Klasifikasi Topik Ayat Al-Qur'an Terjemahan Berbahasa Inggris Menggunakan Metode *Support Vector Machine* Berbasis *Vector Space Model* dan *Word2Vec*

Anisa Salama[1], Adiwijaya[2], Said Al Faraby[3]

[1,2,3]Fakultas Informatika, Universitas Telkom, Bandung
[1]salamaanis@students.telkomuniversity.ac.id, [2]adiwijaya@telkomuniversity.ac.id,
[3]saidalfaraby@telkomuniversity.ac.id

Abstrak

Tujuan diturunkannya Al-Qur'an adalah sebagai petunjuk hidup bagi umat manusia. Al-Qu'an memiliki kandungan makna dan hikmah disetiap ayatnya. Didalam Al-Qur'an terdapat ayat-ayat yang memiliki makna yang tersirat. Al-Qur'an mengandung beberapa topik yang antar surat Al-Qur'an dapat memiliki kemiripan topik dengan surat Al-Qur'an yang lainnya. Pada penelitian ini, dilakukan implementasi metode *Support Vector Machine* dan *Word2vec* pada terjemahan ayat Al-Qur'an berbahasa Inggris yang digunakan untuk pengklasifikasian berdasarkan topik. Kategori topik Al-Qur'an yang digunakan pada penelitian ini dibagi menjadi tiga yaitu perintah, larangan, dan lainnya. Dokumen tersebut diubah kedalam bentuk vektor dengan *tf-idf weighting* dan *Word2vec*. Vektor-vektor kata tersebut dipemetakan berdasarkan nilai kedekatan vektor antar kata pada dokumen. Selanjutnya metode *Support Vector Machine* digunakan untuk mengklasifikasikan topik Al-Qur'an dengan memberikan *hyperplane* pada tiap kategori. Hasil pengujian dari penelitian ini menunjukkan bahwa penerapan *Word2vec* dan *Support Vector Machine* mendapatkan nilai akurasi tertinggi sebesar 0.64 dengan jumlah data *training* sebesar 70% dari keseluruhan dataset.

Kata kunci : Topik Al-Qur'an, Klasifikasi, *Support Vector Machine, Word2vec*

Abstract

The purpose of the Qur'an is as a guide for human's life. Al-Qu'an has meaning and wisdom in every verse. In the Qur'an there are verses that have implied meaning. Al-Qur'an contains several topics which among the Surahs of the Qur'an can have a similarity to the topic of other Al-Qur'an's surah. In this research, the implementation of the Support Vector Machine and *Word2vec* method is used to classify based on topics in the English translation of Al-Qur'an verses. The used topic categories in this study are divided into three namely commands, prohibitions, and others. The document is converted into vector with tf-idf weighting and *Word2vec*. Then the word vectors are mapped based on the value of the proximity of the vectors between words in the document. Then Support Vector Machine method is used to classify the topic of the Qur'an by giving hyperplane in each category. The test results of this study showed that the implementation of *Word2vec* and Support Vector Machine has the highest accuracy results of 0.64 with the amount of *training* data is 70% of the entire dataset.

Keywords: Topics of Al-Qur'an, Classification, Support Vector Machine, *Word2vec*

1. Pendahuluan

Latar Belakang

Al-Qur'an merupakan kitab suci yang diturunkan Allah SWT kepada Nabi Muhammad SAW, yang mana merupakan salah satu mukjizat yang dimiliki oleh Nabi Muhammad SAW. Tujuan diturunkannya Al-Qur'an adalah sebagai petunjuk hidup bagi umat manusia. Membaca, memahami, dan mengamalkan Al-Qur'an pun bernilai ibadah, tak seperti membaca tulisan arab lainnya maupun hadist sekalipun. Al-Qur'an memiliki total 30 juz, 114 surah, dan 6236 ayat[1]. Al-Qu'an memiliki kandungan makna dan hikmah disetiap ayatnya. Didalam Al-Qur'an terdapat ayat-ayat yang memiliki makna yang tersirat[2]. Sehingga dalam memahami Al-Qur'an diperlukannya tafsir Al-Qur'an untuk mengkaji dan mengetahui makna yang terkandung pada ayat-ayat Al-Qur'an[2]. Selain itu dengan sejumlah ayat AlQur'an yang banyak tersebut, mengandung beberapa topik yang antar surat AlQur'an dapat memiliki kemiripan topik dengan surat Al-Qur'an yang lainnya. Dengan permasalahan tersebut dilakukan klasifikasi dengan penerapan metode *Support Vector machinel* (SVM) dan metode *Word2vec*.

Word Embedding merupakan teknik yang merepresentasikan sebuah kata menjadi sebuah vector atau array yang terdiri dari kumpulan angka. Metode Word Embbeding diperkenalkan pertama kali oleh Bengio et all [3]. Dua belas tahun sebelum Mikolov mempublikasikan penelitiannya, pada peneletian tersebut menjelaskan tentang konsep yang disebut sebagai "*learning a distributed representation for word*". Kemudian pada tahun 2008, Ronan dan Jason [4] memperkenalkan konsep *pre-trained* model dan menunjukkan hasil yang memuaskan, bahwa pendekatan dengan konsep *pre-trained* model tersebut sangat cocok diterapkan pada permasalahan NLP. Konsep

pre-trainned model tersebutlah yang digunakan Mikolov ketika menghasilkan sebuah *pre-trained* model yang diberi nama *Word2vec*[5].

Word2vec merupakan metode terbaru *Vector Space Model* yang biasa digunakan untuk mengolah data masukan dalam jumlah besar dengan model prediktif yang sangat efisien untuk mempelajari pola dari data mentah[6]. Dengan representasi vector metode bekerja dengan cara memetakan kata-kata dalam ruang vektor kontinu, dimana metode *Word2vec* akan memproses kata-kata yang serupa secara semantic dan dipetakan dalam ruang vektor yang berdekatan[7]. Sedangkan metode SVM digunakan untuk mengklasifikasi data sesuai dengan topik pada Al-Qur'an. Keunggulan SVM untuk mengolah data berdimensi besar dapat dimanfaatkan, karena sifat data teks yang biasanya berdimensi besar. Penelitian ini akan membahas tentang penerapan metode *Word2vec* dan *Support Vector Machine* menggunakan data terjemahan AL-Qu'an berbahasa Inggris.

Tujuan penelitian ini adalah menghasilkan model vektor ayat Al-Qur'an terjemahan Bahasa inggris yang dapat digunakan untuk melihat topik pada Al-Qur'an. Diharapkan dengan dukungan dari metode *Word2vec* dan *Support Vector Machine*, dapat dihasilkan sistem dengan performansi yang baik untuk memetakan terjemahan Al-Qur'an Berbahsa Inggris berdasarkan dengan topik yang sesuai dengan makna yang terkandung pada ayat-ayat Al-Qur'an. Data yang digunakan dalam penelitian ini adalah terjemahan ayat Al-Qur'an Bahasa Inggris yang terdiri dari tiga kategori yaitu perintah, larangan, dan informasi (lainnya).

Topik dan Batasannya

Berdasarkan latar belakang masalah yang telah diuraikan diatas, maka perumusan masalahnya adalah sebagai berikut :

1. Bagaimana mengimplementasikan metode *Support Vector Machine* dan *Word2vec* pada pengklasifikasian topik ayat Al-Qur'an terjemahan berbahasa Inggris?
2. Bagaimana performansi hasil klasifikasi topik ayat Al-Qur'an terjemahan berbahasa Inggris yang berbasis teks menggunakan metode *Support Vector Machine* dan *Word2vec*?

Adapun Batasan masalah pada tugas akhir ini adalah sebagai berikut:

1. Data set menggunakan data terjemahan Al-Qur'an berbahasa inggris yang berjumlah 781
2. Pengklasifikasian topik ayat alqur'an dibagi menjadi tiga kategori yaitu perintah, larangan, dan informasi

Tujuan

Tujuan dari penelitian tugas akhir ini adalah mengimplementasikan metode *Support Vector Machine* dan *Word2vec* pada pengklasifikasian tejemahan ayat Al-Qur'an berdasarkan topik (perintah, larangan, informasi) untuk menganalisis kinerja dan mengevaluasi performansi sistem yang dibuat dengan metode *Support Vector Machine* dan *Word2vec* tersebut.

Organisasi Tulisan

Organisasi penulisan yang digunakan dalam jurnal Tugas Akhir ini terbagi dalam beberapa pokok bahasan, yaitu: Bagian 2 membahas mengenai studi-studi terkait dengan tugas ahkir ini. Bagian 3 menjelaskan sistem yang akan digunakan untuk melakukan klasifikasi dan mengukur performansi. Bagian 4 menguraikan hasil serta analisis pengujian. Dan kesimpulan dari tugas akhir dibahas pada bagian 5.

2. Kajian Pustaka

2.1. TF-IDF Metode

TF-IDF merupakan metode yang digunakan pada perhitungan VSM untuk menghitung bobot pada suatu kata (term) yang terdapat pada dokumen. Metode ini menggunakan konsep perhitungan frekuensi kemunculan kata (TF) pada suatu dokumen dan inverse dari frekuensi yang mengandung kata tersebut (IDF) [9].
Perhitungan TF-IDF dirumuskan pada persamaan (1) berikut ini:

$$W_{(t,d)} = TF_{(t,d)} . \ Log \frac{N}{DFt} \quad (1)$$

Dimana :
$W_{(t,d)}$: bobot term t pada dokumen d
$TF_{(t,d)}$: jumlah kemunculan term t pada dokumen d
N : jumlah seluruh dokumen yang terambil oleh sistem
DF_t : jumlah dokumen yang memiliki term t dalam koleksi

2.2. *Word2vec*

Word2vec adalah hasil pembelajaran dari algoritma deep *learning*. Tiap kata diwakili oleh vector yang mempunyai nilai dan makna dari setiap kata tersebut. *Word2vec* merupakan bagian dari *word Embedding*. Metode *Word2vec* ini akan mengubah dokumen menjadi ruang vector kata. Metode *Word2vec* ini dapat diimplementasikan pada beberapa tugas *natural language processing* seperti klastering, klasifikasi, klasifikasi sentimen, dan lain-lain[10].

Word2vec mempunyai dua model, yaitu dengan CBOW (*Continuous Bag-of-Words Model*) dan *skipgram (Continuous Skip-gram Model)* [11]. Model CBOW dilakukan dengan cara memproyeksikan vektor kata-kata (Wt-1, wt+1) untuk memprediksi vektor kata target wt. Sedangkan, model Skip-Gram dilakukan dengan cara kebalikannya yaitu memprediksi vektor kata-kata yang ada pada konteks (Wt-1, wt+1) diberikan vektor kata tertentu wt. Model CBOW cenderung lebih mudah diterapkan terhadap informasi distribusional karena semua kata-kata konteks langsung diproses menjadi satu vektor sebelum akhirnya digunakan untuk memprediksi vektor kata target[10].

2.3. Feature selection based on Word2vec

Cara kerja feature selection dengan menggunakan *Word2vec* adalah dengan tahapan awal menghitung vector dokumen yang merepresentasikan dokumen tersebut. Kemudian menghitung jarak antara vector dokumen dengan vector kata pada tiap feature dalam dokumen tersebut.

2.3.1 *Feature Weight*

Sebelum menghitung feature weight, menentukan probabilitas dari feature dokumen:

$$Pc,j(w_i) = \frac{\sum_{l \in c} n_L(w_i)}{\Sigma_{c \epsilon N} \Sigma_{l \in} n_l(w_i)} \quad (2)$$

Dimana:
$Pc,j(w_i)$: probabilitas feature i dari dokumen j dalam kategori c.
$\sum nL(wi)\ l \in c$: jumlah berapa kali fitur ωi muncul pada dokumen.
$\Sigma c \epsilon N \Sigma l \in nl(w_i)$: berapa kali features muncul pada semua dokumen.

Tujuan dari perhitungan feature weight ini adalah untuk memfilter features yang lebih sering muncul dalam kategori tertentu tetapi lebih jarang dalam kategori lain.

2.3.2 Document Vector

Word2vec merupakan word vector representation yang dibuat oleh tim Google Tomas Mikolov buka pada tahun 2013[11][12]. Dengan menggunakan pelatihan model, alat ini mengekspresikan sebuah kata sebagai dimensi-panjang tetap-panjang vektor. Vektor kata yang terlatih memiliki akurasi yang tinggi dan waktu *training* yang rendah, dan banyak digunakan setelah diperkenalkan. Menggunakan vektor kata dari fitur yang diperoleh, vektor kata dari semua fitur dokumen dikalikan dengan bobot dan dijumlahkan untuk mendapatkan vektor dokumen:

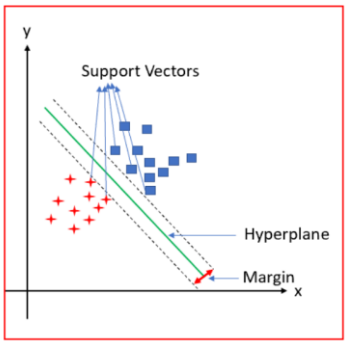
$$vec_{C,j} = \frac{1}{n_j} \Sigma_{i=1}^{m} \omega_i v(w_i) \quad (3)$$

Dimana:

$vec_{C,j}$	:vector dokumen j pada kelas j
$\frac{1}{nj}$	: jumlah feature pada dokumen j
$v(w_i)$	: vector kata dari feature w
ω_i	: bobot dari feature W_i pada dokumen j

2.4. *Support Vector Machine*

Support Vector Machine (SVM) merupakan *supervised learning*. SVM pertama kali diperkenalkan oleh Vapnik. SVM memiliki tujuan untuk mencari *Hyperplane* dengan margin terbesar antara *Hyperplane* dengan support vector (jarak terdekat dengan *Hyperplane*)[13]. Pada gambar 2 menunjukkan struktur SVM terdiri dari dua kelas yaitu kelas -1 dan kelas +1. Kedua kelas data tersebut dipisahkan oleh *Hyperplane*. Data yang terletak paling dekat dengan *Hyperplane* merupakan support vector dengan margin sebagai nilai jarak antara *Hyperplane* dengan *support vector* [13].



Gambar 1. Struktur SVM [8]

Pada metode SVM, proses *learning* pada hyperplane dalam SVM *linear* dilakukan dengan mengubah permasalahan menggunakan beberapa aljabar linier. Disitulah kernel berperan untuk membentuk *hyperplane* dalam sistem[14].

Tabel 1. Tabel Kernel SVM

Jenis Kernel	Definisi
Polynomial	$K(x_i, x_j) = (x_i, x_j + 1)^p$
Gaussian	$K(x_i, x_j) = = exp\ [-\|\| x_i - x_j \|\|^2 / 2\sigma^2]$ $K(x_i, x_j) = exp\ [-\gamma \|\| x_i - x_j \|\|^2, \gamma = 1 / 2\sigma^2]$
Sigmoid	$K(x_i, x_j) = \tan(\alpha x_i, x_j + \beta)$

2.5. Confusion matrix

Confusion matrix merupakan salah satu metode yang digunakan untuk mengukur akurasi dan performansi suatu model. *Confusion matrix* digunakan untuk masalah klasifikasi dimana output yang dihasilkan dapat dari dua tipe kelas ataupun lebih[15]. Pengukuran performansi dengan menggunakann *confusion matrix* memiliki 4 istilah yang merepresentasikan hasil proses klasifikasi yaitu True Positive (TP), True Negative (TN), False Positive (FP), False Negative (FN).

Berdasarkan nilai dari TP, TN, FP, FN tersebut dapat diperoleh nilai *Accuracy, Precision* dan *Recall*. Pengukuran nilai *Accuracy* berfungsi untuk mengetahui efektivitas keseluruhan dari sebuah sistem klasifikasi, nilai *accuracy* dapat diperoleh dengan menggunakan formula pada persamaan 1. Sementara itu, pengukuran nilai *Precision* menggambarkan jumlah data positif yang diklasifikasikan dengan benar dibagi dengan jumlah data yang diberi label positif oleh sistem, nilai *Precision* dapat diperoleh dengan menggunakan formula pada persamaan 2 dan nilai *recall* menggambarkan jumlah data positif yang diklasifikasikan dengan benar dibagi dengan jumlah data positif dalam data dokumen, nilai *recall* dapat diperoleh dengan menggunakan formula pada persamaan 3[16].

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

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

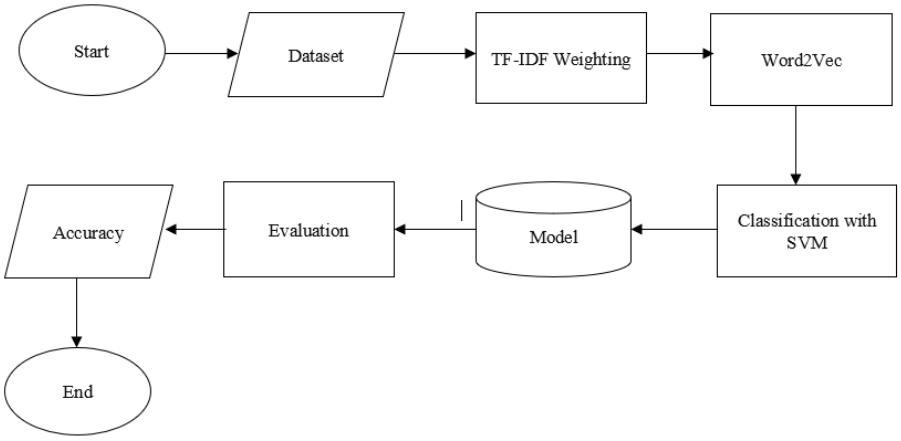
$$Recall = \frac{TP}{TP+FN} \quad (3)$$

$$F1\ Score = 2 * \frac{precision*Recall}{Precision+Recall} \quad (4)$$

Dimana:
TP : jumlah data positif yang terklasifikasi dengan benar oleh sistem
TN : jumlah data negatif yang terklasifikasi dengan benar oleh sistem
FN : jumlah data negatif yang tidak terklasifikasikan dengan benar oleh sistem
FP : jumlah data positif yang tidak terklasifikasikan dengan benar oleh sistem

3. Sistem yang Dibangun

Pada tugas akhir ini, sistem yang dibangun untuk melakukan penelitian terhadap data ayat Al-Qur'an terjemahan Bahasa inggris yang diklasifikasikan dengan menggunakan metode *Vector Space Model* dan *Word2vec* dengan input berupa dataset terjemahan Bahasa inggris Al-Qur'an dalam bentuk .csv serta hasil output berupa nilai performansi keakurasian dari proses klasifikasi tersebut. Secara garis besar, sistem yang akan digunakan terdiri dari tiga tahap, yaitu: tahap *preprocessing*, tahap klasifikasi, dan tahap *learning*. Adapun alur pada sistem ini dapat dilihat pada Gambar 3.



Gambar 2. Alur Sistem

A. Dataset

Dataset yang digunakan pada penelitian ini yaitu menggunakan dataset terjemahan Bahasa Inggris Al-Qur'an Al-Jalalain. Terjemahan Bahasa Inggris digunakan karena Bahasa Inggris merupakan Bahasa Internasional saat ini. Dataset berisi 780 ayat yang terbagi menjadi 3 kategori yaitu perintah, larangan, dan informasi.

B. TF-IDF Weighting

Pada proses perhitungan pembobotan kata menggunakan TF-IDF, bobot dari *TF-IDF Weighting* ini merupakan hasil perkalian dari TF dan juga IDF dengan melakukan perhitungan dengan formula pada persamaan (1).

.

C. *Word2vec*

Setelah vektor dihasilkan melalui *TF-IDF Weighting*, metode *Word2vec* melakukan pemetaan kata secara semantic berdasarkan keterkaitan antar kata pada dokumen.

D. *Learning*

Pada tahap *learning*, dalam membangun sistem klasifikasi dilakukan pemisahan data dari dataset yang dimiliki menjadi data *training*(pelatihan) dan data *testing* (pengujian) yang bertujuan untuk melatih sistem klasifikasi dan mengetahui *Hyperplane* yang paling optimal untuk mengklasifikasikan data pada tiga kelas yakni kelas larangan, kelas perintah, dan kelas lainnya. Proses klasifikasi menggunakan algoritma *Support Vector Machine* (SVM) model *linear*. Klasifikasi SVM ini dilakukan dengan menggunakan *library* pada pemograman *python*.

E. Evaluasi

Pada tahap evaluasi setelah melakukan pengujian terhadap data *training* dan data *testing*, dilakukan perhitungan performansi untuk mengetahui bagaimana kinerja sistem yang telah dibuat dan didapatkan hasil berupa nilai akurasi. Akurasi merupakan ketelitian classifier dalam melakukan klasifikasi, yaitu menyatakan persentase jumlah data uji yang diklasifikasikan dengan benar oleh classifier [17]. Pada penelitian ini kami mengevaluasi kinerja classifier menggunakan empat pengukuran, yaitu accuracy, *recall*, *Precision*, dan F1-measure[18].

4. Evaluasi

4.1 Analisis Pengujian Komposisi Data *Training* dan Data *Testing*

Pengujian perbandingan komposisi data *training* dan data *testing* ini dilakukan dengan menggunakan *percentage-splitting* untuk mengetahui pengaruh pada model yang dibuat oleh sistem klasifikasi. Pengujian ini dilakukan dengan menguji beberapa percentage-splitting dari data *training* dan data *testing* kemudian dibandingkan hasil dari masing-masing komposisi. Data yang ada dibagi menjadi 8 komposisi data *training* dan data *testing*, yaitu 20%-80%, 30%-70%, 40%-60%, 50%-50%, 60%-40%, 70%-30%, 80%-20%, dan 90%-10%. Pengujian percentage-splitting dilakukan pada data *training* berguna agar model yang didapatkan memiliki kemampuan dalam hal generalisasi untuk melakukan klasifikasi data. Hasil pengujian dapat dilihat pada gambar 4 :

Gambar 4. Grafik nilai akurasi *percentage-splitting*

Berdasarkan hasil pengujian yang telah dilakukan, dapat dilihat bahwa semakin banyak jumlah data pada data train maka semakin tinggi akurasi yang didapatkan. Hal tersebut dapat disebabkan karena dengan banyaknya data train, model yang terbentuk dapat menangani lebih banyak keberagaman data yang ada sehingga mampu mengklasifikasikan dengan lebih baik ketika melakukan testing. Pada gambar 4 dapat dilihat bahwa hasil *percentage-splitting* pada komposisi 70% - 30% memiliki nilai akurasi tertinggi, yaitu nilai akurasinya sebesar 0.64 dengan data train berjumlah 546 dan data test berjumlah 234. Sedangkan pada komposisi 80% - 20% memiliki nilai akurasi yang lebih rendah dibandingkan dengan komposisi 70% - 30%, yaitu nilai akurasi sebesar 0.57. Hal ini bisa disebabkan ketika jumlah data train yang digunakan lebih banyak namun karakteristik data train tersebut kurang baik (data memiliki perbedaan yang jelas untuk dapat diklasifikasi) maka model yang dibentuk juga akan menghasilkan model yang kurang baik juga sehingga hasil akurasi yang didapatkan menurun.

4.2 Analisis Pengujian Kernel SVM

Pengujian Kernel SVM dilakukan dengan menggunakan tiga kernel SVM yaitu kernel RBF, Kernel *Sigmoid*, dan Kernel *linear*. Pengujian kernel SVM yang dilakukan bertujuan untuk meliputi pengaruh kernel-kernel SVM yang digunakan untuk pengklasifikasian data, pengaruh kernel SVM pada nilai akurasi yang dihasilkan, serta pengaruh *preprocessing* pada data. Tahap *preprocessing* yang dilakukan pada pengujian ini meliputi *case folding, cleaning data,* tokenisasi, *dan stopwords removal.*

Tabel 2. Hasil Akurasi Pengujian Kernel SVM

Kernel	tf-idf,word2vec without preprocessing			tfidf,word2vec with preprocessing		
	Precision	Recall	F1-Score	Precision	Recall	F1-Score
RBF	0.13	0.33	0.18	0.12	0.33	0.18
Sigmoid	0.13	0.33	0.18	0.12	0.33	0.18
Linear	0.58	0.56	0.57	0.63	0.58	0.60

Berdasarkan hasil pengujian pada tabel 2 menunjukkan bahwa pada kernel model *linear* memberikan hasil yang lebih akurat dibandingkan model RBF dan *Sigmoid*. Dengan nilai *Precision, Recall,* dan *F1-Score* terbaik pada percobaan dengan menggunakan *preprocessing* dan nilai *Precision, Recall,* dan *F1-Score* yang diperoleh yaitu sebesar 0.63, 0.58, dan 0.60. Nilai *Precision* menunjukkan bahwa terdapat 147 data yang dideteksi oleh sistem dan diklasifikasikan pada label benar dari jumlah data testing sebesar 234 data, sedangkan nilai *recall* menunjukkan bahwa terdapat 135 data dengan label benar yang terklasifikasikan sesuai dengan label yang sesungguhnya. Nilai *F1-Score* merupakan perbandingan rata-rata *Precision* dan *Recall* yang dibobotkan. Hal ini diakibatkan karena model *sigmoid* akan lebih cocok dipergunakan untuk model data yang bersifat binary, sedangkan model RBF lebih cocok digunakan apabila data memiliki sifat nilai yang berkelompok secara radial atau periodikal.

4.3 Analisis Hasil Pengujian

Analisis dari pengujian sistem setelah dilakukan beberapa skenario pengujian diatas adalah bahwa pada pengujian komposisi data *training* dan data *testing* (*percentage splitting*), jumlah data yang digunakan untuk *training* dan *testing* mempengaruhi hasil akurasi yang didapatkan. Jumlah data *training* pada tiap kategorinya juga dapat mempengaruhi hasil akurasi pada tiap persentase pengujian pada data *training*. Setelah dilakukan pengujian pada beberapa *percentage splitting*, didapatkan persentase yang paling optimal pada jumlah data *training* sebesar 70%.

Pada pengujian kedua dilakukan pengujian kernel-kernel SVM yaitu RBF, *sigmoid*, dan *linear*. Didapatkan hasil seperti yang ditunjukkan pada tabel 2 bahwa model SVM *linear* menghasilkan nilai *Precision, Recall,* dan *F1-Score* yang lebih baik dibandingkan dengan model RBF dan *sigmoid*. Sehingga pada penerapan klasifikasi teks Al-Qur'an model SVM *linear* akan lebih baik untuk digunakan. Pada tabel 2 juga menunjukkan hasil bahwa *Precision, Recall,* dan *F1-Score* terbaik terdapat pada sistem yang menggunakan tahap *preprocessing*.

5. Kesimpulan dan Saran

5.1 Kesimpulan

Berdasarkan hasil pengujian dan analisis yang telah dilakukan, maka dapat ditarik kesimpulan sebagai berikut :

1. Pengujian *percentage splitting* pada data *training* mempengaruhi hasil akurasi. Pada pengujian ini didapatkan hasil yang paling optimal pada komposisi 70%-30% dengan nilai akurasi 0.64.
2. Model SVM *linear* terpilih menjadi model SVM yang terbaik untuk diterapkan pada klasifikasi terjemahan ayat Al-Qur'an pada penelitian ini, dikarenakan model *linear* memiliki hasil akurasi yang tertinggi dibanding dengan model RBF dan *sigmoid*.
3. Penerapan metode *Word2vec* dalam klasifikasi topik pada terjemahan ayat Al-Qur'an memiliki hasil akurasi yang tidak cukup baik dengan persentase nilai akurasi kurang lebih 64% untuk data *training* sebesar 576 ayat.

5.2 Saran

1. Melakukan pengujian dengan dataset yang lebih besar dan kategori yang lebih luas untuk mengetahui dan membuktikan seberapa efektif metode *Support Vector Machine* dan *Word2vec* dalam melakukan pengklasifikasian teks.
2. Upaya lain pada penelitian yang mendatang ialah dengan menggunakan metode *word Embedding* lainnya seperti *FastText* dan *Glove*.

Daftar Pustaka

[1] Syaamil Qur'an, "Cordova Al-Qur'an and Translation" ,2004.

[2] Prayogo, A. H., and Adiwijaya. (2017). On the Feature Extraction for the English Holy Quran Tafseer Text Classification. 10.

[3] Yoshua Bengio, Ducharme Rejean, Vincent Pascal & Janvin Christian. "A Neural Probabilistic Language Model", 2003. http://www.jmlr.org/papers/volume3/bengio03a/bengio03a.pdf

[4] Collobert Ronan, & Weston Jason. "A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask *Learning*". 2008. https://ronan.collobert.com/pub/matos/2008_nlp_icml.pdf

[5] Tomas Mikolov, Greg Corrado, Kai Chen & Jeffrey Dean. "Efficient Estimation of Word Representations in Vector Space", 2013. https://arxiv.org/pdf/1301.3781.pdf

[6] M. Faruqui and C. Dyer, "Improving Vector Space Word Representations Using Multilingual Correlation", Carnegie Mellon University, 2014 hal.236-244

[7] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space, In Proceedings of Workshop at ICLR, 2013.

[8] Khandelwal, Renu. 2018. *"Support Vector Machine"*. Medium Corporation. https://medium.com/datadriveninvestor/support-vector-machines [diakses 9 Agustus 2019]

[9] Robertson, Stephen., 2005, Understanding Inverse Document Frequency: On Theoretical Arguments for IDF, England Journal of Documentation, Vol. 60, pp. 502 – 520.

[10] Tian, Wenfeng., and Li Hongguang., 2018, A Method of Feature Selection Based on *Word2vec* in Text Categorization: {roceeding of the 37th Chinese Control Conference.

[11] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space, In Proceedings of Workshop at ICLR, 2013.

[12] https://code.google.com/archive/p/*Word2vec*.

[13] Le, Q. Mikolov, T. Distributed Representations of Sentences and Documents. Google Inc, 1600 Amphitheatre Parkway, Mountain View, CA 94043

[14] Yulietha, I. M., Faraby, S. A., & Adiwijaya. (2017). Klasifikasi Sentiment Review Film Menggunakan *Support Vector Machine*. 10.

[15]] E. Prasetyo, Data Mining: Konsep dan Aplikasi menggunakan Matlab, 1 ed. Yogyakarta: Andi Offset, 2012. 17

[16] M. Sokolova dan G. Lapalme, "A sistematic analysis of performance measures for classification tasks," Inf. Process. Manag., vol. 45, no. 4, hal. 427–437, 2009

[17] V. Labatut and H. Cherifi, "Evaluation of Performance Measures for Classifiers Comparison," Ubiquitous Comput. Commun. Journal, pp. 621-34, 2011.

[18] F. Guillet and H. J. Hamilton, Quality Measures in Data Mining, Berlin: Springer, 2007.