

Klasifikasi Teks Multi Label pada Hadis Terjemahan Bahasa Indonesia Menggunakan Chi-Square dan SVM

Fakhri Taufiqurrahman¹, Said Al Faraby², Mahendra Dwifabri Purbolaksono³

^{1,2,3} Universitas Telkom, Bandung

¹fakhritaufig@students.telkomuniversity.ac.id,

²saidalfaraby@telkomuniversity.ac.id, ³mahendradp@telkomuniversity.ac.id

Abstrak

Hadis yaitu pedoman dalam islam setelah Al-Quran yang dijadikan sebagai sumber hukum dalam islam. Akan tetapi terdapat permasalahan ketika menentukan hadis mana saja yang merupakan anjuran, larangan, dan informasi. Oleh karena itu dibutuhkan klasifikasi teks untuk mengelompokkan hadis ke dalam satu atau lebih dari anjuran, larangan, dan informasi, yang disebut dengan klasifikasi *multi-label*. Permasalahan dalam klasifikasi teks yaitu terdapat banyak fitur, sehingga perlu dilakukan seleksi fitur dengan tujuan memangkas fitur yang ada kemudian menentukan fitur paling berpengaruh terhadap kelas target. Pada penelitian ini *Chi-Square* digunakan untuk melakukan seleksi fitur dan *Support Vector Machine (SVM)* untuk melakukan klasifikasi teks. Dengan menggunakan metode evaluasi performa *Macro F1-Score* hasil yang didapat ketika menggunakan *Chi-Square* dan SVM yaitu sebesar 75.32%.

Kata kunci : hadis, klasifikasi teks, *multi-label*, *chi-square*, *support vector machine*.

Abstract

Hadith is a guide in Islam after the Al-Quran which is used as a source of law in Islam. However, there is a problem when determining which hadith are suggestions, prohibitions, and information. Therefore, text classification is needed to classify hadith into one or more suggestions, prohibitions, and information, which is called multi-label classification. The problem in text classification is that there are many features, so it is necessary to select features with the aim of trimming the existing features and then determining which features have the most influence on the target class. In this research, Chi-Square is used to perform feature selection and Support Vector Machine (SVM) to perform text classification. By using the Macro F1-Score performance as evaluation method, the results obtained when using Chi-Square and SVM is 75.32%. **Keywords:** hadith, text classification, multi-label, chi-square, support vector machine.

1. Pendahuluan

Latar Belakang

Hadis adalah kumpulan ucapan, perbuatan, dan ketetapan dari Nabi Muhammad yang merupakan salah satu sumber pedoman didalam islam setelah Al-Quran [1]. Berdasarkan pengertiannya, hadis berfungsi sebagai salah satu norma hukum dalam islam yang di dalamnya terdapat penjelasan mengenai anjuran, larangan, dan informasi yang bermanfaat. Memahami hadis sangat penting karena dapat menuntun umat muslim dalam berperilaku sesuai dengan ajaran Nabi Muhammad. Namun untuk memahami hadis, terdapat kendala dalam menentukan apakah hadis tersebut merupakan anjuran, larangan, atau informasi. Untuk mengelompokkan hadis kedalam salah satu atau lebih dari anjuran, larangan, dan informasi dibutuhkan suatu sistem yang dapat mengelompokkannya.

Terdapat berbagai macam cara yang mampu dilakukan, salah satunya adalah mengelompokkan hadis dengan menggunakan klasifikasi teks karena klasifikasi teks dapat mengelompokkan berbagai dokumen kedalam kelasnya [2]. Dengan menggunakan klasifikasi teks dapat dilakukan analisis untuk mendapatkan sebuah informasi dari dokumen [3]. Namun, salah satu permasalahan dalam klasifikasi teks adalah banyaknya fitur yang ada di dalam dokumen [4]. Sehingga diperlukan proses untuk memilih fitur dalam klasifikasi teks agar mendapatkan fitur yang berpengaruh terhadap kelas target dan fitur yang berpengaruh terlebih dahulu ditentukan sebelum membuat sebuah model klasifikasi.

Feature selection dapat digunakan untuk menentukan fitur yang berpengaruh dan dengan menggunakan *feature selection* dapat meningkatkan efisiensi dan performa dalam pengklasifikasian teks [5]. Terdapat berbagai macam metode seleksi fitur yang dapat dimanfaatkan seperti *Information Gain*, *Mutual Information*, *Chi-Square*, dan *Relief*. Penggunaan *feature selection* dengan menggunakan metode *Information Gain* dan *Chi-Square* menghasilkan hasil yang baik dan dapat meningkatkan performa pada klasifikasi teks [6]. Terdapat berbagai macam metode pengklasifikasi yang dapat dimanfaatkan, salah satunya adalah SVM yang dapat memberikan kinerja yang baik terhadap dataset dengan banyak fitur [7]. Metode ini terbukti menghasilkan performa yang lebih baik dibanding metode NB dan KNN ketika digunakan untuk melakukan klasifikasi teks [8].

Berdasarkan penjelasan sebelumnya, pada penelitian ini dibangun sistem klasifikasi teks *multi-label* terhadap hadis terjemahan bahasa Indonesia ke dalam salah satu atau lebih topik yaitu anjuran, larangan, informasi, yang

disebut dengan klasifikasi *multi-label* menggunakan metode *Chi-Square* untuk melakukan *feature selection* dan metode *Support Vector Machine* (SVM) untuk melakukan klasifikasi.

Tujuan

Pada penelitian ini memiliki tujuan yaitu membangun sistem yang dapat melakukan klasifikasi topik ke dalam satu atau lebih dari anjuran, larangan, dan informasi yang disebut klasifikasi *multi-label* terhadap data hadis Bukhari yang sudah diterjemahkan kedalam bahasa Indonesia dengan *Chi-Square* dan *Support Vector Machine*. Kemudian melakukan analisis terhadap pengaruh penggunaan metode *feature selection Chi-Square* terhadap performa yang dihasilkan, menganalisis pengaruh penggunaan *stopword removal* terhadap performa yang dihasilkan, dan menganalisis pengaruh penggunaan kernel pada metode SVM terhadap performa yang dihasilkan.

Organisasi Tulisan

Setelah pendahuluan, selanjutnya pada bagian kedua akan dijelaskan penelitian terkait. Selanjutnya menerangkan mengenai sistem yang dibuat. Pada tahap empat menerangkan hasil analisis dan evaluasi sistem yang dibangun. Dan pada bagian terakhir yaitu bagian kelima menjelaskan tentang kesimpulan kemudian saran atas hasil penelitian.

2. Studi Terkait

Penelitian yang melakukan klasifikasi terhadap data hadis terjemahan bahasa Indonesia pernah dilakukan sebelumnya. Said Al Faraby [9] dimana melakukan klasifikasi satu label menggunakan metode *Support Vector Machine* dan *Neural Network*. Dilakukan percobaan dengan menggunakan kernel *linear*, *polynomial*, dan RBF yang terdapat pada metode *Support Vector Machine* dengan jumlah dataset sebanyak 1064. Penelitian ini menunjukkan bahwa metode *Support Vector Machine* dengan kernel *linear* mendapatkan performa *F1-Score* terbaik dengan nilai 88%. M. Y. Abu Bakar [10] dimana melakukan klasifikasi satu label dan *multi-label* kemudian metode yang digunakan yaitu *Backpropagation Neural Network* dan *Information Gain*. Penelitian ini mendapatkan nilai *F1-Score* 65.27% dari data *single label* dan 88.42% dari data *multi-label*.

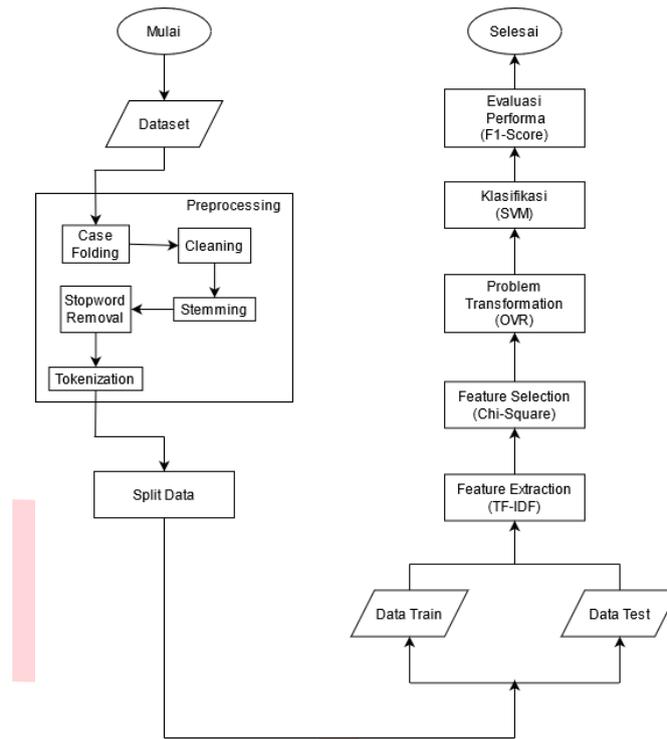
Penelitian dimana metode *Support Vector Machine* digunakan untuk klasifikasi teks pernah dilakukan sebelumnya. M. S. Khorsheed [11], melakukan klasifikasi teks terhadap data Saudi *press agency* menggunakan beberapa metode klasifikasi seperti *Support Vector Machine*, C4.5, dan *Naïve Bayes*. Hasil yang didapat ketika menggunakan metode pemilihan kata CHI, metode *Support Vector Machine* memiliki nilai akurasi paling baik yaitu 72.15%. Penelitian lainnya yang dilakukan oleh A. W. Pradana [12] yang melakukan klasifikasi teks terhadap data tweet berbahasa Indonesia dengan metode TF-IDF dan SVM menghasilkan nilai akurasi 81.44%.

Penggunaan metode *feature selection* terhadap data teks pernah dilakukan oleh Y. Zhai [13]. Penelitian tersebut melakukan klasifikasi teks terhadap data korpus komputer dan buku kedalam dua kelas menggunakan metode seleksi fitur *Information Gain* dan *Chi-Square*. Metode *Support Vector Machine* digunakan ketika melakukan klasifikasi teks. Hasil yang diperoleh menunjukkan bahwa penggunaan *Chi-Square* mendapatkan nilai akurasi yang lebih tinggi, yaitu 86.31% untuk korpus komputer dan 95.29% untuk korpus buku. Penelitian lain [14], ketika melakukan klasifikasi teks terhadap data *Reuters-21578 Collections* dengan menggunakan metode *feature selection* WCP, CHI, IG, dan MI dengan metode klasifikasi NB menunjukkan nilai *F1-Measure* maksimal yang didapat adalah menggunakan WCP sebesar 96%, diikuti oleh CHI sebesar 95%, IG sebesar 90%, dan MI sebesar 90%.

Hadis yang digunakan pada penelitian ini memiliki jumlah data sebanyak 7007 dan memiliki kelas salah satu atau lebih dari anjuran, larangan, dan informasi, yang disebut dengan klasifikasi *multi-label*. Metode *Support Vector Machine* dan *Chi-Square* digunakan dalam penelitian ini. Namun, pada dasarnya *Support Vector Machine* hanya dapat berfungsi untuk klasifikasi biner. Untuk mengatasi hal tersebut digunakan metode *one-versus-rest* untuk mengubah permasalahan klasifikasi *multi-label* menjadi beberapa permasalahan klasifikasi biner [15].

3. Sistem yang Dibangun

Sistem yang dibangun mampu mengerjakan klasifikasi teks *multi-label* data hadis yang sudah diterjemahkan kedalam bahasa Indonesia terhadap salah satu atau lebih dari anjuran, larangan, dan informasi. Metode seleksi fitur *Chi-Square* dan metode klasifikasi *Support Vector Machine* digunakan pada sistem. Sistem terbagi menjadi tahap *preprocessing*, *split data*, *feature extraction*, *feature selection*, *problem transformation*, klasifikasi, dan evaluasi performa. Gambar 1 menunjukkan alur kerja sistem yang dibuat secara umum.



Gambar 1 Gambaran Umum Sistem

Dataset

Dataset yang digunakan adalah dataset hadis Bukhari yang telah diterjemahkan kedalam bahasa Indonesia. Dataset yang digunakan berjumlah 7007 data hadis yang memiliki tiga kelas berdasarkan topik yaitu anjuran, larangan, informasi, dan bersifat *multi-label*. Jumlah hadis kelas anjuran, larangan, dan informasi yang memiliki label satu adalah 1337, 844, dan 6633. Contoh dataset hadis dapat dilihat pada Tabel 1.

Tabel 1 Contoh Dataset

Hadis	Anjuran	Larangan	Informasi
Mencerca orang muslim adalah fasiq dan memeranginya adalah kufur.	0	0	1

Preprocessing

Tahap pertama yang dilakukan adalah *preprocessing* terhadap dataset dengan tujuan agar menjadikan kata lebih terstruktur dan dapat digunakan untuk proses klasifikasi. Terdapat tujuh data yang tidak memiliki kelas atau *missing values*, sehingga data tersebut dihapuskan dari dataset. Berikut merupakan tahapan *preprocessing* yang dilakukan:

1. *Case folding*
 Pada tahap ini data teks diubah kedalam bentuk yang sama dengan mengubah setiap kata menjadi huruf kecil agar memiliki tingkat yang sama.
2. *Cleaning*
 Pada tahap *cleaning*, penghapusan tanda baca dan bilangan dilakukan karena tidak akan digunakan pada proses klasifikasi.
3. *Stemming*
 Pada tahap *stemming* dilakukan pengembalian kata kedalam bentuk dasar kata tersebut. Dalam bahasa Indonesia sebuah kata biasanya memiliki imbuhan seperti “men”, “ter”, “ber”, “ke”, dan sebagainya.
4. *Stopword removal*
 Tahap ini adalah proses untuk mengeliminasi kata yang sering ditemukan dan tidak mempunyai pengaruh seperti kata “walau”, “oh”, “yang”, “dll”, dan sebagainya.
5. *Tokenization*
 Tahap *tokenization* adalah pemotongan kalimat berdasarkan penyusun setiap katanya.

Tabel 2 Contoh Proses *Preprocessing*

Proses	Kalimat Masukan	Hasil Kalimat
Case Folding	Mencerca orang muslim adalah fasiq dan memerangnya adalah kufur.	mencerca orang muslim adalah fasiq dan memerangnya adalah kufur.
Cleaning	mencerca orang muslim adalah fasiq dan memerangnya adalah kufur.	mencerca orang muslim adalah fasiq dan memerangnya adalah kufur
Stemming	mencerca orang muslim adalah fasiq dan memerangnya adalah kufur	cerca orang muslim adalah fasiq dan merang adalah kufur
Stopword Removal	cerca orang muslim adalah fasiq dan merang adalah kufur	cerca orang muslim fasiq merang kufur
Tokenization	cerca orang muslim fasiq merang kufur	'cerca', 'orang', 'muslim', 'fasiq', 'merang', 'kufur'

Split Data

Tahap selanjutnya adalah membagi dataset menjadi data latih dan data uji menggunakan metode *Split Data*. Pembagian data latih dan data uji dilakukan dengan komposisi 80% berbanding 20%. Jumlah data latih adalah 5600 data hadis dan jumlah data uji adalah 1400 hadis.

Feature Extraction

Feature extraction digunakan untuk memberikan pembobotan kata pada setiap dokumen dan menjadikannya bentuk vektor. Metode TF-IDF digunakan pada tahap *feature extraction* yang bekerja dengan cara menyatakan jumlah kemunculan kata terhadap jumlah data. Metode ini digunakan dengan tujuan mengukur bobot kata dalam sebuah data. TF menyatakan total kemunculan *term* pada sebuah data, dimana IDF menyatakan nilai seberapa penting suatu *term* terhadap data [16]. Semakin tinggi frekuensi kemunculan sebuah kata maka semakin tinggi nilai TF, namun pada IDF semakin rendah frekuensi kemunculan sebuah kata, semakin tinggi nilainya. Untuk menghitung nilai TF-IDF dapat menggunakan persamaan 1 [17]:

$$W(d, t) = TF(d, t) * \frac{N}{df(t)} \quad (1)$$

Keterangan :

$W(d, t)$ = Pembobotan *term* t pada data d

$TF(d, t)$ = Total kemunculan *term* t dalam data

d N = Total data keseluruhan

$df(t)$ = Total data dimana terdapat *term* t

Sebagai contoh Tabel 3 merupakan penggunaan metode *TF-IDF* dari potongan kalimat yang ada pada dataset:

Tabel 3 Contoh TF-IDF

Kata	tf		df	idf	tf-idf	
	Kalimat 1	Kalimat 2		log(n/df)	Kalimat 1	Kalimat 2
ambil	1	0	1	0.301	0.301	0
baca	1	1	2	0.0	0	0
sandar	0	1	1	0.301	0	0.301

Feature Selection

Metode *Chi-Square* digunakan untuk memilih fitur yang dianggap penting untuk digunakan pada proses klasifikasi dan dapat menghapus fitur yang tidak berpengaruh terhadap kelas target. *Chi-Square* adalah salah satu metode *feature selection* yang bekerja dengan cara menghitung hubungan antara fitur yang ada terhadap kelas

target [13]. Persamaan yang digunakan untuk menghitung hubungan antar fitur dan kelas target pada metode *Chi-Square* yaitu [18]:

$$X^2(t_k, c_i) = \frac{N(AD - CB)^2}{(A + C)(B + D)(A + B)(C + D)} \quad (4)$$

Keterangan:

N : Total data hadis

A : Total data di kelas c_i yang berisi kata t_k

B : Total data yang berisi kata t_k yang bukan merupakan kelas c_i

C : Total data di kelas c_i yang tidak berisi kata t_k

D : Total data yang bukan merupakan kelas c_i dan tidak berisi kata t_k .

Pada Tabel 4 dapat dilihat contoh perhitungan metode *feature selection Chi-Square* dari potongan kalimat yang ada pada dataset:

Tabel 4 Contoh Perhitungan Chi-Square

Hadis	Anjuran	Larangan	Informasi
cerca orang muslim fasiq merang kufur	0	0	1
jangan kamu pindah pergi hingga kamu dengar suara cium bau	0	1	0

Berdasarkan contoh dataset pada Tabel 4, nilai *Chi-Square* yang akan dicari adalah untuk kata “cerca” dan untuk kelas “Informasi”. Berikut merupakan contoh perhitungan yang dilakukan:

$$X^2(\text{cerca, Informasi}) = \frac{2(1 \cdot 1 - 0 \cdot 0)^2}{(1+0)(0+1)(1+0)(0+1)} = \frac{2}{1} = 2$$

Pada Tabel 5 merupakan hasil dari lima fitur dengan nilai *chi-square* terbaik dari data hadis yang digunakan, dengan total jumlah fitur yang ada pada data latihan sebanyak 8493:

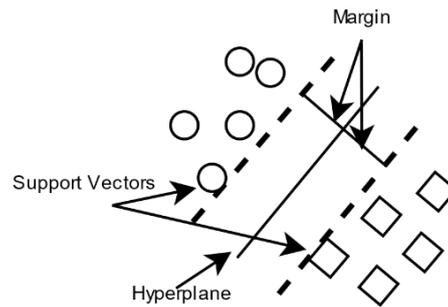
Tabel 5 Nilai Chi-Square

Fitur	Nilai Chi-Square
Jangan	244.46
Larang	186.19
Hendak	60.13
Kalian	56.34
Jual	45.45

Dapat dilihat pada Tabel 5, fitur atau kata “jangan” memiliki nilai *Chi-Square* tertinggi dibanding dengan kata lain. Hal ini menandakan kata tersebut dapat merepresentasikan sebuah kelas dengan baik, dimana kata jangan menggambarkan sebuah larangan.

Support Vector Machine (SVM)

Support Vector Machine yaitu salah satu metode pembelajaran mesin dan termasuk ke dalam *supervised learning*. *Support Vector Machine* berjalan dengan menentukan *hyperplane* yang paling baik dengan memaksimalkan margin antara kelas yang saling berdekatan [19]. Pada gambar 2 dapat dilihat ilustrasi dari metode *Support Vector Machine*.



Gambar 2 Ilustrasi SVM.

Hyperplane merupakan fungsi yang berperan untuk memisahkan antar class label yang ada pada data. Hyperplane yang paling baik dapat ditemukan dengan menghitung margin antar kelas dan mencari maksimalnya. Untuk menghitung hyperplane dapat dijelaskan sebagai berikut [20]:

Sampel pelatihan berupa $\{x_i, y_i\}, \dots, \{x_n, y_n\}$, yang dimaksud $x_i \in R^m$ yaitu atribut serta $y_i \in \{-1,1\}$ yaitu class label [20]. Untuk menghitung hyperplane dapat menggunakan persamaan $w^T x + b = 0$, dimana $w \in R^m$ dan b adalah skalar. Untuk kasus dengan data yang dapat dipisahkan secara linear dengan baik maka akan menghasilkan hyperplane yang optimal dengan menggunakan persamaan 2.

$$\begin{aligned} \text{minimize: } L(w) &= \frac{1}{2} \|w\|^2 \\ y_i(w^T x_i + b) &\geq 1, i = 1, \dots, N. \end{aligned} \tag{2}$$

Untuk kasus sampel yang tidak dapat dipisahkan secara linear dengan baik, perlu dilakukan pengoptimalan dengan menggunakan konsep soft margin. Perhitungan yang dilakukan dapat dilihat pada persamaan 3.

$$\begin{aligned} \text{minimize: } L(w, \xi_i) &= \frac{1}{2} \|w\|^2 + C \sum_{i=1}^N \xi_i \\ y_i(w^T x_i + b) &\geq 1 - \xi_i, \xi_i \geq 0, i = 1, \dots, N \end{aligned} \tag{3}$$

Dimana ξ adalah slack variables yang menggambarkan soft margin, serta C adalah parameter yang digunakan untuk menyeimbangkan margin dan error.

Untuk kasus dengan data yang lebih kompleks dapat menggunakan fungsi kernel pada metode SVM. Penggunaan kernel adalah dengan memetakan data x yang berasal dari fungsi $\Phi(x)$ terhadap ruang dimensi, dimana ruang dimensi yang dimiliki lebih tinggi sehingga hyperplane yang digunakan untuk memisahkan antar kelas dapat dibentuk. Berbagai macam kernel yang ada sebagaimana dirangkum pada Tabel 6.

Tabel 6 Jenis Kernel

Jenis Kernel	Persamaan
Radial Basis Function (RBF)	$K(x_i, x_j) = \exp(-\gamma \ x_i - x_j\ ^2)$
Polynomial	$K(x_i, x_j) = (\gamma(x_i^T x_j + r))^d$
Linear	$K(x_i, x_j) = x_i^T x_j$

Kemudian hasil klasifikasi dari sampel data x diperoleh dari persamaan 4.

$$f(x) = w^T \Phi(x) + b = \sum_{i=1}^N \alpha_i \gamma_i K(x_i, x) + b \tag{4}$$

One-Versus-Rest

Pada dasarnya *Support Vector Machine* hanya mampu mengerjakan klasifikasi biner yaitu klasifikasi dengan data yang terdiri dari dua kelas. Namun pada penelitian ini data yang digunakan memiliki tiga kelas yaitu anjuran, larangan, informasi dan bersifat *multi-label* sehingga metode *one-versus-rest* digunakan untuk mengatasi permasalahan tersebut. Metode ini bekerja dengan cara memisahkan satu kelas dengan kelas lainnya yang terdapat pada dataset dan mengubah permasalahan pada klasifikasi *multi-label* menjadi beberapa permasalahan klasifikasi biner. Model klasifikasi dilatih sebanyak jumlah kelas yang terdapat pada dataset dengan semua data latih kelas label positif dan kelas label negatif [15]. Contoh penggunaan *one-versus-rest* dapat dilihat pada Tabel 7, 8, dan 9.

Tabel 7 Contoh OVR pada Kelas Anjuran

Hadis	Anjuran
Mencerca orang muslim adalah fasiq dan memerangnya adalah kufur.	0

Tabel 8 Contoh OVR pada Kelas Larangan

Hadis	Larangan
Mencerca orang muslim adalah fasiq dan memerangnya adalah kufur.	0

Tabel 9 Contoh OVR pada Kelas Informasi

Hadis	Informasi
Mencerca orang muslim adalah fasiq dan memerangnya adalah kufur.	1

Evaluasi Performa

Untuk mengukur performa dari sebuah klasifikasi dapat menggunakan *confusion matrix* [6]. Dari *confusion matrix* tersebut nilai *F1-Score* dapat diperoleh dengan menggabungkan nilai *precision* dan *recall*. *Precision* adalah jumlah kelas yang diprediksi benar dan terbukti benar. *Recall* adalah jumlah kelas yang memang benar dan berhasil diprediksi benar. Terdapat tiga kelas yang dikategorikan maka nilai *precision* dan *recall* mengikuti jumlah *class* tersebut. Pada persamaan 5, 6, dan 7 merupakan persamaan dari metode evaluasi performa yang digunakan.

$$Precision = \frac{(TP)}{(TP + FP)} \quad (5)$$

$$Recall = \frac{(TP)}{(TP + FN)} \quad (6)$$

$$F1 - Score = \frac{(2 * Precision * Recall)}{(Recall + Precision)} \quad (7)$$

Keterangan :

TP (*True Positive*) : *Class* hadis yang diperkirakan positif serta benar positif

TN (*True Negative*) : *Class* hadis yang diperkirakan negatif serta benar negative

FP (*False Positive*) : *Class* hadis yang diperkirakan positif namun salah

FN (*False Negative*) : *Class* hadis yang diperkirakan negatif namun salah.

4. Evaluasi

4.1 Hasil Pengujian

Pada penelitian ini pembagian data latih dan data uji menggunakan metode *Split Data* dan pembagian data latih sebanyak 80% dan data uji 20%. Metode evaluasi *Macro Precision*, *Macro Recall*, dan *Macro F1-Score* digunakan, karena metode tersebut merupakan metode yang populer digunakan untuk mengevaluasi performa [15].

4.1.1 Pengujian Jumlah Fitur

Skenario pengujian pertama yaitu melakukan perbandingan penggunaan metode *feature selection* dan tanpa menggunakan *feature selection* serta melakukan perbandingan jumlah fitur yang digunakan dengan menggunakan metode *chi-square*. Skenario ini bertujuan untuk mengetahui pengaruh penggunaan *feature selection* pada metode *Support Vector Machine* dan pengaruh jumlah fitur yang digunakan. Perbandingan penggunaan jumlah fitur yang digunakan pada pengujian ini yaitu sebesar: 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90%, dan 100% atau dapat dikatakan tidak menggunakan *feature selection*. Total jumlah fitur yang ada sebanyak 8493 dan metode

klasifikasi *Support Vector Machine* menggunakan kernel *linear*. Tabel 10 adalah hasil pengujian pengaruh jumlah fitur.

Tabel 10 Hasil Pengujian Perbandingan Jumlah Fitur

	Jumlah Fitur	Precision (%)	Recall (%)	F1-Score (%)
Chi-Square	10%	85.17%	63.77%	70.49%
	20%	85.42%	66.78%	73.18%
	30%	85.44%	67.51%	73.84%
	40%	84.35%	68.25%	74.23%
	50%	85.23%	68.13%	74.4%
	60%	85.5%	68.47%	74.72%
	70%	85.87%	68.71%	75.05%
	80%	85.93%	68.91%	75.2%
	90%	85.79%	69.13%	75.32%
Tanpa Feature Selection	100%	85.43%	69.25%	75.3%

Hasil pengujian pada Tabel 10 menunjukkan nilai *F1-Score* terendah didapat ketika menggunakan 10% fitur atau 850 yang menghasilkan nilai sebesar 70.49% dan hasil tertinggi didapat ketika menggunakan 90% fitur atau 7644 yang menghasilkan nilai sebesar 75.32%. Hal ini menunjukkan bahwa tinggi rendahnya nilai *F1-Score* dipengaruhi oleh penggunaan jumlah fitur. Dari hasil pengujian yang dilakukan, rata-rata hasil pengujian yang didapat adalah semakin banyak jumlah fitur yang digunakan akan semakin tinggi nilai *F1-Score*. Dapat dilihat pada Tabel 11 ketika jumlah fitur yang digunakan mulai dari 10% menuju 90% nilai *F1-Score* yang didapat terus mengalami kenaikan. Namun ketika 100% fitur digunakan atau tanpa menggunakan *feature selection* nilai *F1-Score* yang didapat mengalami penurunan menjadi 75.3%. Hal ini menunjukkan metode *Chi-Square* yang digunakan untuk melakukan *feature selection* pada metode *Support Vector Machine* dapat menghilangkan fitur yang tidak berpengaruh dan mampu meningkatkan performa yang dihasilkan. Semakin banyak fitur yang digunakan maka sistem yang dibangun akan menghasilkan performa yang lebih tinggi. Hal ini dikarenakan dengan jumlah fitur yang semakin meningkat, sistem semakin mampu memisahkan antar kelas dengan baik. Akan tetapi, jika terlalu banyak fitur yang digunakan atau sudah melebihi batas optimumnya dapat menyebabkan penurunan performa. Penurunan ini terjadi karena terdapat beberapa fitur atau kata yang tidak merepresentasikan sebuah kelas dengan baik yang tetap digunakan pada proses klasifikasi.

4.1.2 Pengujian Penggunaan Stopword Removal

Kemudian skenario pengujian kedua yaitu melakukan pengujian *stopword removal* pada tahap *preprocessing* untuk mengeliminasi kata yang sering ditemukan dan tidak mempunyai pengaruh. Skenario pengujian ini dilaksanakan menggunakan kernel *linear* dan seleksi fitur *chi-square* dengan total fitur sebanyak sebanyak 90% atau 7644. Hasil pengujian pengaruh *stopword removal* ditunjukkan pada Tabel 11.

Tabel 11 Hasil Pengujian Dengan dan Tanpa Stopword Removal

	Precision (%)	Recall (%)	F1-Score (%)
Dengan Stopword Removal	85.80%	69.13%	75.32%
Tanpa Stopword Removal	85.44%	68.96%	75.19%

Hasil pengujian pada Tabel 11 dapat dilihat bahwa ketika *stopword removal* digunakan nilai *F1-Score* yang dihasilkan adalah 75.32% sedangkan tanpa menggunakan *stopword removal* nilai *F1-Score* yang dihasilkan mengalami penurunan menjadi 75.19%. Pada pengujian yang dilakukan ketika menggunakan *stopword removal* performa yang didapat lebih tinggi dibanding tanpa menggunakan *stopword removal*. Perbedaan performa terjadi disebabkan ketika menggunakan *stopword removal* maka kata yang sering muncul dan dianggap tidak memiliki makna akan dihapuskan yang menyebabkan fitur yang digunakan pada proses klasifikasi menjadi berkurang. Sehingga membuat sistem klasifikasi yang dibangun dapat membedakan tiap kelas dengan lebih baik. Contoh penggunaan *stopword removal* ditunjukkan pada Tabel 12.

Tabel 12 Contoh Stopword Removal

Hadis	Preprocessing	Kelas Larangan		Keterangan
		Aktual	Prediksi	
Melarang kalian untuk berpuasa di dua hari raya ini, salah satu dari hari itu adalah hari raya di mana kalian berbuka setelah kalian berpuasa, sedangkan yang kedua adalah pada hari kalian memakan daging binatang kurban kalian	Dengan <i>Stopword Removal</i>	1	1	Sebagai contoh kata “yang” dihapuskan ketika menggunakan <i>Stopword Removal</i>
	Tanpa <i>Stopword Removal</i>	1	0	Kata “yang” tidak dihapuskan dan tetap digunakan pada proses klasifikasi

Sebagai contoh kata “yang” pada tabel 12 merupakan kata yang tidak mempunyai pengaruh dan merupakan kata yang sering muncul di setiap data, baik di kelas anjuran, larangan, maupun informasi. Jika kata tersebut tidak dihapuskan membuat hasil klasifikasi yang seharusnya termasuk kedalam kelas larangan menjadi salah diklasifikasi. Hal tersebut terjadi karena total data di kelas larangan lebih sedikit dibandingkan pada kelas bukan larangan, sehingga ketika kata tersebut digunakan mendefinisikan kelas bukan larangan. Dengan menggunakan *stopword removal* maka kata tersebut akan dihapuskan dan tidak akan digunakan pada proses klasifikasi.

4.1.3 Pengujian Penggunaan Kernel Pada SVM

Skenario pengujian ketiga atau skenario terakhir yaitu menguji penggunaan tiga jenis kernel yang terdapat pada metode *Support Vector Machine* yaitu *linear*, RBF, dan *polynomial* dengan parameter $C=1$ pada setiap kernel. Pengujian ini menggunakan *Stopword Removal* dan *Chi-Square* dengan jumlah fitur sebanyak 90% fitur atau 7644 karena pada pengujian sebelumnya menghasilkan performa yang paling baik.

Tabel 13 Hasil Pengujian Kernel pada SVM

Kernel	Precision (%)	Recall (%)	F1-Score (%)
Linear	85.79%	69.13%	75.32%
RBF	90.55%	64.56%	72.89%
Polynomial	90.66%	48.69%	56.9%

Pada Tabel 13, dapat dilihat bahwa nilai *F1-Score* terbaik didapat ketika menggunakan kernel *linear* yaitu 75.32%, diikuti oleh kernel RBF dengan nilai *F1-Score* 72.89%, dan terakhir kernel *polynomial* dengan nilai *F1-Score* 56.9%. Kernel *linear* menghasilkan performa yang paling baik karena data yang digunakan memiliki jumlah fitur yang banyak sehingga memetakan data ke dimensi yang lebih tinggi tidak akan terlalu bekerja dengan baik [21]. Dengan karakteristik pada dataset yang memiliki banyak fitur, menjadikan kernel *linear* mendapatkan nilai *F1-Score* terbaik dibanding dengan kernel RBF dan *polynomial*. Sementara itu hasil terendah didapat ketika menggunakan kernel yang lebih kompleks yaitu kernel *polynomial*. Pengujian ini juga membuktikan bahwa dataset yang digunakan dengan jumlah fitur yang banyak mampu dipisahkan secara linear dengan baik.

Kesimpulan dan Saran

Bersumber dari hasil pengujian dan analisis yang sudah dilaksanakan dapat diambil kesimpulan bahwa:

1. Ketika menggunakan seleksi fitur dengan metode *Chi-Square* terhadap data teks yang bersifat *multi-label* dengan menggunakan metode klasifikasi *Support Vector Machine* dapat meningkatkan performa yang dihasilkan dan nilai *F1-Score* yang didapat yaitu 75.32% . Hal ini dikarenakan dengan menggunakan metode *feature selection Chi-Square* maka kata yang tidak merepresentasikan sebuah kelas dengan baik dan kata yang tidak berpengaruh akan dihilangkan.
2. Penggunaan *stopword removal* pada tahap preprocessing dapat meningkatkan performa yang dihasilkan karena kata yang sering ditemukan dan tidak mempunyai pengaruh akan dihapuskan yang menyebabkan sistem yang dibangun dapat memisahkan setiap kelas dengan lebih baik.
3. Penggunaan kernel *linear* yang terdapat pada metode *Support Vector Machine* menghasilkan performa yang terbaik dibanding dengan kernel RBF dan *polynomial* pada data yang mempunyai total fitur yang banyak.

Adapun saran untuk penelitian kedepannya adalah memperbaiki data hadis yang digunakan dengan melakukan pelabelan data oleh ahli dibidang hadis dan ahli data kemudian membangun sistem klasifikasi yang mampu menangani *imbalance* data yang biasa terjadi pada data yang berjumlah banyak.

REFERENSI

- [1] I. R. Ponilan, Adiwijaya, M. A. Bijaksana, and A. S. Raharusun, "Search relevant retrieval on Indonesian translation hadith document using query expansion and smoothing probabilistic model," *J. Phys. Conf. Ser.*, vol. 1192, no. 1, 2019, doi: 10.1088/1742-6596/1192/1/012032.
- [2] M. N. Al-Kabi, H. A. Wahsheh, and I. M. Alsmadi, "A Topical Classification of Hadith Arabic Text," 2014, pp. 12–13.
- [3] Z. Liu, X. Lv, K. Liu, and S. Shi, "Study on SVM compared with the other text classification methods," in *2nd International Workshop on Education Technology and Computer Science, ETCS 2010*, 2010, vol. 1, pp. 219–222, doi: 10.1109/ETCS.2010.248.
- [4] S. Günel, "Hybrid feature selection for text classification," *Turkish J. Electr. Eng. Comput. Sci.*, vol. 20, no. SUPPL.2, pp. 1296–1311, 2012, doi: 10.3906/elk-1101-1064.
- [5] J. Chen, H. Huang, S. Tian, and Y. Qu, "Feature selection for text classification with Naïve Bayes," *Expert Syst. Appl.*, vol. 36, no. 3 PART 1, pp. 5432–5435, 2009, doi: 10.1016/j.eswa.2008.06.054.
- [6] H. M. Abdelaal, B. R. Elemary, and H. A. Youness, "Classification of Hadith According to Its Content Based on Supervised Learning Algorithms," *IEEE Access*, vol. 7, pp. 152379–152387, 2019, doi: 10.1109/ACCESS.2019.2948159.
- [7] H. Fauzan, A. Adiwijaya, and S. Al-Faraby, "Pengklasifikasian Topik Hadits Terjemahan Bahasa Indonesia Menggunakan Latent Semantic Indexing dan Support Vector Machine," *J. Media Inform. Budidarma*, vol. 2, no. 4, p. 131, 2018, doi: 10.30865/mib.v2i4.948.
- [8] N. A. P. Rostam and N. H. A. H. Malim, "Text categorisation in Quran and Hadith: Overcoming the interrelation challenges using machine learning and term weighting," *J. King Saud Univ. - Comput. Inf. Sci.*, no. xxxx, 2019, doi: 10.1016/j.jksuci.2019.03.007.
- [9] S. Al Faraby, E. R. R. Jasin, A. Kusumaningrum, and Adiwijaya, "Classification of hadith into positive suggestion, negative suggestion, and information," *J. Phys. Conf. Ser.*, vol. 971, no. 1, 2018, doi: 10.1088/1742-6596/971/1/012046.
- [10] M. Y. Abu Bakar, Adiwijaya, and S. Al Faraby, "Multi-Label Topic Classification of Hadith of Bukhari (Indonesian Language Translation) Using Information Gain and Backpropagation Neural Network," in *Proceedings of the 2018 International Conference on Asian Language Processing, IALP 2018*, 2018, pp. 344–350, doi: 10.1109/IALP.2018.8629263.
- [11] M. S. Khorsheed and A. O. Al-Thubaity, "Comparative evaluation of text classification techniques using a large diverse Arabic dataset," *Lang. Resour. Eval.*, vol. 47, no. 2, pp. 513–538, 2013, doi: 10.1007/s10579-013-9221-8.
- [12] A. W. Pradana and M. Hayaty, "The Effect of Stemming and Removal of Stopwords on the Accuracy of Sentiment Analysis on Indonesian-language Texts," *Kinet. Game Technol. Inf. Syst. Comput. Network, Comput. Electron. Control*, vol. 4, no. 3, pp. 375–380, 2019, doi: 10.22219/kinetik.v4i4.912.
- [13] Y. Zhai, W. Song, X. Liu, L. Liu, and X. Zhao, "A Chi-Square Statistics Based Feature Selection Method in Text Classification," *Proc. IEEE Int. Conf. Softw. Eng. Serv. Sci. ICSESS*, pp. 160–163, 2018, doi: 10.1109/ICSESS.2018.8663882.
- [14] T. A. G. Sanasam Ranbir Singh, Hema A. Murthy, "Feature Selection for Text Classification Based on Gini Coefficient of Inequality," in *Computational Methods of Feature Selection*, 2020, pp. 273–292, doi: 10.1201/9781584888796-23.
- [15] J. Xu, "An extended one-versus-rest support vector machine for multi-label classification," *Neurocomputing*, vol. 74, no. 17, pp. 3114–3124, 2011, doi: 10.1016/j.neucom.2011.04.024.
- [16] A. W. Romadon, K. M. Lhaksana, I. Kurniawan, and D. Richasdy, "Analyzing TF-IDF and Word Embedding for Implementing Automation in Job Interview Grading," in *2020 8th International Conference on Information and Communication Technology, ICoICT 2020*, 2020, pp. 1–4, doi: 10.1109/ICoICT49345.2020.9166364.
- [17] K. Kowsari, K. J. Meimandi, M. Heidarysafa, S. Mendu, L. Barnes, and D. Brown, "Text classification algorithms: A survey," *Inf.*, vol. 10, no. 4, pp. 1–68, 2019, doi: 10.3390/info10040150.
- [18] S. Bahassine, A. Madani, M. Al-Sarem, and M. Kissi, "Feature selection using an improved Chi-square for Arabic text classification," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 32, no. 2, pp. 225–231, 2020, doi: 10.1016/j.jksuci.2018.05.010.
- [19] D. Meyer, "Support Vector Machines: interface to libsvm in e1071," *Engineering*, 2009.
- [20] Y. Liu and Y. F. Zheng, "One-Against-All Multi-Class SVM Classification Using Reliability Measures," in *Proceedings of International Joint Conference on Neural Networks*, 2005, pp. 849–854.
- [21] Chih-Wei Hsu, Chih-Chung Chang, Chih-Jen Lin, "A Practical Guide to Support Vector Classification," *BJU Int.*, vol. 101, no. 1, pp. 1396–1400, 2008, [Online]. Available: <http://www.csie.ntu.edu.tw/%7B~%7Dcjlin/papers/guide/guide.pdf>.