PENGUKURAN SIMILARITY TEMA PADA JUZ 30 AL QUR'AN MENGGUNAKAN TEKS KLASIFIKASI

Endang Supriyati
Fakultas Teknik, Program Studi Teknik Informatika
Universitas Muria Kudus
Email: endang.supriyati@umk.ac.id

Mohammad Iqbal
Fakultas Teknik, Program Studi Teknik Elektro
Email: mohammad.iqbal@umk.ac.id

ABSTRAK

Klasifikasi teks adalah proses mengkelompokan teks ke satu kelompok atau lebih dari daftar yang telah ditentukan sebelumnya. Statistik ayat-ayat Alquran dipelajari dan dijelaskan dalam literatur, seperti jumlah surah dan ayat, ukuran ayat menurut kata-kata, ukuran ayat dengan huruf dan hubungan ayat satu dengan lainnya. Ini memberi wawasan yang komprehensif tentang struktur ayat. Penelitian ini akan membahas kedekatan tema ayat-ayat pada juz 30 yang berkaitan dengan kiamat, hisab, surga dan neraka. Langkah-langkah penelitian yaitu (1) pengumpulan data (2) *preprocessing*, (3) klasifikasi. Pengumpulan data dilakukan dengan me-record keseluruhan ayat pada juz 30 dilanjutkan dengan mendefiniskan tema tiap ayat yang berkaitan dengan kiamat, hisab, surga dan neraka. Preposesing mempunyai tahapan *tokenizing*, *filtering* dan *stemming*. Kasifikasi dilakukan dengan menggunakan beberapa algoritma seperti Decision Tree, Support Vector Machine (SVM) dan Naïve Bayes (NB). Dari hasil klasifikasi tersebut diperoleh hasil terbaik dengan klasifikasi decision tree akurasi 74,34%, SVM dengan akurasi 75,84%, dan Naïve Bayes dengan akurasi 66,29%.

Kata kunci: data mining, teks klasifikasi, ayat Al Qur'an, similarity, algoritma klasifikasi.

ABSTRACT

Text classification is the process of grouping text into one or more groups of previously selected lists. The statistics of the Qur'anic verses studied and explained in the literature, such as the number of suras and verses, the size of the verse according to the words, the size of the verse with the letters and the relation of verse one. It provides a comprehensive insight into the structure of the verse. This study will discuss the proximity of the verse themes in juz 30 that relate to the apocalypse, reckoning, heaven and hell. The research steps are (1) completion data (2) preprosessing, (3) coverage. The data collection is done by recording the verses on juz 30 times with themes appropriate to the apocalypse, reckoning, heaven and hell. Preposesing give tokenizing stage, filtering and stemming. The classification is done using several algorithms such as Decision Tree, Support Vector Machine (SVM) and Naïve Bayes (NB). From the result of the difference with the decision result. 74,34%, SVM with acceptance 75,84%, and Naïve Bayes with acceptance 66,29%.

Keywords: *text mining, classification, Al Qur'an, text classification*.

1. PENDAHULUAN

Secara alami, manusia menggunakan kata-kata dalam suatu urutan atau struktur. Kata-kata dalam kalimat memiliki semantik dan struktur sintaksis. Natural Language Processing (NLP) merupakan cabang ilmu *AI* (*Artificial Intelligence*) yang berfokus pada pengolahan bahasa natural. Pemrosesan Bahasa Alami atau *Natural Language Processing (NLP)* adalah komponen penting dalam text mining, yang berfokus pada pengolahan bahasa alami (bahasa antar manusia). Bahasa yang diterima oleh komputer butuh untuk diproses dengan baik oleh komputer. Beberapa penelitian di bidang NLP [1] adalah (1) *Question Answering Systems (QAS)*, pada sistem ini user menginputkan pertanyaan dalam teks bahasa natural bukan berbasis keyword. (2) *Summarization*, pembuatan ringkasan dokumen, (3) Machine Translation, mesin penerjemah, https://translate.google.co.id/ sebagai contoh, (4) *Speech Recognition*, mesin pengenalan bahasa, mengartikan pembicaraan siapa saja, (5) *Document classification*, mengelompokan dokumen sesuai dengan tema, misalnya aplikasi spam filtering, klasifikasi artikel berita,

review film, review musik dan sebagainya. Selain itu ada juga penelitian di bidang NLP yang lain seperti *information retrieval*, *information extraction* dan sebagainya. *Information Retrieval* (IR) digunakan untuk menemukan dokumen atau menemukan kembali informasi yang tersimpan dari berbagai sumber yang revelan dengan kebutuhan *user*. *Information extraction* tujuannya adalah untuk mengekstrak secara otomatis informasi terstruktur, data yang sudah terdefinisi dengan baik secara semantik dan secara kontekstual yang sudah terkelompok dari domain tertentu, dengan menggunakan berbagai dokumen tak-terstruktur yang bisa terbaca oleh mesin.[2]

Mengevaluasi kemiripan dalam dokumen banyak digunakan untuk aplikasi yang berkaitan dengan pencarian informasi, pemrosesan bahasa alami, dll. Contoh aplikasi di mana ada kebutuhan untuk mengurutkan, mengklasifikasi atau mengklasifikasikan dokumen berdasarkan jumlah atau tingkat kesamaan meliputi: Pengindeksan database, web, dan mesin telusur, mengelompokkan clustering otomatis dan klasifikasi, dan sebagainya. Evaluasi kemiripan dokumen banyak digunakan untuk pencarian informasi, pemprosesan bahasa alami dan sebagainya. Aplikasi kemiripan dokumen banyak digunakan untuk mengurutkan, klasifikasi atau klasifikasi dokumen. Kemiripan ini berdasarkan jumlah atau tingkat kesamaan. Data mining mempunyai tujuan untuk menemukan pola data, demikian pula text mining bertujuan menemukan pola data. Mayoritas text tidak terstruktur, tidak berformat dan relatif sulit untuk dibandingkan/dicocokkan dengan data yang di simpan dalam database[3]. Bahasa alami adalah sumber infromasi utama tentang penggunaan bahasa. Bahasa alami mewakili bank linguistik/bahasa yang besar yang dapat digunakan untuk menemukan tren, pola atau fenomena linguistik lain yang kemudian digunakan untuk pemprosesan bahasa. Sebagai contoh, *corpora* bahasa dapat mendukung studi terperinci tentang bagaimana kata-kata tertentu digunakan, dengan memberikan contoh-contoh yang luas tentang kalimat bahasa alami dalam konten. Informasi tentang frekuensi kata, korelasi, kolokasi dll dapat diturunkan dari *corpora* dan digunakan untuk membangun model languange statistik, untuk kata disambiguasi atau pengenalan ucapan. [4]

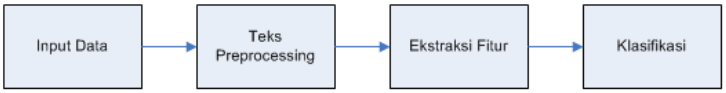
Islam adalah agama yang didasarkan pada dua sumber. Sumber pertama adalah Alquran yang mewakili kata demi kata firman Allah dan wahyu terakhir kepada umat manusia. Oleh karena itu dianggap sebagai sumber sastra yang sangat terhormat. Sumber kedua yang Islam dasarkan adalah Sunnah (ucapan Nabi Muhammad SAW). Sunnah mencakup interpretasi tentang Quran, ajaran dan cara hidup nabi. Alquran sebagai mukjizat Nabi Muhammad SAW berbeda dengan mukjizat pendahulunya (Nabi Musa dan Nabi Isa). Jadi ini adalah keajaiban abadi karena ia mencirikan kesempurnaan linguistik dan tidak tertandingi, nubuat yang benar, dan validasi penemuan ilmiah terkini. Al Quran aslinya ditulis dalam bahasa Arab tanpa menggunakan tanda diakritik. Ortografi Arab sedang dikembangkan pada saat pewahyuan Alquran. Evolusi alfabet Arab dimulai pada abad ke-7 sampai berakhir pada abad ke-8 oleh Al-Khalil ibn Ahmad al-Farahidi. Oleh karena itu, tulisan-tulisan Arab pada saat pewahyuan Quran tidak terbuka dan terbebas dari hamzas dan tanda diakritik. [5]

Dalam Al Quran memiliki apa yang disebut kesatuan subjek. Alquran dibagi menjadi 114 bab (Surah) dan setiap bab terdiri dari sejumlah ayat (ayat). Penelitian ini bertujuan untuk mengklasifikasikan setiap ayat ke subjek yang telah ditentukan, karena Quran sebagai kitab tidak diklasifikasikan pada subjek. Literatur tidak memiliki kajian tentang klasifikasi topikal otomatis ayat Quran. Meski menghadapi tantangan bahasa Arab, ada sejumlah studi tentang klasifikasi teks topik dalam Al Quran.

Penelitian ini bertujuan untuk mengidentifikasi algoritma klasifikasi terbaik untuk mengklasifikasikan ungkapan teks (ayat). Oleh karena itu algoritma klasifikasi (Support Vector Machine (SVM) , Naïve Bayes (NB) dan Decision Tree) diuji. Hasil awal menunjukkan bahwa Support Vector Machine (SVM) adalah yang terbaik.

2. METODE PENELITIAN

Metode penelitian adalah suatu prosedur, cara, atau teknik tertentu dalam memperoleh sesuatu, gambar 1 menujukkan langkah-langkah yang harus dilakukan untuk mendapatkan klasifikasi teks.



Gambar 1. Metode Penelitian Teks Klasifikasi

2.1 Data

Data yang digunakan dalam penelitian ini adalah juz ke-30 Al Quran. Juz ini banyak menyampaikan ayat-ayat yang berkaitan dengan hari akhir. Di juz 30 banyak surat pendek yang merupakan surat golongan Makkiyah. Juz 30 terdiri dari 37 surat dan 564 ayat. [6] Dalam penelitian ini

yang digunakan adalah surat An Naba sampai surat Al Kautsar sehingga jumlah ayat yang dianalisis sejumlah 534 ayat.

2.2 Teks Preprocessing Dan Ekstraksi Fitur

Preprocessing adalah merubah teks menjadi term index dengan tujuan menghasilkan sebuah set term index yang bisa mewakili dokumen. Dalam klasifikasi fase pendahuluan sangat penting yaitu preprocessing teks dan ekstraksi ciri. Dalam proses ini setiap teks diubah dalam bentuk vektor. Setiap dokumen diwakili oleh frekuensi *term* (tema). Langkah-langkah dalam fase ini adalah[7] :

2.1.1 Word Parsing And Tokenization

Yaitu mengubah dokumen menjadi kumpulan term dengan cara menghapus semua karakter dalam tanda baca yang terdapat pada dokumen dan mengubah kumpulan term menjadi *lowercase*. Parsing merupakan proses memecah isi dokumen menjadi unit-unit kecil yang akan menjadi penciri. Misalnya kalimat dengan 100 kata, dipecah menjadi 100 data. Unit terkecil ini yang disebut sebagai token, bagian dasar dalam parsing dari dokumen teks disebut *tokenizer*. Parsing menghasilkan daftar istilah(term) dan informasi tambahan seperti frekuensi yang akan digunakan untuk proses selanjutnya. Tokenisasi adalah proses untuk mengubah kalimat, paragraf, dokumen menjadi teks/token-token/bagian-bagian tertentu. Spasi dan tanda baca digunakan sebagai acuan pemisah antar token. Contoh Tokenisasi : "Aku suka makan bakso pedas di warung bersama teman", menghasilkan 9 token yaitu "aku", "suka", "makan", "bakso", "pedas", "di" , "warung", "bersama", "teman". Token-token ini yang akan di proses untuk analisis lebih lanjut.

2.1.2 Stop-Words Removal

Langkah ini sering disebut filtering. Filtering adalah tahap pemilihan kata-kata penting dari hasil token, yang akan digunakan untuk mewakili dokumen. *Stop-words removal* ini membuang kata-kata yang tidak penting misalnya kata "adalah", "agar", "supaya" dan lainnya. Setiap bahasa mempunyai daftar stop-words removal. Pada tahap ini akan dipunyai database kata-kata yang deskriptif(penting)".

Contoh kalimat : "Aku suka makan bakso pedas di warung bersama teman"
Tokenisasi : "aku", "suka", "makan", "bakso", "pedas", "di" , "warung", "bersama", "teman"
Stop words removal : "aku", "suka", "makan", "bakso", "pedas", "warung", "teman"

2.1.3 Stemming And Lemmatization

Setiap bahasa mempunyai algoritma stemming dan lemmatization. Stemming adalah proses mencari kata dasar dari sebuah kata imbuhan, dengan menghilangkan imbuhan-imbuhan pada kata dalam dokumen. Dalam penelitian ini, proses stemming menggunakan algoritma algoritma yang berbasis Nazief dan Adriani [8]. Contoh stemming : membetulkan -→ betul, berpegangan → pegang. Resiko dari proses stemming adalah hilangnya informasi dari kata yang di-stem. Stemming menyebabkann menurunnya akurasi atau presisi. Keuntungannya adalah bisa meningkatkan kemampuan untuk melakukan *recall*.

Lemmatization adalah sebuah proses untuk menemukan bentuk dasar dari sebuah kata. Lemma adalah bentuk dasar dari sebuah kata yang memiliki arti tertentu berdasar pada kamus. Perbedaan antara *stemming* dan *lemmatization* adalah :

Stemming : memotong akhir kata, dan sering juga membuang imbuhan melalui proses heuristic.
Lemmatization : Serupa stemming, hanya lebih baik hasilnya, karena Memperhatikan kamus dan analisis morfologi dan Menghasilkan kata dasar (lemma)

2.1.4 Term Selection/Feature Extraction

Term yang ditetapkan pada fase sebelumnya masih harus di filter, untuk menhgapus term yang memiliki kemampuan prediksi yang buruk (untuk kelas dokumen)atau sangat berkorelasi dengan persyaratan lain. Tugas ekstraksi fitur juga mengarah pada klasifikasi yang lebih sederhana dan lebih efisien.

Model default pencarian teks (text retrieval) adalah mengubah kata menjadi vektor boolean, artinya setiap dokumen direpresentasikan dengan vektor boolean n-dimensi, dimana n adalah ukuran kosakata, dan setiap nilai memodelkan keberadaan atau tidak adanya istilah kosakata dalam dokumen. Tetapi model yang sering digunakan adalah model berbasis frekuensi seperti model bobot TF-IDF, yang juga

digunakan dalam penenlitian ini. Dalam Al Quran, sebuah ayat bisa Sebuah ayat mencakup kata-kata dengan atau tanpa frekuensi.[9] Adapun rumus bobot term adalah :

Term Weight = wi = tfi * log (D/dfi) (1)

Dimana :

tfi mewakili istilah frekuensi atau berapa kali sebauah term *(i)* muncul dalam sebauah ayat. ***dfi*** = ayat frekuensi atau jumlah ayat yang mengandung term *(i)*. ***D*** = jumlah ayat dalam dokumen korpus.

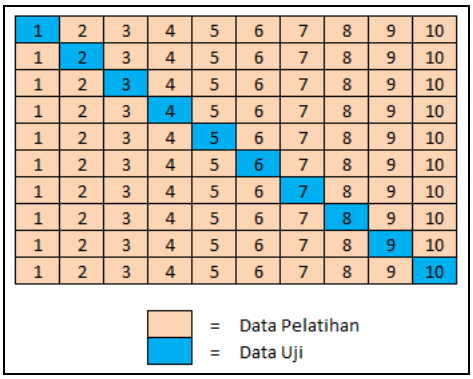
2.3 Klasifikasi

Klasifikasi adalah pembagian sesuatu menurut kelas-kelas. Proses pembagian/pengemlompokan berdasarkan ciri-ciri persamaan dan perbedaan. Dalam penelitian ini akan dilakukan perbandinga beberapa klasifikasi [10] yaitu SVM, NaiveBayes dan Decision Tree. Pada tahap ini dilakukan uji coba dengan langkah sebagai berikut :

a. Menyiapkan fitur-fitur dari hasil presprocessing ke dalam dataset.
b. Untuk setiap dataset, dillakukan pengujian dengan *k-fold cross-validation*, kemudian lakukan klasifikasi dengan SVM, Naive Bayes dan Decision Tree.

2.4 Akurasi K-Fold Cross Validation

K-fold cross validation [11] adalah salah satu teknik untuk mengestimasi tingkat kesalahan. Cara kerja *k-fold cross validation* adalah mengelompokan data latih dan data uji, kemudian dilakukan proses pengujian sebanyak k kali. Contoh dalam 10 *fold cross validation* (seperti gambar 2), data dibagi menjadi 10 fold berukuran sama, sehingga ada 10 subset data untuk mengevaluasi kinerja algoritma. Dari 10 subset data tersebut, 9 *fold* akan digunakan sebagai pelatihan dan 1 *fold* untuk pengujian.



Gambar 2. 10 Fold Cross Validation

Pada pengujian *k-fold cross valiation*, seluruh data secara acak dibagi menjadi K buah subset $\boldsymbol{B_k}$ dengan ukuran yang sama dimana $\boldsymbol{B_k}$ merupakan himpunan bagian dari $\{\boldsymbol{1}, \dots, \boldsymbol{n}\}$ sehingga $\cup_{\boldsymbol{k=1}}^{\boldsymbol{K}} \boldsymbol{B_k} = \{\boldsymbol{1}, \dots, \boldsymbol{n}\}$ dan $\boldsymbol{B_j} \cap \boldsymbol{B_k} = \emptyset(\boldsymbol{j} \neq \boldsymbol{k})$. Kemudian dilakukan iterasi sebanyak K kali. Pada iterasi kw k, subset $\boldsymbol{B_k}$ menjadi test set, sedangkan ubset yang lain menjadi training set. Setelah itu rata-rata *error* dihitung menggunakan daridari K buah iterasi.

3. HASIL DAN PEMBAHASAN

3.1 Data Yang Diolah

Juz 30 terdiri dari 37 surat dan 564 ayat, ketika diolah dengan *word cloud* akan menjadi seperti gambar 2. Gambar 2 menunjukan 50 kata dengan frekuensi yang paling tinggi. Dengan banyaknnya kata yang tidak mendeskripsikan term untuk klasifikasi maka harus dilakukan *preprocessing* untuk mendapatkan term-term yang mampu untuk dilakukan analisa lebih lanjut. Kata yang paling banyak

muncul pada data adalah yang sebanyak 71 kali, dan sebanyak 71. Pada gambar 3, semakin besar ukuran font kata, maka semakin sering muncul.

Gambar 3. Word Cloud Untuk 50 Kata Dengan Frekuensi Tertinggi

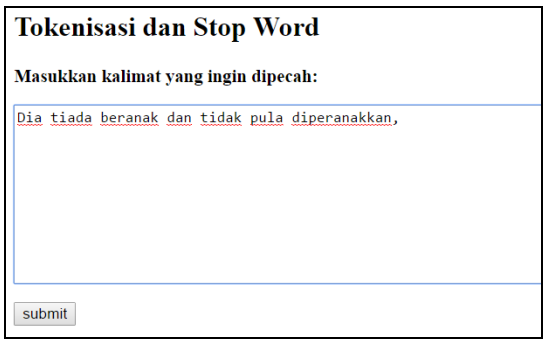
3.2 Alat Yang Digunakan

Alat yang digunakan untuk mengembangkan penelitian ini adalah software PHP untuk text processing, dan tools WEKA 3.8 untuk melakukan klasifikasi.

3.3 Prepocessing

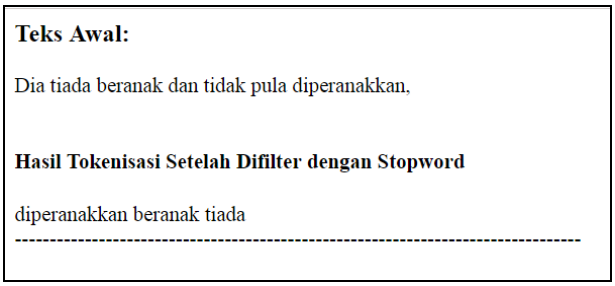
Tahap ini akan dilakukan *preprocessing* sebelum memasuki tahap klasifikasi.

a. Langkah pertama dilakukan proses tokensisasi dan *stop words removal*. Pada gambar 4, ditampilkan form yang harus dimasukkan kalimat yang akan dilakukan tokenisasi dan *stop words removal*. Dalam contoh ini dimasukkan ayat ke-3 dari surat Al Ikhlas.



Gambar 4. Form Input Kalimat Ayang Akan Di Pecah

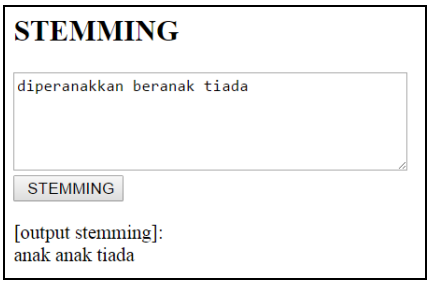
Gambar 5 merupakan hasil dari proses tokenisasi kemudian dilanjutkan dengan stopwords. Awal teks adalah " Dia tiada beranak dan tidak pula diperanakan". Hasil dari proses adalah diperanakkan beranak tiada. Hasil ini masih harus dilanjutkan dengan proses *stemming*.



Gambar 5. Hasil Tokenisasi Dan Stop Words Removal

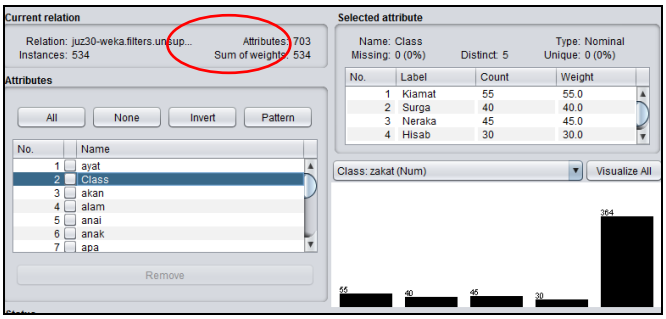
b. Langkah kedua dilakukan proses *stemming*. Teks yang dihasilkan dari proses pertama dilanjutkan dengan proses *stemming*. Hasil yang didapatkan adalah teks kata dasar. Proses

stemming membuang imbuhan pada kata dasar, apakah itu awalan maupun akhiran. Seperti yang terlihat pada gambar 6.



Gambar 6. Hasil S*temming*

c. Ekstraksi Fitur
Proses no.1 dan no.2 dilakukan dengan pemprogram dengan tools PHP, sedangkan untuk mencari ekstraksi *fitur/term* menggunakan tools WEKA didapat 703 *term*, seperti yang terlihat pada gambar 7.



Gambar 7. Lingkaran Merah Menunjukkan Jumlah *Attribute*

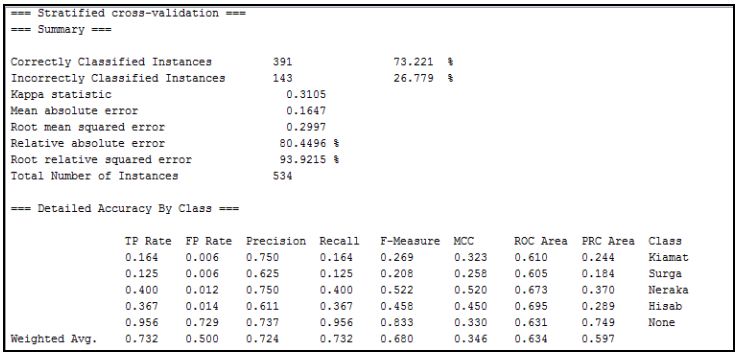
3.4 Klasifikasi

3.4.1 Algoritma Decision Tree

Decision tree adalah mesin prediksi menentukan nilai target (variabel dependen) dari sampel baru berdasarkan berbagai nilai atribut dari data yang ada. Konsep *entropy* digunakan untuk penentuan pada atribut mana sebuah pohon akan terbagi (*split*). Semakin tinggi *entropy* sebuah sampel, semakin tidak murni sampel tersebut. Rumus yang digunakan untuk menghitung *entropy* sampel S adalah sebagai berikut :

$$Entropy\ (S) = -p_1 log_2 p_1 - p_2 log_2 p_2 \quad (2)$$

Dimana $p_1, p_2, \dots p_n$ masing-masing menyatakan proposi kelas 1, kelas 2 …kelas n dalam dalam output.



```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         391               73.221  %
Incorrectly Classified Instances       143               26.779  %
Kappa statistic                          0.3105
Mean absolute error                      0.1647
Root mean squared error                  0.2997
Relative absolute error                 80.4496 %
Root relative squared error             93.9215 %
Total Number of Instances              534

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure  MCC      ROC Area  PRC Area  Class
               0.164     0.006     0.750      0.164     0.269      0.323     0.610     0.244     Kiamat
               0.125     0.006     0.625      0.125     0.208      0.258     0.605     0.184     Surga
               0.400     0.012     0.750      0.400     0.522      0.520     0.673     0.370     Neraka
               0.367     0.014     0.611      0.367     0.458      0.450     0.695     0.289     Hisab
               0.956     0.729     0.737      0.956     0.833      0.330     0.631     0.749     None
Weighted Avg.  0.732     0.500     0.724      0.732     0.680      0.346     0.634     0.597
```

Gambar 8. Hasil Klasifikasi Dengan Decision Tree , *10-Cross Validation*

```
=== Confusion Matrix ===

   a   b   c   d   e   <-- classified as
   9   1   0   0  45 |   a = Kiamat
   0   5   0   2  33 |   b = Surga
   0   0  18   0  27 |   c = Neraka
   0   0   0  11  19 |   d = Hisab
   3   2   6   5 348 |   e = None
```

Gambar 9. Confusion Matrix Untuk Decision Tree, *10-Cross Validation*

Dari hasil decision tree (gambar 8.) didapatkan hasil akurasi 73%, kesalahan klasifikasi sebanyak 26%, dari hasil pengujian 10-*fold cross validation*. Gambar 9, menampilkan *confusion matrix* dari klasifikasi decision tree. Uji coba dilakukan dengan beberapa nilai k, seperti pada tabel 1. Hasil terbaik pada *14-fold cross validation* dengan akurasi 74,34%.

Tabel 1. Hasil akurasi *k-fold validation*

k-Fold Cross	waktu (detik)	Decision Tree (%)
2	5.49	71.16
5	4.31	73.22
8	4.49	73.22
10	4.25	73.22
12	5.83	73.22
14	4.89	74.34
16	4.41	72.65
18	5.05	72.65

3.4.2 Naïve Bayes

Klasifikasi Naïve Bayes didasarkan pada probabilitas bersyarat Bayes (Bayes Rule). Ini memanfaatkan semua atribut yang terdapat dalam data, dan menganalisisnya secara terpisah seolah-olah sama pentingnya dan tidak tergantung satu sama lain.

Bayes Rule rumusnya :

$$P(H|E) = \frac{P(E|H) * P(H)}{P(E)} \quad (3)$$

Dimana :
E = Data dengan *class* yang belum diketahui
H : Hipotesis data merupakan suatu *class* spesifik
P(H|E) :Probabilitas hipotesis *H* berdasar kondisi *E* (posteriori probabilitas)
P(H) : Probabilitas hipotesis *H* (prior probabilitas)
P(E|H) :Probabilitas *E* berdasarkan kondisi pada hipotesis *H*
P(E) : Probabilitas *E*

Rumus Naive Bayes diatas disesuaika sebagai berikut :

$$P(C|F1 \ldots . Fn) = \frac{P(C)\,P(F1 \ldots Fn|C)}{P(F1 \ldots Fn)} \quad (4)$$

Dimana :
C : kelas
F1 ... *Fn* : kriteria yang dibutuhkan untuk melakukan klasifikasi

```
Correctly Classified Instances         354               66.2921 %
Incorrectly Classified Instances       180               33.7079 %
Kappa statistic                          0.3596
Mean absolute error                      0.1622
Root mean squared error                  0.3217
Relative absolute error                 79.2027 %
Root relative squared error            100.8252 %
Total Number of Instances              534

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               0.491    0.067    0.458      0.491    0.474      0.411    0.846     0.439     Kiamat
               0.400    0.057    0.364      0.400    0.381      0.329    0.805     0.337     Surga
               0.489    0.033    0.579      0.489    0.530      0.493    0.836     0.522     Neraka
               0.567    0.054    0.386      0.567    0.459      0.430    0.849     0.429     Hisab
               0.747    0.453    0.779      0.747    0.763      0.288    0.726     0.838     None
Weighted Avg.  0.663    0.326    0.676      0.663    0.668      0.329    0.761     0.710
```

Gambar 10. Hasil Klasifikasi Dengan Naïve Bayes, 10 Cross Validation

```
=== Confusion Matrix ===

  a   b   c   d   e   <-- classified as
 27   2   0   0  26 |   a = Kiamat
  1  16   1   2  20 |   b = Surga
  0   2  22   0  21 |   c = Neraka
  1   2   0  17  10 |   d = Hisab
 30  22  15  25 272 |   e = None
```

Gambar 11. Confusion Matrix Untuk Decision Tree, 10-Cross Validation

Dari hasil Naïve Bayes (gambar 10.) didapatkan hasil akurasi 66%, kesalahan klasifikasi sebanyak 33%, dari hasil pengujian 10-*fold cross validation*. Gambar 11, menampilkan *confusion matrix* dari klasifikasi naïve bayes. Uji coba dilakukan dengan beberapa nilai k, seperti pada tabel 2. Hasil terbaik pada *10-fold cross validation* dengan akurasi 66.29%.

Tabel 2. Hasil Akurasi K-Fold Validation

k-Fold Cross	Waktu (detik)	Naïve Bayes (%)
2	0.44	63.85
5	0.13	66.29
8	0.14	64.60
10	0.19	66.29
12	0.17	65.35
14	0.14	66.29
16	0.14	64.23
18	0.14	66.29

3.4.3 SVM

Support Vector Machine (SVM) adalah sistem pembelajaran yang pengklasifikasiannya menggunakan ruang hipotesis berupa fungsi-fungsi linear dalam sebuah ruang fitur (feature space) berdimensi tinggi. Dalam konsep SVM berusaha menemukan fungsi pemisah (hyperplane) terbaik diantara fungsi yang tidak terbatas jumlahnya. Hyperplane pemisah terbaik antara kedua kelas dapat ditemukan dengan mengukur margin hyperplane tersebut dan mencari titik maksimalnya. Adapun data yang berada pada bidang pembatas disebut support vector. Secara matematika, konsep dasar SVM yaitu :

$$\min \frac{1}{2}|w|^2 \quad (5)$$

$$s.t\ y_i(x_i.w + b) - 1 \geq 0 \quad (6)$$

Dimana $(x_i.w + b) \geq 1$ untuk kelas 1, dan $(x_i.w + b) \leq -1$ untuk kelas 2, x_i adalah data set, y_i adalah output dari data x_i dan w, b adalah parameter yang dicari nilainya.

```
Correctly Classified Instances         405               75.8427 %
Incorrectly Classified Instances       129               24.1573 %
Kappa statistic                          0.452
Mean absolute error                      0.256
Root mean squared error                  0.3403
Relative absolute error                125.0027 %
Root relative squared error            106.6483 %
Total Number of Instances              534

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure  MCC      ROC Area  PRC Area  Class
               0.382     0.021     0.677      0.382     0.488      0.469     0.797     0.373     Kiamat
               0.425     0.018     0.654      0.425     0.515      0.498     0.792     0.378     Surga
               0.467     0.018     0.700      0.467     0.560      0.541     0.820     0.425     Neraka
               0.433     0.018     0.591      0.433     0.500      0.481     0.803     0.348     Hisab
               0.915     0.541     0.784      0.915     0.844      0.432     0.687     0.775     None
Weighted Avg.  0.758     0.375     0.745      0.758     0.740      0.453     0.724     0.651
```

Gambar 11. Hasil Klasifikasi Dengan SVM, 10-Cross Validation

```
=== Confusion Matrix ===

   a   b   c   d   e   <-- classified as
  21   0   0   0  34 |   a = Kiamat
   0  17   0   2  21 |   b = Surga
   0   0  21   0  24 |   c = Neraka
   2   2   0  13  13 |   d = Hisab
   8   7   9   7 333 |   e = None
```

Gambar 12. Confusion Matrix Untuk Decision Tree, 10-Cross Validation

Dari hasil SVM (gambar 11.) didapatkan hasil akurasi 75%, kesalahan klasifikasi sebanyak 24%, dari hasil pengujian 10-*fold cross validation*. Gambar 12, menampilkan *confusion matrix* dari klasifikasi SVM. Uji coba dilakukan dengan beberapa nilai k, seperti pada tabel 3. Hasil terbaik pada *10-fold cross validation* dengan akurasi 75.84%.

Tabel 3. Hasil Akurasi K-Fold Validation

k-Fold Cross	Waktu (detik)	SVM (%
2	4.19	70.59
5	1.03	75.09
8	0.80	73.78
10	0.94	75.84
12	1.42	74.71
14	0.91	75.65
16	0.97	73.4
18	0.70	75.84

4. KESIMPULAN

Tabel 4. Menampilkan hasil akurasi dari klasifikasi Naive Bayes, SVM dan Decision Tree. Akurasi tertinggi diperoleh dengan klasifikasi SVM sebanyak 75.84%. Penelitian ini masih sebatas mengklasifikasikan ayat-ayat yang berkaitan dengan tema akhir jaman. Sehingga perlu dikembangkan menjadi ontology berbasis klasifikasi teks.

Tabel 4. Perbandingan akurasi dari 3 classifier

k-Fold Cross	Akurasi		
	Naïve Bayes	SVM	Decision Tree
2	63.85	70.59	71.16
5	66.29	75.09	73.22
8	64.6	73.78	73.22
10	**66.29**	**75.84**	73.22
12	65.35	74.71	73.22
14	66.29	75.65	**74.34**
16	64.23	73.4	72.65
18	66.29	75.84	72.65

DAFTAR PUSTAKA

[1] Pustejovsky, J., Stubbs A. (2012). Natural Language Annotation for Machine Learning. Beijing: O'Reilly, ISBN13: 9781449306663.
[2] Russel, S. J., Norvig, P. (2010). Artificial Intelligence A Modern Approach. New Jersey: Pearson Education Inc, ISBN-13: 978-0136042594.
[3] Mohammed Naji Al-Kabi, Ghasan K, Riyad Al-Shalabi dkk,2005, "Statistical Classifier of the Holy Quran Verses (Fatiha and Yaseen Chapters", Journal of Applied Science 5 (3): 580-583, ISSN 1812-5654
[4] Wanjiku, N, 2003, "Semantic analysis of kiswahili words using the self organizing map", Nordic J. African Studies, pp 407-425.
[5] Mohammed N. Al-Kabi, Belal M. Abu Ata, Heider A. Wahsheh, Izzat M. Alsmadi, 2013, "A Topical Classification of Quranic Arabic Text", Taibah University International Conference on Advances in Information Technology for the Holy Quran and Its Sciences, December 22 – 25, 2013, Madinah, Saudi Arabia.
[6] http://quran.kemenag.go.id./
[7] Salton, Gerard (1983) Introduction to Modern Information Retrieval, McGraw Hil
[8] Nazief, B. A. A. & Adriani, M. (1996), Confixstripping: Approach to Stemming Algorithm for Bahasa Indonesia. Internal publication, Faculty of Computer Science, University of Indonesia, Depok, Jakarta.
[9] Mohammed Akour, Izzat Alsmadi, Iyad Alazzam, 2014, "MQVC: Measuring Quranic Verses Similarity and Sura Classification Using N-gram", *WSEAS Transactions on Computers, 13*, 485-491.
[10] Jin Huang , Jingjing Lu , Charles X. Ling, 2003, "Comparing Naive Bayes, Decision Trees, and SVM with AUC and Accuracy**",** The Third IEEE International Conference on Data Mining
[11] Juan Diego Rodrı´guez, Aritz Pe´rez, and Jose Antonio Lozano, 2010, "Sensitivity Analysis of k-Fold Cross Validation in Prediction Error Estimation", IEEE Transactions On Pattern Analysis And Machine Intelligence, Vol. 32, No. 3, March 2010.