

# Klasifikasi *Multi-Label* pada Hadis Bukhari dalam Terjemahan Bahasa Indonesia Menggunakan *Mutual Information* dan *Backpropagation Neural Network*

HendroPrasetyo<sup>1</sup>, Adiwijaya<sup>2</sup>, WidiAstuti<sup>3</sup>

<sup>1,2,3</sup>Fakultas Informatika, Universitas Telkom, Bandung

<sup>1</sup>hendroprasetyo@students.telkomuniversity.ac.id, <sup>2</sup>adiwijaya@telkomuniversity.ac.id,

<sup>3</sup>widiwdu@telkomuniversity.ac.id

---

## Abstrak

Hadis adalah segala sesuatu yang disandarkan pada Nabi Muhammad SAW baik perkataan, perbuatan, *taqir* (sikap diam setuju) dan yang lainnya. Hadis merupakan sumber hukum tersendiri bagi umat muslim yang tidak dijelaskan dalam Al Qur'an. Ada banyak hadis yang telah diriwayatkan oleh para ahli hadis, salah satunya adalah hadis shahih Bukhari. Penelitian ini membuat sebuah sistem yang dapat melakukan klasifikasi hadis Bukhari Muslim Terjemahan berbahasa Indonesia. Metode klasifikasi *Backpropagation Neural Network* digunakan karena dapat melakukan klasifikasi data dengan jumlah fitur yang banyak dan beragam, didukung dengan *Mutual Information* sebagai metode seleksi fitur dalam memilih fitur-fitur yang berpengaruh pada setiap label kelas *multi-label*. Pada penelitian ini dilakukan beberapa skenario pengujian dengan memodifikasi tahapan *preprocessing*, seleksi fitur, dan parameter *Backpropagation Neural Network*. Pengujian tersebut menunjukkan bahwa nilai *hamming loss* terbaik adalah sebesar 0,0892 dan waktu komputasi 5284,8 s dengan melibatkan tiga poin pengujian yaitu: *stemming*, *Mutual Information* dan nilai *learning rate* terbaik.

**Kata kunci :** klasifikasi teks, hadis, *backpropagation neural network*, *mutual information*, *multi-label*

---

## Abstract

Hadith is everything that is based on Prophet Muhammad SAW involve words, deeds, *taqir* (silence agree) and others. Hadith is a separate source of law for Muslims which is not explained in the Qur'an. There are many traditions which have been narrated by the experts of hadith, one of which is the hadith of sahih Bukhari. This research makes a system that can classify the Bukhari Muslim Translation of hadith in Indonesian. The classification method of *Backpropagation Neural Network* is used because it can classify data with a large number of diverse features, supported by *Mutual Information* as a feature selection method in selecting features that affect each *multi-label* class label. In this study several test scenarios were carried out by modifying the preprocessing stages, feature selection, and *Backpropagation Neural Network* parameters. The test shows that the best *hamming loss* value is 0.0892 and a computation time of 5284.8 s by involving three test points, namely: *stemming*, *Mutual Information* and the best *learning rate* value.

**Keywords:** text classification, hadith, *backpropagation neural network*, *mutual information*, *multi-label*

---

## 1. Pendahuluan

### Latar Belakang

Hadis adalah segala sesuatu yang disandarkan pada Nabi Muhammad SAW baik perkataan, perbuatan, *taqir* (sikap diam setuju) dan yang lainnya. Hadis merupakan salah satu sumber hukum bagi umat muslim yang tidak dijelaskan dalam Al Qur'an [1] Umumnya hadis dikoleksi oleh beberapa imam besar, salah satunya koleksi hadis yang disusun oleh Imam Bukhari (nama lengkap: Abu Abdullah Muhammad bin Ismail bin Ibrahim bin al-Mughirah al-Ju'fi) yang hidup antara 194 hingga 256 hijriah [2]. Beberapa ahli hadis yang telah meriwayatkan banyak hadis dari Rasulullah Shallallahu 'alaihi wasallam, salah satunya adalah Bukhari. Menurut aplikasi Lidwa, jumlah hadis Shahih Bukhari berjumlah kurang lebih 7.008 hadis termasuk hadis yang berulang.

Hadis shahih Bukhari berisikan hadis-hadis yang menganjurkan kita untuk berbuat kebaikan dan menjauhi larangan sesuai yang diperintahkan oleh Allah Subhanahu wata'ala. Untuk dapat memahami hadis-hadis tersebut dengan jumlah yang banyak maka dibutuhkan sebuah sistem yang dapat mengidentifikasi apakah hadis tersebut bersifat anjuran, larangan atau sebuah informasi sehingga dapat diterapkan dalam kehidupan dengan baik. Suatu hadis mungkin saja hanya merupakan sebuah anjuran, mungkin juga anjuran dan informasi atau bahkan mungkin anjuran, informasi dan larangan. Untuk mengetahui dengan lebih jelas maka perlu untuk dilakukan penelitian dalam pengklasifikasian hadis Shahih Bukhari ini kedalam kategori-kategori yang telah ditentukan.

Atas dasar itu penulis membangun model klasifikasi berdasarkan hadis shahih Bukhari sebanyak 1064 data hadis ke dalam bentuk klasifikasi *multi-label* untuk menentukan kategori atau label dari hadis tersebut. Hadis-hadis tersebut telah memiliki label yang akan digunakan untuk melakukan proses pelatihan pada sistem *Mutual*

*Information*. Metode yang digunakan adalah *Backpropagation Neural Network* dipilih berdasarkan penelitian yang telah dilakukan oleh Muhammad Yuslan Abubakar mengenai Klasifikasi *Multi-Label* hadis Bukhari Terjemahan Bahasa Indonesia menggunakan *Information Gain* dan *Backpropagation Neural Network*[3] yang menghasilkan kesimpulan bahwa penggunaan *Information gain* bekerja efektif pada *multi-label* dengan metode *Backpropagation Neural Network* dan dapat mengklasifikasikan data hadis *multi-label* sebesar 88.42%. Oleh sebab itu penulis melakukan penelitian untuk mencari tahu apakah dengan data *multi-label* dan metode yang sama tetapi dengan seleksi fitur yang berbeda akan menghasilkan nilai yang lebih baik atau sebaliknya. Fitur yang telah didapatkan pada proses seleksi dan ekstraksi fitur akan digunakan sebagai masukan (*input*) pada sistem klasifikasi *Backpropagation Neural Network*.

### Topik dan Batasannya

Ada beberapa masalah yang dibatasi diantaranya adalah penggunaan dataset sejumlah 1064 data hadis terjemahan Bahasa Indonesia dengan tiga label yaitu anjuran, larangan, dan informasi yang diperoleh dari penelitian M. Y. Abu Bakar, Adiwijaya and S. A. Faraby, "*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)*, Bandung, Indonesia, 2018, pp. 344-350. Jumlah data pada setiap kelas tidak merata, berdasarkan data yang tersedia ada salah satu kelas yang memiliki jumlah data sekitar 70% dari keseluruhan data [3]. Sedikitnya jumlah data *multi-label* yang memiliki 3 jenis label sekaligus menjadi salah satu hambatan pada penelitian ini.

### Tujuan

Penelitian ini bertujuan untuk mendapatkan hasil performansi terhadap data *multi-label* dan hasil dari pengaruh penggunaan *Mutual Information* serta hasil klasifikasi dengan metode *Backpropagation Neural Network*, dengan menggunakan metode evaluasi *hamming loss* untuk mengukur seberapa baik sistem dan performansi dari hasil klasifikasi *multi-label* yang didapatkan. *Hamming loss* adalah salah satu metrik evaluasi yang sering digunakan untuk pengujian data *multi-label* dengan cara menghitung *error rate* dari hasil klasifikasi.

### Organisasi Tulisan

Beberapa poin yang akan dijelaskan pada jurnal ini adalah sebagai berikut. Poin pertama menjelaskan latar belakang, batasan, tujuan dan organisasi dari jurnal ini. Kemudian poin kedua terdapat studi literatur terkait dengan penelitian yang sudah dilakukan sebelumnya dan penelitian yang sedang dilakukan serta beberapa tinjauan pustaka yang terkait dengan penelitian. Kemudian poin ketiga akan menjelaskan sistem yang akan dibangun yaitu klasifikasi pada hadis Bukhari terjemahan Bahasa Indonesia dengan menggunakan metode *Mutual Information* dan *Backpropagation Neural Network*. Kemudian poin keempat, akan menjelaskan hasil pengujian dan analisis yang telah dilakukan. Dan yang terakhir poin kelima akan dijelaskan mengenai kesimpulan dan saran yang dihasilkan dari penelitian ini.

## 2. Studi Terkait

### 2.1 Penelitian Terkait

Terdapat beberapa penelitian terkait dengan klasifikasi teks data *multi-label* yang salah satunya telah dilakukan oleh Adhitha Wiraguna [4], yaitu tentang Klasifikasi Topik *Multi-label* pada Hadis Bukhari dalam Terjemahan Bahasa Indonesia Menggunakan Random Forest. Penelitian ini menggunakan metode pendekatan *Problem Transformation* yaitu Binary Relevance (BR) dan Label Powerset (LP) untuk mengadaptasi RF dalam membangun sistem klasifikasi teks *multi-label*. Hasil penelitian menunjukkan bahwa performansi *hamming loss* yang terbaik didapat dari sistem yang menggunakan BR dan tidak menggunakan stemming yaitu sebesar 0,0663.

Penelitian berikutnya adalah penelitian yang telah dilakukan oleh Maria Arista Ulfa [5]. Dengan judul Analisis Sentimen Twitter Menggunakan *Naïve Bayes Classifier* dengan Seleksi Fitur *Mutual Information*. Akurasi rata-rata klasifikasi dengan menggunakan metode NBC dalam mengklasifikasikan *tweet* ke dalam kelas positif dan kelas negatif dengan menggunakan seleksi fitur MI yaitu sebesar 97,9% dan tanpa menggunakan seleksi fitur MI sebesar 96,2%. Oleh karena itu penggunaan seleksi fitur MI dapat menaikkan akurasi sekitar 1,7%.

Kemudian ada juga penelitian yang telah dilakukan oleh E. Jasin [6] klasifikasi hadis anjuran, hadis larangan dan hadis informasi pada hadis shahih Bukhari berdasarkan model *Unigram* menggunakan *Artificial Neural Network* (ANN). E.Jasin juga menggunakan bantuan metode ekstraksi fitur TF-IDF untuk mendapatkan nilai-nilai pada setiap kata. Hasil performansi yang didapatkan dengan *F1-Score* sebesar 85%.

Penelitian terbaru terkait klasifikasi teks data *multi-label* adalah penelitian yang telah dilakukan oleh Muhammad Yuslan Abubakar [3] pada tahun 2018. *Information gain* digunakan sebagai fitur seleksi dan *Backpropagation Neural Network* sebagai *Classifier*, menghasilkan data yang efektif pada penggunaan terhadap data *multi-label* dan berhasil mengklasifikasikan data dengan akurasi sebesar 88,42%.

## 2.2 Text Classification

*Text classification* adalah salah satu teknik dari *text mining* yang bertujuan untuk menentukan kelas atau kategori dari suatu teks yang dapat berbentuk *frase*, kalimat, *paragraph*, atau bahkan dokumen teks. Proses klasifikasi teks memiliki beberapa teknik salah satunya adalah klasifikasi dilakukan berdasarkan data teks yang sudah dilatih pada sistem atau bisa juga disebut sebagai *supervised learning*. Dengan menggunakan *machine learning*, proses klasifikasi ini melibatkan algoritma *classifier* yang digunakan untuk melakukan proses *learning* dan *testing* pada dataset [7].

## 2.3 Preprocessing

*Preprocessing* merupakan salah satu tahapan yang penting untuk data. Data yang digunakan dalam proses *mining* tidak selamanya dalam kondisi yang ideal untuk diproses. Terkadang pada data tersebut terdapat berbagai permasalahan yang dapat mengganggu hasil dari proses *mining* itu sendiri seperti diantaranya adalah *missing value*, data *redundant*, *outliers*, ataupun format data yang tidak sesuai dengan sistem. Oleh karenanya untuk mengatasi permasalahan tersebut dibutuhkan tahap *preprocessing* [2].

*Preprocessing* merupakan salah satu tahapan menghilangkan permasalahan-permasalahan yang dapat mengganggu hasil daripada proses data. Dalam kasus klasifikasi dokumen yang menggunakan data bertipe teks, terdapat beberapa macam proses yang dilakukan umumnya diantaranya *case folding*, *stopword removal*, *stemming*, *tokenization* dan sebagainya [8].

## 2.4 Seleksi Fitur

Setelah didapatkan data melalui proses *preprocessing*, selanjutnya adalah memilih dan menentukan fitur-fitur yang dibutuhkan untuk proses klasifikasi. Seleksi Fitur sangat dibutuhkan pada klasifikasi teks untuk mengurangi dimensi informasi [9]. Seleksi fitur dapat membuat proses klasifikasi menjadi lebih efisien dan efektif dengan mengurangi jumlah data yang dianalisis. *Mutual Information* (MI) merupakan salah satu metode seleksi yang menunjukkan seberapa banyak informasi ada atau tidaknya sebuah term memberikan kontribusi dalam membuat keputusan klasifikasi secara benar atau salah.

*Mutual Information* bekerja apabila ada dua buah pengamatan  $x$  dan  $y$  yang saling bebas memiliki peluang  $P(x)$  dan  $P(y)$ , sedangkan  $P(x,y)$  adalah peluang pengamatan  $x$  dan  $y$  secara bersama-sama. Berikut persamaan(1) Rumus *Mutual Information*  $I(x,y)$  :

$$I(X; Y) = \sum_{y \in Y} \sum_{x \in X} P(x, y) \log_2 \frac{P(x, y)}{P(x) \cdot P(y)} \quad (1)$$

Secara formal, *Mutual Information* membandingkan peluang pengamatan  $x$  dan  $y$  secara bersama dengan peluang pengamatan  $x$  dan  $y$  secara bebas. Jika terdapat hubungan yang kuat antara pengamatan  $x$  dan  $y$ , maka  $P(x,y)$  akan lebih besar dari  $P(x) \cdot P(y)$  dan sebagai akibatnya  $I(x,y) > 0$ . Apabila tidak ada hubungan keterkaitan antara pengamatan  $x$  dan  $y$  maka  $P(x,y) \approx P(x) \cdot P(y)$  dan  $I(x,y) \approx 0$ . [11]

Pada domain diskrit, nilai dari *Mutual Information* antara dua variabel acak didefinisikan sebagai berikut:

$$I(U, C) = \sum_{et \in \{1,0\}} \sum_{ec \in \{1,0\}} P(U = et, C = ec) \log_2 \frac{P(U = et, C = ec)}{P(U = et)P(C = ec)} \quad (2)$$

Persamaan di atas (2), dapat diturunkan menjadi seperti berikut.

$$I(U, C) = \frac{N_{11}}{N} \log_2 \frac{N \cdot N_{11}}{N_1 \cdot N_1} + \frac{N_{01}}{N} \log_2 \frac{N \cdot N_{01}}{N_0 \cdot N_1} + \frac{N_{10}}{N} \log_2 \frac{N \cdot N_{10}}{N_1 \cdot N_0} + \frac{N_{00}}{N} \log_2 \frac{N \cdot N_{00}}{N_0 \cdot N_0} \quad (3)$$

Keterangan :

- $N$  = Jumlah dokumen yang memiliki *et* dan *ec* atau ( $N = N_{00} + N_{01} + N_{10} + N_{11}$ ).
- $N_1$  = Jumlah dokumen yang memiliki *et* atau ( $N_1 = N_{10} + N_{11}$ ).
- $N_0$  = Jumlah dokumen yang memiliki *ec* atau ( $N_0 = N_{01} + N_{00}$ ).
- $N_{10}$  = Jumlah dokumen yang tidak memiliki *et* atau ( $N_{10} = N_{01} + N_{00}$ ).
- $N_{00}$  = Jumlah dokumen yang tidak memiliki *ec* atau ( $N_{00} = N_{10} + N_{00}$ ).

## 2.5 Language Model (N-Gram)

Pada pengurutan kata, model *statistic* yang digunakan biasa disebut *language models* or LMs. Salah satu tekniknya adalah dengan menggunakan N-gram. Nilai N pada N-gram tergantung pada berapa kata terurut yang ingin diambil. N=1 biasa disebut unigram contohnya seperti "*please*", "*turn*", "*your*", "*homework*", untuk N=2

disebut bigram contohnya seperti “please turn”, “turn your”, “your homework”, untuk N=3 disebut trigram contohnya “please turn your”, “turn your homework” [10].

## 2.6 Ekstraksi Fitur

TF-IDF (*Term Frequency Inverse Document Frequency*) adalah metode ekstraksi fitur yang akan digunakan. TF-IDF merupakan salah satu metode pembobotan *term*. Metode ini menghitung TF (*Term Frequency*) yang merupakan jumlah kemunculan tiap *term*(kata) pada setiap dokumen. IDF (*Inverse Document Frequency*) merupakan jumlah dokumen terkait yang mengandung suatu term tertentu [11]. Berikut merupakan persamaan(3) perhitungan bobot pada TF-IDF

$$w_{ij} = tf \times idf = tf_{ij} \times \log D df_i \quad (4)$$

Keterangan :

$w_{ij}$	=	bobot kata $t_j$ terhadap dokumen $d_i$
$tf_{ij}$	=	jumlah kemunculan kata $t_j$ dalam $d_i$
$D$	=	jumlah dokumen
$df_i$	=	jumlah kemunculan <i>term</i> dalam $D$

Input yang digunakan pada proses ekstraksi fitur ini adalah hasil output dari proses *preprocessing* berupa *bag of words* dan hasil fitur *Mutual Information*. *Bag of words* didapatkan dari proses penggabungan semua hasil *preprocessing* data teks hadits, atau pada istilah matematika biasa disebut sebagai *union* [12]. Sedangkan output proses ekstraksi fitur ini adalah berupa bobot dari setiap fitur berdasarkan dokumen terkait yang berbentuk matriks. Matriks adalah kumpulan dari bilangan yang terdiri dari baris dan kolom, dimana bilangan yang ada pada matriks disebut elemen matriks [13]. Elemen dari matriks *word vector* ini adalah berupa bobot dari setiap term atau kata. Selanjutnya, matriks tersebut akan menjadi input bagi sistem klasifikasi *neural network* dalam membangun model *classifier*.

## 2.7 K-Fold Cross-Validation

*K-fold* adalah salah satu teknik *cross-validation* yang digunakan untuk membagi data menjadi data latih dan data uji. *K-fold cross-validation* membagi data secara acak menjadi K bagian berukuran sama. Secara bertahap akan dilakukan pelatihan dan validasi sebanyak K ulangan. Sehingga dalam setiap perulangan K-1 bagian akan menjadi data latih dan 1 bagian sisanya akan digunakan untuk validasi [14]. Perkiraan tingkat kesalahan dihitung dari proporsi kesalahan dari seluruh bagian sampel [15]. Pada penelitian ini digunakan nilai K = 5.

## 2.8 Classifier

*Classifier* yang akan digunakan pada penelitian ini adalah *Backpropagation Neural Network*. *Classifier* ini bekerja dengan cara melakukan dua tahap perhitungan yaitu perhitungan maju yang akan menghitung nilai kesalahan (*error*) antara nilai *output* sistem dengan nilai yang seharusnya dan perhitungan mundur untuk memperbaiki bobot berdasarkan nilai *error* tersebut [16]. Data latih digunakan pada proses *learning* untuk mendapatkan nilai-nilai parameter yang mendapatkan hasil terbaik pada sistem *backpropagation* sistem *neural network*. Hasil terbaik didapatkan ketika nilai *error* yang ada sudah memenuhi standar yang sudah ditentukan atau dengan kata lain nilai *error*-nya sangat kecil [3]. Dalam penelitian ini, nilai yang digunakan sebagai MSE standar adalah sebesar 0.0001.

## 2.9 Metrik Evaluasi

Tahap evaluasi dilakukan terhadap hasil dari klasifikasi. Ini dilakukan untuk mengukur tingkat keefektifan dari sistem [17]. Pada penelitian ini, evaluasi yang akan dilakukan adalah menggunakan standar evaluasi *hamming loss*. Persamaan (5) akan menunjukkan rumus dari *hamming loss* [3].

$$h(loss) = \frac{1}{p} \sum_{i=1}^p \frac{1}{Q} |h(x_i) \Delta Y_i| \quad (5)$$

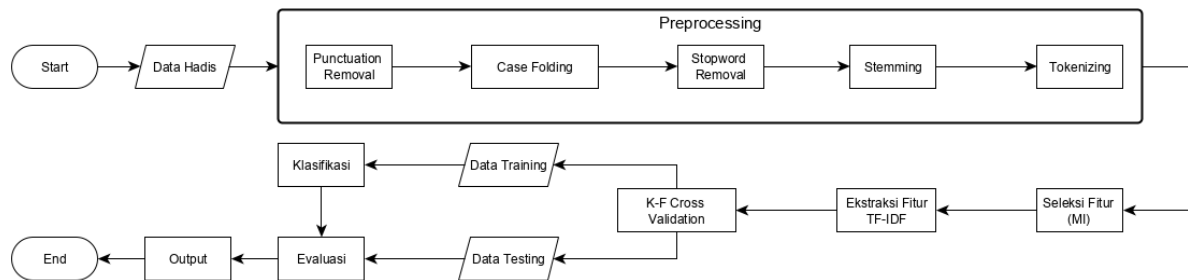
Dimana :

$p$	=	banyak data yang dilakukan klasifikasi
$Q$	=	jumlah label kelas
$ h(x_i) \Delta Y_i $	=	banyaknya kesalahan klasifikasi yang terjadi

*Hamming loss* akan menghitung berapa banyak label kelas yang salah diklasifikasikan. Misal, suatu data yang seharusnya diprediksi kedalam kelas 1 dan 2, namun diprediksi menjadi kelas 2 dan 3. Performansi yang dihasilkan akan sempurna apabila nilai *hamming loss*(h)= 0. Semakin kecil nilai *hamming loss*, maka semakin baik pula performansi sistem.

### 3 Sistem yang Dibangun

Dalam penelitian ini, dibangun sebuah sistem yang dapat melakukan klasifikasi *multi-label* pada data hadis shahih Bukhari ke dalam kelas hadits anjuran, larangan, dan informasi. Gambaran sistem yang dibangun dapat dilihat pada Gambar 3-1 dibawah ini.



**Gambar 3-1 Flowchart sistem**

Data hadis yang akan digunakan adalah hadis shahih Bukhari terjemahan berbahasa Indonesia yang telah diberi label sebelumnya. Sistem terdiri dari 1064 data hadis *multi-label* dengan 289 data memiliki lebih dari satu label. Label kelas pada data *multi-label* terdiri dari 3 kelas yaitu anjuran larangan dan informasi. Masing-masing data hadis *multi-label* dapat berupa salah satu dari ketiga kelas tersebut maupun gabungan dari beberapa kelas. Berikut ini Tabel 3-1 adalah representasi jumlah data yang digunakan pada penelitian ini.

**Tabel 3-1 Representasi Data**

Data	Kelas		
	Anjuran	Larangan	Informasi
Janganlah kalian mengangkat kepala kalian hingga para laki-laki telah duduk.	0	1	1
Janganlah kalian berdusta terhadapku (atas namaku), karena barangsiapa berdusta atas namaku dia akan masuk neraka.	0	1	1
aku melihat Nabi sering beranjak pergi dari sebelah kirinya.	0	0	1

Tahap pertama adalah melakukan normalisasi data menjadi ideal dengan menghilangkan *noise* dengan melakukan *preprocessing*. Pada penelitian ini proses *preprocessing* terurut sebagai berikut.

- 1) *Punctuation removal* untuk menghilangkan tanda baca atau karakter selain teks dengan fungsi. Berikut ini Tabel 3-2 merupakan contoh perubahan teks pada data.

**Tabel 3-2 Punctuation removal**

Data Input	Janganlah kalian berdusta terhadapku (atas namaku), karena barangsiapa berdusta atas namaku dia akan masuk neraka.
Data Output	Janganlah kalian berdusta terhadapku atas namaku karena barangsiapa berdusta atas namaku dia akan masuk neraka

- 2) *Case folding* merupakan proses untuk merubah setiap kata menjadi sama yaitu huruf kecil dengan menggunakan fungsi *lowercase*. Berikut ini Tabel 3-3 merupakan contoh perubahan teks pada data.

**Tabel 3-3 Case folding**

Data Input	janganlah kalian mengangkat kepala kalian hingga para laki-laki telah duduk.
Data Output	janganlah kalian mengangkat kepala kalian hingga para laki-laki telah duduk.

- 3) *Stopword removal* merupakan tahapan menghapus kata-kata yang terlalu umum dan kurang penting, ciri-ciri pada kata ini adalah frekuensi kemunculannya yang cukup banyak dibandingkan dengan kata yang lainnya. Penulis menambahkan beberapa kata pada daftar kata *stopword*, dari jumlah *default* resmi *library* Sastrawi yang berjumlah 126 menjadi 809 kata. Berikut pada tabel merupakan contoh proses *stopwords removal* pada Tabel 3-4.

**Tabel 3-4 Stopword removal**

Data Input	janganlah kalian mengangkat kepala kalian hingga para laki-laki telah duduk
Data Output	jangan mengangkat kepala laki-laki duduk

- 4) *Stemming* adalah proses untuk mengubah kata pada setiap kalimat ke bentuk dasar atau menghapus kata-kata imbuhan. Berikut pada Tabel 3-5 contoh dari proses *stemming*

**Tabel 3-5 Proses Stemming**

Data Input	aku melihat nabi sering beranjak pergi dari sebelah kirinya.
Data Output	aku lihat nabi sering beranjak pergi dari sebelah kiri.

- 5) *Tokenizing* merupakan tahap untuk memengal setiap kata dalam kalimat termasuk karakter. Berikut ini Tabel 3-6 merupakan contoh hasil *tokenize* pada data.

**Tabel 3-6 Tokenizing**

Data Input	aku melihat nabi sering beranjak pergi dari sebelah kirinya.
Data Output	aku   melihat   nabi   sering   beranjak   pergi   dari   sebelah   kirinya

- 6) Tahap fitur seleksi dengan menggunakan metode (MI) *Mutual Information*. Nilai MI dapat diperoleh dengan beberapa nilai pendukung, seperti frekuensi kata tertentu yang ada atau tidak ada pada suatu kelas, frekuensi kelas yang tidak mengandung kata tertentu dan total data pada korpus. Proses perhitungan seleksi fitur dengan MI dapat dilihat pada lampiran 1. Kemudian akan diperoleh semua nilai MI untuk kata-kata yang ada pada masing-masing kelas. Tabel 3-7 berisi contoh tiga kata dengan nilai MI tertinggi yang akan digunakan untuk proses berikutnya.

**Tabel 3-7 Seleksi Fitur**

Fitur	Nilai <i>Mutual Information</i>
janganlah	0.035
bayang	0.063
orang	0.063

- 7) Setelah nilai fitur seleksi didapatkan, selanjutnya adalah pemberian bobot dan membentuk matriks dari fitur-fitur tersebut dengan menggunakan fitur ekstraksi yaitu TF-IDF. Pada Tabel 3-8 berikut, disajikan contoh untuk perhitungan TF-IDF.

**Tabel 3-8 Perhitungan TF-IDF**

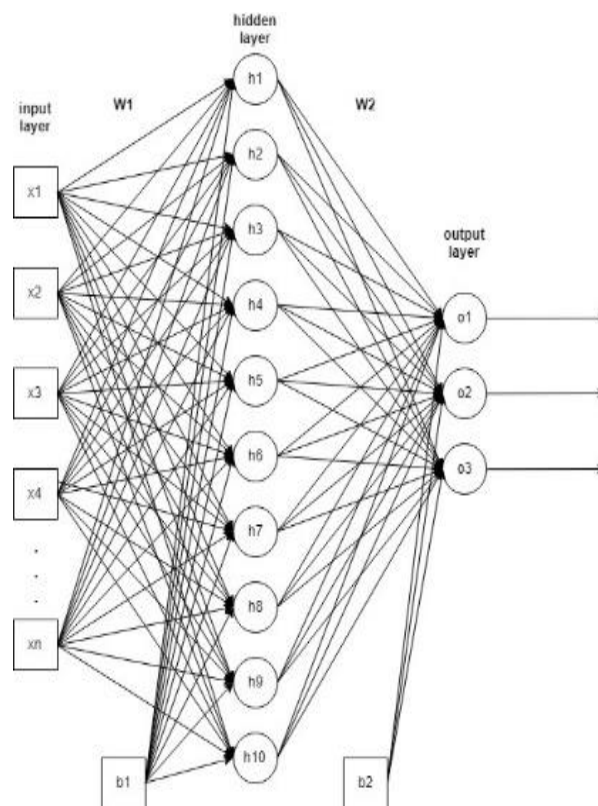
Kata/fitur	tf			df	$\frac{D}{df}$	IDF	W = tfxIDF		
	D1	D2	D3				Dok.1	Dok.2	Dok.3
janganlah	0	2	0	1	3	0,477	0	0,954	0
bayang	1	0	0	1	3	0,477	0,477	0	0
orang	1	1	0	2	1.5	0,176	0,176	0,176	0

Setelah bobot fitur didapatkan untuk setiap kata pada masing-masing dokumen, dibentuklah matriks TF-IDF berukuran  $d \times n$  dimana  $d$  merepresentasikan dokumen dan  $n$  merepresentasikan kata/*term*. Isi dari matriks TF-IDF inilah yang akan menjadi masukan untuk membangun *classifier Backpropagation Neural Network*. Berikut ini Tabel 3-9 memperlihatkan bentuk dari matriks TF-IDF.

**Tabel 3-9 Matriks TF-IDF**

	janganlah	bayang	orang
Dok.1	0	0,477	0.176
Dok.2	0.954	0	0.176
Dok.3	0	0	0

- 8) Data masukan *classifier* didapatkan dari hasil *output* matriks TF-IDF. Besarnya dimensi pada layer *input* sejumlah fitur yang didapatkan pada proses seleksi fitur dan nilai tiap data pada layer *input* akan sama dengan nilai tiap-tiap fitur yang telah didapatkan pada proses ekstraksi fitur. Gambar 3-2 adalah arsitektur *Backpropagation Neural Network* yang dibuat pada penelitian ini.



**Gambar 3-2 Arsitektur Backpropagation Neural Network Multi-label**

Berikut merupakan rumus algoritma dari *Backpropagation Neural Network*,

1. **Perhitungan Maju (*Feedforward*)**

**Pada Hidden Layer**

$$V1 = P * W1 + B1 \quad (6)$$

$$A1 = \frac{1}{1 + e^{-\alpha * V1}} \quad (7)$$

**Pada Output Layer**

$$V2 = W2 * A1 + B2 \quad (8)$$

$$A2 = \frac{1}{1 + e^{-\alpha * V2}} \quad (9)$$

$$E = T - A2 \quad (10)$$

$$MSE = \frac{\sum E^2}{N} \quad (11)$$

**Keterangan :**

- V1 = Nilai hasil hitung dari *hidden layer*
- V2 = Nilai hasil hitung dari *output layer*
- A1 = Nilai keluaran dari *hidden layer*
- A2 = Nilai Keluaran dari *output layer*, (hasil prediksi/klasifikasi)
- E = *Error* antara A2 dengan label sesungguhnya
- MSE = *Mean Squared Error*
- T = Data label sebenarnya pada dataset
- P/X = Inputan data
- W1 = Bobot menuju *hidden layer*
- W2 = Bobot menuju *output layer*
- B1 = Bias menuju *hidden layer*
- B2 = Bias menuju *output layer*

2. **Perhitungan Mundur (*Backward*)**

$$\begin{aligned} D2 &= A2 * (1 - A2) * E & (12) & & D1 &= A1 * (1 - A1) * (W2 * D2) & (17) \\ dW1 &= lr * D1 * P & (13) & & dB1 &= lr * D1 & (18) \\ dW2 &= lr * D2 * A1 & (14) & & dB2 &= lr * D2 & (19) \\ W1 &= W1 + dW1 & (15) & & B1 &= B1 + dB1 & (20) \\ W2 &= W2 + dW2 & (16) & & B2 &= B2 + dB2 & (21) \end{aligned}$$

**Keterangan :**

- D1 = Nilai keluaran dari *hidden layer* untuk *Backpropagation*
- D2 = Nilai keluaran dari *output layer* untuk *Backpropagation*
- dW1 = *Update* bobot menuju *hidden layer*
- dW2 = *Update* bobot menuju *output layer*
- dB1 = *Update* bias menuju *hidden layer*
- dB2 = *Update* bias menuju *output layer*
- lr = *Learning rate*

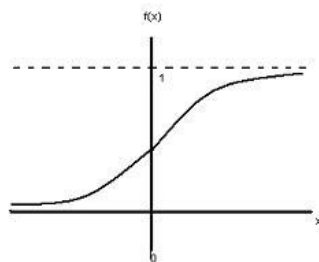
Salah satu fungsi aktivasi yang digunakan adalah fungsi sigmoid biner, yang digunakan untuk *Neural Network* yang membutuhkan nilai *output* yang terletak pada *interval* 0 sampai 1 . Berikut merupakan definisi fungsi sigmoid biner.

$$f(x) = \frac{1}{(1 + e^x)}$$

Turunan :

$$f'(x) = f(x)(1 - f(x))$$

Ilustrasi fungsi sigmoid biner :



**Gambar 3-3 Ilustrasi fungsi sigmoid biner pada range 0-1**

Pada rumus algoritma *Backpropagation Neural Network* fungsi sigmoid biner ini diubah dalam bentuk berikut ini.

$$A1 * (1 - A1) \text{ atau } A2 * (1 - A2)$$

Pada proses *training* dilakukan algoritma perhitungan maju yang akan menghitung nilai kesalahan (*error*) antara nilai *output* sistem dengan nilai yang seharusnya dan perhitungan mundur untuk memperbaiki bobot berdasarkan nilai *error*, sedangkan pada proses *testing* yang dilakukan hanya perhitungan maju saja. Hasil akhir dari pelatihan *Backpropagation* ini adalah bobot-bobot pada **W1**, **W2**, **B1** dan **B2**.



### Arsitektur *Backpropagation Neural Network*

Berikut pada Tabel 3-10 merupakan arsitektur serta nilai pada beberapa parameter *Backpropagation Neural Network* yang perlu ditentukan sebelumnya untuk digunakan dalam proses pengujian.

**Tabel 3-10 Arsitektur *Backpropagation Neural Network***

NO	VARIABLE	NILAI
1	Jumlah <i>neuron</i> pada <i>input layer</i>	Sesuai dengan jumlah fitur yaitu: 1756 (Menggunakan <i>stemming</i> dan MI), 2547 (Menggunakan MI tanpa <i>stemming</i> ), 12730 (Menggunakan <i>stemming</i> tanpa MI)
2	Jumlah <i>neuron</i> pada <i>hidden layer</i>	5 (Nilai Random)
3	Jumlah <i>neuron</i> pada <i>output layer</i>	3 (Nilai Random)
4	Jumlah <i>input layer</i>	1
5	Jumlah <i>hidden layer</i>	1
6	Jumlah <i>output layer</i>	1
7	Nilai <i>learning rate</i>	0,01
		0,02
		0,03
		0,04
		0,05
8	Epoch (Iterate)	2000
9	MSE standar	0,0001
10	(a) Pangkat koefisien (nilai $\epsilon$ saat menghitung A2 pada perhitungan maju)	0,1

*Variable* yang tidak dituliskan pada Tabel 3-10 merupakan *variable* yang tidak perlu ditentukan terlebih dahulu nilainya pada proses perhitungan. Contoh proses perhitungan dapat dilihat pada Lampiran 2.

- 9) Spesifikasi *Hardware* dan *Software* perangkat penelitian.  
Tabel 3-11 merupakan spesifikasi *hardware* yang digunakan dalam proses pembangunan dan sistem pengujian pada penelitian.

**Tabel 3-11 Spesifikasi *Hardware***

#### Perangkat Keras

Laptop Notebook dengan Spesifikasi :
<ul style="list-style-type: none"> <li>▪ Processor : Intel(r) Pentium(r) CPU b960 2.20GHz</li> <li>▪ RAM : 6 gb</li> <li>▪ Memori Hardisk : 320 gb SATA</li> <li>▪ Layar : 8-13 inchi r : 800x600s</li> <li>▪ Keyboard Logitech K115</li> <li>▪ Mouse Wireless Logitech K115</li> </ul>

Tabel 3-12 berikut ini merupakan spesifikasi *software* yang digunakan dalam proses pembangunan dan sistem pengujian pada penelitian.

**Tabel 3-12 Spesifikasi *Software***

No	Perangkat Lunak	Versi
1	Sistem Informasi : Windows	10 Education
2	Browser : Google Chrome	Version 75.0.3770.142
3	Bahasa Pemrograman : Python	Version 3.7.3
4	IDE Python : Jupyter Notebook	Version 5.7.8
5	Microsoft Office	Version 2016

#### 4 Evaluasi

Pengujian perlu dilakukan untuk melihat hasil evaluasi dari sistem yang telah dibangun dan melihat performansi yang dihasilkan oleh sistem. Pengujian akan dilakukan dengan tiga proses skenario yang berbeda yaitu :

1. Skenario Pengujian dengan proses *stemming*
2. Skenario Pengujian dengan proses *Mutual Information*
3. Skenario pengujian dengan nilai learning rate

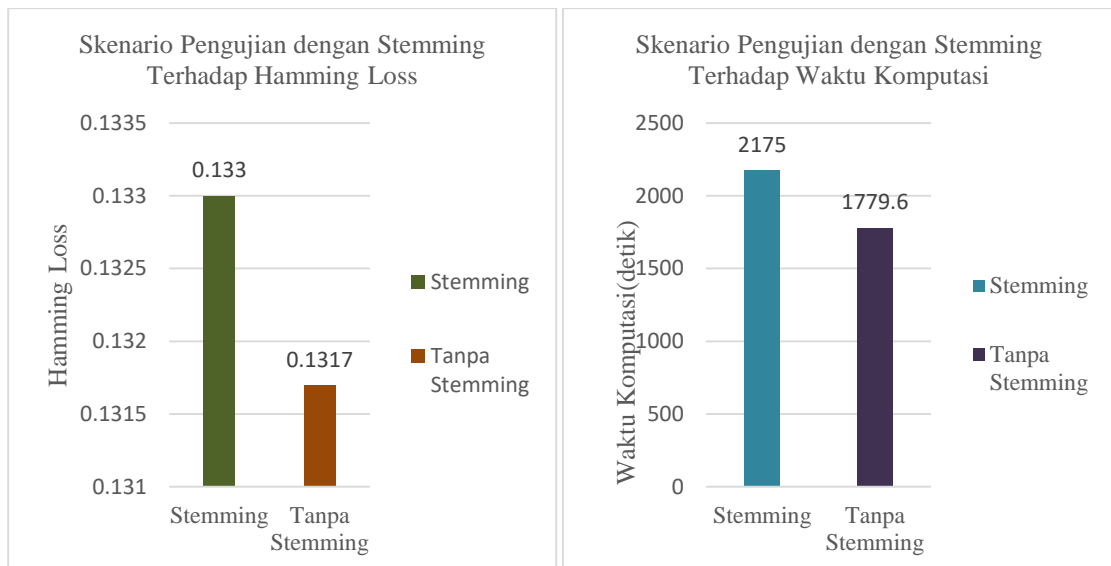
Berikutnya pada poin 4.1 merupakan rincian tahap proses skenario pengujian yang dilakukan.

##### 4.1 Hasil Pengujian

Pengujian terhadap sistem dilakukan untuk mendapatkan dan menganalisis performansi serta *hamming loss* dari sistem. Pengujian ini menggunakan *K-fold cross validation* dengan nilai  $k = 5$ . Data yang digunakan untuk setiap *fold* adalah jumlah 213 dokumen untuk *fold1, fold2, fold3* dan *fold4* sedangkan *fold5* berjumlah 212 dokumen dari total data sejumlah 1064.

1. Skenario pengujian dengan proses *Stemming*

Berikut ini merupakan data dalam bentuk *chart* yang memperlihatkan hasil skenario pengujian.

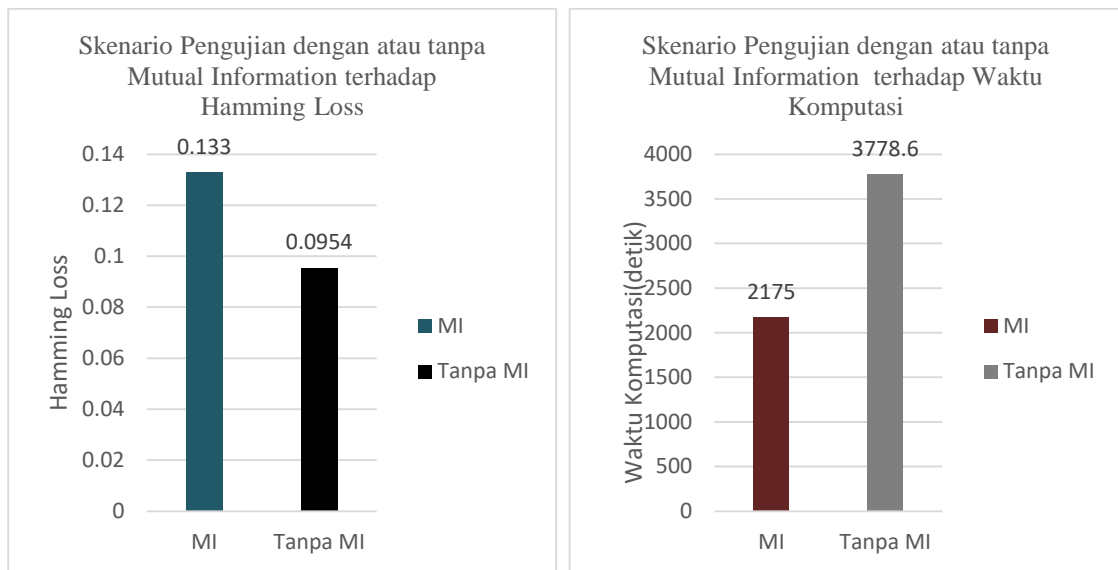


Gambar 4.1 Skenario Pengujian dengan proses *Stemming*

Pada gambar 4.1 adalah hasil skenario pengujian untuk melihat apa, dan ada atau tidaknya pengaruh proses *stemming* dan *non-stemming* terhadap hasil *hamming loss* serta waktu komputasi yang dibutuhkan pada kasus data *multi-label*. Nilai *hamming loss* pada pengujian tanpa proses *stemming* adalah 0,1317 dan dengan komputasi waktu 1779,6 s. Sedangkan nilai *hamming loss* pada pengujian dengan proses *stemming* adalah 0,1330 dan dengan komputasi waktu 2175 s. Maka dapat disimpulkan bahwa pengujian tanpa proses *stemming* lebih baik daripada pengujian menggunakan proses *stemming* dengan selisih *hamming loss* 0,0013 dan selisih waktu 395,4 s. Hal tersebut dapat terjadi karena pada data *multi-label*, perubahan kata yang terjadi pada proses *stemming* akan mempengaruhi proses pengklasifikasian data karena dapat mengubah makna kata dan menghilangkan ciri khusus pada data. Skenario pengujian ini menggunakan data yang juga melalui seleksi fitur *Mutual Information* dengan nilai *threshold* sebesar 0,0001.

2. Skenario pengujian dengan proses *Mutual Information*.

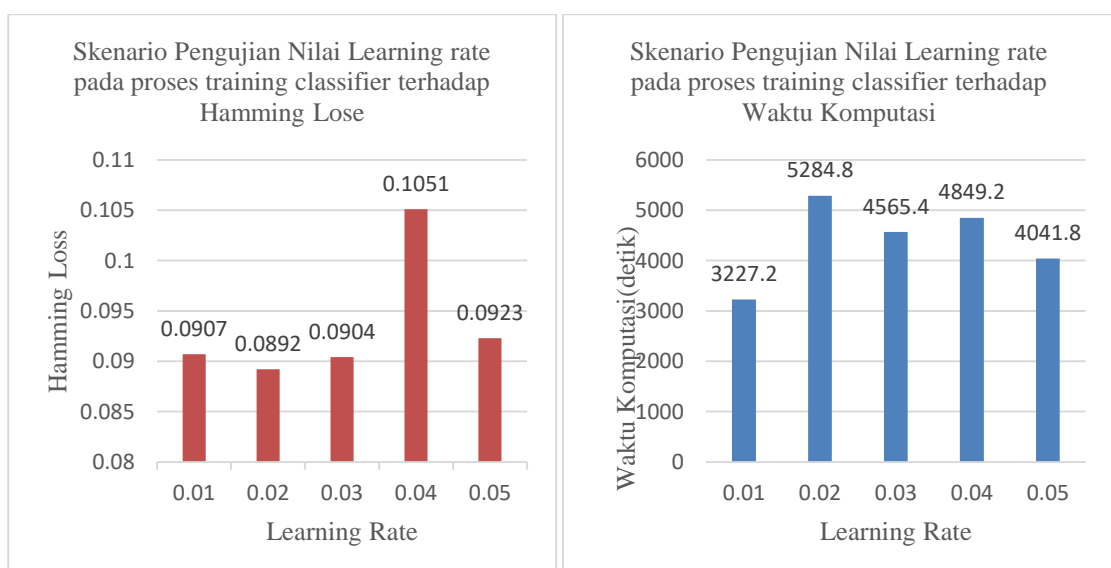
Berikut ini merupakan data dalam bentuk *chart* yang memperlihatkan hasil skenario pengujian.



Gambar 4.2 Skenario pengujian dengan atau tanpa *Mutual Information* terhadap *hamming loss*

Pada Gambar 4.2 Pengujian dengan menggunakan metode *Mutual Information* yang dibatasi dengan nilai *threshold*  $\geq 0,0001$  menghasilkan jumlah fitur sebanyak 1756 dan dengan nilai *hamming loss* sebesar 0,1330 dan dengan waktu 2175 s. Sedangkan pengujian tanpa menggunakan metode *Mutual Information* menghasilkan jumlah fitur sebanyak 12730 dengan nilai *hamming loss* sebesar 0,0954 dan dengan waktu 3778,6 s. Dengan selisih *hamming loss* sebesar 0,0376 maka penggunaan metode seleksi *Mutual Information* pada sisi performansi tidak lebih baik dibandingkan dengan tanpa penggunaan *Mutual Information*, ini terjadi karena kurangnya jumlah fitur memberikan pengaruh pada proses klasifikasi. Sedangkan dari sisi waktu komputasi dengan selisih 1603,6 s dimana penggunaan *mutual information* terbukti lebih cepat karena jumlah fitur yang lebih sedikit. Skenario pengujian ini menggunakan data yang sudah melalui proses *stemming*.

3. Skenario pengujian dengan nilai *learning rate*



Gambar 4.3 Skenario pengujian dengan nilai *learning rate*

Gambar 4.3 merupakan hasil skenario pengujian dengan nilai *learning rate* (salah satu fungsi aktivasi pada proses klaisifikasi *Backpropagation*) terhadap *hamming loss* dengan data *multi-label*. Nilai *learning rate* dimulai dari 0,01 hingga 0,05 dengan perubahan sebesar 0,01. Pada gambar 4.3 menunjukkan nilai *learning rate* 0,02 mendapatkan nilai *hamming loss* terbaik yaitu sebesar 0.0892 dan menunjukkan waktu komputasi selama 5284,8 s. Skenario pengujian ini menggunakan data yang sudah melalui *stemming* dan seleksi fitur *Mutual Information*.

#### 4.2 Analisis Hasil Pengujian

Hasil analisis dari pengujian yang telah dilakukan yaitu, skenario pengujian terhadap proses *stemming* menghasilkan perbandingan bahwa, pengujian tanpa menggunakan proses *stemming* menghasilkan nilai *hamming loss* yang lebih baik dengan selisih *hamming loss* 0,0013. Ini dikarenakan pengujian tanpa proses *stemming* tidak mengubah kata yang dapat menghilangkan makna dan ciri khusus sehingga dapat menaikkan tingkat perhitungan pada proses klasifikasi. Skenario Pengujian terhadap proses metode seleksi fitur *Mutual Information*, menghasilkan perbandingan bahwa penggunaan *Mutual Information* dari sisi performansi tidak lebih baik daripada pengujian tanpa menggunakan seleksi fitur *Mutual Information* dengan selisih nilai *hamming loss* 0,0376. Pengujian terhadap perbedaan nilai *learning rate* dari 0,01 – 0,05 dengan perubahan 0,01 terhadap *hamming loss* menghasilkan nilai *learning rate* 0,02 sebagai nilai *hamming loss* terbaik yaitu sebesar 0,0892.

#### 5 Kesimpulan

Berdasarkan dari beberapa seknario pengujian dan analisis yang telah dilakukan, maka dapat diambil kesimpulan sebagai berikut. Nilai *hamming loss* menggunakan seleksi fitur MI yang dibatasi dengan nilai  $\text{threshold} \geq 0,0001$  menghasilkan jumlah fitur sebanyak 1756 dan nilai *hamming loss* sebesar 0,1330 serta waktu 2175 s. Penggunaan seleksi fitur MI dari sisi performansi *hamming loss* tidak lebih baik daripada penggunaan tanpa seleksi fitur MI namun pada sisi waktu komputasi penggunaan MI lebih baik dengan selisih waktu 1603,6 s lebih cepat dibandingkan tanpa menggunakan seleksi fitur MI, ini terjadi karena jumlah fitur yang berkurang mempengaruhi proses pencarian fitur saat klasifikasi.

Penggunaan proses *stemming* pada data *multi-label* tidak lebih baik pengaruhnya terhadap proses klasifikasi dibandingkan dengan proses tanpa *stemming* dengan selisih nilai *hamming loss* 0,0013. Ini terjadi karena proses *stemming* akan menghilangkan ciri khas dari fitur untuk setiap label data. Hasil *hamming loss* terbaik yaitu sebesar 0,0892 dengan melibatkan tiga poin pengujian yaitu penggunaan: *stemming*, *Mutual Information* dan nilai *learning rate* terbaik.

Adapun saran untuk untuk penelitian lebih lanjut mengenai klasifikasi hadis *multi-label* dan fitur seleksi yang terkait dengan penelitian ini adalah sebagai berikut. Perlu dilakukan tahap *preprocessing* yang lebih baik agar dapat menghasilkan data yang lebih berkualitas, meminimalkan *noise* sekecil mungkin dan memenuhi kaidah ejaan yang benar.. Selain iu, dipandang perlu menerapkan cara untuk menangani ketidakseimbangan proporsi data pada suatu kelas. Selanjutnya, diharapkan dapat meningkatkan proses pelabelan data yang lebih baik untuk meningkatkan kualitas model klasifikasi.

#### Daftar Pustaka

- [1] Zein, Muhammad Ma'Shum."Ulumul Hadis dan Musthalah Hadis."(2008)
- [2] Sacra, Syair Audi Liri."Klasfikasi Anjuran,Larangan, dan Informasi pada Hadis Shahih Bukhari menggunakan Naïve Bayes Classifier". *E-Proceeding of Engineering*, 4(3), 2017
- [3] M. Y. Abu Bakar, Adiwijaya and S. A. Faraby, "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), Bandung, Indonesia, 2018, pp. 344-350.
- [4] Wiraguna,Adhitia." Klasifikasi Topik Multi Label pada Hadis Bukhari dalam Terjemahan Bahasa Indonesia Menggunakan Random Forest ".*e-Proceeding of Engineering* 6(1), 2019
- [5] Arista, Ulfa Maria."Analisis Sentimen Twitter Menggunakan Naïve Bayes Classifier dengan Seleksi Fitur Mutual Information". *J-COSINE* 2(2), 2018
- [6] E. Jasin, S. Faraby, and A. Adiwijaya. "Classification of hadith into positive suggestion, negative suggestion, and information". International Conference on Data and Information Science, Bandung, Indonesia, 2017

- [7] A. Kusumaningrum, S. Faraby, and A. Adiwijaya. Klasifikasi anjuran, larangan, dan informasi pada hadits shahih bukhari menggunakan metode support vector machine. *eProceedings of Engineering*, 4(3), 2017.
- [8] Tri, Wahyuni Rizki, "Penerapan Algoritma Cosine Similarity dan Pembobotan TF-IDF pada Sistem Klasifikasi Dokumen Skripsi". *Jurnal Teknik Elektro* 9(1), 2017
- [9] Sushma G. Autade, Dr. Gayatri M. Bhandari, "A Survey on a Bayesian Classification Approach Using Class-Specific Features for Text Categorization," *Imperial Journal of Interdisciplinary Research (IJIR)*, Vol-3, Issue-3, 2017
- [10] M. S. Mubarak, Adiwijaya, and M. D. Aldhi, "Aspect-based sentiment analysis to review products using Naïve Bayes," *AIP Conf. Proc.*, vol. 1867, no. 1, p. 20060, Aug. 2017.
- [11] A. Yusuf dan T. Priambadha, "Support Vector Machines yang didukung K-Means Clustering dalam klasifikasi dokumen," *JUTI*, vol. XI, no. 1, pp. 13-16. 2013
- [12] Puspongoro, N. H., Djuraidah, A., Fitrianto, A., & Sumertajaya, I. M. (2019). Geo-additive Models in Small Area Estimation of Poverty. *Journal of Data Science and Its Applications*, 2(1), 59-67.
- [13] Manik, A., Adiwijaya, A., & Utama, D. Q. (2019). Classification of Electrocardiogram Signals using Principal Component Analysis and Levenberg Marquardt Backpropagation for Detection Ventricular Tachyarrhythmia. *Journal of Data Science and Its Applications*, 2(1), 78-87.
- [14] S. Prangga, "Optimasi Parameter pada Support Vector Machine menggunakan Pendekatan Metode Taguchi untuk Data High-Dimensional," pp. 33-37, 2017
- [15] K. Yang, H. Wang, G. Dai, S. Hu, Y. Zhang, and J. Xu Determining the repeat number of cross-validation. In *Biomedical Engineering and Informatics (BMEI), 2011 4<sup>th</sup> International Conference on*, volume 3, pages 1706–1710. IEEE, 2011.
- [16] A. I. Suyanto and A. I. Msc. Searching, reasoning, planning, learning. *Informatika Bandung*, pages 184-186, 2014
- [17] I. Dilrukshi, K. De Zoysa, and A. Caldera. Twitter news classification using svm. In *Computer Science & Education (ICCSE), 2013 8<sup>th</sup> International Conference on*, pages 287-291. IEEE, 2013
- [18] Adiwijaya, *Matematika Diskrit dan Aplikasinya*, Bandung: Alfabeta, 2016
- [19] Naf'an, M. Z., Bimantara, A. A., Larasati, A., Risondang, E. M., & Nugraha, N. A. S. (2019). Sentiment Analysis of Cyberbullying on Instagram User Comments. *Journal of Data Science and Its Applications*, 2(1), 88-98.
- [20] Firmansyah, A.F.B. and Pramana, S., 2018. Ensemble Based Gustafson Kessel Fuzzy Clustering. *Journal of Data Science and Its Applications*, 1(1), pp.1-9.
- [21] Arifin, A.H.R.Z., Mubarak, M.S. and Adiwijaya, A., 2016, September. Learning struktur bayesian networks menggunakan novel modified binary differential evolution pada klasifikasi data. In *Indonesia Symposium on Computing (IndoSC) 2016*.
- [22] Adiwijaya, *Aplikasi Matriks dan Ruang Vektor*, Yogyakarta: Graha Ilmu, 2014.
- [22] Nhita, F., Saepudin, D. and Wisesty, U.N., 2015, December. Comparative study of moving average on rainfall time series data for rainfall forecasting based on evolving neural network classifier. In *2015 3rd International Symposium on Computational and Business Intelligence (ISCBI)* (pp. 112-116). IEEE.

