TELEMATIKA, Vol. 13, No. 02, JULI, 2016, Pp. 123 – 133
ISSN 1829-667X

TEXT CLASSIFICATION USING NAIVE BAYES UPDATEABLE ALGORITHM IN SBMPTN TEST QUESTIONS

Ristu Saptono[(1)], Meiyanto Eko Sulistyo[(2)], Nur Shobriana Trihabsari[(3)]
[1,3)]Program Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam,
Universitas Sebelas Maret, Surakarta
[2)]Program Studi Teknik Elektro
Fakultas Teknik, Universitas Sebelas Maret, Surakarta
e-mail : ristu.saptono@staff.uns.ac.id[(1)], mekosulistyo@staff.uns.ac.id[(2)],
nanashobriana@student.uns.ac.id[(3)]

Abstract

Document classification is a growing interest in the research of text mining. Classification can be done based on the topics, languages, and so on. This study was conducted to determine how Naive Bayes Updateable performs in classifying the SBMPTN exam questions based on its theme. Increment model of one classification algorithm often used in text classification Naive Bayes classifier has the ability to learn from new data introduces with the system even after the classifier has been produced with the existing data. Naive Bayes Classifier classifies the exam questions based on the theme of the field of study by analyzing keywords that appear on the exam questions. One of feature selection method DF-Thresholding is implemented for improving the classification performance. Evaluation of the classification with Naive Bayes classifier algorithm produces 84,61% accuracy.

Keywords: classification, Naive Bayes Updateable, text mining, DF-Thresholding

Abstrak

Klasifikasi dokumen merupakan minat dalam penelitian text mining. Klasifikasi dapat dilakukan berdasarkan topik, bahasa, dan sebagainya. Penelitian ini dilakukan untuk mengetahui bagaimana Naive Bayes Updateable melakukan dalam mengklasifikasikan soal ujian SBMPTN berdasarkan tema. Model kenaikan satu algoritma klasifikasi sering digunakan dalam klasifikasi teks klasifikasi memiliki kemampuan untuk belajar dari data baru memperkenalkan dengan sistem bahkan setelah classifier telah diproduksi dengan data yang ada. Klasifikasi mengklasifikasikan pertanyaan ujian berdasarkan tema bidang studi dengan menganalisis kata kunci yang muncul di soal ujian. Salah satu metode seleksi fitur DF-Thresholding diimplementasikan untuk meningkatkan kinerja klasifikasi. Evaluasi klasifikasi dengan algoritma classifier Naive Bayes menghasilkan akurasi 84,61%.

Kata Kunci: klasifikasi, Naive Bayes Updateable, text mining, DF-Thresholding

1. PENDAHULUAN

Salah satu seleksi bersama untuk penerimaan mahasiswa baru disuatu PTN dilakukan dengan ujian tertulis secara nasional yang dikenal dengan istilah SBMPTN. Oleh Majelis Rektor Perguruan Tinggi Negeri Indonesia (MRPTNI), SBMPTN 2015 diselenggarakan secara nasional. Ujian tertulis menggunakan soal ujian yang dikembangkan sedemikian rupa sehingga memenuhi persyaratan reliabilitas dan validitas yang memadai. Soal SBMPTN dirancang untuk mengukur kemampuan dasar yang dapat memprediksi keberhasilan calon mahasiswa disemua program studi, yakni kemampuan penalaran tingkat tinggi (*higher order thinking*), yang meliputi potensi akademik, penguasaan bidang studi dasar, bidang sains dan teknologi (saintek) serta bidang sosial dan humaniora (soshum). Selain mengikuti ujian tertulis, peserta yang memilih program studi Ilmu Seni dan/atau Keolahragaan diwajibkan mengikuti ujian keterampilan (A. Hamzah, 2012). Materi yang diujikan diantaranya, Tes Kemampuan dan Potensi Akademik (TKPA), Tes Kemampuan Dasar Sains dan Teknologi (TKD Saintek), dan Tes Kemampuan Dasar Sosial dan Humaniora (TKD Soshum). TKD Saintek terdiri atas mata uji Matematika,

Biologi, Kimia, dan Fisika. TKD Soshum terdiri atas mata uji Sosiologi, Sejarah, Geografi, dan Ekonomi. Soal ujian SBMPTN itu sendiri memiliki standar tertentu.

Untuk soal ujian bidang studi, saintek, dan soshum, tiap soalnya memiliki tema. Tiap soal memiliki istilah unik yang hanya mengacu pada suatu tema tertentu. Oleh karena itu, setiap soal memiliki informasi yang dapat digunakan untuk mengidentifikasi termasuk ke dalam tema bidang studi apakah soal tersebut. Untuk membantu peserta ujian lolos seleksi ini perlu diadakan proses latihan ujian. Sehingga dibutuhkan sebuah sistem yang mampu memberikan berbagai soal-soal latihan ujian yang memiliki standar dan tema-tema yang sesuai dengan ujian SBMPTN. Selain itu, sistem yang dapat di-*update* setiap waktu dan mampu secara otomatis mengklasifikasikan soal ujian ke dalam tema-tema tertentu akan sangat mempermudah dalam otomatisasi penyusunan naskah soal latihan ujian ini.

Penelitian ini akan membahas pengklasifikasian otomatis soal ujian SBMPTN berdasarkan tema bidang studi. Tema-tema bidang studi yang digunakan diambil dari buku-buku bidang studi siswa SMA kelas X, XI, dan XII kurikulum 2006 yang diterbitkan oleh Pusat Perbukuan Departemen Pendidikan Nasional tahun 2009. Kasus klasifikasi soal ujian SBMPTN merupakan *single-labeled classification* karena untuk setiap soal akan memiliki satu tema. Salah satu kelompok klasifikasi berbasis *numeris* dengan pendekatan berbasis *probabilistic* yaitu *Naïve Bayes Classifier* (NBC) akan digunakan dalam penelitian ini. NBC memiliki beberapa kelebihan antara lain, sederhana, cepat dan memiliki akurasi yang tinggi. Metode NBC untuk klasifikasi atau kategorisasi teks menggunakan atribut kata yang muncul dalam suatu dokumen sebagai dasar klasifikasinya. Sebuah penelitian menunjukkan bahwa meskipun asumsi independensi antar kata dalam dokumen tidak sepenuhnya dapat dipenuhi, tetapi kinerja NBC dalam klasifikasi relatif sangat bagus [2]. Selain itu, NBC berkerja dengan baik untuk data numerik dan tekstual, mudah diimplementasikan dibandingkan dengan metode klasifikasi lain [3]. Metode NBC untuk klasifikasi teks telah dilakukan oleh beberapa peneliti. Wibisono di tahun 2005 meneliti metode NBC untuk kategorisasi berita menggunakan 291 dokumen training dan 291 dokumen testing mendapatkan hasil akurasi 86.9%, jika 70% dokumen training dan 30% dokumen testing akurasinya naik jadi 90,23%. Penelitian Wulandini dan Nugroho tahun 2009 membandingkan metode klasifikasi teks NBC dengan method Support Vector machine (SVM), C4.5 dan K-Nearest Neighbour (K-NN). Hasil penelitian ini menunjukkan akurasi masing-masing metode diurutkan dari yang terbaik adalah SVM akurasi 92%, NBC akurasi 90% C4.5 akurasi 77.5% dan yang terendah K-NN akurasi 50% (H. Mario Naga Mait, 2012). Karakteristik utama atau kesulitan dari klasifikasi teks adalah tingginya dimensi ruang atribut/ banyaknya *term*/ istilah/ kata yang ada dalam suatu dokumen. Metode *feature selection* otomatis dapat membantu menghilangkan non-*informative term*/ kata yang dianggap tidak penting berdasarkan frekuensi kemunculannya di dalam dokumen (I. Rish, 2001).

Data soal ujian SBMPTN setiap tahun akan terus bertambah variasinya. Setiap variasi soal ujian ini harus disimpan sebagai pengetahuan sistem dalam melakukan pengklasifikasian selanjutnya. Terdapat beberapa algoritma yang dapat menangani data *batch-based* secara bertahap ini, salah satunya adalah algoritma *Naive Bayes Updateable*. Algoitma ini merupakan *incremental version* dari *Naive Bayes Classifier* (J. Asian, dkk, 2003). Seperti hasil dari penelitian Ade dan Deshmukh, algoritma *Naive Bayes Updatable* baik diterapkan untuk data yang secara real time terus bertambah. Tehnik *incremental learning* semacam ini sangat pentng untuk menangani data yang cukup besar karena dua alasan, yang pertama karena tidak memungkinkan bahwa dapat mengumpulkan semua data sebelum proses *training*, dan yang kedua memodifikasi sistem dengan penambahan data *training* akan berdampak baik dalam membangun sistem (M. Adriani, dkk, 2007.

2. DASAR TEORI

2.1. *Stemming*

Stemming merupakan suatu proses dalam *information retrieval* yang mengubah suatu kata (*term*) dari bentuk kata berimbuhan menjadi kata dasar (*root word*) menggunakan aturan-aturan tertentu

sesuai dengan bahasa yang digunakan [8]. Dalam Bahasa Indonesia proses *stemming* sangat penting untuk diterapkan karena imbuhan (*affixes*) baik yang terdiri dari awalan (*prefixes*), sisipan (*infixes*), akhiran (*suffixes*) dan *confixes* (kombinasi dari awalan dan akhiran) pada kata turunan [9]. Untuk *stemming* bahasa Indonesia sendiri terdapat beberapa algoritma antara lain, *Arifin and Setiono's Algorithm*, *Vega's Algorithm*, dan *Ahmad*, *Yuso*, *and Sembok's Algorithm* [10].

2.2. Tokenizing

Tokenizing*/tokenisasi* merupakan suatu proses memotong teks input menjadi potongan – potongan kata / token dengan mengingat urutannya pada teks yang ditokenisasi dan pada saat yang sama membuang karakter tertentu, seperti tanda baca [11]. Hasil dari proses tokenisasi ini adalah berupa kata-kata yang terpisah. Contoh masukan dan keluaran dari proses tokenisasi dapat dilihat pada gambar 1.

Input:

Tumbuhan paku heterospor mempunyai karakter sebagai berikut, kecuali

Output:

tumbuhan	paku	heterospor	mempunyai	karakter
sebagai	berikut	kecuali		

Gambar 1. Proses Tokenisasi

2.3. Stopword Removal

Salah satu langkah penting dalam text mining *adalah menghilangkan* stopword*.* Stopword *adalah daftar kata yang umum digunakan, memiliki fungsi penting dalam sebuah teks namun tidak memiliki arti [12]* Stopword *terdiri dari kata sambung, kata depan, kata hubung, kata ganti orang, dan kata umum lain yang sering digunakan namun tidak memiliki arti penting.* Stopword *dalam suatu teks akan dihilangkan untuk mengurangi* noise term*. Sehingga tersisa* keyword*.*

2.4. Document Frequency Thresholding

Salah satu metode feature selection *adalah* Document Frequency Thresholding *untuk mengeliminasi* term *yang dianggap tidak penting agar mendapatkan kosakata yang diaggap mampu menjadi pembeda antar kelas.* Document frequency *(DF) merupakan jumlah dokumen yang di dalamnya terdapat suatu kata [5]. DF dihitung untuk setiap* term *yang unik dan menghapus* term *dari daftar atribut/ kosakata yang memiliki frekuensi kurang dari* threshold *yang telah ditentukan sebelumnya atau memiliki frekuensi lebih dari* threshold *yang telah ditentukan sebelumnya.* Term*/ kata/* keyword *yang memiliki frekuensi terlalu kecil atau jarang muncul dalam dokumen memiliki kemungkinan besar tidak memberikan informasi penting bagi dokumen terhadap suatu kelas. Begitu pula term yang memiliki frekuensi tinggi atau terlalu sering muncul dalam dokumen, term tersebut akan dianggap sebagai istilah yang biasa dipakai dalam kalimat sehingga tidak memberikan informasi penting bagi dokumen terhadap suatu kelas.*

2.5. Naive Bayes Classifier

Konsep klasifikasi oleh Naive Bayes atau yang biasa disebut Naive Bayes Classifier diasumsikan bahwa pengklasifikasiaannya berdasarkan efek dari suatu nilai atribut sebuah kelas sedangkan atribut yang diberikan adalah bebas dari atrbut-atribut lain atau bersifat independen. NBC merupakan metode klasifikasi yang berasal dari teorema Bayes. Ciri utama dari NBC adalah asumsi yang sangat kuat akan independensi dari masing-masing kondisi [13], dimana setiap atribut yang membangun setiap kelasnya bersifat saling lepas. Nilai yang dimiliki fitur-fitur NBC ini dapat berupa data statis atau berupa kategori. sehingga dalam pengerjaannya sudah didapatkan hasil yang pasti juga.

Jika data yang akan diklasifikasinya nilai atributnya bersifat kontinu atau nilainya terus-menerus berhubungan maka distribusi yang digunakan adalah distribusi Gaussian. Ciri khas dari distibusi

ini adalah menggunakan 2 parameter yaitu mean (μ) dan variansi (σ^2). Mean merupakan nilai rata-rata dari atribut yang memiliki data kontinu rumusnya ditunjukkan oleh persamaan 1. Parameter kedua yaitu variansi (σ^2) didapatkan dengan persamaan 2. Dari kedua parameter tersebut, kemudian terdapat suatu fungsi yang akan menghitung nilai dentitas. Persamaan 3 menyatakan fungsi yang akan menghitung nilai dentitas untuk mengekspresikan probailitas relatifnya [14]. Setelah nilai probabilitas relatif masing-masing atribut terhadap setiap kelasnya didapatkan, selanjutnya dilakukan perhitungan nilai likelihood dengan mengalikan seluruh kemungkinannya. Persamaan likelihood dapat dilihat pada persamaan 4. Kemudian normalisasi dengan persamaan 5 dilakukan untuk mendapatkan persentase nilai probabilitas setiap kelasnya [15].

$$mean(\mu) = \frac{\sum_{i=1}^{n} x_i}{n} \quad (1)$$

n : jumlah data dalam satu kelas
x_i : nilai atribut ke-*i*

$$\sigma^2 = \frac{1}{n-1}\sum_{i=n}^{n}(x_i - \mu)^2 \quad (2)$$

μ : mean
n : jumlah data dalam satu kelas
x_i : nilai atribut ke-*i*

$$\varphi_{\mu,\sigma}(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (3)$$

μ : *mean*
σ^2 : variansi
x : nilai atribut

$$P(x_i|y_j) = P(x_1|y_j) * P(x_2|y_j) * P(x_3|y_j) * \dots \quad (4)$$

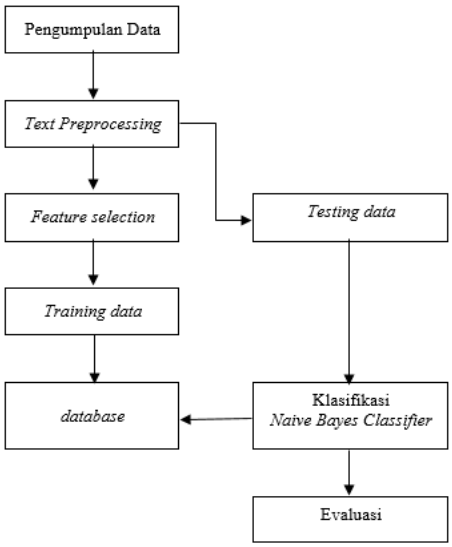
$$normal\,P(x_i|y_j)_1 = \frac{P(x_i|y_j)_1}{(P(x_i|y_j)_1 + P(x_i|y_j)_2 + \dots + P} \quad (5)$$

2.6. Naive Bayes Updateable

Naive Bayes Updateable *merupakan bentuk* incremental *dari* Naive Bayes Classifier *[6], model* Incremental classification *dapat melakukan* learning *menggunakan satu* instance *atau satu* batch *dalam satu waktu [7]. Proses* incremental learning *dalam kasus ini adalah data* training *yang secara bertahap bertambah/* increment*, karena setiap* batch *data* testing *yang masuk akan digunakan sebagai* knowledge *untuk klasifikasi berikutnya.*

3. METODOLOGI

Pelaksanaan penelitian ini melibatkan beberapa sistematika penelitian dari pengambilan data tekstual, *text preprocessing*, *training data*, klasifikasi data hingga evaluasi hasil. Gambar 2 menunjukkan diagram alir penelitian.



Gambar 2 Diagram alir metodologi penelitian

3.1. Pengumpulan Data

Data yang digunakan bersumber dari buku kumpulan soal-soal ujian SBMPTN, soal soal latihan ujian SBMPTN atau *tryout*, dan tema-temanya didapatkan dari buku bidang studi.

3.2. *Text Preprocessing*

Sebelum suatu data teks mengalami pengolahan data, terdapat beberapa langkah *preprocessing* untuk mendapatkan kata kunci. Serangkaian langkah yang masuk ke dalam *preprocessing* diataranya *stemming.* Pada satu soal berupa teks dilakukan *stemming* dengan algoritma Nazief & Adriani yang diambil dari *library* Sastrawi. Proses ini akan menghilangkan imbuhan pada kata sehingga tersisa kata dasar atau root word. Setelah didapatkan teks yang hanya tersusun dari kata dasar selanjutnya, dilakukan penghapusan kata-kata yang tidak diperlukan. Kata-kata yang tidak diperlukan ini biasa disebut kata sambung atau *stopword*, oleh sebab itu proses ini disebut *stopword removal*. Hasil dari proses ini adalah teks yang terdiri dari kata kunci. Kemudian dilakukan tokenisasi. Proses ini akan mentransformasikan bentuk teks / *string* menjadi token-token, satu kata satu token. Selain itu pada proses ini juga terjadi penghapusan tanda baca sehingga hanya menghasilkan token yang berisi kata kunci.
Proses *spelling check* tidak dilakukan dalam pada tahap *text peprocessing.* Akibat dari tidak adanya proses ini akan adalah kata dengan salah penulisan akan dianggap sebagai kata baru. Hal ini dibiarkan karena terlalu jarang terjadi kesalahan penulisan dalam teks naskah soal ujian SBMPTN.

3.3. Feature Selection

*Token-token kata kunci dihitung frekuensinya atau nilai TF (*term frequency*)-nya. Nilai TF digunakan sebagai bobot untuk setiap kata kunci terhadap suatu dokumen (dalam hal ini soal ujian). Kemudian dihitung pula nilai DF (*Document Frequency*). DF dari sebuah fitur menunjukan jumlah dokumen dimana sebuah kata kunci tersebut muncul. DF bersifat* class independent*. DF banyak digunakan dalam* dimensionality reduction *[16]. Biasanya, kata yang jarang muncul atau nilai DF-nya kecil akan memberikan informasi spesifik dan tidak mempengaruhi kinerja klasifikasi* secara global, sedangkan kata yang terlalu sering muncul atau nilai DF-nya besar menunjukkan bahwa kata tersebut merupakan kata yang biasa digunakan sehingga tidak mampu memberikan informasi spesifik. Tetapi kata yang terlalu jarang muncul juga dianggap tidak penting. Oleh sebab itu kata yang terlalu sering muncul dan terlalu jarang muncul dihilangkan dari data set. Tehnik eliminasi kata dengan nilai DF dan batas nilai tertentu ini merupakan tehnik feature reduction yang paling sederhana.
Dengan mereduksi fitur yang digunakan dalam proses klasifikasi, akan meningkatkan kinerja klasifikasi. Terdapat 3 syarat suatu data input dinyatakan sangat membantu dalam proses klasifikasi, yaitu [17]:

1. *Concentration degree*: dalam suatu data set dengan berbagai kelas atau kategori, jika fitur kata muncul di satu atau sedikit kelas tapi tidak muncul di kelas lain, fitur kata tersebut memberikan informasi spesifik yang kuat dan sangat membantu dalam proses klasifikasi.
2. *Disperse degree*: jika suatu fitur kata muncul di satu kelas, fitur ini memiliki korelasi yang kuat dengan kelas tersebut. Sehingga, fitur sangat membantu proses klasifikasi apabila persebarannya berpencar untuk berbagai kelas.
3. *Contribusion degree*: jika sebuah fitur kata memiliki korelasi yang tinggi dengan satu kelas, maka fitur tersebut memiliki informasi yang penting dan sangat membantu dalam proses klasifikasi.

Dari 3 prinsip di atas dan konsep DF *Threshold* yang sebelumnya telah dibahas, dapat dirumuskan [17]:

1. *Concentration degree*: Menggunakan formula 6 untuk menunjukkan rasio seberapa tinggi konsentasi suatu fitur dalam satu kelas.

$$\frac{DF(t,c_i)}{(1+\sum_{j=1|j\neq i}^{n} DF(t,c_j))} \quad (6)$$

2. *Disperse degree*: Terdapat *m* kelas yang berbeda, C= $\{c_1, c_2, \ldots, c_m\}$. Menggunakan formula $DF(t,c_i)/N(c_i)$. Dengan $N(c_i)$ adalah jumlah dokumen di kelas c_i, Semakin besar nilai rasio ini maka nilai persebaran fitur makin tinggi.
3. *Contribusion degree*: *Expectation Crossing Entropy* (ECE) dengan mempertimbangkan hubungan antara kemunculan fitur dan kelas, melalui perhitungan informasi suatu fitur muncul di dalam suatu kelas. Penelitian ini menggunakan formula ECE yang disederhanakan untuk menentukan nilai kontribusi suatu fitur terhadap suatu kelas. Formula yang telah disederhanakan seperti yang ditunjukkan oleh persamaan 7.

$$P(c_i,t)log\frac{P(c_i|t)}{p(c_j)} \quad (7)$$

Sehingga didapatkan persamaan 8 sebagai metode fitur seleksi kedua yang digunakan untuk mereduksi fitur berdasarkan 3 prinsip suatu fitur dikatakan sangat membantu dalam proses klasifikasi.

$$DFM(t,c_i) = \frac{DF(t,c_i)}{(1+\sum_{j=1|j\neq i}^{n} DF(t,c_j))} + p(t|c_i) + P(c_i,t)log\frac{P(c_i|t)}{p(c_j)} \quad (8)$$

Hasil perhitungan dengan menggunakan persamaan 8 adalah nilai yang menunjukkan tingkat kontribusi suatu kata kunci terhadap suatu kelas.

3.4. *Training Data*

Proses pelatihan data tekstual dilakukan dengan data sebanyak 299 data soal ujian SBMPTN. Data ini kemudian dilatih untuk mengidentifikasi tema bidang studi dengan mengklasifikannya ke dalam 68 tema. Klasifikasi dilakukan dengan metode *Naive Bayes Classifier*. *Input* yang digunakan untuk pelatihan ini adalah soal ujian berupa teks dan tema untuk masing-masing soal ujian. Proses *training* ini menghasilkan kombinasi nilai *mean* dan variansi untuk setiap *feature* (kata kunci) dalam satu kelas.

3.5. Klasifikasi Soal Ujian SBMPTN (*Naive Bayes Classifier*)

Nilai bobot (TF) dari kata kunci yang telah diseleksi kemudian dihitung nilai probabilitasnya dengan *Naive Bayes*. Oleh karena nilai bobot (TF) bersifat kontinu maka rumus distribusi *Gaussian* yang digunakan dalam klasifikasi menggunakan Naive Bayes berikut ini tahap-tahap pengklasifikasiannya:

1. Menghitung prediksi dengan *Naive Bayes Gaussian* menggunakan fungsi Dentitas Gauss pada persamaan 3. Dengan menggunakan nilai *mean* dan varian yang telah didapatkan dari proses *training* selanjutnya dihitung nilai dentitas Gaussnya yang menunjukkan nilai probabilitas relatif suatu fitur. Sama halnya dengan *mean* dan variansi, nilai dentitas Gauss ini juga akan dimiliki setiap fitur terhadap masing-masing kelas.

2. Menghitung tingkat *likelihood* dengan persamaan 4. Nilai-nilai dentitas Gauss kemudian dikalikan sehingga menghasilkan nilai probabilitas yang akan menunjukkan di kelas mana data yang diinputkan ini memiliki kemungkinan terbesar.
3. Menormalisasi nilai probabilitas dengan persamaan 5. Menentukan nilai probabilitas akhir dilakukan dengan normalisasi dari nilai probabilitas sebelumnya, sehingga dapat diketahui berapa % kah suatu data memiliki kemungkinan untuk masuk ke dalam suatu kelas tertentu.

Proses *updateable* dilakukan berdasarkan waktu atau *time based* dengan sejumlah data *training* awal sebagai *knowledge* dan waktu tertentu sebagai batas penambahan *knowledge*. Misalnya dalam penelitian ini digunakan sebanyak 299 soal ujian sebagai *training data*, kemudian dimasukkan 15 data *testing*, pada pukul 23.00 dilakukan proses *update* data *training* dengan 15 data soal ujian ini. Sehingga data *training* akan terus bertambah. Jumlah data *testing* yang masuk tidak dibatasi jumlahnya.

3.6. Evaluasi

Untuk mengevaluasi kinerja klasifikasi dilakukan dengan menghitung F1 *measure* yang merupakan kombinasi dari *precision* dan *recall*.

$$precision = \frac{number\ of\ correct\ positive\ predictions}{number\ of\ positive\ predictions} \quad (9)$$

$$recall = \frac{number\ of\ correct\ positive\ predictions}{number\ of\ positive\ examples} \quad (10)$$

$$F1\ measure = \frac{2 * precision * recall}{precision + recall} \quad (11)$$

4. HASIL DAN PEMBAHASAN

4.1. Pengumpulan Data

Data yang digunakan bersumber dari Buku Top Fokus SBMPTN IPC 2015 oleh Genta Group Production. Data soal-soal SBMPTN yang diambil dari tahun 2009 sampai 2014 untuk bidang studi Matematika, Fisika, Kimia, Biologi, Sejarah, Geografi, Ekonomi, dan Sosiologi. Jumlah data soal ujian yang diambil untuk SBMPTN tahun 2009 dan 2010 masing-masing sebanyak 120 soal. Dan untuk soal ujian SBMPTN tahun 2011 sampai 2014 diambil sebanyak masing-masing 40 soal. Jadi jumlah total soal yang diambil adalah 400 soal ujian. Soal-soal ujian ini dikategorikan ke dalam 171 tema dari buku-buku mata pelajaran yang diterbitkan oleh Pusat Perbukuan Departemen Pendidikan Nasional Tahun 2009 sesuai dengan kurikulum 2006 (KTSP). Syarat data yang akan digunakan adalah minimal terdapat 3 buah soal untuk setiap kategori atau bidang studi. Sehingga total soal ujian yang dipakai adalah 309 soal ujian dengan 68 tema.

4.2. Text Preprocessing

Tahap text preprocessing dilakukan dengan memanfaatkan library Sastrawi. Stemming kata menggunakan kelas StemmerFactory() menghasilkan string yang terdiri dari kata dasar dengan huruf kecil secara keseluruhan. Selain itu tanda baca dihilangkan pada fungsi ini. String ini kemudian difilter dengan kelas StopWordRemoverFactory() untuk dihilangkan stopword atau kata sambung yang tidak diperlukan. Proses ini menghasilkan string kata dasar tanpa stopword. Selanjutnya bentuk string diubah menjadi token-token atau array dengan menggunakan kelas TokenizerFactory() dari library Sastrawi. Setelah itu dilakukan penghapusan karakter selain huruf sehingga hanya menghasilkan token yang berisi kata. Setiap kata hasil text preprocessing ini disimpan di dalam database. Dalam implementasinya proses ini menghasilkan 1.497 kata.

4.3. Feature Selection

Seleksi fitur dilakukan saat training data. Satu data training yang masuk akan dilakukan text preprocessing terlebih dahulu untuk mendapatkan kata dasar, kemudian kumpulan kata dasar ini direduksi lagi untuk menghilangkan noise term agar fitur-fitur yang digunakan untuk proses klasifikasi hanya terdiri dari fitur yang mampu menjadi pembeda antar kelas. Dengan kata lain fitur yang akan digunakan ini merupakan fitur yang memiliki nilai kontribusi tinggi terhadap suatu kelas sehingga membantu proses klasifikasi. Tahap seleksi fitur ini yang pertama menggunakan DF-Thresholding yang bersifat class independent. Batas bawah dan batas atas yang digunakan adalah 1 dan 20. Jumah data soal ujian untuk masing-masing kelas dalam penelitian ini tidak sama. Dalam satu kelas paling sedikit terdapat 3 soal ujian dan paling banyak terdapat 13 soal ujian. Nilai DF terendah adalah 1 dan nilai DF tertinggi adalah 47. Seleksi kedua dilakukan dengan metode DFM yang bersifat class dependent dan berdasarkan peningkatan jumlah DF. Nilai threshlod DFM yang digunakan adalah 0,3. Angka 1 sampai 20 dan 0,3 dipilih sebagai threshold karena menghasilkan akurasi paling baik seperti yang ditunjukkan pada tabel 1. Dari 1.497 kata hasil text preprocessing didapatkan 1.394 keywords hasil seleksi fitur dengan dua tahap tersebut.

Tabel 1 Pengaruh pemilihan angka *threshold* terhadap akurasi dengan 299 data *training* dan 23 data *testing*

DF *Threshold*	DFM *Threshold*	Kata Terseleksi	Akurasi (%)
1 – 20	0,2	1438	78,26
	0,3	1394	82,61
	0,4	1297	77,27
1 - 30	0,2	1445	72,73
	0,3	1399	82,61
	0,4	1298	73,91
2 - 20	0,2	555	73.91
	0,3	511	73.91
	0,4	414	65.22
2 - 30	0,2	565	69,57
	0,3	517	73,91
	0,4	416	60,87

4.4. Training Data

Proses pelatihan data tekstual dilakukan dengan data sebanyak 299 data soal ujian SBMPTN. Data yang dibutuhkan untuk proses training ini adalah kombinasi teks soal ujian dengan temanya masing-masing. Teks soal ujin yang telah di-preprocessing untuk setiap kata pada satu kelas dihitung nilai mean dan variansinya. Perhitungan ini dilakukan untuk setiap fitur kata pada satu kelas. Kombinasi nilai mean dan varian inilah yang merupakan hasil dari training data.

*4.5. Klasifikasi Soal Ujian SBMPTN (*Naive Bayes Classifier*)*

Dengan menggunakan fitur-fitur yang terseleksi, dilakukan perhitungan nilai probabilitas. Persamaan yang digunakan untuk menghitung probabilitas ini adalah formula dentitas gauss yang ditunjukkan oleh persamaan 3. Sedangkan nilai fitur yang digunakan selama perhitungan adalah nilai TF atau *Term Frequency*.

4.6. Evaluasi

Evaluasi yang dilakukan pada penelitian ini menggunakan perhitungan akurasi hasil klasifikasi *precision*, *recall*, dan *f-measure*. Proses pengujian menggunakan *supplied test* dengan 2 kali *memorizing* dan 2 kali *testing*. Jumlah data *training* 299 soal ujian dengan jumlah data *testing* pertama dan kedua masing-masing 10 dan 13. Hasil evaluasi sistem secara keseluruhan dapat

dilihat pada tabel 2. Berdasarkan tabel 2 *memorizing* I nilai *precision* dari sistem 98,01%, nilai *recall* 98,37%, dan nilai *F-Measure* adalah 97,94% dengan akurasi 97,66%. Untuk *testing* I nilai *precision* 77,78%, nilai *recall* 77,78%, dan nilai *F-Measure* adalah 77,78% dengan akurasi 80%. *Memorizing* II nilai *precision* dari sistem adalah 97,48%, nilai *recall* 98,37%, dan nilai *F-Measure* adalah 97,56% dengan akurasi 97,41%. Dan *testing* II nilai *precision* dari sistem adalah 81,82%, nilai *recall* 90,91%, dan nilai *F-Measure* adalah 81,82% dengan akurasi 84,61%.

Tabel 2. Hasil Evaluasi Klasifikasi

No	Data *Training*	Data *Testing*	Akurasi (%)		*Precision* (%)		*Recall* (%)		*F-Measure* (%)	
			M	T	M	T	M	T	M	T
I	299	10	97,66	80,00	98,01	77,78	98,37	77,78	97,94	77,78
II	309	13	97,41	84,61	97,48	81,82	98,37	90,91	97,56	81,82

Pengujian menggunakan *supplied test* dilakukan bukan menggunakan *cross validation* karena jumlah data yang terlalu kecil dibandingkan dengan jumlah kelasnya. Untuk 322 soal dengan 68 tema nilai *k-fold* yang digunakan setidaknya 68, jika 322 dibagi dengan nilai *k* maka jumlah data *training* terlalu sedikit.dan data *testing* tidak seimbang. Sehingga tidak memungkinkan menerapkan metode *cross validation*.

Akurasi sistem pada saat proses *memorizing* lebih tinggi dari pada saat proses *testing* karena pada proses *testing* terdapat fitur-fitur dari dokumen atau soal ujian yang diklasifikasikan belum ada di dalam *database*. Sehingga mengurangi nilai probabilitas satu soal ujian terhadap suatu kelas. Hal ini yang memungkinkan satu soal ujian gagal diklasifkasikan dengan benar. Sedangkan pada proses *memorizing* semua fitur telah ada dalam *database* sehingga dapat diklasifikasikan dengan benar.

5. PENUTUP

5.1. Kesimpulan

Pengklasifikasian soal ujian SBMPTN dengan algoritma *Naive Bayes* yang dilakukan pada penelitian ini dapat berjalan dengan baik, sebagian besar data soal ujian yang diklasifikasikan sesuai dengan pengklasifikasian secara manual dengan nilai akurasi 84,61%. Soal ujian yang gagal diklasifkasikan dengan benar ini disebabkan oleh adanya kata kunci baru yang memiliki kemungkinan sebagai kata kunci penting tetapi tidak dapat digunakan karena belum terdaftar pada data training, selain itu kegagalan klasifikasi juga dapat disebabkan oleh keterbatasan data *training* yang dimiliki, seperti dapat dilihat pada proses *memorizing* terdapat beberapa soal ujian yang tidak mampu diingat oleh sistem dikarenakan *keyword* yang dimiliki kelas sebenarnya masih kurang beragam maupun kurang frekuensinya sehingga mempengaruhi derajat konsentrasi, derajat persebaran dan derajat kontribusinya.

5.2. Saran

Untuk pengembangan sistem yang dibuat pada penelitian ini, disarankan untuk menambah fungsi identifikasi sinonim Bahasa Indonesia, agar jika kata kunci yang masuk memiliki sinonim dengan kata dalam database tidak dianggap sebagai kata baru sehingga membantu dalam proses pengklasifikasian, dan penentuan tema untuk setiap soal dilakukan oleh pakar atau orang yang ahli dan bertanggung jawab dalam penyusunan soal ujian SBMPTN. Selain itu, dengan menambahkan satu tahap klasifikasi berdasarkan bidang studi sebelum klasifikasi terhadap tema dilakukan, diharapkan mampu meningkatkan akurasi. Karena dengan mengklasifikasikan ke dalam bidang studi terlebih dahulu akan mempersempit area klasifikasi terhadap tema.

DAFTAR PUSTAKA

A. Hamzah, 2012, "Klasifikasi Teks dengan Naive Bayes Classifier (NBC) untuk Pengelompokan Teks Berita dan Abstrak Akademis," *Prosiding Seminar Nasional Aplikasi Sains & Teknologi (SNAST) Periode III,* pp. 269-277.

C. D. Manning, P. Raghavan dan H. Schutze, 2009, "An Introduction to Information Retrieval", Cambridge: Cambridge University Press.

E. R. Anandita, 2013, "Klasifikasi Tebu dengan Menggunakan Algoritma Naive Bayes Classifier pada Dinas Kehutanan dan Perkebunan,".

H. Mario Naga Mait, 2012, "Making Stemming Synonym Indonesian Using Algorithm Nazief And Adriani,".

I. Rish, 2001, "An Empirical Study of the Naive Bayes Classifier," IBM Research Division Thomas J. Watson Research Center, Yorktown Heights.

J. Asian, H. E. Williams dan S. M. M. Tahaghoghi, 2003, "Stemming Indonesian,".

M. Adriani, J. Asian, B. Nazief, S. M. M. Tahaghoghi dan H. E. Williams, 2007, "Confix Stripping: Aproach to Stemming Algorithm for Bahasa Indonesia," *Internal Publication, Faculty of Computer Science, University of Indonesia,* vol. 6, no. 4, pp. 1-33.

P. R. Deshmukh dan R. Ade, 2014, "Classification of Students Using Psychometric tests with the Help of Incremental Naive Bayes Algorithm," *International Journal of Computer Applications,* vol. 89.

R. Arthana, 2012, "Stopword Bahasa Indonesia (dan Implementasi pada Apache Lucene)," [Online]. Available: http://www.rey1024.com/2012/06/stop-word-bahasa-indonesia-dan-implementasi-pada-apache-lucene/

R. Parimala dan R. Nallaswamy, , 2012, "A Study on Analysis of SMS Classification Using Document Frequency Threshold," *I.J. Information Engineering and Electronic Business,* pp. 44-50.

S. N. Fais A, M. Aditya D dan S. Mulya I, , 2014, "Klasifikasi Calon Pendonor darah dengan Metode Naive Bayes Classifier," Informatika, Program Teknologi Informasi dan Ilmu Komputer, Universitas Brawijaya,, Malang.

S. Natalius, 2010, "Metode Naive Bayes Classifier dan Penggunaanya pada Klasifikasi Dokumen," Institut Teknologi Bandung, Bandung.

SBMPTN, 2015, "Informasi Umum Seleksi Bersama Masuk Perguruan Tinggi Negeri Tahun 2015," 2015. [Online]. Available: https://sbmptn.or.id/?mid=13#a1. [Diakses 18 September 2015].

V. Korde dan C. N. Mahender, "Text Classification and Classifiers: a Survey," *International Journal of Artificial Intelligence & Applications (IJAIA),* vol. 3, no. 2, 2012.

W. Zheng dan G. Feng, 2014, "Feature Selection Method Based on Improved Document

Frequency," *Telkomnika,* vol. 12, pp. 905-910.

WEKA, 2015, "Weka Classifiers.Bayes," [Online]. Available: http://weka.sourceforge.net/doc.dev/weka/classifiers/bayes/NaiveBayesUpdateable.html. [Diakses 20 Oktober 2015].

Y. Yang dan J. O. Pedersen, 1997, "A Comparative Study on Feature Selection in Text Categorization," *International Conference on Machine Learning (ICML),* pp. Vol. 97, pp. 412-420.