

**Klasifikasi Multilabel pada Hadis Bukhari dalam  
Terjemahan Bahasa Indonesia Menggunakan *Information  
Gain* dan *Backpropagation Neural Network***

**Tugas Akhir**  
**diajukan untuk memenuhi salah satu syarat**  
**memperoleh gelar sarjana**  
**dari Program Studi Teknik Informatika**  
**Fakultas Informatika**  
**Universitas Telkom**

**1301141071**  
**Muhammad Yuslan Abubakar**



**Program Studi Sarjana Teknik Informatika**  
**Fakultas Informatika**  
**Universitas Telkom**  
**Bandung**

**2018**

# LEMBAR PENGESAHAN

**Klasifikasi Multilabel pada Hadis Bukhari dalam Terjemahan Bahasa Indonesia  
Menggunakan *Information Gain* dan *Backpropagation Neural Network***

***Multilabel Topic Classification for Hadith of Bukhari in Indonesian Translation using  
Information Gain and Backpropagation Neural Network***

**NIM: 1301141071**

**Muhammad Yuslan Abubakar**

Tugas akhir ini telah diterima dan disahkan untuk memenuhi sebagian syarat memperoleh  
gelar pada Program Studi Sarjana Teknik Informatika

Fakultas Informatika

Universitas Telkom

Bandung, 28 Mei 2018

Menyetujui

Pembimbing I

Pembimbing II

Prof. Dr. Adiwijaya, S.Si., M.Si.

NIP: 00740046

Said Al Faraby, S.T., M.Sc.

NIP: 15890019

Ketua Program Studi  
Sarjana Teknik Informatika,

Said Al Faraby, S.T., M.Sc.

NIP: 15890019

## LEMBAR PERNYATAAN

Dengan ini saya, Muhammad Yuslan Abubakar, menyatakan sesungguhnya bahwa Tugas Akhir saya dengan judul ” **Klasifikasi Multilabel pada Hadis Bukhari dalam Terjemahan Bahasa Indonesia Menggunakan *Information Gain* dan *Backpropagation Neural Network*** ” beserta dengan seluruh isinya adalah merupakan hasil karya sendiri, dan saya tidak melakukan penjiplakan yang tidak sesuai dengan etika keilmuan yang berlaku dalam masyarakat keilmuan. Saya siap menanggung resiko/sanksi yang diberikan jika dikemudian hari ditemukan pelanggaran terhadap etika keilmuan dalam buku TA atau jika ada klaim dari pihak lain terhadap keaslian karya.

Bandung, 28 Mei 2018

Yang Menyatakan,

Muhammad Yuslan Abubakar

# Klasifikasi Multilabel pada Hadis Bukhari dalam Terjemahan Bahasa Indonesia Menggunakan *Information Gain* dan *Backpropagation Neural Network*

Muhammad Yuslan Abubakar<sup>1</sup>, Adiwijaya<sup>2</sup>, Said Al Faraby<sup>3</sup>

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

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

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

---

## Abstrak

Hadis merupakan sumber hukum dan pedoman kedua bagi umat Islam setelah Al-Qur'an. Banyak sekali hadis-hadis yang telah diriwayatkan oleh para ahli hadis selama ini. Penelitian ini membangun sebuah sistem yang dapat melakukan klasifikasi terhadap hadis Bukhari terjemahan berbahasa Indonesia berupa *backpropagation neural network* dengan dibantu *information gain* sebagai seleksi fitur untuk data multilabel dan *single label* yang belum pernah dilakukan sebelumnya. Hasil yang didapatkan bahwa sebesar 94.56% data hadis multilabel dapat diklasifikasikan dengan benar dan 65.71% data hadis *single label* diklasifikasikan dengan benar menggunakan bantuan *information gain* dalam pemilihan fitur.

**Kata kunci :** hadis, *backpropagation neural network*, *information gain*, multilabel, seleksi fitur, klasifikasi teks

---

## Abstract

Hadith is the second source and legal guidance after Al-Qur'an for muslim. There are so many hadith that have been narrated by the experts of hadith so far. This research build a system that can classify shahih hadith of Bukhari in the form of *backpropagation neural network* with the help of *information gain* as feature selection for multilabel and single label data which has never been done before. Results show that 94.56% of multilabeled hadith can be correctly classified and 65.71% of single label hadith data is correctly classified using *information gain* as feature selection.

**Keywords:** hadith, *backpropagation neural network*, *information gain*, multilabel, feature selection, text classification

---

## 1. Pendahuluan

Hadis merupakan sumber hukum kedua setelah Al-Qur'an sebagai pedoman hidup umat Islam [1]. Banyak sekali hadis yang telah diriwayatkan oleh para ahli hadis dan telah dikelompokkan kedalam beberapa kategori untuk memudahkan orang lain memahami hadis-hadis tersebut. Salah satu ahli hadis yang telah meriwayatkan banyak hadis dari Rasulullah SAW adalah Bukhari.

Hadis digunakan untuk membantu umat Islam khususnya dalam memahami ilmu-ilmu agama Islam. Banyak hadis-hadis yang menganjurkan kita untuk berbuat dalam kebaikan. Ada juga yang melarang dalam hal berbuat dosa kepada Allah SWT. Untuk dapat mengetahui hal tersebut, dibutuhkan suatu sistem untuk mengidentifikasi apakah hadis tersebut bersifat anjuran, larangan atau sekedar informasi saja. Dari hal itu, timbulah pertanyaan bagaimana sebuah hadis terjemahan berbahasa Indonesia dapat diklasifikasikan kedalam bentuk yang lebih khusus yaitu berupa hadis anjuran, hadis larangan dan hadis informasi. Suatu hadis bisa saja merupakan salah satu jenis dari ketiga kelas tersebut namun tidak menutup kemungkinan bahwa termasuk gabungan dari ketiganya.

Namun saat ini belum ditemukan adanya penelitian yang melakukan klasifikasi hadis kedalam beberapa kelompok klasifikasi atau yang biasa disebut dengan *multilabel classification*. Hal ini diperlukan mengingat bahwa suatu hadis tidak hanya mengandung informasi saja melainkan adanya pesan khusus yang ingin disampaikan seperti bentuk anjuran dan larangan mengenai suatu topik keilmuan tertentu.

Atas dasar itulah penulis akan memfokuskan penelitian dalam membuat model klasifikasi berdasarkan hadis Bukhari sebanyak 1064 data hadis kedalam bentuk klasifikasi multilabel. Hadis-hadis tersebut telah diberikan label sebelumnya dengan tujuan untuk melakukan proses pelatihan pada sistem. *Information Gain* dipilih sebagai seleksi fitur karena metode ini bekerja dengan cara memberikan nilai kepada setiap fitur dan hanya akan dipilih fitur-fitur yang berkorelasi terhadap kelas yang sesuai. Adapun metode yang akan digunakan adalah *Backpropagation Neural Network* untuk menyelesaikan masalah yang telah dipaparkan sebelumnya. Metode *Backpropagation Neural Network* dipilih berdasarkan penelitian yang telah dilakukan sebelumnya oleh Min-Ling Zhang dan

Zhi-Hua Zhou mengenai *Multilabel Neural Networks with Applications to Functional Genomics and Text Categorization* [9]. Hasil yang didapatkan dari penelitian itu menunjukkan bahwa metode *Backpropagation Neural Network* dapat melakukan klasifikasi teks multilabel lebih baik dibandingkan dengan beberapa metode lain. Fitur yang telah didapatkan pada proses seleksi dan ekstraksi fitur/ciri akan digunakan sebagai masukan (*input*) pada sistem klasifikasi *Backpropagation Neural Network*. Sementara itu, hasil yang didapatkan dari proses klasifikasi merupakan label kelas dari masing-masing data.

Beberapa masalah yang dibatasi pada penelitian ini adalah penggunaan *dataset* sebesar 1064 data hadis terjemahan berbahasa Indonesia dengan tiga label yaitu anjuran, larangan dan informasi. Proporsi data untuk masing-masing kelas juga tidak merata, terbukti dengan adanya satu kelas yang memiliki jumlah data sekitar 90% dari keseluruhan data. Sulitnya mencari data dengan 3 jenis label tersebut juga menjadi salah satu hambatan dalam penelitian ini.

Metode evaluasi diperlukan untuk mengukur seberapa baik sistem yang telah dibangun dan diterapkan pada penelitian ini. *Hamming Loss* dipilih sebagai metrik evaluasi untuk mengukur performansi dari hasil klasifikasi multilabel yang telah didapatkan. *Hamming Loss* merupakan salah satu metrik evaluasi yang umum digunakan untuk pengujian data multilabel dengan cara menghitung *error rate* dari hasil klasifikasi.

Beberapa bagian selanjutnya pada jurnal ini adalah sebagai berikut. Pada bagian kedua, terdapat studi terkait yang menjelaskan mengenai penelitian sebelumnya yang berkaitan dengan penelitian yang sedang dikerjakan saat ini serta beberapa tinjauan pustaka terkait penelitian. Pada bagian ketiga, akan dijelaskan mengenai sistem yang akan dibangun pada penelitian ini yaitu klasifikasi pada hadis Bukhari terjemahan berbahasa Indonesia dengan menggunakan metode *Information Gain* dan *Backpropagation Neural Network*. Kemudian pada bagian keempat, akan dipaparkan hasil pengujian dan analisis hasil yang telah didapatkan. Selanjutnya pada bagian kelima, akan dijelaskan mengenai kesimpulan yang didapatkan dari penelitian ini.

## 2. Studi Terkait

Hadis dapat diklasifikasikan ke banyak kelompok khusus. Seperti yang telah dijelaskan pada [1], hadis dapat dikategorikan sebagai hadis shahih, maudu', dhaif dan hasan. Hadis juga dapat dikategorikan sebagai hadis anjuran, hadis larangan dan hadis informasi seperti yang dijelaskan oleh Eliza [2] dan Andina [3]. Klasifikasi hadis kedalam beberapa kelompok diperlukan untuk bisa mendapatkan informasi yang lebih khusus dari hadis tersebut. Dengan mengetahui kelompok suatu hadis, diharapkan pencarian informasi terkait hadis tersebut dapat dengan mudah dilakukan.

Penelitian terkait klasifikasi hadis dalam terjemahan berbahasa Indonesia sebelumnya adalah untuk melakukan klasifikasi apakah hadis tersebut merupakan hadis kuat (shahih) atau lemah. Sejauh ini klasifikasi hadis dengan kelas hadis informasi, hadis larangan dan hadis anjuran telah dilakukan pada penelitian [2] dan [3]. Penelitian tersebut melakukan klasifikasi hanya kedalam satu kelas dari tiga kelas yang ada atau dengan kata lain model klasifikasi yang dibangun hanya klasifikasi *multiclass*. Adapun kebanyakan penelitian serupa yang sebelumnya telah dilakukan masih dalam bentuk Bahasa Arab.

Kawther A. Aldhlan [1], telah melakukan suatu mekanisme untuk meningkatkan performansi pada klasifikasi hadis. Metode yang digunakan adalah *Decision Tree* karena menurutnya merupakan pendekatan yang tepat untuk klasifikasi hadis dikarenakan kemudahan induksi aturan dan hasil interpretasi. Metode ini juga digunakan karena memiliki kemampuan yang dapat mengatasi nilai (*value*) yang hilang namun usaha yang dibutuhkan untuk mencapai hal itulah yang dianggap sebagai kelemahannya. Mengabaikan nilai yang hilang ini dapat menyebabkan keputusan yang kritis seperti kesalahan klasifikasi hadis. Untuk mengatasi hal ini, mereka menggunakan suatu mekanisme yang disebut dengan *Missing Data Detector* (MDD).

Sedangkan Al-Kabi [4], menggunakan metode TF-IDF dengan bantuan Microsoft Visual Basic Programming Language. Alasan penggunaan bahasa pemrograman ini adalah karena mendukung tulisan/teks Arab dan data dalam bentuk dokumen/file. Adapun klasifikasi yang dilakukan adalah kedalam 8 bentuk kelas yaitu "*Knowledge*", "*Praying*", "*Eclipse*", "*Call to Prayer*", "*Faith*", "*Good Manners*", "*Fasting*" dan "*Almsgiving*". Hadis yang digunakan dalam klasifikasi ini adalah hadis Bukhari dengan rata-rata akurasi yang didapat adalah sebesar 83.2%. Hasil akurasi yang didapatkan ini bergantung penuh pada proses *stopwords* dan *stemming*.

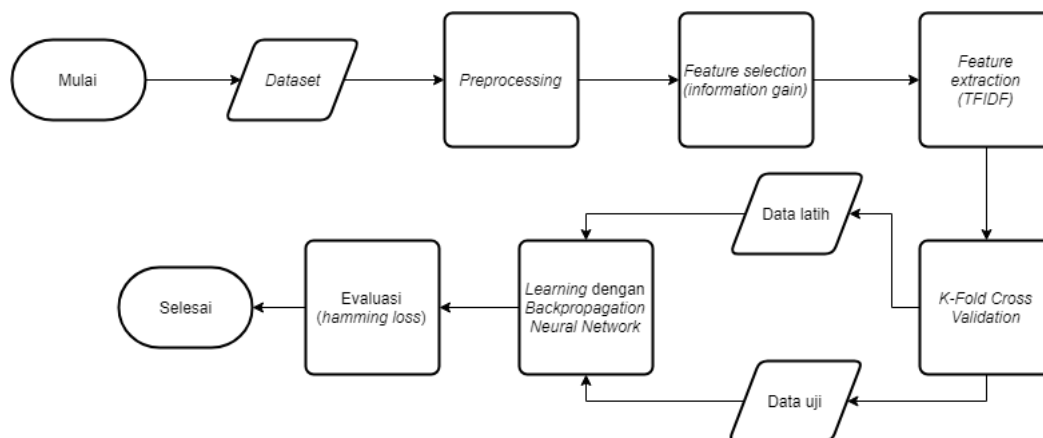
Eliza J [2] melakukan klasifikasi hadis anjuran, hadis larangan dan hadis informasi pada hadis shahih Bukhari berdasarkan model Unigram menggunakan *Artificial Neural Network* (ANN). Ia juga menggunakan bantuan metode ekstraksi fitur TF-IDF untuk mendapatkan nilai-nilai pada setiap kata. Hasil performansi yang didapatkan dengan *F1-Score* sebesar 85%. Penelitian yang serupa juga telah dilakukan oleh Andina K [3] pada tahun 2017 dengan menggunakan bantuan metode *Support Vector Machine*. *K-Fold* digunakan untuk melakukan pembagian data menjadi data latih dan data uji. Rata-rata performansi yang didapatkan dengan menggunakan metode ini adalah sebesar 88%.

Untuk kasus *multi label* pada teks, sudah ada beberapa penilitan terkait hal tersebut. Min-Ling Zhang dan Zhi-Hua Zhou [9] pada tahun 2006 telah melakukan penelitian terkait *multi label* pada kategorisasi teks. Metode yang digunakan berupa *neural network* yang bernama BP-MLL (*Bacpropagation for Multilabel Learning*). Metode ini berasal dari algoritma *backpropagation* yang populer dengan menggunakan fungsi *error* yang menangkap karakteristik pembelajaran *multi label*, yaitu label dari suatu *instance* (fitur) harus diberi peringkat lebih tinggi daripada yang tidak termasuk dalam *instance* tersebut. Mereka menggunakan metode tersebut dikarenakan hasil akurasi yang didapat lebih tinggi daripada metode *multilabel-learning* lainnya seperti BoosTexter, ADTBoost.MH, Rank-SVM dan BasicBP.

### 3. Sistem yang Dibangun

#### 3.1 Rancangan Sistem

Pada pengerjaan penelitian ini, sistem akan mampu melakukan klasifikasi hadis terjemahan berbahasa Indonesia terhadap 3 (tiga) kelas seperti yang telah didefinisikan sebelumnya. Sistem akan terbagi kedalam beberapa proses utama yaitu proses *preprocessing* yang akan menghasilkan data bersih, dilanjutkan dengan proses *feature selection* yang akan memilah fitur, proses ekstraksi fitur yang akan memberikan nilai untuk tiap-tiap fitur dan yang terakhir adalah proses klasifikasi yang akan menentukan kelas dari tiap hadis. Secara umum, sistem akan ditunjukkan pada Gambar 1.



Gambar 1. Flowchart sistem

#### 3.2 Dataset

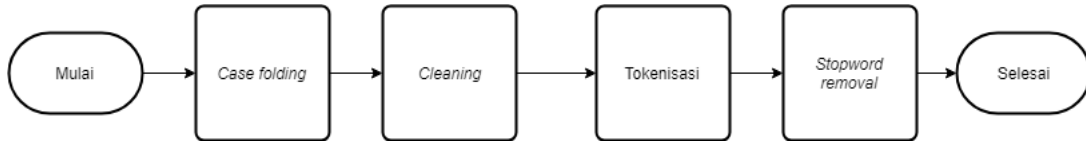
*Dataset* yang digunakan pada penelitian ini adalah berupa hadis shahih Bukhari terjemahan berbahasa Indonesia sebanyak 1064 data multilabel serta 455 data *single label* yang telah diberi label sebelumnya. Label kelas pada data multilabel terdiri dari 3 (tiga) kelas yaitu kelas anjuran, kelas larangan dan kelas informasi sedangkan pada data *single label* terdiri dari 5 (lima) kelas yaitu kelas 1 adalah tentang "iman", kelas 2 adalah tentang "ilmu", kelas 3 adalah tentang "wudhu", kelas 4 adalah tentang "shalat", dan kelas 5 adalah tentang "waktu-waktu shalat". Adapun masing-masing data hadis multilabel dapat berupa salah satu dari ketiga kelas tersebut maupun gabungan dari beberapa kelas. Representasi data tersebut akan ditampilkan seperti pada Tabel 1 berikut.

Tabel 1. Representasi Data

Data	Kelas 1	Kelas 2	Kelas 3	Kelas Kitab
Iman itu ada enam puluh lebih cabangnya, dan malu adalah salah satu cabang iman	0	0	1	1

### 3.3 Preprocessing

*Preprocessing* digunakan untuk menghilangkan *noise* yang terdapat pada *dataset*. Proses lengkap dari *preprocessing* digambarkan pada Gambar 2 berikut.



**Gambar 2.** Tahapan *Preprocessing*

Beberapa proses dari tahapan *preprocessing* diatas akan dijelaskan seperti berikut.

1. *Case folding*, merupakan proses untuk merubah setiap kata menjadi huruf kecil dan untuk menyeragamkan setiap kata yang ada. *Case folding* pada penelitian ini menggunakan bantuan fungsi `str.lower()` yang ada pada bahasa pemrograman Python.
2. *Cleaning*, merupakan proses untuk menghilangkan tanda baca dan atau simbol yang ada pada data. Proses ini memerlukan bantuan *library* yang ada pada bahasa pemrograman Python yaitu *Regular Expression*.
3. Tokenisasi, merupakan proses yang akan memenggal setiap kata yang terdapat pada setiap data masukan. Proses tokenisasi pada penelitian ini menggunakan fungsi `str.split()` yang ada pada bahasa pemrograman Python.
4. *Stopword removal*, merupakan proses untuk menghilangkan kata-kata yang dianggap kurang penting pengaruhnya. Biasanya ciri-ciri dari kata-kata ini adalah frekuensi kemunculannya yang cukup sering dibandingkan dengan kata-kata lain.

### 3.4 Feature Selection

Setelah didapatkan data bersih melalui proses *preprocessing*, langkah selanjutnya adalah memilih dan menentukan fitur-fitur yang dibutuhkan untuk proses klasifikasi. Tahapan ini disebut juga dengan *feature selection*. Tahapan ini akan menentukan fitur/kata apa saja yang relevan dengan masing-masing kelas. Seleksi fitur yang digunakan adalah *information gain* dimana seleksi fitur ini akan menghitung nilai dari setiap kata dalam sekumpulan kata. Kata-kata dengan nilai *information gain* yang tinggi akan menjadi fitur utama untuk setiap data dalam proses klasifikasi. Persamaan (1) berikut ini akan menjelaskan perhitungan untuk menghitung nilai *information gain*.

$$I(C,A) = H(C) - H(C|A), \quad (1)$$

dimana  $H(C) = -\sum_{c \in C} p(C) \log p(C)$  merupakan entropi dari kelas dan  $H(C|A) = -\sum_{c \in C} p(C|A) \log p(C|A)$  merupakan kondisional entropi dari tiap kelas terhadap atribut/kata [5].

Semakin besar nilainya maka semakin menandakan bahwa kata tersebut berpengaruh pada kelas-kelas yang ada. Setelah itu, kata-kata dengan nilai diatas standar (*threshold*) akan dipilih sebagai fitur utama dalam melakukan klasifikasi. Adapun *threshold* ini dapat ditentukan dengan melihat plot dari setiap kata yang didapatkan.

### 3.5 Feature Extraction

*Term Frequency-Inverse Document Frequency* atau biasa disebut dengan TF-IDF adalah salah satu metode ekstraksi fitur yang populer digunakan khususnya pada klasifikasi teks. TF-IDF merepresentasikan suatu matriks dimana setiap baris pada matriks tersebut sebagai data dan kolom yang ada sebagai kata atau fitur (*term*) [8]. Matriks ini akan diisi oleh suatu nilai atau bobot setiap kata terhadap suatu dokumen. Bobot yang telah didapatkan akan menjadi masukan untuk sistem klasifikasi *neural network*. Adapun TF (*Term Frequency*) merupakan jumlah kemunculan tiap kata pada setiap dokumen dan IDF (*Inverse Document Frequency*) merupakan jumlah dokumen terkait yang mengandung suatu kata tertentu [3]. Perhitungan bobot pada TF-IDF didapatkan seperti pada persamaan (2) berikut.

$$w_{ij} = tf \times idf = tf_{ij} \times \log \frac{D}{df_i}, \quad (2)$$

dimana :

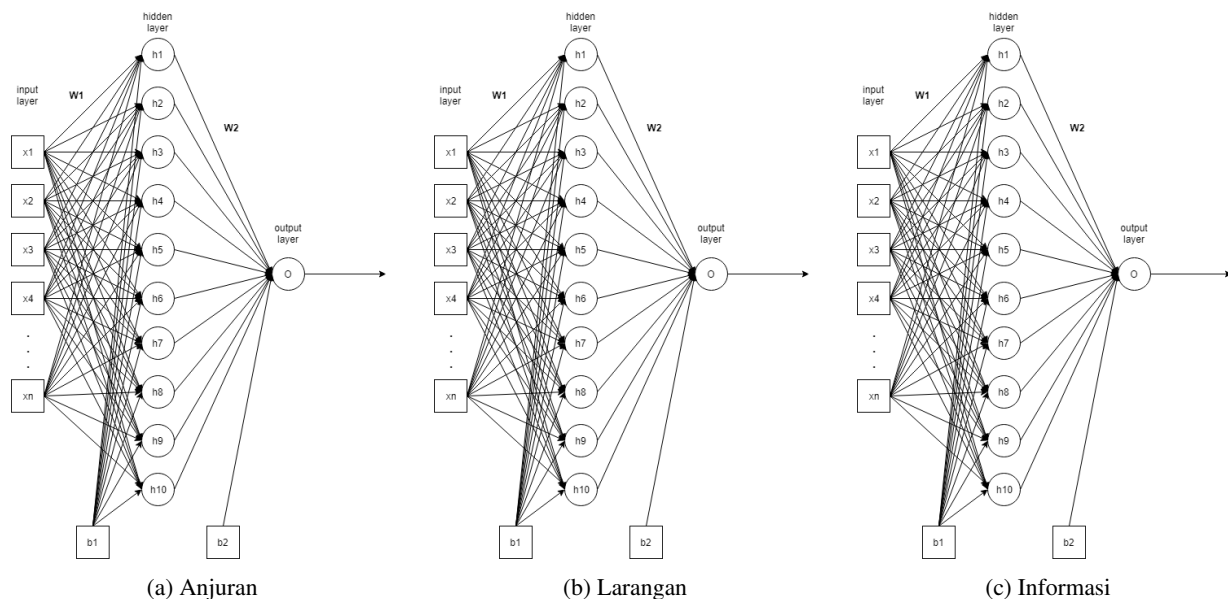
$w_{ij}$  = bobot kata  $t_j$  terhadap dokumen  $d_i$   
 $tf_{ij}$  = jumlah kemunculan kata  $t_j$  dalam dokumen  $d_i$   
 $D$  = jumlah dokumen  
 $df_i$  = jumlah kemunculan kata dalam dokumen  $D$

### 3.6 K-Fold Cross Validation

Pada *k-fold cross validation*, kumpulan data  $D$  dipartisi secara acak menjadi  $k$ -bagian dengan ukuran yang kira-kira sama,  $D = \{D_1, D_2, D_3, \dots, D_k\}$ . Setiap satu  $k$ -bagian pada satu perulangan  $k$  akan digunakan sebagai data uji dan sisanya akan digunakan sebagai data latih untuk *classifier* yang dibangun. Perkiraan tingkat kesalahan dihitung dari proporsi kesalahan dari seluruh bagian sampel [7]. Pada penelitian kali ini digunakan nilai  $k = 5$ .

### 3.7 Klasifikasi

Pada penelitian ini, *classifier* yang dibangun adalah *Backpropagation Neural Network*. *Classifier* ini berkerja dengan melakukan dua tahap perhitungan yaitu perhitungan maju yang akan menghitung nilai kesalahan (*error*) antara nilai keluaran sistem dengan nilai yang seharusnya dan perhitungan mundur untuk memperbaiki bobot berdasarkan nilai *error* tersebut [6]. Data latih digunakan pada proses *learning* untuk mendapatkan nilai-nilai parameter yang mengeluarkan hasil terbaik pada sistem *Backpropagation Neural Network*. Hasil terbaik didapatkan ketika nilai *error* yang ada sudah memenuhi standar atau dengan kata lain nilai *error*-nya sangat kecil. Gambar ?? berikut ini merupakan arsitektur *backpropagation neural network* yang dibangun pada penelitian ini.



**Gambar 3.** Arsitektur *Backpropagation Neural Network* untuk Data Multilabel

Dapat dilihat pada Gambar 3 diatas bahwa terdapat 3 (tiga) arsitektur *backpropagation neural network* yang dibangun. Masing-masing arsitektur jaringan berfungsi melatih data kedalam tiap kelasnya masing-masing. Adapun langkah-langkah pelatihan pada *classifier* ini seperti berikut [6].

1. Untuk setiap model arsitektur, tentukan matriks masukan ( $P$ ).
2. Inisialisasi nilai *learning rate*, nilai ambang MSE dan epoch maksimal untuk kondisi berhenti serta nilai awal bobot dan bias untuk setiap layer. Nilai *learning rate* yang digunakan = 0.08, nilai ambang MSE = 0.01, jumlah epoch maksimal = 1000 dan nilai untuk bobot serta bias akan dibangkitkan secara random menggunakan *random uniform* dengan rentang nilai [0,1] dan menggunakan *random seed* = 1. *Random seed* ini berguna agar setiap kali nilai acak dibangkitkan akan selalu sama dengan sebelumnya.
3. Hitung nilai keluaran pada *hidden layer* dengan menggunakan fungsi aktivasi sigmoid seperti yang ditunjukkan pada persamaan 3 lalu selanjutnya hasil keluaran dari *hidden layer* digunakan sebagai masukan untuk menghitung hasil keluaran pada *output layer* menggunakan persamaan 4.



$$A1 = \frac{1}{1 + e^{-(\alpha * W1 * P)}} + B1 \quad (3)$$

$$A2 = \frac{1}{1 + e^{-(\alpha * W2 * A1)}} + B2 \quad (4)$$

4. Hitung nilai *error* dengan membandingkan hasil keluaran dari *output layer* dengan label asli setiap data tersebut menggunakan persamaan 5 berikut.

$$E = T - A2 \quad (5)$$

5. Lakukan pelatihan mundur untuk memperbaharui nilai-nilai bobot sebelumnya. Adapun perhitungan untuk perbaikan bobot-bobot ditunjukkan seperti pada persamaan-persamaan berikut.

$$D2 = (A2 * (1 - A2)) * E \quad (6)$$

$$D1 = (A1 * (1 - A1)) * (W2 * D2) \quad (7)$$

$$dW1 = lr * D1 * P \quad (8)$$

$$dB1 = lr * D1 \quad (9)$$

$$dW2 = lr * D2 * P \quad (10)$$

$$dB2 = lr * D2 \quad (11)$$

6. Setelah selesai satu pelatihan (satu epoch), maka hitung nilai MSE dengan persamaan 12 berikut,

$$MSE = \frac{\sum E^2}{N} \quad (12)$$

dimana  $N$  merupakan banyak data.

7. Seluruh bobot akan diperbaharui nilainya oleh perhitungan mundur sebelumnya dengan menerapkan persamaan-persamaan berikut ini.

$$W1 = W1 + dW1; \quad W2 = W2 + dW2; \quad W2 = W2 + dW2; \quad B2 = B2 + dB2; \quad (13)$$

### 3.8 Pengukuran Performansi

Salah satu cara mengetahui apakah sistem yang dibangun sudah baik atau belum adalah dengan mengukur performansi klasifikasi yang dihasilkan. Pada kasus multilabel, *hamming loss* merupakan salah satu metrik pengukuran performansi yang sering digunakan. *Hamming loss* akan menghitung nilai kesalahan (*error*) dari hasil klasifikasi. Performansi yang dihasilkan akan sempurna apabila nilai  $hloss(h) = 0$ . Semakin kecil nilai  $hloss(h)$  yang dihasilkan, maka semakin baik pula performansi sistem. Persamaan (14) akan menunjukkan perhitungan dari  $hloss(h)$ .

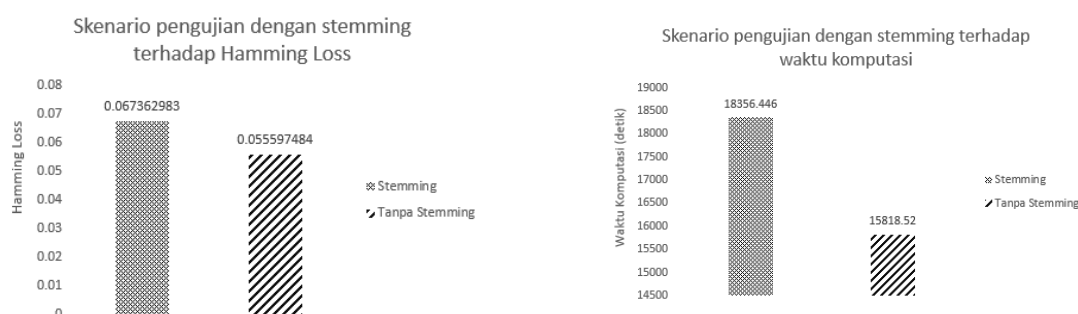
$$hloss(h) = \frac{1}{p} \sum_{i=1}^p \frac{1}{Q} |h(x_i) \Delta Y_i| \quad (14)$$

dimana  $p$  merupakan banyak data yang dilakukan klasifikasi dan  $Q$  adalah jumlah label kelas. Sedangkan  $|h(x_i) \Delta Y_i|$  merupakan banyaknya kesalahan klasifikasi yang terjadi.

## 4. Evaluasi

Untuk melihat hasil evaluasi pada *Backpropagation Neural Network* yang dibangun, dilakukan beberapa skenario pengujian. Pengujian pertama dilakukan untuk mengetahui pengaruh dari efek *stemming* pada data hadis Bukhari terjemahan berbahasa Indonesia terhadap *hamming loss* dan waktu komputasi yang dihasilkan. Untuk melakukan proses *stemming*, digunakan bantuan *library* Sastrawi.Stemmer.StemmerFactory(). Pengujian selanjutnya dilakukan untuk melihat pengaruh perubahan batas ambang *threshold* pada *information gain* terhadap *hamming loss* yang didapatkan. Terakhir adalah pengujian dilakukan pada perubahan nilai  $\alpha$  yang terdapat pada fungsi aktivasi sigmoid terhadap hasil *hamming loss* pada data multilabel dan perubahan nilai *learning rate* yang terdapat pada proses *learning* sistem terhadap hasil F1-Score yang didapatkan pada data *single label*. Berikut adalah beberapa skenario pengujian untuk data multilabel dan *single label*.

### 1. Skenario Pengujian dengan Proses *Stemming*



Gambar 4. Skenario pengujian dengan proses *stemming* pada multilabel

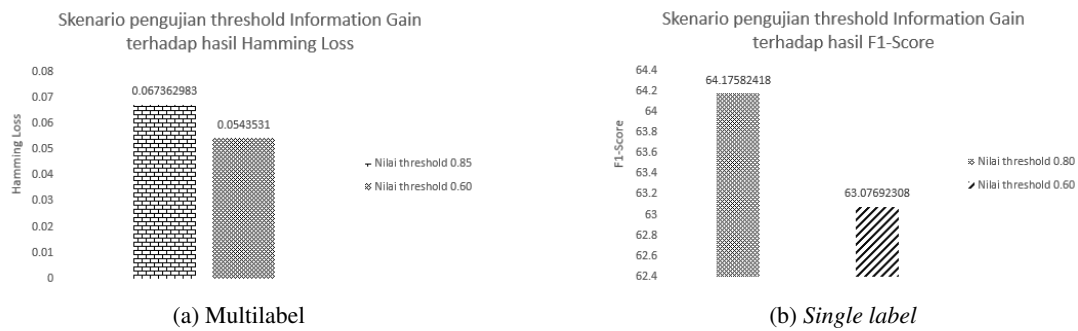


Gambar 5. Skenario pengujian dengan proses *stemming* pada *single label*

Pada Gambar 4 dilakukan skenario pengujian proses *stemming* dan *non-stemming* terhadap hasil *hamming loss* dan waktu komputasi yang didapatkan pada data multilabel. Hasil *hamming loss* menunjukkan bahwa pengujian tanpa proses *stemming* mendapatkan hasil yang lebih baik dibandingkan dengan menggunakan proses *stemming* yaitu sebesar 0.055597484 atau dapat dikatakan bahwa sekitar 94.44% data berhasil diklasifikasikan dengan benar. Sedangkan untuk waktu komputasi, pengujian tanpa proses *stemming* juga mendapatkan hasil yang lebih baik dibandingkan dengan menggunakan proses *stemming* yaitu selama 15818.52 detik. Hal ini dikarenakan pada data multilabel, setiap kata tidak bisa digeneralisasi dengan menggunakan proses *stemming* karena dapat menghilangkan ciri khusus pada data multilabel hadis Bukhari seperti pada label anjuran, kata "-lah" menunjukkan perintah anjuran seperti kata "tinggalkan-lah". Kata ini akan berbeda maknanya jika digeneralisasi dengan menggunakan proses *stemming*.

Adapun untuk pengujian pada data *single label*, didapatkan hasil F1-Score sebesar 64.17582418% data berhasil diklasifikasikan dengan benar dan waktu komputasi yang lebih baik yaitu selama 4183.39 detik dengan menggunakan proses *stemming* dibandingkan tanpa menggunakan proses *stemming* seperti yang ditunjukkan pada Gambar 5. Ini disebabkan bahwa imbuhan pada suatu kata dapat digeneralisasi menjadi satu kata dengan menggunakan proses *stemming* seperti pada data dengan label shalat, kata "shalat-lah" dan "menyalatkan" dapat digeneralisasi menjadi kata "shalat".

### 2. Skenario Pengujian dengan Threshold Information Gain

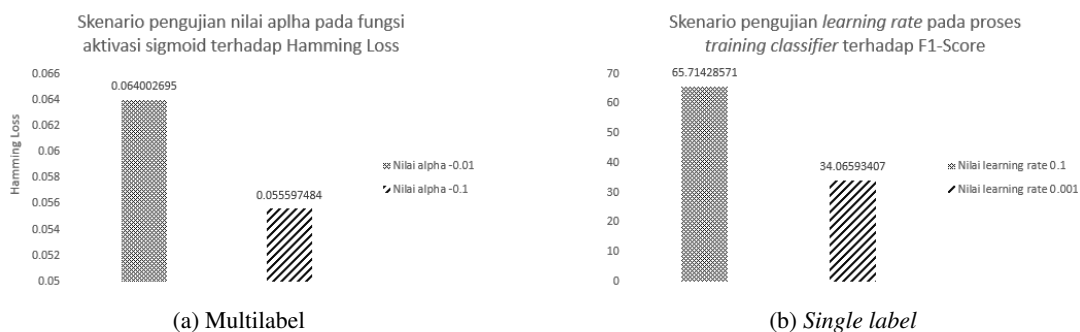


**Gambar 6.** Skenario Pengujian dengan Threshold Information Gain

Pada Gambar 6(a) menunjukkan bahwa skenario pengujian pada data multilabel dengan menggunakan ambang batas *threshold* sebesar 0.6 lebih baik dibandingkan dengan menggunakan ambang batas *threshold* sebesar 0.85 dengan hasil *hamming loss* sebesar 0.0543531 atau dapat dikatakan bahwa sekitar 94.56% data berhasil diklasifikasikan dengan benar. Hal ini dikarenakan kata-kata atau fitur yang terdapat pada masing-masing label memiliki nilai *information gain* yang sama rata dan sama-sama berpengaruh untuk masing-masing label.

Sedangkan untuk data *single label* menunjukkan bahwa hasil F1-Score yang didapatkan lebih baik ketika menggunakan ambang batas *threshold* sebesar 0.8 dibandingkan dengan menggunakan ambang batas *threshold* sebesar 0.6 dengan nilai F1-Score sebesar 64.17582418% data dapat diklasifikasikan dengan benar seperti yang ditunjukkan pada Gambar 6(b). Hal ini dikarenakan fitur dengan nilai *information gain* diatas 0.8 sangat memengaruhi hasil klasifikasi dibandingkan dengan beberapa fitur yang memiliki nilai *information gain* dibawah 0.8.

### 3. Skenario Pengujian dengan Nilai Alpha dan *Learning Rate*



**Gambar 7.** Skenario Pengujian dengan Nilai Alpha dan *Learning Rate*

Pada Gambar 7(a) terlihat bahwa pengujian terhadap nilai alpha pada fungsi aktivasi sigmoid pada data multilabel dengan nilai alpha sebesar -0.1 mendapatkan hasil *hamming loss* yang lebih baik dibandingkan nilai alpha sebesar -0.01 dengan nilai hasil *hamming loss* sebesar 0.055597484 atau sebesar 94.44% data berhasil diklasifikasikan dengan benar. Ini karena nilai alpha  $\leq 1$  akan mengakibatkan grafik fungsi sigmoid yang lebih landai sehingga perubahan nilainya lebih kecil untuk nilai alpha yang kecil.

Sedangkan untuk data *single label* pada Gambar 7(b) menunjukkan bahwa nilai *learning rate* sebesar 0.1 pada proses *training classifier* mendapatkan hasil F1-Score lebih baik dibandingkan nilai *learning rate* sebesar 0.001 dengan hasil F1-Score sebesar 65.71428571% data berhasil diklasifikasikan dengan benar. Ini dikarenakan nilai *learning rate* yang sangat kecil akan mengakibatkan proses *learning* yang semakin lambat sehingga ketika proses *learning* tidak akan dapat mencapai batas MSE standar ketika dibatasi dengan nilai epoch tertentu.

## 5. Kesimpulan

Dari hasil beberapa skenario pengujian yang telah dilakukan pada penelitian ini, dapat disimpulkan bahwa pengaruh nilai ambang batas *threshold* pada *information gain* berbeda untuk data multilabel dan data *single label*. Pada data multilabel, fitur yang memengaruhi setiap label tidak bisa menghasilkan hasil performansi yang baik pada *classifier* jika hanya mengambil fitur dengan nilai *information gain* diatas 0.8. Sedangkan pada data *single label*, hasil performansi yang dihasilkan lebih baik dengan mengambil beberapa fitur saja yang benar-benar memiliki nilai *information gain* tinggi atau diatas 0.8. Pengaruh penggunaan *information gain* bekerja efektif pada data *single label*.

Adapun untuk penggunaan proses *stemming*, hasil performansi terbaik didapatkan pada data *single label* dan tidak berlaku untuk data multilabel. Hal ini dikarenakan pada data multilabel, proses *stemming* akan menghilangkan ciri khas dari fitur untuk setiap label data. Sedangkan untuk data *single label*, generalisasi kata dengan proses *stemming* dapat berpengaruh terhadap hasil performansi sistem yang lebih baik.

Beberapa saran yang dapat diterapkan untuk pengembangan penelitian selanjutnya adalah seperti lebih memperhatikan label data khususnya pada data *single label* yang saat ini masih banyak terdapat label yang kurang sesuai dengan datanya. Selanjutnya adalah mencoba untuk menganalisa makna kata pada data multilabel agar hasil yang didapatkan bisa lebih baik.

## Daftar Pustaka

- [1] K. A. Aldhlan, A. M. Zeki, A. M. Zeki, and H. A. Alreshidi. Novel mechanism to improve hadith classifier performance. In *Advanced Computer Science Applications and Technologies (ACSAT), 2012 International Conference on*, pages 512–517. IEEE, 2012.
- [2] E. R. R. Jasin. Klasifikasi anjuran, larangan dan informasi pada hadis sahih al-bukhari berdasarkan model unigram menggunakan artificial neural network (ann). 2017.
- [3] A. Kusumaningrum. Klasifikasi informasi, anjuran dan larangan pada hadits shahih bukhari menggunakan metode support vector machine. 2017.
- [4] M. Naji Al-Kabi, G. Kanaan, R. Al-Shalabi, S. I. Al-Sinjilawi, and R. S. Al-Mustafa. Al-hadith text classifier. *Journal of Applied Sciences*, 5:584–587, 2005.
- [5] A. I. Pratiwi et al. On the feature selection and classification based on information gain for document sentiment analysis. *Applied Computational Intelligence and Soft Computing*, 2018, 2018.
- [6] A. I. Suyanto and A. I. Msc. Searching, reasoning, planning, learning. *Informatika Bandung*, pages 184–186, 2014.
- [7] 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 4th International Conference on*, volume 3, pages 1706–1710. IEEE, 2011.
- [8] M. Zareapoor and K. Seeja. Feature extraction or feature selection for text classification: A case study on phishing email detection. *International Journal of Information Engineering and Electronic Business*, 7(2):60, 2015.
- [9] M.-L. Zhang and Z.-H. Zhou. Multilabel neural networks with applications to functional genomics and text categorization. *IEEE transactions on Knowledge and Data Engineering*, 18(10):1338–1351, 2006.

## Lampiran

Lampiran dapat berupa detil data dan contoh lebih lengkapnya, data-data pendukung, detail hasil pengujian, analisis hasil pengujian, detail hasil survey, surat pernyataan dari tempat studi kasus, screenshot tampilan sistem, hasil kuesioner dan lain-lain.