Coding : Jurnal Komputer dan Aplikasi
Volume 07, No. 2 (2019), hal 49-60 ISSN 2338-493X

IMPLEMENTASI METODE *NAIVE BAYES CLASSIFIER* PADA SISTEM PENGKLASIFIKASI BERITA OTOMATIS BERBASIS *WEBSITE* (STUDI KASUS: BERITA LOKAL DARI MEDIAMASSA *ONLINE* KALIMANTAN BARAT)

Ines Rasila[1], Uray Ristian[2]
[1,2,]Jurusan Rekayasa Sistem Komputer, Fakultas MIPA Universitas Tanjungpura
Jl. Prof. Dr. H. Hadari Nawawi, Pontianak
Telp./Fax.: (0561) 577963
e-mail: [1]inesrasila@student.untan.ac.id, [2]eristian@siskom.untan.ac.id

Abstrak

Media massa online seperti Pontianak Post, Tribun Pontianak, dan The Tanjungpura Times selalu memuat berita-berita terkini seputar Provinsi Kalimantan Barat. Proses klasifikasi berita selama ini masih dilakukan secara manual oleh tenaga khusus sehingga memakan waktu dan tenaga. Jumlah berita yang terkumpul dari text mining dengan menggunakan Node.js adalah sebanyak 18.794 berita. Pengujian performa dilakukan dengan membagi data sebanyak 936 sebagai data latih (training set) dan 405 data sebagai data uji (testing set) dengan rasio pembagian 70:30. Performa model yang dihasilkan dengan menggunakan metode Naive Bayes Classifier (NBC) mendapatkan akurasi sebesar 98,90% dari total 9 kelas yang diuji. Model tersebut kemudian digunakan untuk mengklasifikasi 17.453 berita di luar training set dan testing set. Hasil klasifikasi berita dengan menggunakan NBC menunjukkan bahwa sebanyak 4.821 berita diklasifikasikan ke dalam topik politik dan pemerintahan, sebanyak 2.475 berita diklasifikasikan ke dalam topik event, pariwisata dan olahraga, sebanyak 2.254 berita diklasifikasikan ke dalam topik pemilu, sebanyak 1.885 berita diklasifikasikan ke dalam topik hukum dan kriminalitas, sebanyak 1.561 berita diklasifikasikan ke dalam topik pendidikan, sebanyak 1.400 berita diklasifikasikan ke dalam topik lalu-lintas dan transportasi, sebanyak 1.179 berita diklasifikasikan ke dalam topik bencana alam, sebanyak 663 berita diklasifikasikan ke dalam topik kesehatan, dan sebanyak 488 berita diklasifikasikan ke dalam topik narkoba.

Kata Kunci: *Naive Bayes Classifier, Text Mining, Klasifikasi Berita, Node.js, Python*

1. PENDAHULUAN

Berita adalah laporan mengenai peristiwa atau pendapat yang memiliki nilai penting, menarik sebagian besar peminat-nya, masih baru atau aktual dan dipubli-kasikan secara luas melalui media massa periodik [1]. Berita pada awalnya bersumber dari surat kabar, radio dan televisi namun seiring berkembangnya teknologi dan bertambahnya pengguna internet di Indonesia yang menurut Kementrian Komunikasi dan Informatika Republik Indonesia pengguna internet tahun 2017 diperkirakan mencapai 143,26 juta atau setara dengan 54,68 persen dari total penduduk Indonesia maka dari itu kini berita juga bersumber dari media massa *online* [2]**.**

Media massa *online* merupakan salah satu sumber informasi dari internet yang memiliki banyak pembaca. Tak terkecuali di Kalimantan Barat, ada banyak media massa *online* lokal yang selalu memuat berita-berita terkini seputar Provinsi Kalimantan Barat. Selain mudah didapat, dengan media internet cakupan pembaca dapat lebih luas. Beberapa contoh media massa *online* lokal Kalimantan Barat adalah Pontianak Post, Tribun Pontianak, Kalbar Online, The Tanjungpura Times, Equator dan lain-lain. Ketersediaan berita di Internet yang berlimpah akan menyulitkan masyarakat untuk mengaksesnya jika berita tersebut tidak diatur secara layak. Pengaturan berita yang umum adalah dengan melakukan klasifikasi pada masing-masing artikel berita tersebut. Klasifikasi tersebut dapat didasarkan pada kondisi yang ada dalam masyarakat ataupun menurut standar khusus. Jumlah klasifikasi tersebut sifatnya selalu berkembang. Proses klasifikasi dilakukan dengan melibatkan

tenaga khusus yang memahami proses klasifikasi suatu artikel berita. Akan lebih efisien apabila mengklasifikasikan berita dapat dilakukan secara otomatis.

Dalam penelitian yang berjudul "Klasifikasi Artikel Berita Berbahasa Indonesia Berbasis Naive Bayes Classifier Menggunakan Confix-Stripping Stemmer" data artikel berita diambil dari media massa online Kompas dengan total data sebanyak 1.995 yang terdiri dari 12 kategori dengan rasio pembagian 70% sebagai data latih dan 30% untuk data uji di setiap kategori. Hasil evaluasi kinerja aplikasi yang dibangun dengan menggunakan data latih yaitu akurasi sebesar 86.738%, *precision* 87.20%, *recall* 86.70%, dan *f-measure* 86.80% (Arifiyanti, 2014). Selanjutnya, penelitian yang berjudul "Klasifikasi Berita Indonesia Menggunakan Metode Naive Bayesian Classification dan Support Vector Machine dengan Confix Stripping Stemmer" metode Naive Bayes menghasilkan akurasi 82.2%, *precision* 83.9%, *recall* 82.2%, dan *f-measure* 82.4% (Ariadi, 2015). Penelitian lain yang mengklasifikasikan data teks menggunakan metode Naive Bayes Classifier penelitian berjudul "Klasifikasi Data Forum Dengan Menggunakan Metode Naive Bayes Classifier" dimana penelitian ini mengklasifikasikan pertanyaan atau pernyataan yang dituliskan oleh pengguna pada suatu forum secara otomatis, hasil tingkat akurasinya mencapai 73% [3].

Berdasarkan latar belakang yang telah dijabarkan, maka akan dibuat suatu sistem untuk mengklasifikasi teks berita secara otomatis yang bersumber dari tiga media massa *online* Kalimantan Barat yaitu Tribun Pontianak, Pontianak Post dan The Tanjungpura Times dengan menggunakan metode *Naive Bayes Classifier* (NBC). Metode NBC dipilih dalam penyelesaian masalah klasifikasi berita dikarenakan metode NBC dalam melakukan klasifikasi atau kategorisasi teks menggunakan atribut kata yang muncul dalam suatu dokumen sebagai dasar klasifikasinya. Sistem yang akan dibangun pada penelitian ini merupakan aplikasi berbasis *website*.

2. LANDASAN TEORI

2.1 Text Mining

Penambangan teks *(text mining)* adalah proses ekstraksi pola berupa informasi dan pengetahuan yang berguna dari sejumlah besar sumber data teks, seperti dokumen Word, PDF, kutipan teks, dan lain-lain. Jenis masukan untuk penambangan teks ini disebut data tak terstruktur dan merupakan pembeda utama dengan penambangan data yang menggunakan data terstruktur atau basis data sebagai masukan [4]. Penambangan teks dapat dianggap sebagai proses dua tahap yang diawali dengan penerapan struktur terhadap sumber data teks dan dilanjutkan dengan ekstraksi informasi dan pengetahuan yang relevan dari data teks terstruktur ini dengan menggunakan teknik dan alat yang sama dengan penambangan data.

2.2 Text Preprocessing

Text preprocessing bertujuan untuk mempersiapkan teks menjadi data yang akan mengalami pengolahan pada tahapan berikutnya [5]. Tahapan *text preprocessing* adalah sebagai berikut:

1. *Case Folding*
2. *Tokenizing*
3. *Filtering*
4. *Stemming*

Salah satu *library* yang biasa digunakan dalam melakukan proses *stemming* Bahasa Indonesia adalah *library* Sastrawi di mana *library* ini menerapkan Algoritma Nazief dan Andriani. Semua tahapan *preprocessing* yang dilakukan pada penelitian ini menggunakan *library* Sastrawi pada aplikasi Python.

2.3 *Naive Bayes Classifier*

Naive Bayes Classifier (NBC) merupakan pengklasifikasian dengan metode probabilitas dan statistik yang dikemukan oleh ilmuwan Inggris Thomas Bayes, yaitu memprediksi probabilitas di masa depan berdasarkan pengalaman di masa sebelumnya. Metode NBC menempuh dua tahap dalam proses klasifikasi teks, yaitu tahap pelatihan dan tahap klasifikasi. Pada tahap pelatihan dilakukan proses analisis terhadap sampel dokumen berupa pemilihan *vocabulary* atau kumpulan kosakata, yaitu kata yang mungkin muncul dalam koleksi dokumen sampel yang sedapat mungkin dapat menjadi representasi dokumen. Selanjutnya adalah penentuan probabilitas prior bagi tiap kategori berdasarkan sampel dokumen. Pada tahap klasifikasi ditentukan nilai kategori dari suatu dokumen

berdasarkan *term* yang muncul dalam dokumen yang diklasifikasi [6].

Secara umum Teorema Bayes dapat dinotasikan sebagai berikut [7]:

$$P(A|B) = \frac{P(A)P(B|A)}{P(B)} \quad (1)$$

Keterangan:
$P(A)$ = probabilitas terjadinya A
$P(B)$ = probabilitas terjadinya B
$P(A|B)$ = probabilitas terjadinya A dengan syarat B
$P(B|A)$ = probabilitas terjadinya B dengan syarat A

Pada saat klasifikasi, algoritma akan mencari probabilitas tertinggi dari semua kategori dokumen yang diujikan. Adapun persamaannya adalah sebagai berikut:

$$V_{MAP} = \underset{v_j \in V}{argmax}\ P(v_j) \prod_i P(a_i \mid v_j) \quad (2)$$

Keterangan:

V_{MAP} = probabilitas tertinggi dari semua kategori
$P(v_j)$ = probabilitas terjadinya kategori tertentu dari sekumpulan data
$P(a_i \mid v_j)$ = probabilitas terjadinya kata a_i pada suatu dokumen dengan kategori v_j

Probabilitas terjadinya kategori tertentu $P(v_j)$ dari sekumpulan data *(prior probability)* yang dimasukkan dapat dihitung dengan persamaan:

$$P(v_j) = \frac{|doc\ j|}{|training|} \quad (3)$$

Keterangan:
$|doc\ j|$ = jumlah teks atau dokumen yang memiliki kategori j
$|training|$ = jumlah teks atau dokumen dalam contoh yang digunakan untuk *training*

Probabilitas suatu kata menunjukkan kecenderungan pada kategori tertentu *(likelihood)* dalam suatu teks atau dokumen dapat dihitung dengan persamaan:

$$P(a_i \mid v_j) = \frac{|n_i + 1|}{|n + kosakata|} \quad (4)$$

Keterangan:
n_i = jumlah kemunculan kata a_i dalam dokumen yang berkategori v_j
n = banyaknya seluruh kata dalam dokumen dengan kategori v_j
$kosakata$ = banyaknya kata dalam contoh pelatihan

2.4 *Confusion Matrix* dan Pengukuran Performa

Kinerja sistem klasifikasi menggambarkan seberapa baik sistem dalam mengklasifikasikan data. *Confusion matrix* merupakan salah satu metode yang dapat digunakan untuk mengukur kinerja suatu metode klasifikasi. Pada dasarnya *confusion matrix* mengandung informasi yang membandingkan hasil klasifikasi yang dilakukan oleh sistem dengan hasil klasifikasi yang seharusnya [8]. *Confusion Matrix* terdapat empat istilah sebagai representasi hasil proses klasifikasi yaitu:

1. *True Positive* (TP) adalah jumlah data positif yang diklasifikasikan sebagai data positif.
2. *True Negative* (TN) adalah jumlah data negatif yang diklasifikasikan sebagai data negatif.
3. *False Positive* (FP) adalah jumlah data negatif yang diklasifikasikan sebagai data positif.
4. *False Negative* (FN) adalah jumlah data positif yang diklasifikasikan sebagai data negatif.

Nilai akurasi menggambarkan seberapa akurat sistem dapat mengklasifikasikan data secara benar yaitu membandingkan data yang terklasifikasi benar dengan keseluruhan data. Nilai akurasi dapat diperoleh dengan persamaan berikut:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \times 100\% \quad (5)$$

Nilai *precision* menggambarkan jumlah data positif yang terklasifikasi benar dibagi dengan total data yang terklasifikasi positif. Nilai *precision* dapat diperoleh dengan persamaan berikut:

$$Precision = \frac{TP}{FP+TP} \times 100\% \quad (6)$$

Nilai *recall* menunjukkan jumlah data positif terklasifikasi positif secara benar oleh sistem. Nilai *recall* dapat diperoleh dengan persamaan berikut:

$$Recall = \frac{TP}{FN+TP} \times 100\% \quad (7)$$

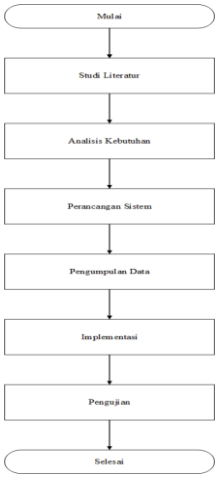
F-measure merupakan nilai yang didapatkan dari pengukuran *precision* dan *recall* antara *predicted class* dengan *actual class* yang terdapat pada data masukan. Nilai *f-measure* dapat diperoleh dengan persamaan berikut:

$$F = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (8)$$

2.5 Web Scraping

Web Scraping adalah proses otomatis yang melibatkan beberapa jumlah *parsing* data untuk memperoleh hanya informasi yang dibutuhkan. *Web scraping* mengambil HTML lengkap dari sebuah situs *web*, menguraikannya menjadi objek dengan menggunakan sejumlah perpustakaan yang tersedia, mengisolasi dan memproses data yang diinginkan. Terbebas dari keterbatasan dan ketersediaan API untuk mengambil data [8].

3. METODE PENELITIAN



Gambar 1. Diagram Alir Penelitian

3.1 Studi Literatur

Penelitian dimulai dengan melakukan studi literatur. Hal-hal yang dilakukan yaitu membaca bahan-bahan referensi berupa buku-buku, jurnal ilmiah penelitian sebelumnya.

3.2 Analisis Kebutuhan

Tahap selanjutnya adalah melakukan analisis kebutuhan perangkat keras dan perangkat lunak.

3.3 Perancangan Sistem

Setelah analisis kebutuhan selesai dilakukan, maka selanjutnya adalah tahap perancangan sistem yang terdiri dari perancangan aplikasi *data mining* & *data processing*, aplikasi *front end*, dan perancangan basis data.

3.4 Pengumpulan Data

Pengumpulan data berita menggunakan *text mining* dengan teknik *web scraping*. Data berita yang digunakan bersumber dari tiga media massa *online* yang memberitakan berita-berita terkait Kalimantan Barat. Tahap pengumpulan data dilakukan setelah aplikasi *data mining* atau *scraping module* selesai dibuat. *Scraping module* berjalan pada sebuah *private server*. Berita-berita yang sudah terkumpul akan dilakukan proses *training* atau data latih secara manual dengan mengumpulkan kata-kata kunci yang sebelumnya ditentukan topik apa saja yang akan diangkat ke dalam penelitian terlebih dahulu.

3.5 Implementasi

Setelah perancangan selesai dilakukan, penelitian dilanjutkan ke tahap implementasi yaitu pembuatan aplikasi berdasarkan rancangan.

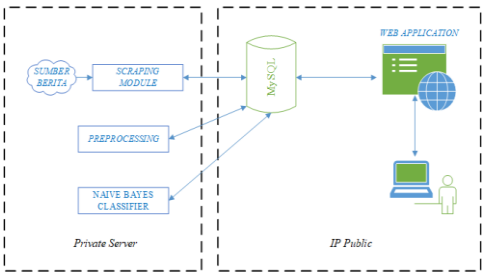
3.6 Pengujian

Tahap terakhir adalah tahap pengujian dengan menggunakan metode *black box*. Analisis pengujian performa yang dilakukan menghitung akurasi, *precision*, *recall* dan *F-Measure*.

4. PERANCANGAN

4.1 Diagram Blok

Diagram blok dibuat untuk memetakan proses kerja suatu sistem untuk memudahkan dalam memahami alur kerja suatu sistem. Bagian utama atau fungsi yang diwakili oleh blok yang dihubungkan dengan garis yang menunjukkan hubungan dari blok-blok tersebut yang menggambarkan rancangan sistem secara keseluruhan. Diagram blok sistem dapat dilihat pada Gambar 2 berikut.

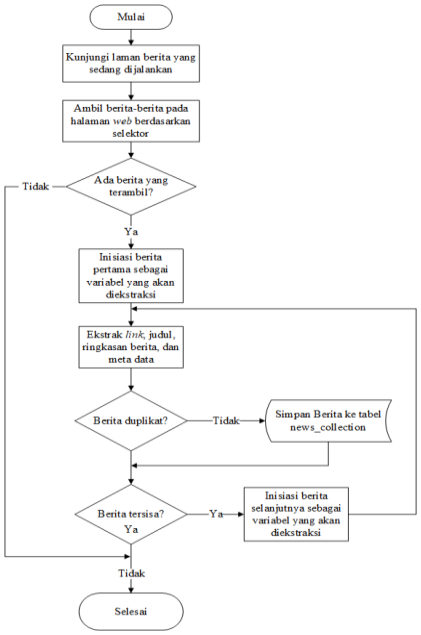


Gambar 2. Diagram Blok Sistem

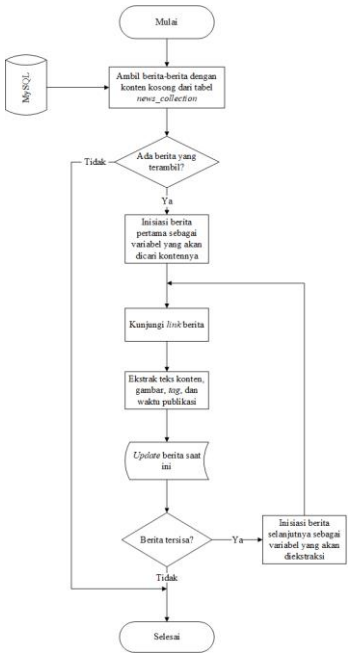
Data berita diambil dari sumber berita dengan menggunakan *scraping module* yang kemudian disimpan di dalam *database*. *Scraping module* merupakan kumpulan kode program Node.js yang berfungsi untuk melakukan *text mining* atau yang di dalam penelitian ini adalah *web scraping.* Perhitungan metode NBC dilakukan oleh *naive bayes classifier module* yang mengolah teks hasil *preprocessing* yang dilakukan oleh *text preprocessing module*. Pengguna berinteraksi dengan sistem pada sisi aplikasi *front end* berbasis *website* yang diakses melalui IP *public.* Proses pelatihan data atau *data training* dilakukan oleh admin pada aplikasi *website.* Aplikasi *website* dan aplikasi pada *private server* tidak berhubungan secara langsung. Kedua aplikasi tersebut menggunakan *database* yang sama sebagai media penyimpanan data.

4.2 Diagram Alir

Scraping module dibagi menjadi 2 yaitu *scraping module link collector* dan *scraping module content collector*. *Scraping module link collector* untuk mengambil *link* berita, judul, ringkasan berita dan meta data. Diagram alir *scraping module link collector* dapat dilihat pada Gambar 3. *Scraping module content collector* untuk mengambil konten berita dari *link-link* yang sudah dikumpulkan dari *scraping module link collector*. Diagram alir *scraping module content collector* dapat dilihat pada Gambar 4.

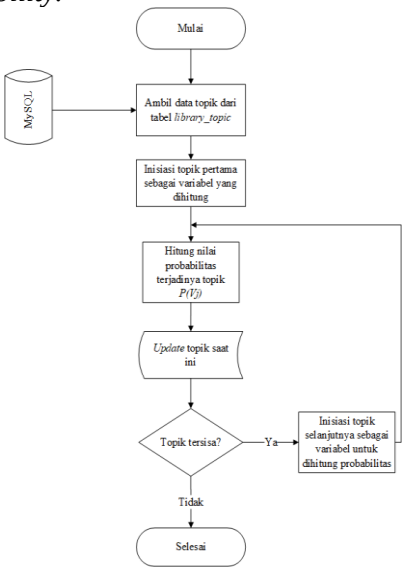


Gambar 3. Diagram Alir *Scraping Module Link Collector*

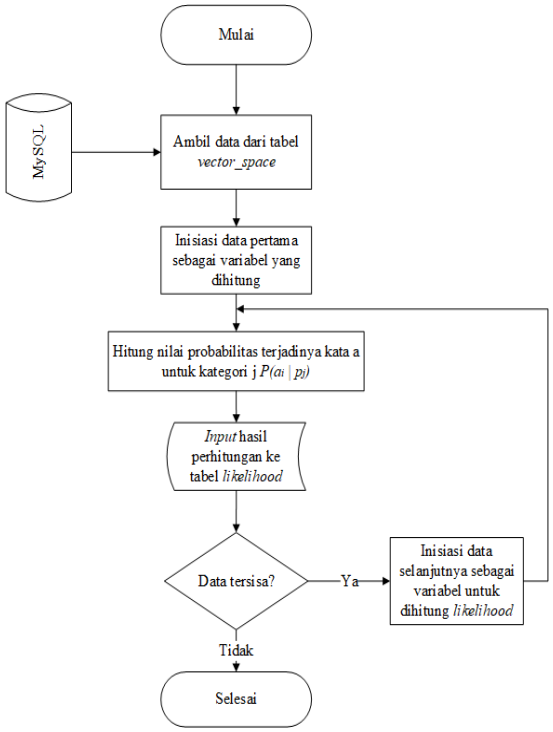


Gambar 4. Diagram Alir *Scraping Module Content Collector*

Naive bayes classifier module terbagi menjadi 3 program yaitu program untuk menghitung *prior probability*, *likelihood* dan prediksi. Gambar 5 merupakan diagram alir perhitungan *prior probability.* Program perhitungan *prior probability* dimulai dengan mengambil semua topik atau *class* yang tersimpan di dalam *database* dan menghitung jumlah berita dari data latih *(training set)* untuk setiap topik. Nilai *prior probability* yang telah dihitung kemudian disimpan ke dalam *database.* Proses akan terus dilakukan sampai semua topik telah memiliki nilai *prior probability.*



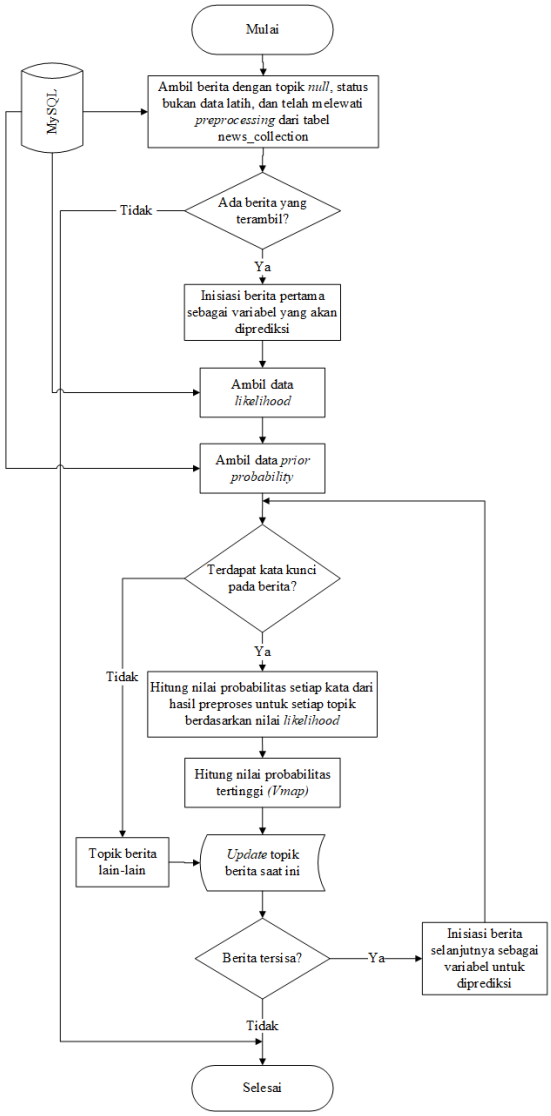
Gambar 5. *Flowchart* NBC *Prior Probability*



Gambar 6. *Flowchart* NBC *Likelihood*

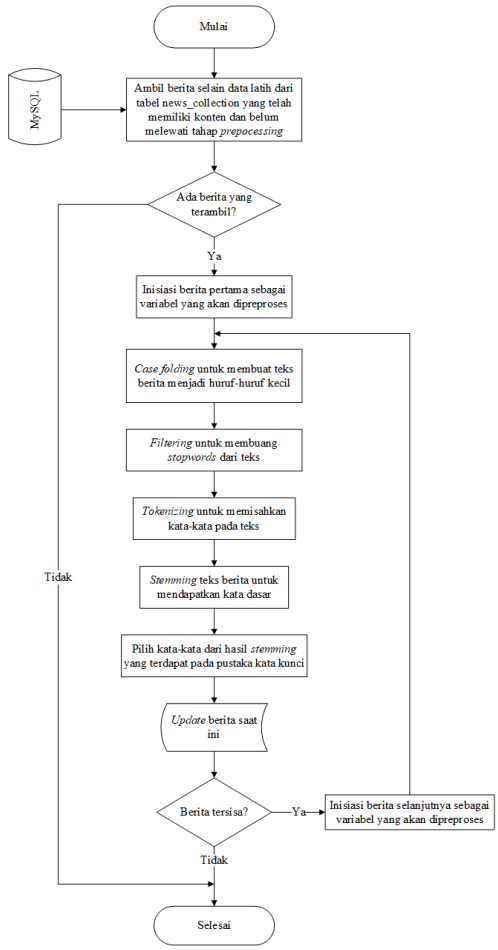
Gambar 6 merupakan diagram alir perhitungan *likelihood*. Perhitungan *likelihood* dimulai dengan mengambil data *vector space* yang berisi jumlah kemunculan kata kunci pada suatu dokumen berita yang memiliki topik tertentu yang dihasilkan dari proses *training*. Data *vector space* ini dikelompokkan berdasarkan kata kunci data topik berita. Setiap kata kunci kemudian dihitung probabilitasnya pada setiap topik yang kemudian menghasilkan nilai *likelihood*. Nilai *likelihood* untuk setiap kata pada setiap topik yang telah dihitung kemudian disimpan ke dalam *database*. Proses penghitungan *likelihood* akan selesai jika probabilitas kata kunci pada setiap topik telah dihitung. Perhitungan *likelihood* hanya berjalan satu kali, jika terjadi perubahan pada *training set* maka proses ini harus diulang kembali.

Perhitungan prediksi topik berita diatur untuk berjalan setiap 5 menit secara terus-menerus. Program perhitungan prediksi topik ini akan menunjukkan kualitas dari model *(prior probability* dan *likelihood)* yang dihasilkan dari proses *training*. Diagram alir yang menunjukkan proses perhitungan topik berita dapat dilihat pada Gambar 7 berikut.



Gambar 7. *Flowchart* NBC Prediksi Topik

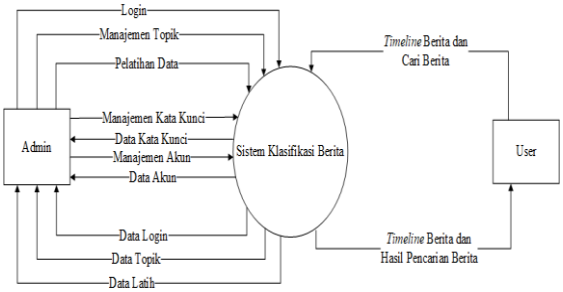
Proses *preprocessing* yang dilakukan oleh *text preprocessing module* dimulai dengan mengambil data berita yang tersimpan di dalam *database* yang belum melewati tahap *preprocessing*. Berita-berita yang akan dipraproses harus sudah memiliki konten berita terlebih dahulu dan bukan merupakan data latih. Jika tidak terdapat berita yang terambil dari *database* dengan kriteria-kriteria tersebut maka proses selesai. Jika terdapat berita yang terambil dengan kriteria-kriteria tersebut maka *preprocessing* akan dilanjutkan. Beberapa tahapan dari *text preprocessing* yaitu *case folding*, *filtering*, *tokenizing*, dan *stemming*. Diagram alir *text preprocessing* dapat dilihat pada Gambar 8.



Gambar 8. Diagram Alir *Text Preprocessing*

4.3 Data Flow Diagram (DFD)

Data Flow Diagram (DFD) merupakan alat perancangan sistem yang berorientasi alur data dengan konsep dekomposisi yang dapat digunakan untuk penggambaran analisis maupun rancangan sistem. DFD menggambarkan aliran data dalam proses tergantung pada *input* dan *output*.

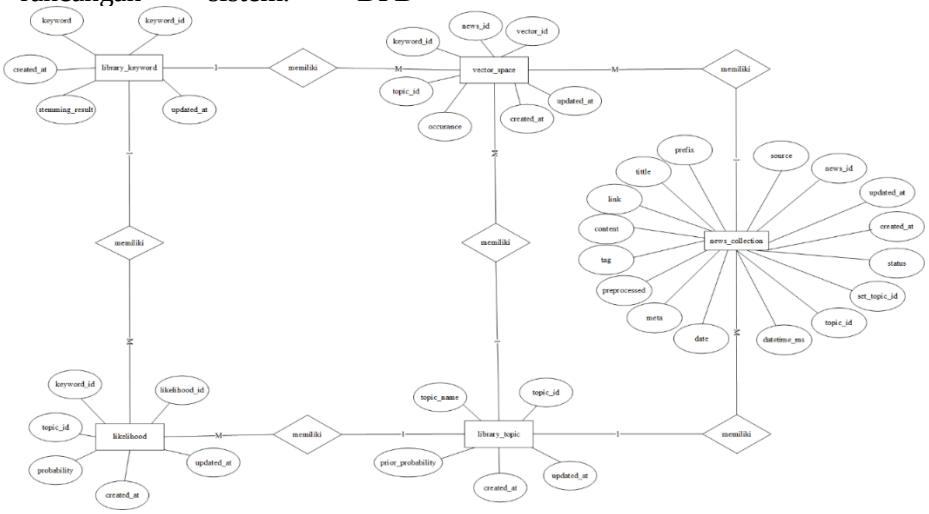


Gambar 9. *Data Flow Diagram* Level 0

Gambar 9 merupakan DFD level 0 dari sistem. Dari gambar tersebut dapat diketahui bahwa terdapat dua entitas yaitu admin dan *user*. Masing-masing dari entitas memiliki akses yang berbeda. Untuk masuk sebagai admin, diharuskan untuk *login* terlebih dahulu. Admin memiliki akses untuk melakukan manajemen topik, pelatihan data, manajemen kata kunci dan manajemen akun. Sedangkan *user* tidak memerlukan *login*, hanya memiliki akses untuk melihat *timeline* berita & hasil pencarian berita.

4.4 Entity Relationship Diagram

Basis data yang dibangun pada penelitian ini menggunakan 5 entitas yaitu *library_keyword* untuk menyimpan kata kunci dari latih data, *likelihood* menyimpan nilai probabilitas kata kunci, *vector_space* untuk menyimpan nilai kemunculan kata kunci, *library_topic* menyimpan data topik, *news_collection* menyimpan data berita. ERD penelitian ini dapat dilihat pada Gambar 10.

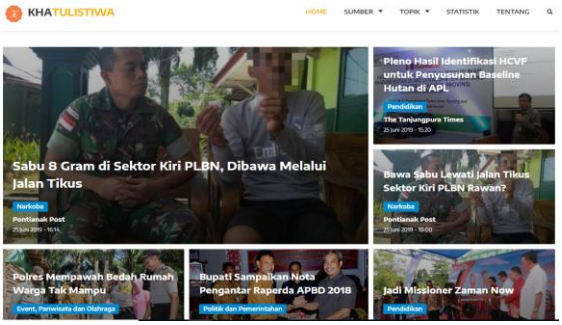


Gambar 10. Rancangan *Entity Relationship Diagram (ERD)*

5. HASIL DAN PEMBAHASAN

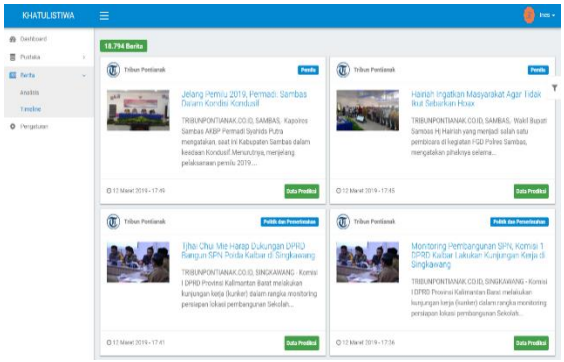
5.1 Implementasi Antarmuka Aplikasi

Halaman *homepage user* dapat diakses oleh pengguna tanpa harus *login* melalui IP *public*. Halaman ini memuat berita-berita berita terbaru dari berbagai sumber berita yang diurutkan dari waktu publikasi terbaru hingga waktu publikasi paling lama. Terdapat menu lainnya yang dapat dipilih oleh pengguna yaitu melihat berita berdasarkan sumber, berdasarkan topik, dan statistik berita dengan tampilan seperti pada Gambar 11 berikut.



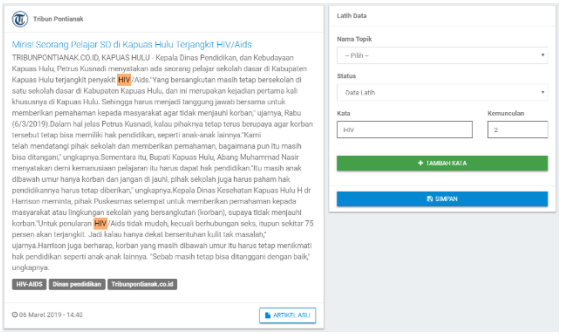
Gambar 11. Halaman *Homepage User*

Halaman selanjutnya yaitu halaman *timeline* berita admin, pada halaman ini menampilkan seluruh berita berdasarkan data berita yang paling baru hingga yang paling lama. Tampilan halaman *timeline* berita admin dapat dilihat pada gambar 12 berikut.



Gambar 12. Halaman *Timeline* Berita Admin

Halaman selanjutnya adalah halaman untuk membaca berita dan melakukan proses *training*. Admin dapat memilih kata-kata pada teks berita yang akan dijadikan sebagai kata kunci dan jumlah kemunculan kata-kata tersebut di dalam berita. Tampilan halaman *training* berita dapat dilihat pada Gambar 13 berikut.

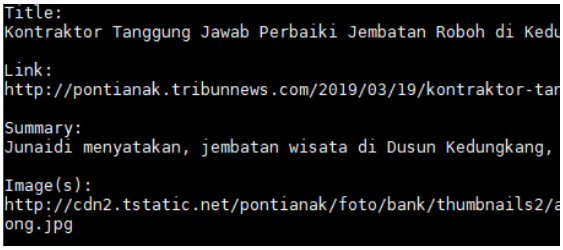


Gambar 13. Halaman *Training* Berita

5.2 Implementasi *Web Scraping*

Pengumpulan data dokumen berita pada penelitian ini adalah menggunakan teknik *web scraping*. Modul *scraping* dibangun menggunakan Node.js dan berjalan terus-menerus secara otomatis pada *server* untuk melakukan *text mining*.

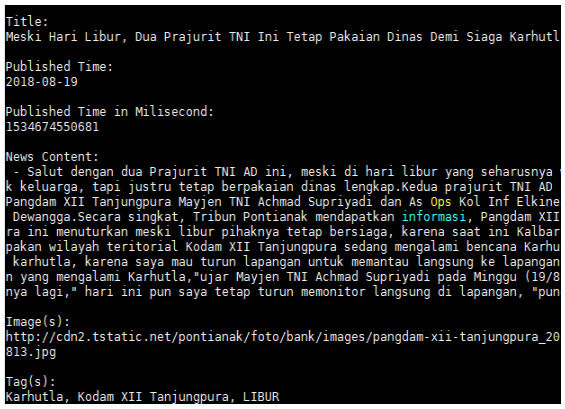
Modul *link collector* berjalan setiap 2 menit sekali. Modul ini hanya bertugas untuk mengumpulkan *link-link* berita pada halaman utama atau halaman yang telah ditentukan tanpa mengumpulkan konten berita. Hasil implementasi modul *link collector* dapat dilihat pada Gambar 14 berikut.



Gambar 14. Hasil *Mining* dari Modul *Link Collector*

Gambar 14 menunjukkan proses pengumpulan *link* berita dari sumber Tribun Pontianak yang dilakukan oleh program. Program/*script* berhasil mengekstraksi informasi-informasi dari halaman *website* Tribun Pontianak yaitu berupa judul berita, *link* berita, ringkasan berita/*summary*, dan *url* gambar. Dalam satu kali berjalan, *script* dapat mengambil lebih dari satu berita sehingga mengurangi kemungkinan adanya berita yang terlewatkan untuk diambil.

Modul *content collector* berjalan setiap 5 menit sekali. Modul ini bertugas untuk mengumpulkan konten berita dari *link-link* dari hasil proses pengumpulan *link* oleh modul *link collector*. Hasil implementasi modul *content collector* dapat dilihat pada Gambar 15 berikut.

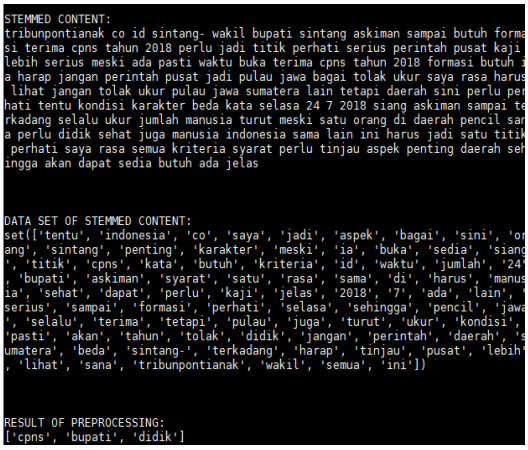


Gambar 15. Hasil *Mining* dari Modul *Content Collector*

Gambar 15 menunjukkan proses pengambilan konten berita dari sebuah berita yang dipublikasikan oleh Tribun Pontianak. *Script* berhasil mengekstraksi informasi-informasi yaitu teks konten berita, waktu publikasi berita, *url* gambar yang ada pada berita, dan *tag* berita.

5.3 Implementasi *Text Preprocessing*

Text preprocessing module berjalan setiap 5 menit sekali secara terus-menerus. Hasil implementasi *preprocessing* yang dilakukan oleh *text preprocessing module* dapat dilihat pada Gambar 16 berikut.



Gambar 16. Hasil *Preprocessing* Konten Berita

Gambar 16 menunjukkan hasil *preprocessing* dari sebuah konten berita. Hasil *preprocessing* yang berubah kumpulan kata-kata diseleksi kembali untuk mendapatkan kata kunci yang terdapat pada teks. Hasil akhir dari *preprocessing* didapatkan kata kunci "cpns", "bupati", dan "didik" terdapat pada konten atau teks berita.

5.4 Pengujian

5.4.1 *Naive Bayes Classifier* (NBC)

Pembagian jumlah dokumen data latih dan data uji penelitian ini menggunakan proporsi 70:30 seperti yang dilakukan pada referensi jurnal penelitian sebelumnya. Jumlah berita keseluruhan yang dijadikan sebagai data latih atau *training set* pada penelitian ini adalah 936. Jumlah berita keseluruhan yang dijadikan sebagai data uji atau *testing set* pada penelitian ini adalah sebanyak 405 berita.

Sebanyak 936 *training set* dan 405 *testing set* tersebut dibagi secara merata ke dalam 9 topik atau *class*. Sebanyak 998 kata kunci berhasil dikumpulkan dari proses *training* yang telah dilakukan. Rincian jumlah *training set*, *testing set* dan kata kunci dapat dilihat pada Tabel 1 berikut.

Tabel 1. Rincian Data Latih dan Data Uji

No.	Nama Topik	Data Latih	Data uji	Kata Kunci
1.	Bencana Alam	104	45	44
2.	*Event*, Pariwisata dan Olahraga	104	45	164
3.	Hukum dan Kriminalitas	104	45	117
4.	Kesehatan	104	45	169
5.	Lalu-lintas dan Transportasi	104	45	109
6.	Narkoba	104	45	32
7.	Pemilu	104	45	70
8.	Pendidikan	104	45	201
9.	Politik dan Pemerintahan	104	45	115

Beberapa kata kunci seperti kata "korban" tidak hanya ditemukan pada satu topik saja. Kata korban terdapat pada 3 topik yaitu bencana alam, hukum dan kriminalitas, dan lalu-lintas dan transportasi. Hal ini menyebabkan kolom kata kunci pada Tabel 1 jika ditotalkan akan lebih dari jumlah total kata kunci yaitu 998 kata kunci.

Nilai *prior probability* untuk setiap *class* adalah sama yaitu 0,1111. Hal ini disebabkan oleh jumlah *training set* untuk setiap *class* berjumlah sama yaitu 104 berita. Nilai *prior probability* dapat dilihat pada Tabel 2 berikut.

Tabel 2. *Prior Probability*

No.	Nama Topik	Probabilitas
1.	Bencana Alam	0,1111
2.	*Event*, Pariwisata dan Olahraga	0,1111
3.	Hukum dan Kriminalitas	0,1111
4.	Kesehatan	0,1111
5.	Lalu-lintas dan Transportasi	0,1111
6.	Narkoba	0,1111
7.	Pemilu	0,1111
8.	Pendidikan	0,1111
9.	Politik dan Pemerintahan	0,1111

5.4.2 *Text Mining*

Proses *text mining* yang dilakukan dari tanggal 28 Mei 2018 dihitung sampai dengan tanggal 12 Maret 2018 menghasilkan total 18.794 berita. Tribun Pontianak merupakan situs berita yang paling banyak dalam mempublikasikan berita yaitu dengan jumlah berita terkumpul sebanyak 16.008 berita. Pontianak Post berada pada posisi kedua dengan jumlah berita terkumpul sebanyak 2.505 berita. The Tanjungpura Times merupakan situs berita yang paling sedikit mempublikasikan berita dengan jumlah berita terkumpul sebanyak 281 berita. Rekapitulasi hasil *text mining* berdasarkan sumber berita dapat dilihat pada Gambar 17 berikut.

Gambar 17. Rekapitulasi Hasil *Text Mining* Berdasarkan Sumber Berita

Bulan Juli 2018 merupakan bulan dengan pengumpulan berita tertinggi. Sedangkan, Bulan Mei 2018 merupakan bulan dengan pengumpulan berita paling rendah dikarenakan proses *text mining* dimulai pada tanggal 20 Mei yang merupakan waktu pertama kali dimulainya proses *text mining*. Rekapitulasi hasil pengumpulan berita dari proses *text mining* untuk setiap bulannya dapat dilihat pada Gambar 18 berikut.

No.	Bulan	Jumlah Berita
1	Mei 2018	340
2	Juni 2018	1.831
3	Juli 2018	2.328
4	Agustus 2018	1.502
5	September 2018	1.423
6	Oktober 2018	2.227
7	November 2018	1.838
8	Desember 2018	2.254
9	Januari 2019	2.071
10	Februari 2019	2.155
11	Maret 2019	819
	TOTAL	18.788

Gambar 18. Rekapitulasi Hasil *Text Mining* Berdasarkan Bulan

5.4.3 Hasil Klasifikasi NBC

Terdapat total 17.453 berita di luar data latih dan data uji yang diklasifikasikan menggunakan NBC. Hasil rekapitulasi klasifikasi topik berita dapat dilihat pada Tabel 3 berikut.

Tabel 3. Hasil Klasifikasi NBC

No.	Nama Topik	Jumlah Berita	Persentase
1.	Politik dan Pemerintahan	4.821	27,62%
2.	*Event*, Pariwisata dan Olahraga	2.475	14,18%
3.	Pemilu	2.254	12,91%
4.	Hukum dan Kriminalitas	1.885	10,80%
5.	Pendidikan	1.561	8,94%
6.	Lalu-lintas dan Transportasi	1.400	8,02%
7.	Bencana Alam	1.179	6,76%
8.	Lain-lain	727	4,17%
9.	Kesehatan	663	3,8%
10.	Narkoba	488	2,8%
Total		17.453	100%

Terdapat 727 berita atau 4,17% dari total keseluruhan dengan topik lain-lain yang tidak diklasifikasi oleh sistem. Sistem tidak mengklasifikasi suatu berita jika di dalam teks berita tersebut tidak terdapat satu pun kata kunci yang telah dihimpun dari proses *training*. Hal

ini menunjukkan bahwa diperlukannya pengayaan kata-kata kunci pada *database* melalui proses *training* sehingga sistem dapat mengklasifikasi lebih banyak topik berita.

5.4.4 Performa NBC

Perhitungan performa menunjukkan bahwa *class* yang memiliki akurasi paling tinggi adalah kesehatan dengan nilai akurasi yang sebesar 99,75%. Nilai akurasi paling rendah dimiliki oleh *class* politik dan pemerintahan dengan nilai akurasi 97,28%. Nilai *precision* paling tinggi dimiliki oleh *class* kesehatan, narkoba, dan pendidikan dengan nilai *precision* 100%. Nilai *precision* paling rendah dimiliki oleh *class* politik dan pemerintahan yaitu sebesar 82,69%. Nilai *recall* paling tinggi dimiliki oleh *class event*, pariwisata dan olahraga, dan kesehatan yaitu sebesar 97,78%. Nilai *recall* yang paling rendah dimiliki oleh *class* hukum dan kriminalitas yaitu 91,11%. Nilai *F-measure* didapatkan dari nilai *precision* dan *recall*. Nilai *F-measure* paling tinggi terdapat pada *class* kesehatan yaitu sebesar 98,88%. Nilai *F-measure* paling rendah dimiliki oleh *class* politik dan pemerintahan yaitu 88,66%.

Hasil perhitungan performa *Naive Bayes Classifier (NBC)* yang dihasilkan pada penelitian ini dapat dilihat pada Tabel 4 berikut

Tabel 4. Performa Pengujian NBC

Topik/*Class*	Akurasi	*Precision*	*Recall*	*F-measure*
Bencana Alam	99,01%	97,67%	93,33%	95,45%
Event, Pariwisata dan Olahraga	99,26%	95,65%	97,78%	96,70%
Hukum dan Kriminalitas	98,02%	91,11%	91,11%	91,11%
Kesehatan	99,75%	100,00%	97,78%	98,88%
Lalu-lintas dan Transportasi	98,77%	93,48%	95,56%	94,51%
Narkoba	99,51%	100,00%	95,56%	97,73%
Pemilu	99,26%	97,73%	95,56%	96,63%
Pendidikan	99,26%	100,00%	93,33%	96,55%
Politik dan Pemerintahan	97,28%	82,69%	95,56%	88,66%
Rata-Rata	98,90%	95,37%	95,06%	95,14%

6. KESIMPULAN

Berdasarkan penelitian yang telah dilakukan dalam membangun sistem pengklasifikasi berita otomatis yang dimulai dari tahap studi literatur, analisis kebutuhan, perancangan sistem, pengumpulan data, implementasi sistem, hingga tahap pengujian dapat diambil kesimpulan-kesimpulan sebagai berikut:

1. *Text mining* dilakukan menggunakan bahasa pemrograman *server-side* yang di dalam penelitian ini menggunakan bahasa pemrograman Node.js. Sumber daya yang dibutuhkan untuk dapat menjalan *script mining* adalah sebuah *private server*. Pemilihan elemen-elemen HTML yang memuat informasi yang akan diambil sangat penting untuk mendapatkan data yang tepat dari halaman *website* tersebut.
2. Jumlah berita yang dikumpulkan dari proses *text mining* yang dihitung pada rentang waktu 20 Mei 2018 sampai dengan 12 Maret 2019 adalah sebanyak 18.794 berita. Sebanyak 16.008 berita dikumpulkan dari situs Tribun Pontianak, 2.505 berita dikumpulkan dari situs Pontianak Post, dan 281 berita dikumpulkan dari situs The Tanjungpura Times.
3. Hasil klasifikasi berita dengan menggunakan *Naive Bayes Classifier* menunjukkan bahwa sebanyak 4.821 berita ke topik politik dan pemerintahan, 2.475 berita ke topik *event*, pariwisata dan olahraga, 2.254 berita ke topik pemilu, 1.885 berita ke topik hukum dan kriminalitas, 1.561 berita ke topik pendidikan, 1.400 berita ke topik lalu-lintas dan transportasi, 1.179 berita ke topik bencana alam, 663 berita ke topik kesehatan, dan 488 berita ke topik narkoba. Terdapat 727 berita yang tidak berhasil diklasifikasi oleh sistem pada penelitian ini.
4. Klasifikasi berita dengan menggunakan *Naive Bayes Classifier* pada 405 data uji berhasil mengklasifikasi dengan benar 385 berita dan 20 prediksi yang salah. Performa

pengujian berdasarkan Tabel 5.21 menunjukkan bahwa rata-rata nilai *accuracy* dari 9 *class* yang diuji adalah 98,90%, rata-rata nilai *precision* adalah 95,37%, rata-rata nilai *recall* adalah 95,06%, dan rata-rata nilai *F-measure* adalah 95,14%.

7. SARAN

Berdasarkan penelitian yang telah dilakukan oleh penulis terdapat beberapa hal yang patut diperhatikan dan penulis sarankan agar penelitian ini dapat dikembangkan pada penelitian-penelitian selanjutnya yaitu antara lain:

1. Proses *training* berita dilakukan secara otomatis oleh sistem dengan bantuan manusia untuk menentukan *actual class*. Proses pemilihan kata kunci sepenuhnya dilakukan oleh sistem atau yang disebut dengan *unsupervised learning*.
2. Menambah jumlah *class* yang diangkat ke dalam penelitian.
3. Menambah fitur klasifikasi lain selain topik berita seperti klasifikasi sentimen berita.
4. Menambahkan fitur *text summarization* untuk membantu pengguna dalam menyimpulkan isi berita.

8. DAFTAR PUSTAKA

[1] J. Wahyudi, Komunikasi Jurnalistik: Pengetahuan Praktis Kewartawanan, Suratkabar-Majalah, Radio, dan Televisi., Bandung: Alumni, 1991.

[2] Kominfo, "Jumlah Pengguna Internet 2017 Meningkat, Kominfo Terus Lakukan Percepatan Pembangunan Broadband," 2 February 2018. [Online]. Available: https://kominfo.go.id/index.php/content/detail/12640/siaran-pers-no-53hmkominfo022018-tentang-jumlah-pengguna-internet-2017-meningkat-kominfo-terus-lakukan-percepatan-pembangunan-broadband/0/siaran_pers. [Accessed 1 August 2018].

[3] A. Indriani, "Klasifikasi Data Forum Dengan Menggunakan Metode Naive Bayes Classifier," *Seminar Nasional Aplikasi Teknologi Informasi (SNATI)*, pp. G5-G10, 2014.

[4] R. Feldman and J. Sanger, Text Mining Handbok: Advanced Approaches in Analyzing Unstructed Data, New York: Cambridge University Press, 2007.

[5] W. F. Mahmudy, "Klasifikasi Artikel Berita Secara Otomatis Menggunakan Metode Naive Bayes Classifier Yang Dimodifikasi," pp. 1-10, 2014.

[6] A. Hamzah, "Klasifikasi Teks Dengan Naive Bayes Classifier (NBC) Untuk Pengelompokkan Teks Berita Dan Abstract Akademis," *Prosiding Seminar Nasional Aplikasi Sains & Teknologi (SNAST)* , 2012.

[7] E. Prasetyo, Data Mining: Konsep dan Aplikasi Menggunakan Matlab, Yogyakarta: Andi Publisher, 2013.

[8] R. Mitchell, Web Scraping With Java, Birmingham: Packt Publishing Ltd, 2013.