekerja walaupun terjadi kesalahan/kerusakan

sampat tingkat tertentu. *Neural network* dapat melakukan generalisasi artinya jaringan dapat mengolah data yang belum pernah dilatihkan padanya berdasarkan apa yang telah diperoleh selama proses pembelajaran. (Krieger, 1996)

Oleh karena alasan diatas, penelitian ini akan dilakukan dengan *neural network* untuk mengklasifikasi kategori suatu hadist berupa larangan atau perintah. Hadist yang digunakan adalah hadist riwayat dari Bukhari dan Muslim.

## 2. METODE PENELITIAN

Berikut adalah tahapan penelitian seperti gambar 1. dibawah.



Gambar 1. Tahapan Penelitian

### 2.1 Penelitian dan Pelabelan Data

Penelitian ini menggunakan dataset Hadis Shahih Bukhari dan Muslim yang berjumlah 7008 matan dan dilabeli menurut jenis klasifikasi larangan atau perintah.

### 2.2 Pre-/Processing

Pada tahap *pre-processing* ini menggunakan *library StemmerFactory* pada modul *sastrawi* di python 3.7. Ada 3 tahap *pre-processing text* yang dilakukan, yaitu *stemming*, *remove stopword*, dan *tokenized*.

#### 2.2.1 Stemming

Metode *stemming* yang digunakan yaitu *StemmerFactory* yang berasal dari *library sastrawi* pada python 3.7 yang kemudian di eksekusi kedalam kalimat menggunakan *create stemmer*.

#### 2.2.2 Remove Stopword

Metode *remove stopword* yang digunakan yaitu *StopWordRemoverFactory* yang berasal dari *library sastrawi* pada *python 3.7* yang kemudian di eksekusi kedalam kalimat menggunakan *create stop word remover*.

#### 2.2.3 Tokenized

Metode *tokenized* yang digunakan yaitu *word\_tokenize* yang berasal dari *library NLTK* pada *python 3.7* yang kemudian di eksekusi kedalam kalimat.

### 2.3 Feature Dalam Eksperimen

*Feature* yang digunakan yaitu *word vector* yang berasal dari *library gensim* pada *Python 3.7*. Pada penelitian ini, *word vector* digunakan untuk mencari nilai *vocabulary* dan *similarity* dari dataset hadist. Selanjutnya *word vector* akan digunakan sebagai *one hot encoding* yang akan mengkonversi teks menjadi nilai biner untuk dimasukkan ke dalam *neural network*.

Nilai *Vocabulary* adalah nilai perbendaharaan kata yang terdapat pada *dataset* berupa nilai ordinat *x* dan *y*, sedangkan nilai *similarity* adalah nilai keterkaitan kata pada *vocabulary* yang telah dikonversi dari teks menjadi nilai biner.

## 2.4 Data Split

Data *split* digunakan untuk membagi data menjadi data *training* dan data *testing* menggunakan *library scikit-learn* pada *library Python* yang akan digunakan dalam permodelan *neural network*

## 2.5 Neural Network

Permodelan *neural network* sebagai arsitektur dalam eksperimen dengan menggunakan *library Keras* pada *Python 3.7*. Untuk menambahkan layer menggunakan *dense*, selanjutnya *input* yang telah ditentukan berdasarkan *size similarity* pada *word vector*, untuk aktivasi input layer dan hidden layer menggunakan fungsi aktivasi *relu*, lalu pada *output layer* menggunakan fungsi aktivasi *sigmoid*. Selanjutnya untuk *optimizer* menggunakan *adam*, *binary crossentropy* sebagai *loss*, lalu berikutnya menentukan nilai *epoch* dan *batch size*. Dibawah ini merupakan gambar 2. untuk membangun arsitektur *neural network*.

Input layer (Dense)	—————>	Aktivasi Relu
Size Input	—————>	Size Similarity in Word Vector
Hidden layer (Dense)	—————>	Aktivasi Relu
Output layer (Dense)	—————>	Aktivasi Sigmoid
Optimizer	—————>	Adam
Loss	—————>	Binary Crossentropy
Epoch	—————>	Value Epoch
Batch Size	—————>	Value Batch Size

Gambar 2. Pembangunan *Neural Network*

## 2.6 Evaluasi

Tahap evaluasi eksperimen dibuat untuk mendapatkan nilai akurasi. Berikut merupakan fungsi yang digunakan untuk mencari nilai akurasi:

$$Accuracy = \frac{(TP+TN)}{(TP+TN+FP+FN)} \times 100 \% \quad (1)$$

Keterangan:

*TP* = True Positive

*TN* = True Negative

*FP* = False Positive

*FN* = False Negative

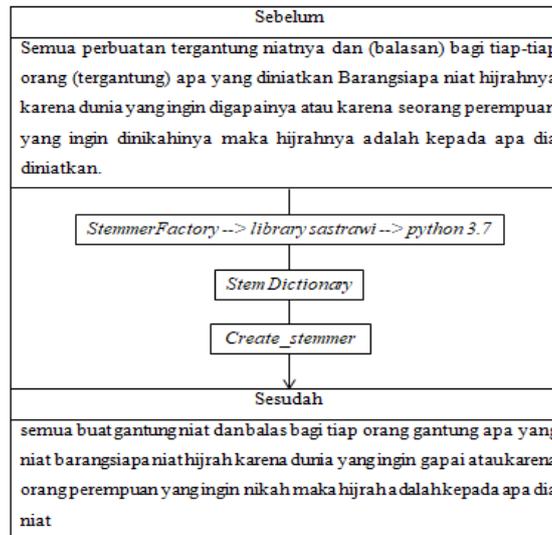
## 3. HASIL DAN PEMBAHASAN

### 3.1 Hasil Pre-processing Text

Berikut di bawah ini adalah hasil *pre-processing text*.

#### 3.1.1 Hasil Stemming

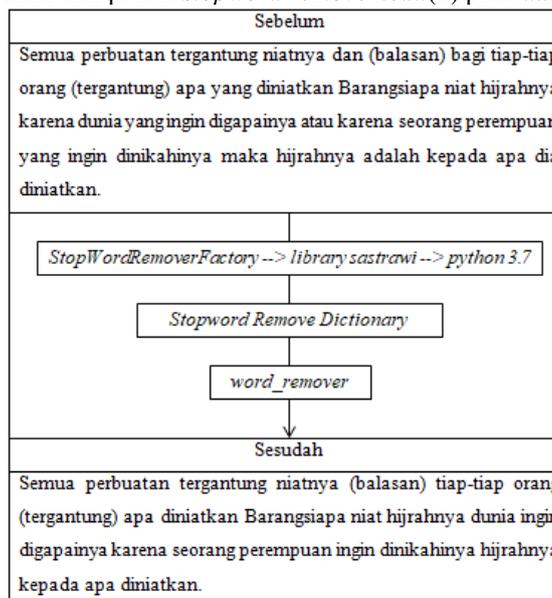
Gambar 3 merupakan hasil *stemming head(1)* pada *dataset* hadist.



Gambar 3. Hasil Stemming

### 3.1.2 Hasil Remove Stopword

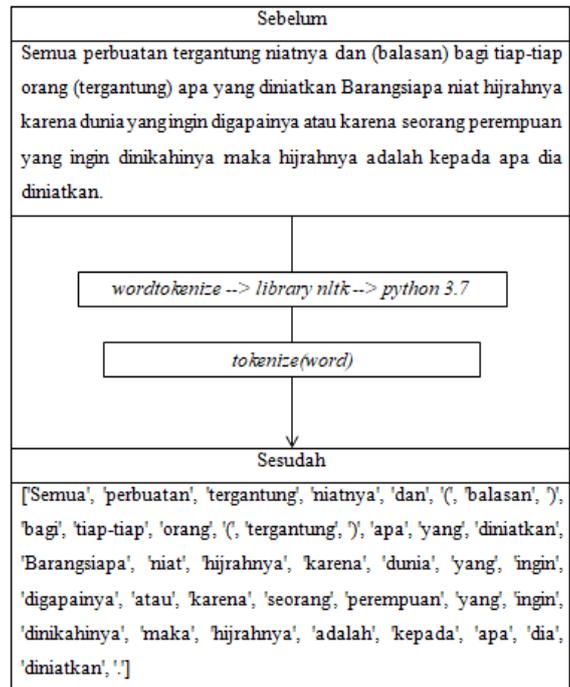
Pada gambar 4 adalah hasil dari proses *stopword remove head(1)* pada *dataset* hadist.



Gambar 4. Hasil Stopword Remove

### 3.1.3 Hasil Tokenize

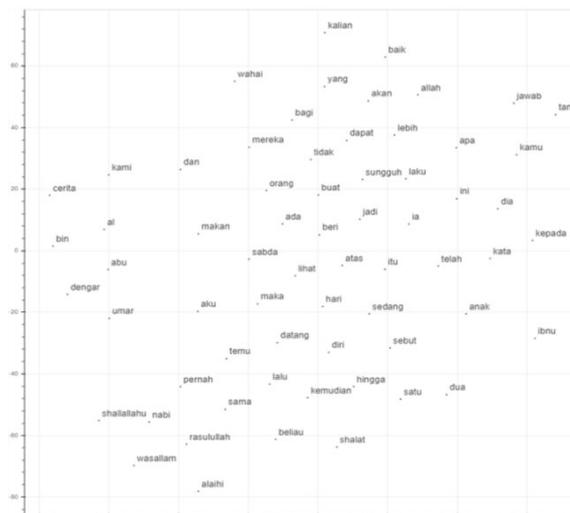
Selanjutnya pada gambar 5 adalah contoh dari tahapan *tokenize head(1)*:



Gambar 5. Hasil Tokenize

### 3.2 Penggunaan Word Vector Model

Penggunaan *word vector* yang digunakan dalam penelitian ini yaitu dengan pengaturan *min\_count = 5* sehingga menghasilkan *vocab = 3393*. Berikut adalah tabel 4.4 yang merupakan ordinat dari *vocabulary* dan pada gambar 4.2 merupakan visualisasi *vocabulary* dengan menambahkan *min\_count = 1000* sehingga menghasilkan 65 *vocab* yang lebih halus untuk dapat dilihat.



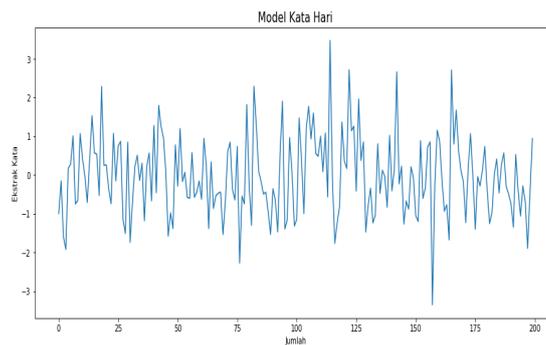
Gambar 6. Hasil Visualisasi Vocabulary

#### 3.2.1 Hasil Similarity Kata

Penggunaan *word vector* yang digunakan untuk mencari *similarity* dalam penelitian ini yaitu dengan pengaturan *windows = 5* dan *size = 200*. Berikut adalah contoh hasil *similarity* pada kata 'hari', seperti pada tabel 2. Sedangkan gambar 7. merupakan visualisasi *similarity* dari kata hari.

Tabel 2. Hasil Similarity Kata 'Hari'

-0.998	-0.15458	-1.60573	-1.92564	0.171547
0.2764	1.009715	-0.74964	-0.66444	1.068439
0.4179	-0.04054	-0.70896	0.424856	1.530188
0.5693	0.534064	-0.52959	2.284214	0.250277
0.2571	-0.35988	-0.74414	1.074334	-0.15327
0.7496	0.866513	-1.14856	-1.51139	0.845582
-1.739	-0.69451	0.20521	0.508369	-0.13977
0.2997	-1.18372	0.227631	0.562575	-0.66795
.....				
0.8793	-0.12627	-0.93852	-0.77086	-1.67769
2.7103	0.79881	1.67535	0.608546	0.101442
-0.178	-1.23129	0.161919	1.066376	-0.04765
-1.4	-0.0445	-0.28541	0.121079	0.734709
-0.395	-1.25611	-0.98406	0.012469	0.411275
-0.465	0.272096	0.565929	-0.30323	-0.49207
-0.741	-1.3506	0.526849	-0.39069	-1.05952
-0.282	-0.69368	-1.89614	-0.44613	0.939504



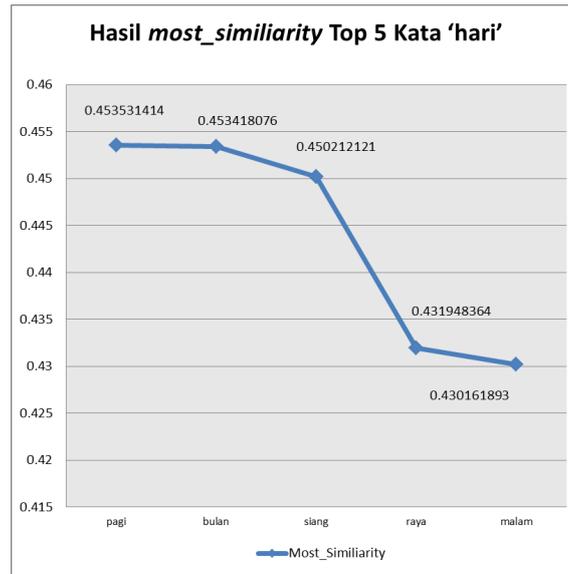
Gambar 7. Hasil Visualisasi Similarity 'Hari'

### 3.2.2 Hasil Most Similarity Kata

Kemudian didapat pula *most similarity* yang merupakan nilai keterkaitan suatu kata dengan kata lainnya. Dibawah ini adalah contoh *most similarity top 5* dari kata 'hari' yang terdapat pada tabel 3. dan visualisasi *most similarity Top 5* dari kata 'hari' pada gambar 4.4 .

Tabel 3. Hasil *Most similarity Top 5 'hari'*

Kata	Most Similarity
pagi	0.453531414
bulan	0.453418076
siang	0.450212121
raya	0.431948364
malam	0.430161893

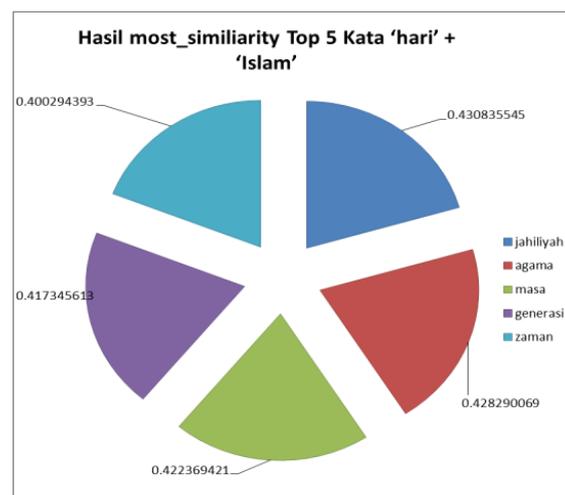


Gambar 8. Hasil visualisasi similarity 'hari'

Selanjutnya adalah contoh *most\_similarity* top 5 dari kata 'hari' + 'islam' yang terdapat pada tabel 4 dan visualisasi *most\_similarity* pada gambar 8.

Tabel. 4 Hasil *Most\_similarity* Top 5 'hari'

Kata	Most Similarity
jahiliyah	0.430835545
agama	0.428290069
masa	0.422369421
generasi	0.417345613
zaman	0.400294393

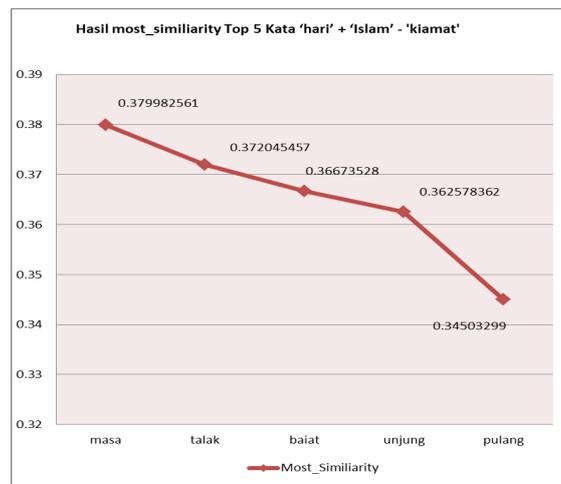


Gambar 9. Hasil visualisasi Most\_similarity 'hari' + 'islam'

Selanjutnya merupakan contoh *most\_similarity* top 5 dari kata ‘hari’ + ‘islam’ – ‘kiamat’ yang terdapat pada tabel 5 dan *visualisasi most\_similarity Top 5* ‘hari’ + ‘islam’ – ‘kiamat’ pada gambar 10.

Tabel. 5 Hasil *Most\_similarity Top 5* ‘hari’+ ‘islam’ – ‘kiamat’

Kata	Most Similarity
masa	0.379982561
talak	0.372045457
baiat	0.36673528
ujung	0.362578362
pulang	0.34503299



Gambar 10. Hasil *visualisasi* ‘hari’ + ‘islam’ – ‘kiamat’

### 3.3. Data Split

Data *split* merupakan validasi silang dengan data *train* sebanyak 6307 *samples* dan data *test* sebanyak 701 *samples* (6307 *samples* data *train* + 701 *samples* data *test* = 7008 *samples* data), dengan kata lain data *test* sebesar 10% dan 90% data digunakan untuk proses *training*.

### 3.4. Penggunaan Neural Network

Jenis *neural network* yang dianggap cocok pada penelitian ini adalah *multilayer perceptron* karena dapat membangun beberapa *layer* yang cukup sederhana dalam menyelesaikan masalah pengklasifikasian pada penelitian ini.

Dalam membangun sebuah model *multilayer perceptron* terdapat beberapa parameter tetap yang dipakai untuk penelitian yaitu model *sequential, dense* untuk menambahkan *layer*, nilai *input* yang dimasukan adalah 200 yang berasal dari jumlah ekstrak *feature selection* pada eksperimen, *relu* sebagai fungsi aktivasi pada *layer* pertama dan perantara, *sigmoid* sebagai fungsi aktivasi pada *output layer*, *optimizer* menggunakan *adam*, *loss* menggunakan *binary crossentropy* karena memiliki label 0 dan 1, nilai *epoch* sebesar 25 dan nilai *batch size* sebesar 1024.

Sedangkan untuk parameter yang diubah dalam penelitian ini adalah jumlah *layer* dan jumlah *neurons* pada *hidden layer* untuk mendapatkan nilai akurasi yang paling baik. Berikut adalah hasil akurasi pengujian dengan *neural network* dengan 3 eksperimen berbeda:

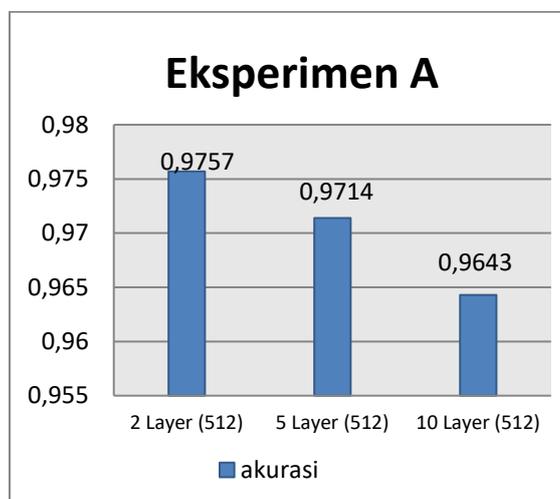
#### a. Eksperimen A

Dibawah ini adalah tabel 6 dan gambar 11 yang merupakan hasil eksperimen A, eksperimen A menggunakan *neurons* sebanyak 128 dengan beberapa variasi *layer* seperti dibawah ini.

Tabel 6. Hasil Eksperimen A

Layer	Neurons	Akurasi
-------	---------	---------

2	128	0,9757
5	128	0,9714
10	128	0,9643



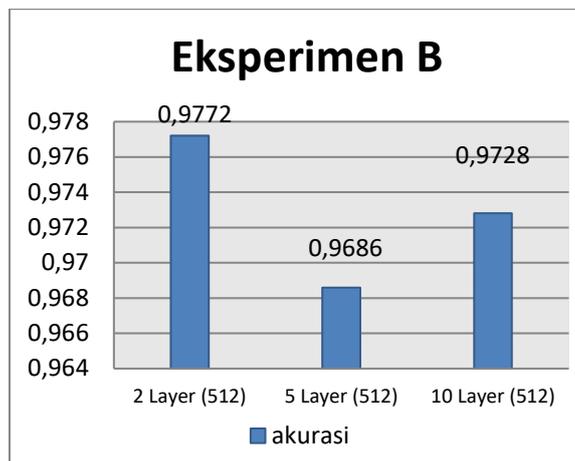
Gambar 11 Hasil Visualisasi Eksperimen A

**b. Eksperimen B**

Selanjutnya dibawah ini merupakan tabel hasil eksperimen B seperti pada tabel 7 dan visualisasi eksperimen B seperti pada gambar 12. Eksperimen B menggunakan *neurons* sebanyak 256 dengan beberapa variasi *layer* seperti dibawah ini.

Tabel 7 Hasil eksperimen B

Layer	Neurons	Akurasi
2	256	0,9772
5	256	0,9686
10	256	0,9728



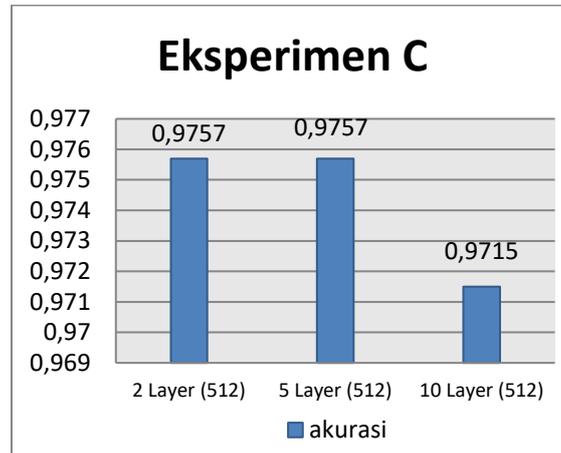
Gambar 12. Hasil Visualisasi Eksperimen B

**c. Eksperimen C**

Kemudian dibawah ini adalah tabel hasil eksperimen C seperti pada tabel 8 dan visualisasi eksperimen C seperti pada gambar 12. Eksperimen C menggunakan *neurons* sebanyak 512 dengan beberapa variasi *layer* seperti dibawah ini

Tabel 8. Hasil Eksperimen C

Layer	Neurons	Akurasi
2	512	0,9757
5	512	0,9757
10	512	0,9715



Gambar 12. Hasil Visualisasi Eksperimen C

Dari hasil penelitian diatas diperoleh rekapitulasi seperti pada tabel 9, dimana eksperimen B dengan jumlah dua layer dan 256 neurons menghasilkan akurasi sebesar 97,72% yang menjadi model arsitektur terbaik dalam penelitian ini.

Tabel 9. Rekapitulasi Hasil Eksperimen

Eksperimen	Layer	Neurons	Akurasi
A	2	128	0.9757
	5	128	0.9714
	10	128	0.9643
B	2	256	<b>0.9772</b>
	5	256	0.9686
	10	256	0.9728
C	2	512	0.9757
	5	512	0.9757
	10	512	0.9715

### 3.5. Evaluasi

Dari hasil visualisasi arsitektur eksperimen B didapat nilai *confusion matrix* seperti pada tabel 10 dibawah ini.

Tabel 10 Hasil *Confusion Matrix*

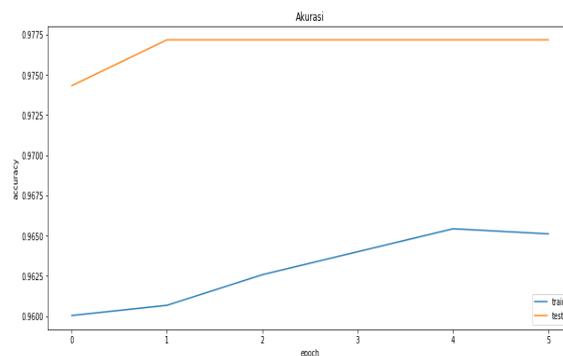
	Actual 0	Actual 1

Pred. 0	22 (TN)	13 (FP)
Pred. 1	3 (FN)	663 (TP)

Dari hasil *confusion matrix* dapat dihitung nilai akurasi dengan menggunakan fungsi (3.1).

$$\begin{aligned}
 \text{Accuracy} &= \frac{(TP+TN)}{(TP+TN+FP+FN)} \times 100 \% \\
 &= \frac{(663+22)}{(663+22+3+13)} \times 100 \% \\
 &= 97,72\%
 \end{aligned}$$

Berikut adalah gambar 13 yang merupakan visualisasi hasil *history* dari akurasi pada penelitian ini.



Gambar 13. Hasil *History* Akurasi

Pada *history* hasil akurasi terlihat nilai akurasi pada data training lebih kecil daripada nilai akurasi pada data *testing*, artinya data yang diolah pada proses *training* tidak *overfit* sehingga menghasilkan nilai yang baik pada data *testing*. Nilai akurasi maksimal yang dihasilkan oleh eksperimen B dengan dua *layer* dan 256 *neurons* sebesar 97,72 %.

#### 4. KESIMPULAN

Kesimpulan yang didapat pada penelitian ini berdasarkan analisis dan pembahasan antara lain:

- Dengan melakukan teks *pre-processing* yang baik dapat memberikan hasil yang optimal dalam menganalisa kata dalam kalimat.
- Penggunaan *word vector* sebagai *feature selection* dapat membantu *neural network* untuk mengekstrak sebuah kalimat.
- *Neural network* dapat digunakan sebagai pengklasifikasi suatu hadist masuk kedalam kategori larangan atau perintah.
- Arsitektur eksperimen B dengan menggunakan dua *layer* dan 256 *neurons* pada *hidden layer* menghasilkan nilai akurasi paling tinggi yaitu sebesar 97,72 %.

## REFERENSI

- Abdelaal, H. M., & Youness, H. A. (2019). *Hadith Classification using Machine Learning Techniques According to its Reliability*. 22(3), 259–271.
- Faraby, S. Al, Riviera, E., & Jasin, R. (2018). *Classification of hadith into positive suggestion, negative suggestion, and information*.
- Ghufran, M., & Faraby, S. Al. (2018). *Penerapan Particle Swarm Optimization Pada Feedforward Neural Network Untuk Klasifikasi Teks Hadis Bukhari Terjemahan Bahasa Indonesia*. 2(4), 165–173.
- Hastuti, K. (2012). Analisis komparasi algoritma klasifikasi data mining untuk prediksi mahasiswa non aktif. *Seminar Nasional Teknologi Informasi & Komunikasi Terapan*.
- Kusumaningrum, A., & Al-faraby, S. (2017). *Klasifikasi Informasi , Anjuran dan Larangan pada Hadits Shahih Bukhari menggunakan Metode Support Vector Machine* . 4(3), 5014–5023.
- M, H., & M.N, S. (2015). A Review on Evaluation Metrics for Data Classification Evaluations. *International Journal of Data Mining & Knowledge Management Process*, 5(2), 01–11. <https://doi.org/10.5121/ijdkp.2015.5201>
- Nuha, U., & Rochmawati, N. (2019). *Klasifikasi Kesahihan Hadits Berdasarkan Perawi Hadits Menggunakan Principal Component Analysis ( PCA ) dan Backpropagation Neural Network ( BPNN )*. 01, 138–143.
- Park, S., Byun, J., Baek, S., Cho, Y., & Oh, A. (2018). *Subword-level Word Vector Representations for Korean*. 2429–2438.
- Riviera, E., Jasin, R., & Al-faraby, S. (2017). *Klasifikasi Anjuran , Larangan dan Informasi pada Hadis Sahih Al-Bukhari berdasarkan Model Unigram menggunakan Artificial Neural Network ( ANN )*. 4(3), 4683–4694.
- Yuslan, M., & Bakar, A. (2018). Multi-Label Topic Classification of Hadith of Bukhari ( Indonesian Language translation ) using Information Gain and Backpropagation Neural Network. *2018 International Conference on Asian Language Processing (IALP)*, 344–350.
- Krieger, C. (1996). *Neural Networks in Data Mining*.
- Jarvis, E. D., Güntürkün, O., Bruce, L., Csillag, A., Karten, H., Kuenzel, W., ... & Striedter, G. (2005). Avian brains and a new understanding of vertebrate brain evolution. *Nature Reviews Neuroscience*, 6(2), 151-159.