yKLASIFIKASI BERITA BERBAHASA INDONESIA BERBASIS *MULTI-WORD* MENGGUNAKAN *METODE MULTINOMIAL NAIVE BAYES* (MNB)

SKRIPSI

Oleh:

MUHAMAD ARIEF YAENUDIN

0710963023-96

PROGRAM STUDI ILMU KOMPUTER
JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS BRAWIJAYA
MALANG
2012


UNIVERSITAS BRAWIJAYA

KLASIFIKASI BERITA BERBAHASA INDONESIA BERBASIS *MULTI-WORD* MENGGUNAKAN *METODE MULTINOMIAL NAIVE BAYES* (MNB)

SKRIPSI

Sebagai salah satu syarat untuk memperoleh gelar
Sarjana Komputer dalam bidang Ilmu Komputer

Oleh:

MUHAMAD ARIEF YAENUDIN

0710963023-96

PROGRAM STUDI ILMU KOMPUTER
JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS BRAWIJAYA
MALANG
2012


UNIVERSITAS BRAWIJAYA

LEMBAR PENGESAHAN SKRIPSI

KLASIFIKASI BERITA BERBAHASA INDONESIA BERBASIS *MULTI-WORD* MENGGUNAKAN *METODE MULTINOMIAL NAIVE BAYES* (MNB)

Oleh :

MUHAMAD ARIEF YAENUDIN
0710963023 – 96

Setelah dipertahankan di depan Majelis Penguji
Pada tanggal 2 April 2012
dan dinyatakan memenuhi syarat untuk memperoleh gelar Sarjana dalam bidang Ilmu Komputer

Pembimbing I	Pembimbing II
Drs. Achmad Ridok, M.Kom	**Yusi Tyroni M., S.Kom, MS**
NIP. 19680825 199403 1 002	**NIP. 19800228 200604 1 001**

Mengetahui,
Ketua Jurusan Matematika Fakultas MIPA
Universitas Brawijaya Malang

Dr. Abdul Rouf Alghofari, M.Sc
NIP. 19670907199203 1 001


UNIVERSITAS BRAWIJAYA

LEMBAR PERNYATAAN

Saya yang bertanda tangan di bawah ini :

Nama	: **Muhamad Arief Yaenudin**
NIM	: **0710963023 - 96**
Jurusan	: **Matematika**
Program Studi	: **Ilmu Komputer**
Penulis Skripsi Berjudul	: **Klasifikasi Berita Berbahasa Indonesia Berbasis *Multi-word* Menggunakan Metode Multinomial Naive Bayes**

Dengan ini menyatakan bahwa :

1. **Isi dari Skripsi yang saya buat adalah benar – benar karya sendiri dan tidak menjiplak karya orang lain, selain nama – nama yang termaktub di isi dan tertulis di daftar pustaka dalam Skripsi ini.**
2. **Apabila di kemudian hari ternyata Skripsi yang saya tulis terbukti hasil jiplakan, maka saya akan bersedia menanggung segala resiko yang akan saya terima.**

Demikian pernyataan ini dibuat dengan segala kesadaran.

Malang, Juni 2012
Yang menyatakan,

Muhamad Arief Yaenudin
NIM. 0710963023

KLASIFIKASI BERITA BERBAHASA INDONESIA BERBASIS *MULTI-WORD* MENGGUNAKAN METODE MULTINOMIAL NAIVE BAYES (MNB)

ABSTRAK

Berita sudah menjadi kebutuhan masyarakat Indonesia sehari–hari. Berita yang disajikan dalam bentuk media *online* berupa dokumen yang jumlahnya sangat banyak dan sulit sekali jika dokumen – dokumen berita tersebut diklasifikasikan secara manual

Penelitian ini membahas mengenai penerapan klasifikasi berita berbahasa Indonesia berbasis *multi-word* menggunakan metode Multinomial Naive Bayes(MNB), dimana dokumen yang digunakan berasal dari situs media *online* surat kabar Kompas.com. Tahapan-tahapan yang dilakukan dalam sistem ini adalah, pertama dilakukan proses *parsing* kalimat yaitu memotong suatu dokumen menjadi kalimat-kalimat penyusunnya, tahap kedua yaitu melakukan proses *Preprocessing* yang memiliki sub proses *case folding* yaitu mengubah semua huruf menjadi huruf kecil, *tokenizing* yaitu proses penguraian kata, *filtering* yaitu mengambil kata-kata yang penting/relevan dan penghilangan *stopword*, *stemming* mereduksi kata ke bentuk dasarnya, tahap ketiga ekstraksi *multi-word* yaitu menemukan *multi-word* dari setiap dokumen , tahap keempat yaitu melakukan perhitungan frekuensi dari masing-masing *multi-word*, dan tahap terakhir yaitu klasifikasi menggunakan metode *MNB* (*Multinomial Naïve Bayes*). Hasil pengujian dan evaluasi menunjukkan bahwa sistem ini menghasilkan nilai rata-rata *precision* sebesar 0.840, rata-rata *recall* sebesar 0.7 dan rata-rata F_1 *measure* sebesar 0.705714. Evaluasi dilakukan untuk mengetahui pengaruh jumlah data latih terhadap efektifitas dari sistem.

INDONESIAN NEWS CLASSIFICATION WITH MULTI-WORD BASED USING MULTINOMIAL NAIVE BAYES (MNB) METHOD

ABSTRACT

News has become Indonesian society needs in their daily life. News is presented in the form of documents online media is a large scale of documents and very difficult to be clasified manually.

This research discusses about the application of the indonesian news classification with multi-word based using Multinomial Naive Bayes(MNB) method, where the document that is used from the online media sites Kompas.com. There are some steps in the system, first step is parsing sentences which is cutting a document into sentences. Second steps is Preprocessing which has sub process Case Folding that changes all letters to lowercase, Tokenizing is parsing the word ,Filtering is to take the unique words and stopword removal, Stemming is reduce the word to its basic form. Third step is extraction multi-word that is find the multi-word of every document. Fourth step is calculating of the frequency of each multi-word. The last step is classification using MNB (Multinomial Naïve Bayes) method. Test and evaluation result show that this system produces an average precision 0.840, average recall 0.7 and average F_1 measure 0.705714. The evaluation used to determine the effect of training data on the effectiveness of the system.

KATA PENGANTAR

Alhamdulillah, Puji syukur penulis ucapkan kepada Allah SWT karena dengan limpahan rahmat, karunia, hidayah dan petunjuk-Nya, penulis dapat menyelesaikan skripsi yang berjudul **"Klasifikasi Berita Berbahasa Indonesia Berbasis Mult-word Menggunakan Metode *Multinomial Naive Bayes* (MNB)"** yang mana skripsi ini merupakan salah satu syarat kelulusan di program studi Ilmu Komputer Universitas Brawijaya.

Terselesaikannya skripsi ini tidak hanya hasil dari penulis semata. Dalam penyusunan skripsi ini, penulis juga banyak diberi bantuan, saran, motivasi, inspirasi, penghiburan dan doa dari berbagai pihak. Oleh karena itu penulis ingin mengucapkan terima kasih yang sebesar – besarnya kepada :

1. Drs. Achmad Ridok, M.Kom, selaku pembimbing utama yang telah meluangkan waktu memberikan pengarahan dan bimbingan kepada penulis
2. Yusi Tyroni.,SKom, MS, selaku pembimbing pendamping yang telah meluangkan waktu memberikan pengarahan dan bimbingan kepada penulis
3. Dr. Abdul Rouf Alghofari,M.Sc, selaku Ketua Jurusan Matematika Fakultas MIPA Universitas Brawijaya Malang.
4. Drs. Mardji,MT, selaku Ketua Program Studi Ilmu Komputer Jurusan Matematika Fakultas MIPA Universitas Brawijaya Malang.
5. Kedua orang tua, dan keluarga besar penulis yang tidak pernah berhenti memberikan do‟a, semangat dan dukungan kepada penulis.
6. Seluruh Bapak dan Ibu Dosen, khususnya Dosen Program Studi Ilmu Komputer yang telah memberikan ilmu – ilmu yang bermanfaat dengan sabar dan ikhlas kepada penulis
7. Aprilia Zuliharti, Ahmad Azwar Anas, Mas dedi, Sari, Ade Asti, Jimmy Yoedi, Agung, Soni, Ivan, Diks, Isa, Madya, Hugi, Susan serta seluruh teman – teman Ilmu Komputer 2007 yang telah berjuang bersama terutama Kelas A yang selama di bangku perkuliahan selalu bersama – sama dikala suka maupun duka.

8. Kakak – kakak serta adik kelas dari program studi Ilmu Komputer yang telah memberikan saran dan motivasi kepada penulis.
9. Dan semua pihak yang terlibat baik secara langsung maupun tidak langsung yang tidak dapat disebutkan satu per satu. Terima kasih atas semua bantuan yang telah diberikan.

Semoga penulisan skripsi ini bermanfaat. Penulis menyadari bahwa penulisan skripsi ini masih jauh dari sempurna dan memiliki banyak kekurangan. Oleh karena itu penulis mengharapkan saran dan kritik yang membangun dari pembaca. Dan semoga skripsi ini bermanfaat bagi pembaca. Penulis dapat dihubungi melalui email arief.feira.88@gmail.com.

Malang, Juni 2012

Penulis

DAFTAR ISI

DAFTAR GAMBAR


UNIVERSITAS BRAWIJAYA

DAFTAR TABEL

DAFTAR PERSAMAAN

DAFTAR *SOURCE CODE*

BAB I
PENDAHULUAN

1.1. Latar Belakang

Berita adalah informasi baru atau informasi mengenai sesuatu yang sedang terjadi, disajikan lewat bentuk cetak, siaran, internet, atau dari mulut ke mulut (William, 2008).

Berita merupakan kebutuhan penting masyarakat sehari–hari. Berita memberikan informasi kepada masyarakat berupa peristiwa/kejadian apa saja yang sedang terjadi pada saat itu. Perkembangan berita meningkat seiring berjalannya waktu. Berita disampaikan melalui beberapa media, antara lain media cetak, televisi, radio dan yang paling modern adalah berita dapat dinikmati secara *online* dengan mengakses situs yang memuat berita *online*. Berita yang disajikan dalam bentuk media *online* berupa dokumen yang jumlahnya sangat banyak dan sulit sekali jika dokumen–dokumen berita tersebut diklasifikasikan secara manual, sehingga kebutuhan akan proses pengklasifikasian berita secara otomatis, yaitu penggolongan suatu berita ke dalam suatu kategori diperlukan untuk mempermudah pencarian berita mengenai suatu kejadian tertentu.

Text mining merupakan salah satu cara yang dapat mengatasi permasalahan tersebut. Pada *text mining* terdapat dua teknik pembelajaran, yaitu *unsupervised learning* dan *supervised learning*. *Clustering* adalah contoh dari *unsupervised learning*, dimana sekelompok data langsung dikelompokkan berdasarkan tingkat kemiripannya tanpa dilakukan supervisi. Sedangkan Klasifikasi merupakan bentuk dari *supervised learning* yang merupakan salah satu teknik dalam pembelajaran mesin untuk membentuk model yang merupakan fungsi dari data latihan *(training set)*. Pada *supervised learning*, data *training* yang digunakan terdiri dari beberapa pasangan nilai-nilai masukan dan nilai keluaran (nilai dari atribut tujuan). Model yang terbentuk dari data *training* digunakan sebagai dasar pengetahuan untuk mengklasifikasikan data-data yang baru *(testing set)* (Even dan Zohar, 2002).

Terdapat beberapa metode pengklasifikasian, diantaranya *neural network, decision tree, rocchio classifiers, single pass clustering, naïve bayes classifier,* dan *Support Vector Machine.*

Pada penelitian ini, digunakan suatu metode lain dalam menyelesaikan permasalahan pengklasifikasian dokumen berita berbahasa Indonesia yaitu dengan pendekatan algoritma *Multinomial Naïve Bayes (MNB)*. Metode *MNB* merupakan salah satu variasi lain dari *naïve bayes* yang merupakan algoritma yang menerapkan metode *probabilistic learning method*. *Naive Bayes* sering digunakan untuk mengatasi permasalahan klasifikasi teks karena teknik komputasinya sangat efisien dan mudah diimplementasikan (Kibriya dan Geoffry, 2004).

Metode *Multinomial Naïve Bayes (MNB)* pernah digunakan pada penelitian sebelumnya, yaitu oleh Ni'am Shofi Nurdwianto, 2007 dalam permasalahan yang sama yaitu klasifikasi berita berbahasa indonesia, namun pada penelitian sebelumnya *term* yang digunakan adalah satu kata / *single-word* sedangkan dalam penelitian ini *term* yang digunakan adalah *multi-word*. Selain perbedaan *term*, dalam penelitian ini juga terdapat proses ekstraksi *multi-word* dimana tidak dilakukan pada penelitian sebelumnya.

Pada metode *naïve bayes* kemunculan *term* pada dokumen latih diperhitungkan sedangkan pada metode *MNB* selain term pada dokumen latih, frekuensi kemunculan *term* pada dokumen uji juga diperhitungkan. Sehingga dapat dokatakan model *MNB* dianggap sebagai model klasifikasi yang lebih akurat untuk *data set* yang mempunyai variasi besar pada panjang dokumen (McCallum dan Nigam, 1998).

Penelitian ini menggunakan fitur *multi-word* sebagai acuan untuk melakukan klasifikasi dalam metode *Multinomial Naïve Bayes (MNB)*. Frasa/ *Multi-word* adalah gabungan dua kata atau lebih yang bersifat nonpredikatif (www.bahasakita.com/2011/05/10/kata-frasa-klausa-dan- kalimat/). Dalam sumber lain, menurut kamus bahasa Indonesia, frasa adalah sebuah istilah satuan linguistik yang lebih besar dari kata dan lebih kecil dari klausa dan kalimat (Sari Ernawati dkk, 2009). Fitur *multi-word* adalah metode baru yang praktis untuk pemilihan fitur dalam kategorisasi dokumen. *Multi-word* memiliki banyak potensi dalam rekayasa bahasa (*language engineering*). Misalnya, dalam *natural language generation* (generasi bahasa alami) oleh komputer, *multi-word* yang digunakan untuk membuat output suara komputer seperti yang diucapkan manusia secara alami. Dalam aplikasi praktis dari pengenalan suara dan pengenalan

karakter optik, *multi-word* dapat digunakan untuk solusi dalam kesamaran/ketidakjelasan karakter yang dapat membangun model bahasa statistik untuk prediksi karakter. *Multi-word* juga dapat digunakan dalam parsing sintaks, komputasi leksikografi (Wen Zhang, 2008).

Penelitian ini menggunakan strategi *Decomposition* untuk menentukan *multi-word* yang akan digunakan. Strategi *Decomposition* adalah strategi untuk mengambil multi-word dengan frekuansi kata terpendek didalam multi-word, artinya, *multi-word* yang digunakan merupakan gabungan kata dengan panjang maksimal dua kata.

Pemilihan fitur dengan menggunakan *multi-word* dikarenakan memiliki beberapa keunggulan. Keunggulan representasi *multi-word* mencakup setidaknya tiga aspek. Pertama, memiliki dimensi lebih rendah daripada kata individu/*single-word* tetapi kinerja dapat diterima dengan baik, dengan menggunakan fitur *multi-word* dapat mencapai akurasi ketepatan sampai 0,8673. (87%). Kedua, *multi-word* mudah untuk melakukan proses pembelajaran/*learning* dari dokumen-dokumen oleh korpus pembelajaran tanpa dukungan dari tesaurus, kamus atau ontologi. Ketiga, *multi-word* lebih semantik/bermakna karena merupakan unit yang memiliki nilai "pembeda" lebih besar dari kata individu/*single-word* (Taketoshi Yoshida, 2008).

Dalam penelitian ini akan dilakukan proses *stemming* yaitu menghilangkan kata imbuhan pada suatu kata, hal ini bertujuan untuk mendapatkan kata dasar. *Stemming yang* digunakan dalam penelitian ini menggunakan algoritma Nazief-Andriani. Algoritma *stemming* Nazief-Andriani mempunyai kebenaran *stemming* sekitar 93% untuk dokumen berbahasa indonesia. (Asian,William dan Tahaghoghi, 2005).

Berdasarkan latar belakang yang telah dipaparkan, maka judul yang diambil dalam skripsi ini "**KLASIFIKASI BERITA BERBAHASA INDONESIA BERBASIS *MULTI-WORD* MENGGUNAKAN *METODE MULTINOMIAL NAIVE BAYES* (MNB)**".

1.2. Rumusan Masalah

Rumusan masalah dari penulisan skripsi ini adalah:

1. Bagaimana penerapan pengklasifikasian berita berbahasa Indonesia berbasis *multi-word* menggunakan metode *Multinomial Naïve Bayes (MNB).*
2. Mengukur hasil akurasi sistem dari proses pengklasifikasian berita berbahasa Indonesia berbasis *multi-word* menggunakan metode *Multinomial Naïve Bayes (MNB).*

1.3. Batasan Masalah

Batasan masalah pada penulisan skripsi ini adalah :

1. Klasifikasi hanya pada dokumen berita berbahasa Indonesia.
2. Dokumen berita berbahasa indonesia yang akan digunakan dalam skripsi ini berupa dokumen dalam format *file text* berekstensi txt *(*.txt*) dimana berita yang diolah hanya isi berita.
3. Pada proses *stemming* tidak memperhitungkan adanya infiks (sisipan). Proses *stemming* yang dibangun hanya melakukan penghilangan prefiks dan sufiks.
4. Kategori berita yang digunakan terdiri dari 4 kategori, yaitu bisnis, edukasi, olahraga dan sains.
5. Dokumen berita berbahasa indonesia yang digunakan dalam skripsi ini hanya bersumber dari http://www.kompas.com.
6. *Multi-word* yang digunakan sebagai fitur hanya merupakan *multi-word* bebas yaitu gabungan dua kata tanpa terikat dengan aturan *multi-word* sintaksis (frasa) dalam bahasa Indonesia.
7. Tidak dilakukan perbandingan dengan sistem pengelompokkan berita sebelumnya.
8. Metode *parsing* kalimat yang digunakan adalah metode sederhana yaitu mengidentifikasi adanya tanda baca akhir suatu kalimat tanpa memperhitungkan kondisi kalimat.
9. Perhitungan akurasi dalam evaluasi menggunakan metode *precision*, *recall* dan *F1-measure.*
10. Proses *stemming* menggunakan algoritma algoritma Nazief-Andriani.
11. *Multi-word* yang digunakan dalam sistem ini dibatasi hanya dua kata.

1.4. Tujuan

Tujuan dari penulisan skripsi ini adalah:

1. Menerapkan metode *Multinomial Naïve Bayes (MNB)* untuk klasifikasi dokumen berita berbahasa Indonesia berbasis *multi-word* dalam kategori yang sesuai dan telah ditentukan.
2. Mengetahui akurasi kinerja sistem klasifikasi dokumen berita berbahasa Indonesia berbasis *multi-word* menggunakan metode *Multinomial Naïve Bayes (MNB).*

1.5. Manfaat

Manfaat yang dapat diambil dari skripsi ini adalah untuk memudahkan proses pengklasifikasian kategori dokumen berita berbahasa indonesia secara otomatis dan diharapkan mempunyai nilai keakuratan yang baik, sehingga memudahkan pencarian berita sesuai dengan kategori tertentu.

1.6. Sistematika Penulisan

Untuk memberikan gambaran tentang skripsi ini, berikut disajikan secara garis besar pembahasan dari keseluruhan isi laporan skripsi untuk setiap bab, sebagai berikut :

1. **BAB I : PENDAHULUAN**
   Berisi latar belakang penulisan, permasalahan yang dihadapi, tujuan, batasan masalah, dan manfaat serta sistematika penulisan skripsi.

2. **BAB II : DASAR TEORI**
   Menguraikan teori tentang berita, *text mining*, dan Metode *Multinomial Naïve Bayes (MNB)* serta teori-teori yang berhubungan dengan penggunaan metode pengklasifikasian berita berbahasa indonesia dengan Metode *MNB.*

3. **BAB III : METODOLOGI DAN PERANCANGAN**
   Berisi metode-metode yang digunakan dalam membuat sistem yang mampu mengklasifikasikan kategori berita berbahasa Indonesia ke dalam suatu kategori dengan Metode *MNB*.

4. **BAB IV : HASIL DAN PEMBAHASAN**
   Berisi tentang penjelasan implementasi sistem dan hasil pengujian yang dilakukan dari penerapan pengklasifikasian kategori berita berbahasa Indonesia dengan Metode *MNB*.

5. **BAB V : PENUTUP**
   Bab ini berisi kesimpulan yang diperoleh dari hasil pengujian dan saran-saran untuk pengembangan lebih lanjut.

BAB II
TINJAUAN PUSTAKA

2.1. Pengertian Berita

Berita adalah hasil rekonstruksi tertulis dari realitas sosial yang terdapat dalam kehidupan. Itulah sebabnya ada orang yang beranggapan bahwa penulisan berita lebih merupakan pekerjaan merekonstruksikan realitas sosial ketimbang gambaran dari realitas itu sendiri. Yang disebut berita adalah laporan tentang sebuah peristiwa. Dengan perkataan lain, sebuah peristiwa tidak akan pernah menjadi berita bila peristiwa tersebut tidak dilaporkan (Basuki, 1983).

Dalam literatur lain, Menurut Kamus Besar Bahasa Indonesia tahun 2008, berita adalah cerita atau keterangan mengenai kejadian atau peristiwa yang hangat. Sedangkan menurut Wahyudi (2002), berita adalah laporan tentang peristiwa atau pendapat yang memiliki nilai penting, menarik bagi sebagian khalayak, masih baru dan dipublikasikan melalui media massa periodik.

Secara umum, unsur-unsur berita yang selalu ada pada sebuah berita (Basuki, 1983) yaitu :

1. ***Headline*** **(Judul Berita)**
   Sering juga dilengkapi dengan anak judul. Ia berguna untuk menolong pembaca agar segera mengetahui peristiwa yang akan diberitakan, dan menonjolkan satu berita dengan dukungan teknik grafika.
2. ***Deadline***
   Pada bagian ini terdiri atas nama media massa, tempat kejadian dan tanggal kejadian. Tujuannya adalah untuk menunjukkan tempat kejadian dan inisial media.
3. ***Lead*** **(Teras Berita)**
   Biasanya ditulis pada paragraph pertama sebuah berita. Ia merupakan unsur yang paling penting dari sebuah berita, yang menentukan apakah isi berita akan dibaca atau tidak. Ia merupakan sari pati sebuah berita, yang melukiskan seluruh berita secara singkat.

4. ***Body* (Tubuh Berita)**
   Isinya menceritakan peristiwa yang dilaporkan dengan bahasa yang singkat, padat, dan jelas. Dengan demikian *body* merupakan perkembangan berita.

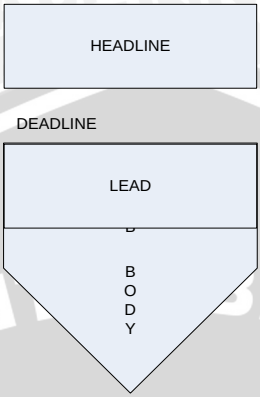
Di dalam suatu berita juga terdapat unsur-unsur lain. Unsur-unsur tersebut sering dikenal dengan *5W 1H* , *5W 1H* yaitu (Budiman, 2005):

a. *What* - apa yang terjadi di dalam suatu peristiwa?
b. *Who* - siapa yang terlibat di dalamnya?
c. *Where* - di mana terjadinya peristiwa itu?
d. *When* - kapan terjadinya?
e. *Why* - mengapa peristiwa itu terjadi?
f. *How* - bagaimana terjadinya?

2.1.1.Struktur Berita

Struktur berita sangat ditentukan oleh format berita yang akan ditulis. Struktur berita langsung berbeda dengan berita ringan (*straight* news) dan berita kisah. Tetapi, untuk berita langsung struktur yang lazim hanya satu, yaitu piramida terbalik (Basuki, 1983).

Lead menunjukkan bagian permulaan berita yang paling penting. Sedangkan piramida terbalik menunjukkan bagian yang penting dari sebuah berita pada bagian awal dan makin ke bawah makin kurang penting. Dengan perkataan lain, seiring dengan menyempitkan piramida terbalik, berkurang pula arti penting beritanya. Struktur seperti ini, di samping memudahkan mengenali inti berita, juga memudahkan pemotongan bagian yang tidak mungkin termuat. Struktur berita dengan piramida terbalik dapat dilihat pada Gambar 2.1.



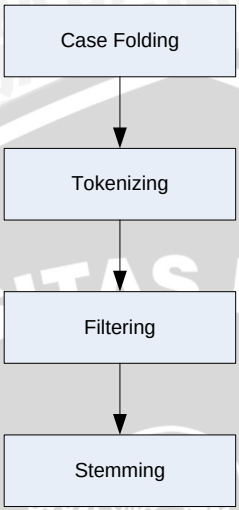
Gambar 2. 1 Struktur Berita

(Sumber: Ditjen Pendidikan Tinggi Dep P dan K, 1978)

Struktur-struktur berita tersebut bisa dipandang sebagai kerangka berita, yang akan diisi dengan fakta. Dalam mengisi kerangka berita, satu hal yang perlu diperhatikan adalah keterkaitan ide yang dikandung satu alinea dengan ide yang dikandung alinea berikutnya. Kalau keterkaitan itu tidak ada, maka ceritanya akan tersendat-sendat, tidak mengalir. Pengalaman menunjukkan, hanya berita yang terasa mengalir saja yang disenangi oleh khalayak.

2.2. Text Preprocessing

Struktur data yang baik dapat memudahkan proses komputerisasi secara otomatis. Pada *text mining*, informasi yang akan digali berisi informasi-informasi yang strukturnya tidak beraturan. Oleh karena itu, diperlukan proses pengubahan bentuk menjadi data yang terstruktur sesuai kebutuhannya untuk proses dalam *data mining*, yang biasanya akan menjadi nilai-nilai numerik. Proses ini sering disebut *Text Preprocessing*. Setelah data menjadi data terstruktur dan berupa nilai numerik maka data dapat dijadikan sebagai sumber data yang dapat diolah lebih lanjut. Untuk dapat memproses suatu dokumen teks diperlukan beberapa tahapan yang dapat dilihat pada Gambar 2.2



Gambar 2. 2 *Text Processing*

2.2.1.*Case Folding*

Case folding yaitu pengubahan karakter huruf menjadi huruf kecil (Garcia, 2005). Hanya huruf 'a' sampai 'z' yang diterima, sedangkan karakter selain huruf dihilangkan dan dianggap sebagai *delimiter*, yaitu karakter dasar yang sudah tidak dapat diturunkan lagi.

2.2.2.*Tokenizing*

Tokenizing adalah proses untuk mengambil kata dan istilah sederhana dari sebuah dokumen (Baldi, 2003). Kata dan istilah sederhana itu berupa potongan-potongan kata tunggal yang menyusun suatu dokumen. Pada tahap ini, dilakukan pemotongan (*parsing*) terhadap kata-kata tunggal tersebut menjadi kumpulan *token.*

2.2.3.*Filtering*

Filtering adalah proses menentukan *term-term* apa saja yang akan digunakan untuk merepresentasikan dokumen. Selain untuk menggambarkan isi dokumen, term ini juga berguna untuk

membedakan dokumen yang satu dengan dokumen lainnya pada koleksi dokumen (Garcia, 2005). Proses ini dilakukan dengan mengambil kata-kata penting dari hasil *token* dan menghapus *stopwords*. *Stopwords* adalah kata kata yang tidak merefleksikan isi dokumen, contohnya "yang", "di", "dari", "oleh", dan sebagainya. Umumnya *stopwords* berupa kata seruan, kata hubung, kata sambung, kata ganti, dan kata tidak penting lainnya. Daftar *stopwords* yang digunakan diambil dari hasil penelitian yang dilakukan oleh Tala (2003) dan dapat dilihat pada Lampiran 1.

2.2.4.*Stemming*

Stemming adalah proses untuk mereduksi kata ke bentuk dasarnya (Garcia, 2005). Pada tahap ini dicari *root* kata dari tiap kata hasil *filtering*. Pada tahap ini dilakukan proses pengembalian berbagai bentukan kata ke dalam suatu representasi yang sama. Tahap ini kebanyakan dipakai untuk teks berbahasa inggris dan lebih sulit diterapkan pada teks berbahasa Indonesia. Hal ini dikarenakan bahasa Indonesia tidak memiliki rumus bentuk baku yang permanen.

2.3. *Stemming* Bahasa Indonesia

2.3.1. Struktur Morfologi Kata Bahasa Indonesia

Morfologi adalah bagian dari ilmu bahasa yang membicarakan atau yang mempelajari seluk beluk kata serta pengaruh perubahan-perubahan bentuk kata terhadap golongan dan arti kata, atau dengan kata lain dapat dikatakan bahwa morfologi mempelajari seluk-beluk bentuk kata serta fungsi perubahan-perubahan bentuk kata itu (Ramlan, 1995).

Morfologi kata bahasa Indonesia bisa terdiri dari struktur *infleksional* dan *derivasional*. *Infleksional* adalah struktur yang paling sederhana yang dinyatakan dalam penambahan sufiks dimana tidak mempengaruhi arti sebenarnya dari kata dasar yang dilekati (Tala, 2003). Sufiks *infleksional* dapat dibagi menjadi 2 jenis :

1. Sufiks *-lah, -kah, -pun, -tah*. Sufiks ini sebenarnya adalah partikel yang tidak mempunyai arti. Keberadaannya pada suatu kata adalah untuk penekanan. Contoh :

dia + kah → diakah
duduk + lah → duduklah

2. Sufiks *-ku, -mu, -nya*. Sufiks ini berfungsi sebagai kata ganti kepunyaan. Contoh :
tas + ku → tasku
buku + mu → bukumu

Sufiks-sufiks diatas dapat melekat pada kata dasar secara bersama-sama. Adapun aturan urutannya adalah sufiks pada jenis kedua selalu diletakkan sebelum sufiks jenis pertama. Sehingga struktur morfologi pada kata *infleksional* adalah :

Infleksional = (kata dasar +kata ganti) | (kata dasar + partikel) |
(kata dasar + kata ganti + partikel)

Penambahan sufiks *infleksional* tidak akan merubah bentuk dasar dari kata berimbuhan (Tala, 2003). Dengan kata lain, tidak ada penghilangan atau peleburan kata dasar pada kata berimbuhan. Kata dasar dapat ditentukan dengan mudah pada struktur *infleksional*.

Struktur *derivasional* dalam bahasa Indonesia terdiri dari prefiks, sufiks dan kombinasi dari keduanya. Prefiks yang sering dipakai adalah : *ber-*, *di-*, *ke-*, *meng-*, *peng-*, *per-*, *ter-*. Contoh penggunaan prefiks adalah :

ber	+	lari	→	berlari
di	+	makan	→	dimakan
ke	+	kasih	→	kekasih
meng	+	ambil	→	mengambil
peng	+	atur	→	pengatur
per	+	lebar	→	perlebar
ter	+	baca	→	terbaca

Beberapa prefiks seperti *ber-*, *meng-*, *peng-*, *per-*, *ter* mungkin akan berubah menjadi beberapa bentuk yang berbeda. Bentuk dari setiap prefiks bergantung pada karakter pertama dari kata dasar yang dilekatinya. Tidak seperti struktur *infleksional*, pada

struktur *derivasional* pengucapan kata mungkin berubah setelah adanya penambahan prefiks. Seperti contoh menyapu yang terdiri dari prefiks meng- dan kata dasar sapu. Prefiks meng- berubah menjadi meny- dan karakter pertama dari kata dasar mengalami peleburan.

Sufiks *derivasional* adalah *-i, -kan, -an* (Tala, 2003). Contoh penggunaan sufiks *derivasional* adalah :

gula	+	i	→	gulai
makan	+	an	→	makanan
sampai	+	kan	→	sampaikan

Berbeda dengan penggunaan prefiks, penambahan sufiks tidak akan mengubah bentuk dasar dari suatu kata. Seperti disebutkan sebelumnya, struktur *derivasional* juga terdiri dari konfiks, yaitu gabungan dari prefiks dan sufiks yang melekat secara bersama-sama pada suatu kata. Contoh :

per	+	main	+	an	→	permainan
ke	+	kalah	+	an	→	kekalahan
ber	+	jatuh	+	an	→	berjatuhan
meng	+	ambil	+	i	→	mengambili

Tidak semua prefiks dan sufiks dapat dikombinasikan menjadi sebuah konfiks. Ada beberapa kombinasi prefiks dan sufiks yang tidak diperbolehkan. Kombinasi tersebut ditunjukkan pada tabel 2.1.

Tabel 2. 1 Pasangan Konfiks yang tidak diperbolehkan

Prefiks	Sufiks
Ber	i
Di	an
Ke	i | kan
Meng	an
Peng	i | kan
Ter	an

Prefiks/konfiks dapat ditambahkan pada suatu kata yang telah terdapat konfiks/prefiks, yang menghasilkan struktur prefiks ganda. Seperti pada pembentukan sebuah konfiks, pada pembentukan prefiks ganda, tidak semua prefiks/konfiks dapat ditambahkan pada kata yang telah mendapatkan prefiks/konfiks. Ada beberapa aturan dalam urutan pembentukan prefiks ganda. Aturan-aturan tersebut ditunjukkan pada tabel 2.2.

Tabel 2. 2 Urutan prefiks ganda

Prefiks 1	Prefiks 2
Meng di ter ke	per ber

Struktur morfologi pada kata derivasional adalah :

Derivasional = (prefiks + kata dasar) | (kata dasar + sufiks) | (prefiks + kata dasar + sufiks) | (prefiks 1 + prefiks 2 + kata dasar) | (prefiks 1 + prefiks 2 + kata dasar + sufiks).

Struktur lain yang mungkin terjadi dalam morfologi bahasa Indonesia adalah penambahan sufiks *infleksional* pada struktur *derivasional*, yang dinamakan multiple sufiks. Sehingga dapat disimpulkan secara umum struktur morfologi kata bahasa Indonesia adalah :

Sturtur morfologi = [prefiks 1] + [prefiks 2] + kata dasar + [sufiks] + [kata ganti] + [partikel].

keterangan :
s[...] menunjukkan opsi/pilihan.

2.3.2.Proses *Stemming* Nazief-Andriani

Tahap-tahap yang dilakukan pada algoritma Bobby Nazief dan Mirna Andriani adalah sebagai berikut :

1. Mencari kata yang akan di-*stemming* dalam kamus. Jika ditemukan dalam kamus, dapat dikatakan bahwa kata tersebut merupakan kata dasar, dan algoritma berhenti.
2. *Inflection suffixes* (*-lah, -kah, -ku, -mu,* atau *–nya*) dihapus. Jika berhasil dan akhirannya adalah partikel (*-lah* atau *–kah*), langkah ini diulangi untuk menghapus *inflectional possesive pronoun suffixes* (*-ku, -mu, -nya*).
3. Hapus *Derivation Suffixes* (*-i, -an* atau *-kan*). Jika kata ditemukan di kamus, maka algoritma berhenti. Jika tidak maka ke langkah 3a:
   a. J ika *-an* telah dihapus dan huruf terakhir dari kata tersebut adalah *-k*, maka *-k* juga ikut dihapus. Jika kata tersebut ditemukan dalam kamus maka algoritma berhenti. Jika tidak ditemukan maka lakukan langkah 3b.
   b. Akhiran yang dihapus (*-i, -an,* atau *–kan*) dikembalikan, lanjut ke langkah 4.
4. Hapus *Derivation Prefix*. Jika pada langkah 3 ada sufiks yang dihapus maka pergi ke langkah 4a, jika tidak pergi ke langkah 4b. :
   a. Jika akhiran telah dihapus pada langkah 3, kemudian kombinasi awalan-akhiran yang tidak diijinkan telah dicek menggunakan daftar pada tabel 2.8. Jika dalam kamus ditemukan maka algoritma berhenti. Jika tidak pergi ke langkah selanjutnya.

Tabel 2. 3 Kombinasi awalan dan akhiran yang tidak diijinkan

Awalan	Akhiran yang tidak diijinkan
be-	-i
di-	-an
ke-	-i, -kan
me-	-an
se-	-i, -kan
te-	-an

b. Jika awalan saat ini sama dengan awalan yang sebelumnya, maka algoritma ini berhenti.
c. Jika tiga awalan sebelumnya telah dihapus, maka algoritma berhenti.
d. Tipe awalan dijelaskan pada langkah di bawah ini :
    i. Jika awalan *di*, *ke*-, atau *se*-, maka tipe awalan berturut-turut adalah *di*,*ke*, atau *se*.
    ii. Jika awalan adalah *te*- seperti yang terdapat pada tabel 2.9, awalan *be*-, *me*-, atau *pe*-, maka diperlukan proses tambahan untuk menentukan tipe awalannya.

Tabel 2. 4 Menentukan tipe awalan untuk kata berawalan te-

Karakter yang mengikuti				Tipe awalan
Set 1	Set 2	Set 3	Set 4	
"-r"-	"-r-"	-	-	tidak ada
"-r-"	Huruf vokal	-	-	ter-luluh
"-r"	Bukan (huruf vokal atau "-r")	"-er-"	Huruf vokal	ter
"-r"	Bukan (huruf vokal atau "-r")	"-er-"	Bukan huruf vokal	ter-
"-r"	Bukan (huruf vokal atau "-r")	Bukan "-er-"		ter
Bukan (huruf vokal atau "-r")	"-er-"			tidak ada
Bukan (huruf vokal atau "-r")	"-er-"	Bukan huruf vokal		te

    iii. Jika dua karakter pertama bukan *di*-, *ke*-, *se*-, *te*-, *be*-, *me*- atau *pe*- maka algoritma berhenti.
e. Jika tipe awalan adalah "none", maka algoritma berhenti. Jika tipe awalan tidak "none", maka tipe awalan ada di tabel 2.10 awalan yang ditemukan dihapus.

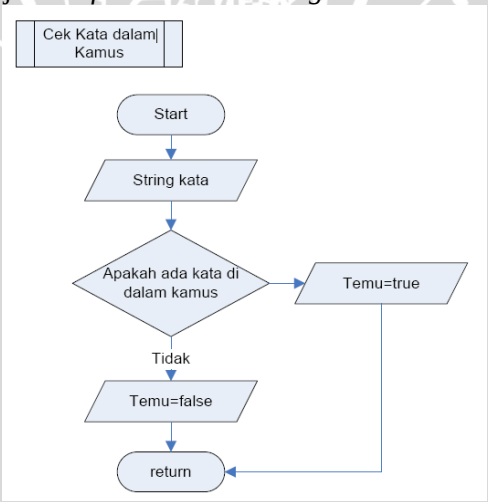
Tabel 2. 5 Menentukan awalan dari tipe awalan

Tipe awalan	Awalan yang harus dihapus
Di	di-
Ke	Ke-
Se	Se-
Te	Te-
Ter	Ter-
Ter-luluh	Ter-

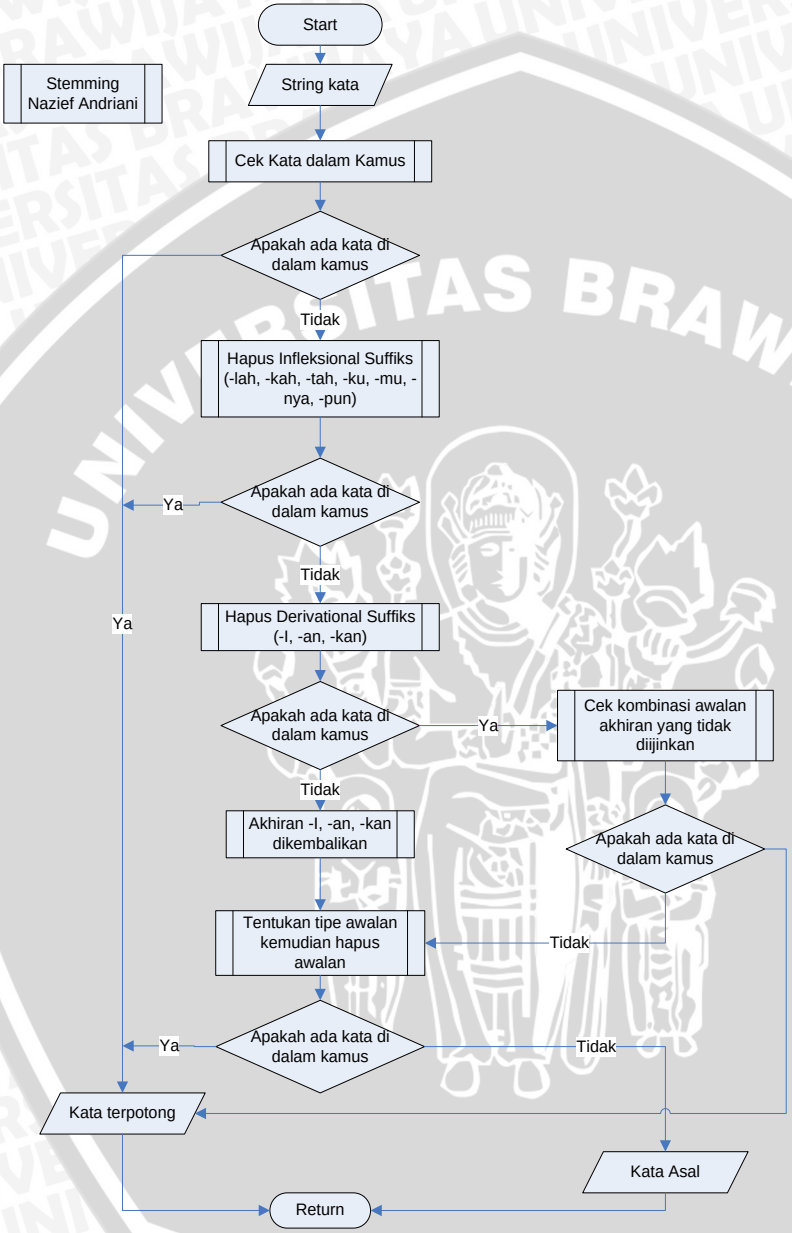
f. Jika kata dasar tidak ditemukan, langkah 4 dilakukan untuk menghapus awalan berikutnya secara berulang.
g. Melakukan *recoding.* Langkah ini tergantung pada tipe awalan, dan dapat mengakibatkan awalan yang berbeda.

2. Jika semua langkah sudah selesai tetapi tidak berhasil, maka kata awal dianggap kata dasar (Jelita Asian, Hugh. E Williams & S.M.M Tahaghoghi, 2005).

Gambar 2.3 menunjukkan *flowchart* cek kamus, sedangkan gambar 2.7 menunjukkan *flowchart stemming* Nazief-Andriani.



Gambar 2. 3 *Flowchart* cek kamus (Sari, 2011)



Gambar 2. 4 *Flowchart stemming* Nazief-Andriani (Sari, 2011)

2.4. Klasifikasi Teks

Klasifikasi adalah proses pengelompokan dokumen kedalam kelas berbeda, dalam tahapannya tiap dokumen *d* menunjuk pada satu kelas tertentu maka dibutuhkan proses untuk menggali informasi dari dokumen tersebut. Dokumen tersebut harus dapat merepresentasikan dari kelasnya sehingga tiap kata yang muncul dalam dokumen mempunyai nilai. Klasifikasi memiliki dua proses yaitu membangun model klasifikasi dari sekumpulan kelas data yang sudah didefinisikan sebelumnya (*training data set*) dan menggunakan model tersebut untuk klasifikasi tes data serta mengukur akurasi model. Model klasifikasi dapat disajikan dalam berbagai macam model klasifikasi seperti *decision trees*, *Bayesian classification*, *K-Nearest Neighbourhood classifier*, *neural network classification* (*IF-THEN*) *rule*, *Rocchio Classifier*. Klasifikasi dapat dimanfaatkan dalam berbagai aplikasi seperti *diagnose medis*, *selective marketing*, pengajuan kredit perbankan, *news categorization*, *email filtering*, dan lainnya (Rachli, 2007).

2.5. *Machine Learning* Untuk Klasifikasi

Dalam bidang klasifikasi dokumen, *machine learning* dapat dilakukan dengan dua cara yaitu dengan menggunakan pendekatan *supervised learning* dan *unsupervised learning*. Pada *unsupervised learning*, pengelompokan kelompok tidak melalui proses pengenalan ciri-ciri suatu topik dokumen (Turney, 2002).

Pada Sebastiani (2002), salah satu metode yang digunakan untuk melakukan klasifikasi dokumen adalah Naïve Bayes. Metode ini mencapai nilai akurasi tertinggi yaitu 81,5% saat melakukan klasifikasi ke dalam 10 topik. Penelitian lain dilakukan pada (Nigam, Laverty,& McCallum, 1999), dua metode yang digunakan adalah Naïve Bayes dan Maximum Entropy. Dengan menggunakan metode Naïve Bayes, hasil akurasi tertinggi yang dihasilkan adalah 86,9%. Sementara dengan menggunakan Maximum Entropy nilai akurasi yang dihasilkan dapat mencapai 92.18%.

2.5.1.*Naive Bayes Classifier*

Naïve bayes classifier merupakan salah satu metode *machine learning* yang dapat digunakan untuk klasifikasi suatu dokumen.Teorema *Bayes* berawal dari persamaan 2.1.

$$P(A \mid B) = \frac{P(B \cap A)}{P(B)} \qquad (2.1)$$

dimana P(A | B) artinya peluang A jika diketahui keadaan B. Kemudian dari persamaan 2.1 kita mendapatkan persamaan 2.2.

$$P(B \cap A) = P(B \mid A) \cdot P(A) \qquad (2.2)$$

Sehingga didapatkan teorema *Bayes* pada persamaan 2.3.

$$P(A \mid B) = \frac{P(B \mid A) \cdot P(A)}{P(B)} \qquad (2.3)$$

Naïve bayes classifier termasuk ke dalam algoritma pembelajaran *bayes*. Algoritma pembelajaran *bayes* menghitung probabilitas eksplisit untuk menggambarkan hipotesa yang dicari. Suatu data pada *naïve bayes classifier* direpresentasikan dengan konjungsi dari nilai-nilai atribut dan sebuah fungsi target f(x) yang dapat memiliki nilai apapun dari himpunan set domain V (Dumais dan Mehran, 1998). Sistem dilatih menggunakan data latih lengkap berupa pasangan nilai-nilai atribut dan nilai target kemudian sistem akan diberikan sebuah data baru dalam bentuk $<a_1, a_2, a_3, \ldots, a_n>$ dan sistem diberi tugas untuk menebak nilai fungsi target dari data tersebut (Mitchell, 1997).

Naïve bayes classifier member nilai target kepada data baru menggunakan nilai V_{MAP}, yaitu nilai kemungkinan tertinggi dari seluruh anggota himpunan set domain V dirumuskan pada persamaan 2.4.

$$V_{MAP} = \operatorname{argmax}_{v_j \in V} P(v_j \mid a_1, a_2, a_3, \ldots, a_n) \qquad (2.4)$$

Teorema *Bayes* kemudian digunakan untuk menulis ulang persamaan 2.4 menjadi persamaan 2.5.

$$V_{MAP} = \text{argmax}_{vj \in V} \frac{P(a_1, a_2, a_3, ..., a_n | V_j) P(V_j)}{P(a_1, a_2, a_3, ..., a_n)} \quad (2.5)$$

Karena $P(a_1, a_2, a_3, ..., a_n)$ nilainya konstan untuk semua sehingga persamaan 2.5 dapat ditulis menjadi 2.6.

$$V_{MAP} = \text{argmax}_{vj \in V} P(a_1, a_2, a_3, ..., a_n | V_j) P(V_j) \quad (2.6)$$

Tingkat kesulitan menghitung $P(a_1, a_2, a_3, ..., a_n | V_j) P(V_j)$ menjadi tinggi karena jumlah *term* $P(a_1, a_2, a_3, ..., a_n | V_j) P(V_j)$, bisa menjadi sangat besar. Ini disebabkan jumlah *term* tersebut sama dengan jumlah kombinasi posisi kata dikali dengan jumlah kategori. *Naïve bayes classifier* menyederhanakan hal ini dan bekerja dengan dasar asumsi bahwa atribut-atribut yang digunakan bersifat *conditionally independent* antara satu dan yang lainnya, dengan kata lain dalam setiap kategori, setiap term *independent* satu sama lain. Sehingga menjadi persamaan 2.7.

$$P(a_1, a_2, a_3, ..., a_n | V_j) = \prod_i P(a_1 | v_j) \quad (2.7)$$

Subsitusi persamaan 2.7 dengan persamaan 2.6 menjadi persamaan 2.8.

$$V_{NB} = \text{argmax}_{vj \in V} P(v_j) \prod_i P(a_1 | v_j) \quad (2.8)$$

V_{NB} adalah nilai probabilitas hasil perhitungan *naïve bayes classifier* untuk nilai fungsi target yang bersangkutan. Frekuensi kemunculan kata menjadi dasar perhitungan nilai dari $P(v_j)$ dan $P(a_1 | v_j)$. Himpunan set dari nilai-nilai probabilitas ini berkorespondensi dengan hipotesa yang ingin dipelajari. Hipotesa kemudian digunakan untuk mengklasifikasi data-data baru. Pada pengklasifikasian teks, perhitungan persamaan 2.7 dapat didefinisikan:

$$P(V_j) = \frac{docs_j}{|D|} \quad (2.9)$$

$$P(W_j \mid V_j) = \frac{n_k + 1}{n + |kata|} \qquad \textbf{(2. 10)}$$

Keterangan:

1. $Docs_j$: kumpulan dokumen yang memiliki kategori v_j .
2. |D| : jumlah dokumen yang digunakan dalam pelatihan (kumpulan data latih).
3. n : jumlah total kata yang terdapat di dalam kata tekstual yang memiliki nilai fungsi target yang sesuai.
4. n_k : jumlah kemunculan kata pada W_k semua data tekstual yang memiliki nilai fungsi target yang sesuai.
5. |kata| : jumlah kata yang berbeda yang muncul dalam seluruh data tekstual yang digunakan

Perbedaan antara *naïve bayes classifier* dengan metode pembelajaran lainnya terletak pada proses pembangunan hipotesa. Pada *naïve bayes classifier*, hipotesa langsung dibentuk tanpa proses pencarian (*searching*), hanya dengan menghitung frekuensi kemunculan suatu kata dalam data latih, sedangkan pada metode pembelajaran lainnya biasanya dilakukan pencarian hipotesa yang sesuai dari ruang hipotesa (Mitchell, 1997).

Ringkasan algoritma untuk *Naïve Bayes Clasifier* adalah sebagai berikut :

```
A. Proses pelatihan.
   Input adalah dokumen-dokumen contoh yang
   Telah diketahui kategorinya:
    1. |kata|
    2. Untuk setiap kategori vj lakukan:
       a. docsj
       b. Hitung P(vj) dengan persamaan 2.9
       c. Untuk setiap kata wk pada |kata|
          lakukan:
             i.    Hitung P(wk|vj ) dengan
                   persamaan 2.10
B. Proses klasifikasi.
   Input adalah dokumen yang belum diketahui
   kategorinya:
     1. Hasilkan VNB sesuai dengan persamaan
        2.8 dengan menggunakan P(vj) dan P(wk|vj )
        yang telah diperoleh dari pelatihan.
```

Algoritma *Naïve Bayes Classifier*

2.5.2.*Multinomial Naïve Bayes (MNB)*

Multinomial Naïve Bayes merupakan salah satu variasi lain dari metode *naïve bayes*. Model *MNB* mengambil frekuensi jumlah kata yang muncul pada sebuah dokumen. Dalam model *MNB* sebuah dokumen terdiri dari beberapa kejadian kata dan di asumsikan panjang dokumen tidak bergantung pada kelasnya. Dengan menggunakan asumsi Bayes yang sama bahwa kemungkinan tiap kejadian kata dalam sebuah dokumen adalah bebas tidak terpengaruh dengan konteks kata dan posisi kata dalam dokumen. Dalam metode *MNB* kemunculan setiap kata pada dokumen uji selalu diperhitungkan (McCallum dan Nigam, 1998).

Dimisalkan kategori suatu dokumen dengan lambang *C* dan *N* merupakan jumlah kata pada kategori yang sesuai. Kemudian *MNB* menempatkan dokumen uji t_i pada kategori yang mempunyai kemungkinan probabilitas tertinggi $\Pr(c \mid t_i)$ dengan menggunakan teorema *Bayes* (Kibriya dan Geoffry, 2004). Dirumuskan pada persamaan 2.11.

$$\mathbf{Pr(c \mid t_t) = \frac{Pr(c)\, Pr(t_i \mid c)}{Pr(t_i)}, c \in C} \qquad \textbf{(2. 11)}$$

Prior kategori $\Pr(c)$ dapat dihitung dengan membagi jumlah dokumen dari kategori c dengan total dokumen. $\Pr(t_i|c)$ adalah probabilitas dari pengambilan dokumen seperti t_i di kategori c dan dikalkulasikan pada persamaan 2.12.

$$\mathbf{Pr(t_i \mid c) = (\sum_n f_{ni})! \prod_n \frac{Pr(w_n \mid c)^{f_{ni}}}{f_{ni}!}} \qquad \textbf{(2. 12)}$$

Dimana f_{ni} adalah jumlah kata pada dokumen uji t_i dan $\Pr(w_n \mid c)$ merupakan probabilitas kata pada kategori c yang dapat dicari dengan persamaan 2.13.

$$\mathbf{Pr(w_n \mid c) = \frac{1+F_{nc}}{N+\sum_{x=1}^{N} F_{xc}}} \qquad \textbf{(2. 13)}$$

Dimana F_{nc} adalah jumlah kata x pada dokumen latih yang termasuk dalam kategori c. Normalisasi Pr(t_i) dapat dihitung dengan dengan persamaan 2.14.

$$\mathbf{Pr}(\ t_i\ ) = \sum_{k=1}^{|c|} \mathbf{Pr}\ (k)\ \mathbf{Pr}(t_i \mid k) \qquad \textbf{(2. 14)}$$

Komputasi $(\sum_n f_{ni})!$ dan $\prod_n f_{ni}!$ pada persamaan 2.12 dapat dihilangkan tanpa mempengaruhi hasil karena keduanya tidak tergantung pada kategori c dan persamaan 2.12 dapat ditulis kembali menjadi persamaan 2.15.

$$\mathbf{Pr}(t_i \mid c) = \alpha \prod_n \mathbf{Pr}(\ w_n \mid c)^{f_{ni}} \qquad \textbf{(2. 15)}$$

Perbedaan metode *Multinomial Naïve Bayes* dengan *Naïve Bayes* yaitu pada proses pembuatan model klasifikasi, kedua metode memiliki rumusan yang sama yaitu mengacu pada teorema *bayes* yang menghitung *posterior* dan *prior*. Pada *prior* kedua metode menggunakan perhitungan yang sama yaitu menghitung peluang semua dokumen yang berkelas tertentu terhadap semua kelas. Akan tetapi, perhitungan *posterior* yaitu peluang suatu term jika dikenakan kelas tertentu pada kedua metode memiliki perhitungan yang berbeda. Adapun algoritma menggunakan rumus-rumus yang telah diterangkan di atas dapat dilihat pada contoh algoritma berikut.

```
A. Proses pelatihan.
   Input adalah dokumen-dokumen contoh yang
   Telah diketahui kategorinya:
    1. |kata|
    2. Untuk setiap kategori c lakukan:
       a. docsj
       b. Hitung Pr(c) dengan persamaan 2.9
       c. Untuk setiap kata wk pada |kata|
          lakukan:
            i. Hitung P(wn|c) dengan persamaan
               2.13

B. Proses klasifikasi.
   Input adalah dokumen yang belum diketahui
   kategorinya:
    1. Untuk setiap kategori c hitung Pr(ti|c)
       dengan persamaan 2.15
```

```
2. Hitung total Pr(tᵢ) dengan persamaan
   2.14 dengan menggunakan Pr(tᵢ|c) yang
   Telah diperoleh dari semua kategori.
3. Hasilkan Pr(c|tᵢ) dengan persamaan 2.11
   Untuk dicari nilai yang paling besar.
```

Algoritma *Multinomial Naïve Bayes*

2.6. Definisi Frasa Dalam Bahasa Indonesia

Menurut kamus bahasa Indonesia, frasa atau frase adalah sebuah istilah linguistik. Lebih tepatnya, frasa merupakan satuan linguistik yang lebih besar dari kata dan lebih kecil dari klausa dan kalimat.

Dalam literatur lain dijelaskan bahwa frasa adalah satuan konstruksi yang terdiri dari dua kata atau lebih yang membentuk satu kesatuan (Keraf, 1984:138). Frasa juga didefinisikan sebagai satuan gramatikal yang berupa gabungan kata yang bersifat nonprediktif, atau lazim juga disebut gabungan kata yang mengisi salah satu fungsi sintaksis di dalam kalimat (Chaer, 1991:222). Frasa juga diartikan sebagai satuan linguistik yang secara potensial merupakan gabungan dua kata atau lebih, yang tidak mempunyai ciri-ciri klausa (Cook, 1971: 91; Elson and Pickett, 1969: 73). Menurut Prof. M. Ramlan, frasa adalah satuan gramatik yang terdiri atas satu kata atau lebih dan tidak melampaui batas fungsi atau jabatan (Ramlan, 2001:139). Artinya sebanyak apapun kata tersebut asal tidak melebihi jabatannya sebagai Subjek, predikat, objek, pelengkap, atau pun keterangan, maka masih bisa disebut frasa.

Contoh:

a. gedung sekolah itu
b. yang akan pergi
c. sedang membaca
d. sakitnya bukan main
e. besok lusa
f. di depan.

Jika contoh itu diletakkan dalam kalimat, kedudukannya tetap pada satu jabatan saja, misalnya

a. Gedung sekolah itu(S) luas(P).
b. Dia(S) yang akan pergi(P) besok(Ket).
c. Bapak(S) sedang membaca(P) koran sore(O).
d. Pukulan Budi(S) sakitnya bukan main(P).
e. Besok lusa(Ket) aku(S) kembali(P).
f. Bu guru(S) berdiri(P) di depan(Ket).

Jadi, walau terdiri dari dua kata atau lebih tetap tidak melebihi batas fungsi. Dalam penelitian ini, yang dimaksud dengan *multi-word* adalah frasa dengan urutan kata yang terdapat dalam sebuah kalimat pada dokumen. Dengan kata lain *multi-word* merupakan frasa bebas yaitu gabungan dua kata atau lebih tanpa terikat aturan frasa yang sebenarnya.

2.7. Representasi Teks Dalam *Multi-word*

Justeson mengusulkan Metode untuk ekstraksi *multi-word* berdasarkan struktur sintaksis (Justeson, 1995). Metode tersebut adalah strategi yang dikembangkan untuk menggunakan ekstraksi *multi-word* pada suatu dokumen di tingkat semantik yang berbeda.

2.7.1.Ekstraksi Multi-word (Multi-word extraction)

Secara umum, ada dua jenis metode yang dikembangkan untuk proses ekstraksi *multi-word*. Salah satunya adalah metode linguistik, yang memanfaatkan sifat struktural *multi-word* dalam kalimat untuk ekstraksi *multi-word* dari dokumen (Bourigault, 1992). Sebagai contoh, Smadja menggunakan *relative offset* dari kemunculan posisi dua kata pada semua dokumen korpus untuk menentukan apakah itu merupakan suatu *multi-word* (Smadja, 1993). Dan hasilnya menunjukkan bahwa metode itu bekerja dengan baik untuk *multi-word* tetap tetapi tidak dapat mengatasi dengan *multi-word* yang memiliki panjang yang berubah - ubah. Metode yang lainnya adalah metode statistik, berdasarkan korpus pembelajaran menggunakan MI (*Mutual Information*) untuk penemuan kemunculan pola suatu kata. Sebagai contoh, Zhang et al mengajukan sebuah metode yang didasarkan pada adaptasi MI

(*Mutual Information*) dan ketergantungan konteks untuk ekstraksi *multi-word* dari *Corpus* Bahasa Cina, dan mereka melaporkan bahwa metode mereka cukup efisien dan kuat untuk ekstraksi *multi-word* Bahasa Cina (Zhang, 2000). tetapi, metode mereka yang menentukan konteks ketergantungan dari sepasang kata, dan pengaturan parameter terlalu kompleks sehingga metode ini tidak menghasilkan kinerja yang kuat. Beberapa metode lain juga menggabungkan kedua *linguistic knowledge* dan perhitungan statistik untuk ekstraksi *multi-word* (Fahmi, 2005)

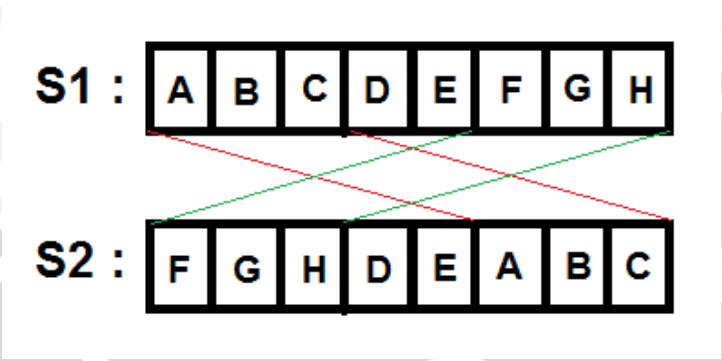
Metode ekstraksi *multi-word* yang digunakan dalam penelitian ini , Justeson dan Katz mengusulkan suatu strategi yaitu pengulangan kandidat *multi-word* karena pada umumnya *multi-word* yang bukan merupakan kandidat topik dari suatu dokumen tidak akan diulang lebih dari dua kali pada suatu dokumen. Selain itu, dari sudut pandang kategorisasi teks, jika frekuensi suatu *multi-word* terlalu kecil, maka *multi-word* tersebut tidak memiliki kekuatan diskriminatif (pembeda) untuk menentukan kategori dokumen. Untuk mengurangi kompleksitas komputasi tersebut , maka ditemukan sebuah solusi yaitu dengan mengekstrak pengulangan pola pada dua kalimat pada dokumen terlebih dahulu lalu meneruskan bagian dari kalimat tadi ke proses ekstraksi *pattern*. Sebagai contoh, jika kita memiliki dua kalimat sebagai berikut:

- *Standard oil co and bp north America inc said they plan to form a venture to manage the money market borrowing and investment activities of both companies.*

- *The venture will be called bp/standard financial trading and will be operated by standard oil under the oversight of a joint management committee.*

Dari dua kalimat di atas," *standard oil* " akan diekstrak sebagai *multi-word* untuk direpresentasikan pada dokumen di dalam *text collection*. Ide dasar untuk menemukan pola pengulangan dari dua kalimat adalah pencocokan string. Sebagai contoh, dengan asumsi kita kita memiliki dua kalimat yaitu S_1 {ABCDEFGH} dan

S_2 {FGHEDABC} dimana huruf kapital mewakili kata dalam setiap kalimat seperti yang ditunjukkan pada Gambar 2.2



Gambar 2. 5 Pengulangan pola yang sesuai pada dua dokumen

Setiap kata dalam kalimat S_2 akan digunakan untuk mencocokkan satu per satu kata pada kalimat S_1 dan pola yang sama dalam dua kalimat tersebut akan diambil sebagai *multi-word*.

2.7.2. *Decomposition Strategy*

Penelitian ini menggunakan strategi *Decomposition* untuk menentukan *multi-word* yang akan digunakan. Strategi *Decomposition* adalah strategi untuk mengambil multi-word dengan frekuensi kata berurutan terpendek didalam *multi-word*, artinya, *multi-word* yang digunakan merupakan gabungan kata dengan panjang maksimal dua kata. Misalnya," *U.S. agriculture department* "akan dihilangkan dari dan akan digantikan oleh" '*U.S. agriculture*" dan" *agriculture department* ". Setelah mendapatkan urutan kata menjadi menjadi *multi-word* dengan strategi ini, proses akan dilanjutkan dengan penghitungan frekuensi *multi-word* (TF) dari dokmen. Frekuensi adalah petunjuk penting untuk menentukan tingkat relevansi dari *multi-word* dengan topik dokumen, yaitu, kategori dokumen.

2.8. Evaluasi

Untuk mengevaluasi kesamaan diantara dokumen – dokumen dapat diukur berdasar *recall*, *precision* dan *F-measure*. *Recall* adalah tingkat keberhasilan mengenali suatu kelas yang harus dikenali. *Precision* adalah tingkat ketepatan hasil klasifikasi dari seluruh dokumen. *F-measure* merupakan nilai yang mewakili keseluruhan kinerja sistem dan merupakan penggabungan nilai *recall* dan *precision* dalam sebuah nilai (Destuardi dan Sumpeno, 2009).

Untuk mengevaluasi performa efektifitas dari sistem klasifikasi teks digunakan suatu standar yang disebut matrix confusion. *Matrix Confusio*n berisi informasi menegenai klasifikasi yang sebenarnya dan prediksi klasifikasi yang dilakukan oleh sistem (Hamilton, H dan Olive, W, 2003). Tabel 2.11 menunjukan *Matrix Confusion* (Lewis, 1995).

Tabel 2. 6 *Matrix Confusion*

Hasil klasifikasi dari sistem	Hasil klasifikasi dari ahli	
	YES	NO
YES	*True positive*	*False Positive*
NO	*False Negative*	*True Negative*

- *True positive* menunjukan bahwa dokumen yang termasuk dalam hasil klasifikasi oleh sistem memang merupakan anggota klasifikasi. Pengklasifikasi sebenarnya ya dan pengklasifikasi oleh sistem ya.
- *False Positive* menunjukkan bahwa dokumen yang termasuk dalam hasil klasifikasi oleh sistem ternyata bukan merupakan anggota klasifikasi. Pengklasifikasi sebenarnya bukan dan pengklasifikasi oleh sistem ya.
- *False Negative* menunjukkan bahwa dokumen yang tidak termasuk dalam hasil klasifikasi oleh sistem ternyata seharusnya merupakan anggota klasifikasi. Pengklasifikasi sebenarnya ya dan pengklasifikasi oleh sistem bukan.
- *True Negative* menunjukkan bahwa dokumen yang tidak termasuk dalam hasil klasifikasi sistem ternyata seharusnya bukan merupakan anggota klasifikasi. Pengklasifikasi sebenarnya bukan dan pengklasifikasi oleh sistem bukan.

$$Precision = \frac{true\ \ positive}{true\ \ positive\ + false\ \ positive} \quad \textbf{(2.16)}$$

$$Recall = \frac{true\ \ positive}{true\ \ positive\ + false\ \ negative} \quad \textbf{(2.17)}$$

$$F1\ Measure = \frac{2\ x\ recall\ \ x\ precision}{recall\ + precision} \quad \textbf{(2.18)}$$

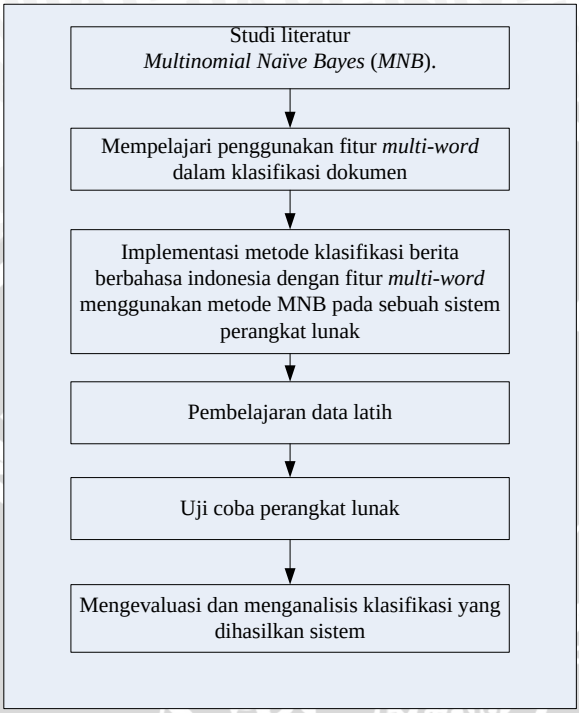
BAB III
METODOLOGI DAN PERANCANGAN

Pada bab metodologi dan perancangan ini akan dibahas metode, rancangan yang digunakan dan langkah langkah yang dilakukan dalam penelitian pembuatan klasifikasi dokumen berita berbahasa Indonesia berbasis *multi word* dengan menggunakan metode *Multinomial Naïve Bayes* (*MNB*).

Penelitian dilakukan dengan tahapan-tahapan berikut ini :

1. Mempelajari metode dan proses yang akan digunakan dalam pengklasifikasian dokumen menggunakan fitur *multi word*.
2. Mempelajari metode yang akan digunakan pada sistem pengklasifikasian dokumen ini, yaitu metode *Multinomial Naïve Bayes* (*MNB*).
3. Menganalisa dan merancang perangkat lunak pengklasifikasian dokumen menggunakan fitur frasa dengan metode *MNB*.
4. Membuat perangkat lunak berdasarkan analisis dan perancangan yang telah dilakukan.
5. Melakukan proses pelatihan (pembelajaran) terhadap perangkat lunak dengan memasukkan sejumlah dokumen berita berbahasa indonesia yang diperoleh dari sumber data tertentu sebagai data latih.
6. Melakukan uji coba perangkat lunak yang telah dibuat menggunakan dokumen tes berupa data dokumen berita berbahasa indonesia sebagai data uji.

Langkah-langkah pembuatan perangkat lunak pengklasifikasian dokumen berita menggunakan fitur *multi-word* dengan metode *MNB* dapat dilihat pada Gambar 3.1.



Gambar 3. 1 Alur penelitian

3.1. Analisis Data

Pada penelitian ini data yang digunakan berupa dokumen berita berbahasa indonesia yang diambil dari sebuah situs yang memuat kumpulan berita secara *online* beserta kategorinya, yaitu *http://www.kompas.com/*. Sumber tersebut merupakan salah satu dari situs berita *online* yang dapat dipercaya, hingga saat ini situs tersebut masih aktif dalam memberikan informasi berita secara *online* oleh karena itu situs tersebut dapat dijadikan refensi dalam penelitian ini.

Data yang dikumpulkan diambil dari berita yang diterbitkan dari bulan September 2011 sampai dengan November 2011. Pada situs *http://www.kompas.com/* telah memberikan kategori label pada berita yang diterbitkannya. Adapun Kategori tersebut adalah nasional, regional, internasional, megapolitan, bisnis, olahraga, sains, travel, oase dan edukasi. Namun pada penelitian ini proses

pengkategorisasian hanya menggunakan 4 kategori yaitu olahraga, bisnis, sains, dan edukasi, hal ini dimaksudkan untuk memperoleh data latih (*training set*) yang tepat dan untuk mempermudah pengujian kebenaran dan keakuratan pada data *testing*.

Dokumen yang diambil dari kompas dibagi menjadi dua, yaitu sebagai data latih atau *training* dan data uji atau *testing*. Data berita yang diambil dari website harian kompas sebanyak 360 dokumen terdiri dari 300 data latih dan 60 data uji.

3.2. Perancangan Sistem Secara Keseluruhan

Pada sub-bab perancangan sistem secara Keseluruhan, analisis sistem ini akan dibahas mengenai semua hal yang diperlukan dalam proses pembuatan perangkat lunak pengklasifikasian dokumen menggunakan fitur *multi-word* dengan metode *MNB*.

3.2.1. Deskripsi Umum Sistem

Dalam penelitian ini, akan dibangun suatu perangkat lunak yang dapat digunakan untuk pengklasifikasian dokumen berita berbahasa indonesia secara otomatis pada file dokumen berformat *file teks* dengan ekstensi txt (**.txt*) sebagai masukan. Metode yang digunakan dalam sistem ini adalah *Multinomial Naïve Bayes* (*MNB*) berbasis *multi-word*.

Pengklasifikasian berita yang dibuat terdapat dua tahap. Tahap pertama adalah proses pembelajaran atau pelatihan terhadap sekumpulan dokumen berita (*training set*) dan tahap selanjutnya adalah proses pengklasifikasian berita yang belum diketahui kategorinya (*testing set*) berdasarkan pengetahuan yang telah terbentuk dari *training set* sehingga didapatkan kategori dari berita tersebut.

Pada tahap pembelajaran, proses-proses yang dilakukan adalah :

1. *User* memasukkan teks berita yang akan dijadikan data latih.
2. *User* menentukan kategori berita yang telah dimasukkan.
3. Teks berita akan dipotong (parsing) menjadi kalimat-kalimat terpisah sesuai dengan urutan pemotongan dan dilanjutkan dengan dilanjutkan dengan penghapusan angka dan tanda baca.

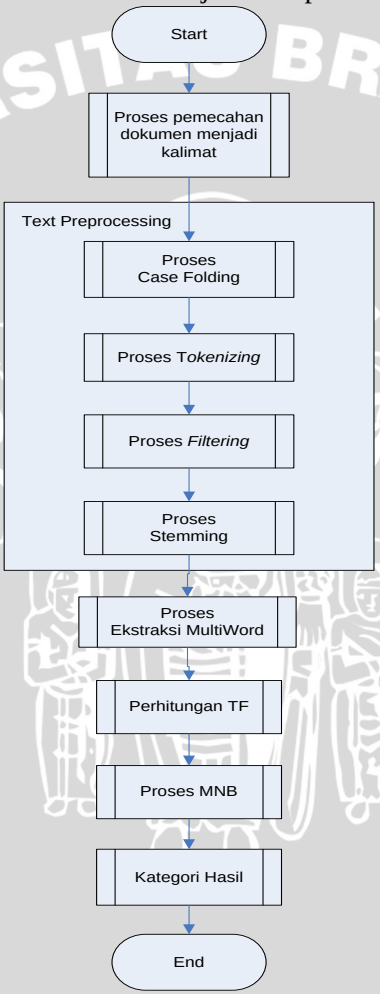
4. Sistem melakukan tahap *preprocessing* terhadap setiap kelimat yang telah dipotong (*parsing*) , diawali dengan proses *case folding* yaitu mengubah semua huruf di dalam data latih menjadi huruf *non capital*.
5. Kemudian proses *tokenizing* yaitu pemotongan (*parsing*) teks dalam data latih menjadi kata tunggal.
6. Proses selanjutnya adalah *filtering*, yaitu menghilangkan *stopword* yang ada dalam data latih dan dilanjutkan dengan penghapusan angka dan tanda baca.
7. Hasil dari *filtering* kemudian dilakukan *stemming* untuk menghilangkan kata berimbuhan.
8. Proses selanjutnya adalah ekstraksi *multi-word* dimana akan dihasilkan *multi-word* yang sesuai dari perbandingan setiap kalimat pada dokumen latih teks berita
9. Hitung frekuensi dari masing-masing *multi-word* yang telah ditentukan pada dokumen latih teks berita.
10. Sistem kemudian melakukan perhitungan nilai probabilitas dalam setiap kategori yang terdapat dalam data latih.

Dalam pengujian, langkah-langkah yang dilakukan dalam pengklasifikasian dokumen berita adalah :

1. *User* memasukkan teks dokumen yang akan diklasifikasikan.
2. Teks dokumen akan dipotong (parsing) menjadi kalimat-kalimat terpisah sesuai dengan urutan pemotongan.
3. Sistem melakukan tahap *preprocessing*, yang diawali dengan proses *case folding*, mengubah semua huruf yang ada di dalam dokumen menjadi huruf *non capital*.
4. *Tokenizing*, pemotongan (*parsing*) tiap-tiap kata yang menyusun dokumen menjadi kata tunggal.
5. *Filtering*, penghilangan *stopword* yang terdapat pada dokumen dan dilanjutkan dengan dilanjutkan dengan penghapusan angka dan tanda baca.
6. *Stemming* untuk menghilangkan kata berimbuhan.
7. Proses selanjutnya adalah ekstraksi *multi-word* dimana akan dihasilkan *multi-word* yang sesuai dari perbandingan setiap kalimat pada dokumen uji teks berita
8. Dilakukan perhitungan nilai probabilitas dokumen uji berdasarkan probabilitas masing-masing kategori.

9. Hasil probabilitas yang paling tinggi akan menjadi kategori dari dokumen uji. Gambar 3.2 menunjukkan alur deskripsi umum system

Adapun gambar yang menjelaskan mengenai Alur deskripsi umum klasifikasi dokumen berita berbahasa indonesia berbasis *multi-word* dengan metode MNB dijelaskan pada Gambar 3.2



Gambar 3. 2 Alur deskripsi umum klasifikasi dokumen berita berbahasa indonesia berbasis *multi-word* dengan metode MNB

3.2.2.Batasan Sistem

Sistem yang akan dibuat memiliki batasan-batasan sebagai berikut :

1. Sistem hanya menangani dokumen teks berita berbahasa Indonesia saja.
2. Sumber berita hanya diperoleh dari situs berita berbahasa Indonesia, yaitu www.kompas.com.
3. Sistem hanya dapat menangani dokumen berformat *file text* dengan ekstensi txt (*.*txt*).
4. Penghilangan *stopword* yang digunakan terkait dengan bahasa indonesia, karena dalam penelitian ini yang dipilih hanya dokumen yang berbahasa Indonesia, maka *stopword*-nya juga dalam bahasa Indonesia.
5. Sistem hanya mengklasifikasikan berita berbahasa Indonesia tidak dilakukan pemisahan struktur kalimat, semua kata dalam kalimat dianggap mempunyai kedudukan yang sama.
6. Proses klasifikasi dilakukan secara *single process* dokumen uji.

3.3. Perancangan Proses

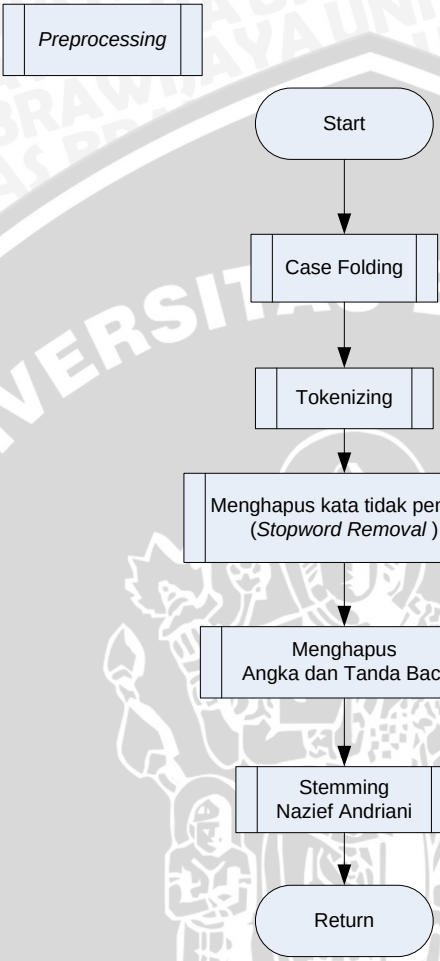
Pada bagian ini dinjelaskan secara rinci mengenai proses yang dilakukan sistem pengklasifikasian dokumen otomatis secara beruntun dan sistematis. Pertama sistem melakukan pemecahan dokumen menjadi kalimat-kalimat yang terpisah.Tahap selanjutnya adalah *preprocessing* pada tiap-tiap kalimat didalam setiap dokumen. Selanjutnya sistem malakukan ekstraksi *multi-word* dari setiap dokumen dan merepresentasikannya di tabel *multi-word*. Frekuensi *multi-word* akan dihitung berdasarkan jumlah *multi-word* yang sesuai pada setiap dokumen. Selanjutnya, sistem melakukan tahap klasifikasi untuk menentukan kategori dokumen dengan menggunakan metode *Multinomial Naïve Bayes* (*MNB*).

3.3.1. Tahap *Preprocessing*

Proses klasifikasi dokumen yang dilakukan oleh sistem diawali dengan memparsing dokumen kedalam kalimat-kalimat terpisah berdasarkan susunan dari dokumen tersebut. Hal ini dilakukan agar mempermudah dalam proses ekstraksi *multi-word* pada setiap kalimat sehingga dihasilkan *multi-word* yang sesuai dari hasil perbandingan setiap kalimat.

Tahap selanjutnya adalah *preproscessing* meliputi *case folding, stopword removal, stemming,* dan *tokenizing*. Proses *case folding* adalah pengubahan karakter huruf menjadi huruf kecil (Garcia, 2005), Selanjutnya adalah proses *tokenizing* yaitu memecah kata menjadi kata-kata, selanjutnya adalah proses *filtering* dengan penghilangan kata-kata yang tidak penting yaitu *stopword removal*, *stopword* yang digunakan merupakan *stopword* dalam bahasa Indonesia (Tala, 2003). Proses tersebut dilanjutkan dengan penghapusan angka dan tanda baca. Setelah proses *filtering* dilakukan *stemming* dengan menggunakan algoritma Nazief-Andriani. Gambar 3.4 menunjukkan *flowchart* dari *preprocessing*.

Gambar 3. 3 *Flowchart* proses awal



Gambar 3. 4 *Flowchart* proses *Preprocessing*

1. Pemecahan Kata

Pada tahap ini dilakukan proses ekstraksi dokumen menjadi kalimat-kalimat terpisah sesuai dengan susunannya. Hal ini bertujuan agar mempermudah dalam proses ekstraksi *multi-word* dari masing-masing kalimat pada dokumen tersebut.

Contoh :
Pembelian hak cipta buku pelajaran selama 15 tahun untuk memperbanyak pilihan masyarakat dalam menggunakan buku sekolah elektronik terus dilakukan pemerintah. Setelah buku pelajaran SD-SMA terpenuhi, fokus pembelian hak cipta buku pelajaran dilanjutkan hingga terpenuhinya buku-buku pelajaran SMK dan pendidikan anak usia dini.

Setelah pemecahan kalimat :

- Pembelian hak cipta buku pelajaran selama 15 tahun untuk memperbanyak pilihan masyarakat dalam menggunakan buku sekolah elektronik terus dilakukan pemerintah.
- Setelah buku pelajaran SD-SMA terpenuhi, fokus pembelian hak cipta buku pelajaran dilanjutkan hingga terpenuhinya buku-buku pelajaran SMK dan pendidikan anak usia dini.

2. Case Folding

Pada tahap ini dilakukan proses mengubah semua huruf yang terdapat dalam dokumen menjadi huruf kecil.

Contoh :
Pembelian hak cipta buku pelajaran selama 15 tahun untuk memperbanyak pilihan masyarakat dalam menggunakan buku sekolah elektronik terus dilakukan pemerintah.

Setelahproses *Case Folding* :
pembelian hak cipta buku pelajaran selama 15 tahun untuk memperbanyak pilihan masyarakat dalam menggunakan buku sekolah elektronik terus dilakukan pemerintah.

3. Tokenizing

Pada tahap ini, dilakukan pemotongan (*parsing*) terhadap kata-kata tunggal tersebut menjadi kumpulan *token*.

- pembelian
- hak
- cipta
- buku
- pelajaran
- selama
- 15
- tahun
- untuk
- memperbanyak
- pilihan
- masyarakat
- dalam
- menggunakan
- buku
- sekolah
- elektronik
- terus
- dilakukan
- pemerintah.

4. Proses *Filtering*

Filtering yang dilakukan dalam penelitian ini adalah dengan melakukan penghapusan terhadap kata-kata yang tidak relevan (*stopword*) dan mengambil kata-kata penting saja. Oleh karena itu, tahap ini disebut juga dengan *stopword removal*. Selanjutnya dilanjutkan dengan penghapusan angka dan tanda baca.

Contoh hasil *filtering* dari hasil *tokenizing* :

1. pembelian
2. hak
3. cipta
4. buku
5. pelajaran
6. pilihan
7. masyarakat
8. buku
9. sekolah
10. elektronik
11. pemerintah

5. Proses *Stemming*

Stemming yang digunakan adalah menggunakan algoritma Nazief-Andriani. Proses *stemming* digunakan untuk mendapatkan kata dasar dari setiap kata. Kata-kata tersebut dicocokkan dengan kamus *stemming*, dimana kamus *stemming* merupakan kamus yang berisi kata dasar, diambil dari Kamus Besar Bahasa Indonesia *online* (KBBI *online*). Jika kata yang dimaksud cocok dengan yang terdapat di kamus maka proses selesai, jika tidak maka akan masuk kedalam proses *stemming* dengan menggunakan algoritma Nazief-Andriani.

Contoh hasil *stemming* :

1. beli
2. hak
3. cipta
4. buku
5. ajar
6. pilih
7. masyarakat
8. buku
9. sekolah
10. elektronik
11. pemerintah

3.3.2. Tahap Ekstraksi *Multi-word*

Dalam proses ini setiap kalimat yang telah melewati proses *preprocessing* akan dibandingkan antara kalimat satu dengan kalimat yang lain dalam satu dokumen. Apa bila terdapat *pattern* kata yang sama maka kata tersebut menjadi kandidat *multi-word*, namun apa bila tidak maka perbandingan dilanjutkan pada *pettern* kata selanjutnya. Proses ini berjalan secara *sequential* pada setiap kalimat yang dibandingkan.

Kesamaan *pettern* kata antara dua kalimat dapat dihitung menggunakan variabel *k*. Ketika ditemukan kata sama antara dua kalimat, maka variabel *k* akan bernilai satu, ketika ditemukan lagi kata yang sama setelah kata sebelumnya (berurutan/ *sequential*) maka akan membuat nilai *k* menjadi dua, ketika nilai k sama dengan dua maka kesamaan pattern kata tersebut akan diekstrak menjadi sebuah kandidat *multi-word*.

```
Input:
     s1, the first sentence
     s2, the second sentence
Output:
     Multi-word extracted from s1 and s2.
Procedure:
     s1 = {w1,w2,…,wn}, s2 = {w1',w2',…,wm'}, k=0
     For each word wi in s1
         For each word wj in s2
            While(wi equal to wj)
              k++
            End while
            If k==2
              extract the words from wi to wi+k to
form a multi-word candidate
              k = 0
            End if
          End for
     End for
```

Algoritma Ekstraksi *Multi-word*

3.3.3. Perhitungan TF *Multi-word*

TF adalah *term frequency* atau jumlah kemunculan *term (multi-word)* dalam isi dokumen. Dari ekstraksi *multi-word* maka akan dihitung jumlah kemunculan *multi-word* tertentu dalam dokumen tersebut. Nilai kemunculan tersebut menjadi nilai *TF*.

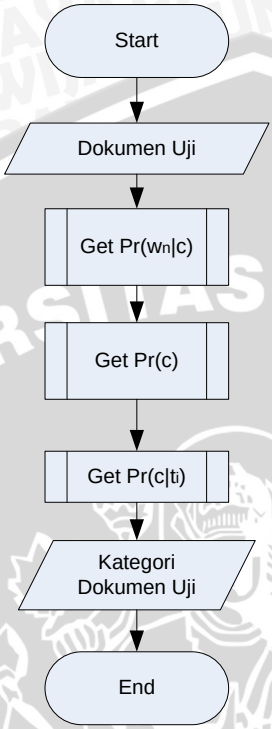
1. beli hak : 2
2. hak cipta : 2
3. cipta buku : 2
4. buku pelajaran : 3

3.3.4. Pengklasifikasian Dokumen

Proses pengklasifikasian untuk dokumen uji menggunakan algoritma *MNB* dapat dilakukan dengan langkah-langkah berikut :

1. Menghitung peluang $Pr(w_n|c)$ yaitu peluang setiap *multi-word* w_n dalam setiap kategori *c* pada data latih sebagai pembelajaran klasifikasi, dengan persamaan 2.13.
2. Mencari peluang setiap kategori Pr(c) untuk kategori *c* dengan persamaan 2.9.
3. Menghitung max $Pr(c|t_i)$ yaitu peluang kategori *c* dari dokumen uji t_i dengan persamaan 2.11.
4. Nilai $Pr(c|t_i)$ tertinggi dari setiap kategori yang merupakan kategori dari dokumen uji tersebut.

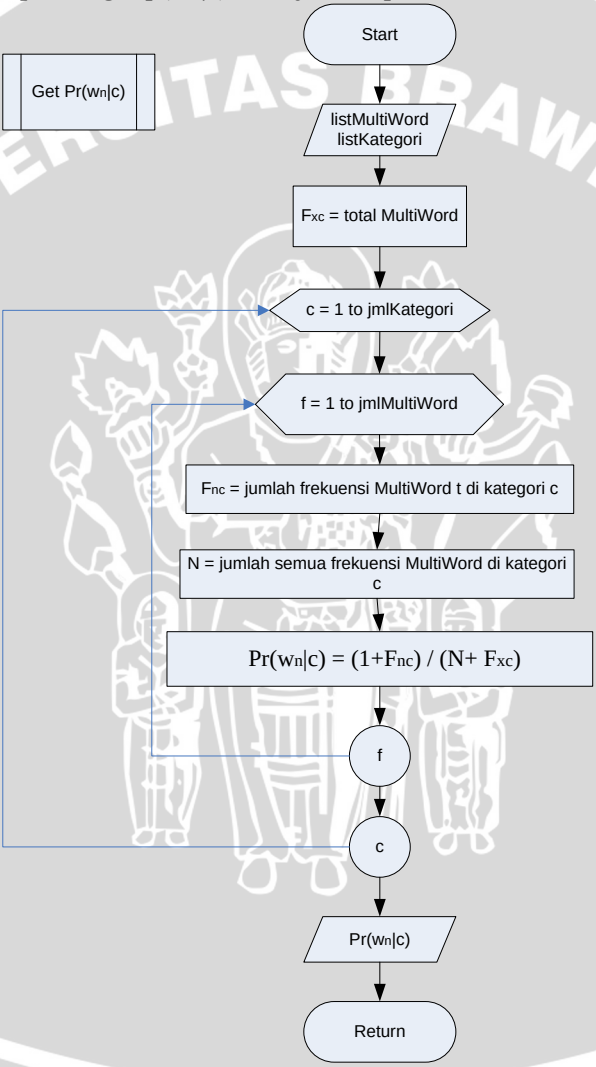
Flowchart proses klasifikasi dengan algoritma *MNB* ditunjukkan pada Gambar 3.5.



Gambar 3. 5 Algoritma *MNB*

1. **Proses Get Pr(w_n|c)**

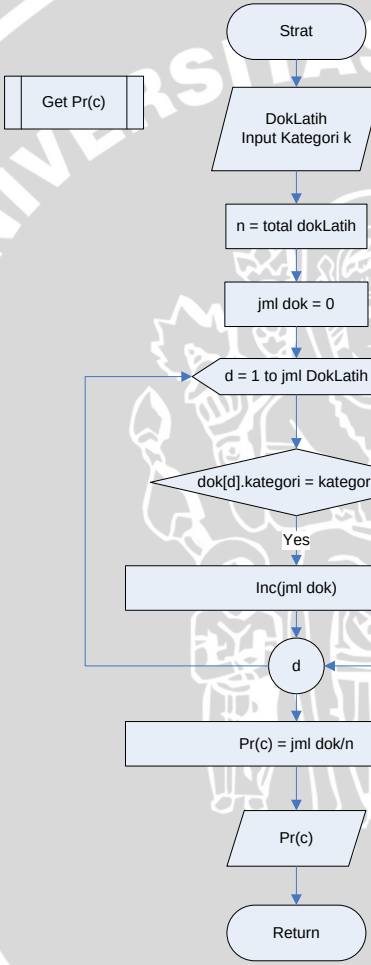
   Pada proses get Pr(w_n|c) dilakukan perhitungan peluang setiap *term multi-word* dalam setiap kategori, dengan persaman 2.13. Dari *term* yang ada di setiap dokumen latih dicari peluangnya di setiap kategori. Flowchart proses get p(w_n|c) ditunjukkan pada Gambar 3.6.



Gambar 3. 6 Flowchart Proses Get Pr(w_n|c)

2. **Proses Get Pr(c)**

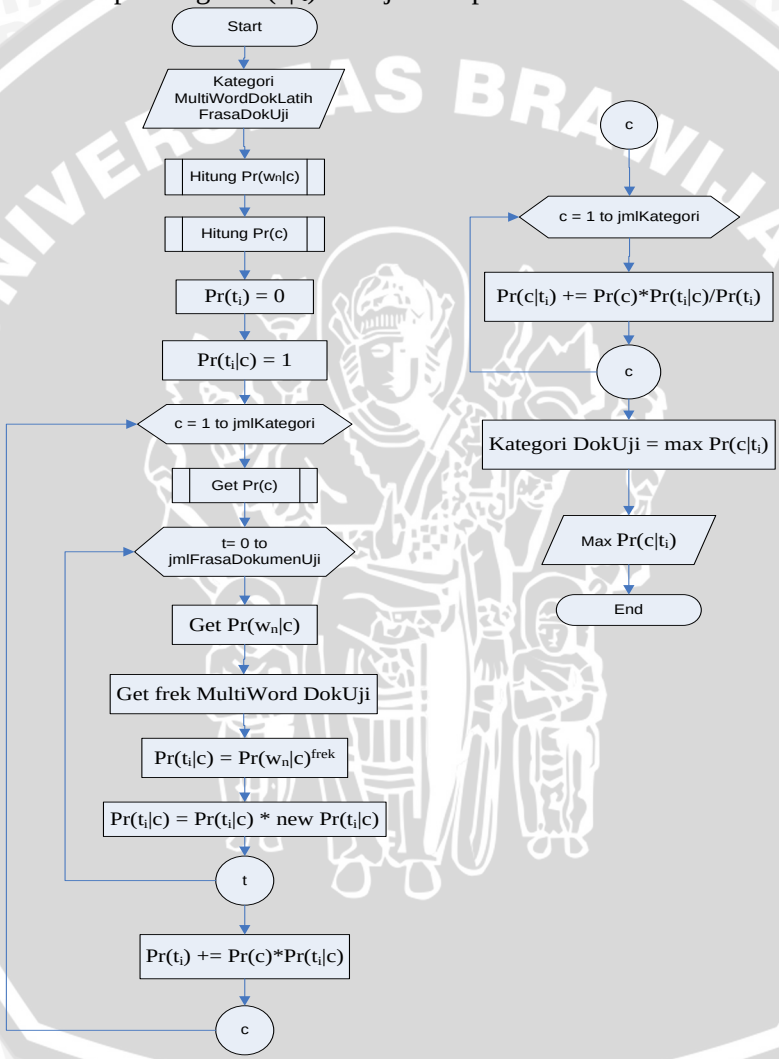
Pada proses get Pr(c) dilakukan perhitungan peluang setiap kategori dalam dokumen latih. Dengan persamaan 2.9, dihitung jumlah dokumen dalam setiap kategori dibagi jumlah semua dokumen dalam dokumen *corpus*. Flowchart proses get Pr(c) ditunjukkan pada Gambar 3.7.



Gambar 3. 7 Flowchart Proses Get Pr(c)

3. **Proses Get Pr(c|t_i)**

Pada proses get Pr(c|t_i) dilakukan perhitungan peluang setiap kategori dengan dokumen latih dengan persaman 2.11. Dari *term multi-word* yang ada di setiap dokumen *latih* dicari peluangnya di setiap kategori. Flowchart proses get Pr(c|t_i) ditunjukkan pada Gambar 3.8.



Gambar 3. 8 Flowchart Proses Get Pr(c|t_i)

3.4. Contoh Perhitungan Manual

Berikut ini diberikan contoh, yaitu terdapat empat buah dokumen latih yang berasal dari empat kategori yang berbeda dan sebuah dokumen uji yang kategorinya tidak diketahui. Berikut ini adalah daftar dokumen latih dan dokumen uji :

D1*
Baterai litium yang digunakan dalam komputer jinjing masih menjadi penentu utama ukuran dan berat komputer. Baterai ini menggunakan cairan elektrolit yang mudah menguap dan berbahaya.Panas yang dihasilkan baterai bisa memicu kebakaran komputer jinjing. Selain di komputer, baterai model ini banyak digunakan dalam mobil listrik. Selain mahal dan berat, efisiensi baterai ini sangat rendah. Untuk mengatasi, sejumlah peneliti dari Universitas Leeds, Inggris, yang dipimpin Ian Ward, meneliti baterai yang menggunakan gel sebagai pengganti cairan elektrolit. "Gel polimer ini mirip seperti lapisan film yang solid meski 70 persen berupa elektrolit cair," kata Ward kepada BBC, Sabtu (10/9/2011). Penggunaan gel akan mencegah pemanasan pada baterai yang dapat mencapai puluhan derajat celsius. Selain akan menghasilkan baterai berukuran lebih kecil dan lebih aman, para peneliti menjanjikan harga baterai ini lebih murah 10-20 persen.
Kategori : Sains

D2*
Memasuki bulan Oktober 2011, Bank Indonesia (BI) menerbitkan tiga peraturan baru yang berhubungan dengan devisa ekspor Indonesia. Kepala Biro Humas BI, Difi Ahmad Johansyah, di Jakarta, Jumat (30/9/2011), menjelaskan, tiga peraturan itu adalah Peraturan Bank Indonesia (PBI) Devisa Hasil Ekspor, PBI Lalu Lintas Devisa, dan PBI Devisa Utang Luar Negeri. "Akan ada siaran pers pada hari Senin pekan depan," kata Difi yang sedang berada di Bandung saat dihubungi Kompas, Jumat (30/9/2011). BI dalam berbagai kesempatan telah menyosialisasikan devisa hasil ekspor. Aturan itu mewajibkan pembayaran ekspor melalui bank di dalam negeri.
Kategori : Bisnis

D3*
Rektor Universitas Indonesia Gumilar R Somantri dinilai tidak memiliki kepekaan lagi. Sensitivitas semestinya dijadikan pertimbangan dalam menentukan pemberian gelar doktor honoris causa. Di sela-sela persiapan mengikuti pertemuan organisasi perburuhan internasional (ILO) di Jerman, Senin (5/9/2011), Ketua Umum Asosiasi Pengusaha Indonesia (Apindo) Sofjan Wanandi mengatakan, "Sebenarnya Rektor UI tidak sensitif memberikan gelar ini, apalagi dengan adanya kejadian-kejadian yang menimpa tenaga kerja Indonesia belakangan ini!" Penilaian terhadap Rektor UI Gumilar R Somantri ini terkait pemberian gelar doktor kehormatan atau honoris causa (HC) kepada Raja Arab Saudi Abdullah bin Abdul Aziz al-Saud. Menurut Sofjan, banyak tokoh international yang lebih berhak menerima gelar ini daripada Raja Arab Saudi karena lebih banyak berbuat dan lebih menguntungkan Indonesia dalam berinvestasi.
Kategori : Edukasi

D4*
Indonesia merebut dua medali emas dari nomor kata beregu putra dan putri pada ajang Kejuaraan Dunia Karate World Karate Federation di Istanbul, Turki, Minggu (18/9/2011). Pada kejuaraan yang diikuti 36 negara ini, trio putra dan putri Indonesia mengalahkan Mesir dengan angka telak 5-0. Trio putri, yang terdiri dari Dewi, Yulianti dan Sisilia, merebut emas lebih dulu dengan memeragakan Haiku', sedangkan trio putra (Faisal Zainudiin, Aswar, dan Fidelys Lolobua) memeragakan jurus Kururunfa. Berbeda dengan di nomor kata, di nomor kumite karateka Indonesia gagal mempersembahkan medali. Beberapa karateka Indonesia berhasil melaju ke babak ketiga dan keempat, tetapi gagal di babak repechages. "Kami akan evaluasi hasil dari Istanbul Open ini," ujar Manajer Karate SEA Games Indonesia Zulkarnaen Purba, di Istanbul.
Kategori : Olahraga

DU**
Kenshi Indonesia mendapatkan tiga emas di hari terakhir kompetisi yang berlangsung di GOR Ciracas, Jakarta Timur, Minggu (20/11/2011). Dengan raihan tiga emas ini, maka total medali emas yang didapat Indonesia dari cabang olahraga Kempo menjadi delapan, melampaui target 5 medali emas. Ketiga medali emas didapat dari kenshi Rini Imelda Samol dari kelas 54 kilogram randori putri, beregu campuran dantai embu grup Kyu Kenshi dan beregu campuran dantai embu grup Yudansha. Di nomor beregu campuran dantai embu grup Kyu Kenshi diperkuat delapan kenshi, yakni Jani M Bone, Ferdy Firmanda, Ashari Ridho, Helmi Yanuar, Dewi Aristiani, Vonny Suzendra, Yunika Asyora, dan Leni Marlinah. Sementara beregu campuran embu grup Yudhansa diperkuat Menah Suprianah, Siti Nurhayati, Jenneth P Dethan, Dwi Putri, Mulya Sitanggang, Arif Satria, Aljufri, dan Arif Nurachman. "Hasil ini cukup memuaskan meski ada beberapa nomor yang lepas. Memuaskan karena kita berhasil melampaui target," kata manajer tim Kempo Indonesia .
Kategori : ?

Ket :

* = Dokumen latih (D1-D4)

** = Dokumen latih (DU)

Dari dokumen-dokumen tersebut, langkah pertama adalah memecah setiap dokumen menjadi kalimat-kalimat yang sesuai dengan susunannya. Selanjutnya akan dilakukan proses *preprosesing* yaitu proses *case folding, tokenizing, filtering, stemming.* Berikut ini adalah daftar hasil dari seluruh langkah awal :

D1
1. baterai litium komputer jinjing penentu utama ukuran berat komputer 2. baterai cairan elektrolit mudah uap bahaya 3. panas hasil baterai bisa picu kebakaran komputer jinjing 4. komputer baterai model mobil listrik 5. mahal berat efisiensi baterai rendah 6. peneliti universitas leeds inggris pimpin ian ward teliti baterai gel cairan elektrolit 7. gel polimer lapisan film solid persen elektrolit cair ward bbc 8. gel cegah panas baterai derajat Celsius 9. baterai ukuran kecil aman peneliti janji harga baterai murah persen.
Kategori : Sains

D2
1. masuk bulan oktober bank indonesia bi terbit atur baru hubung devisa ekspor indonesia 2. kepala biro humas bi difi ahmad johansyah jakarta jumat atur atur bank indonesia pbi devisa hasil ekspor pbi lintas devisa pbi devisa utang luar negeri 3. siar pers senin pekan difi bandung hubung kompas jumat 4. bi sempat sosial devisa ekspor 5. atur wajib bayar ekspor bank negeri
Kategori : Bisnis

D3
1. rektor universitas indonesia gumilar r somantri nilai milik peka 2. sensitivitas timbang tentu beri gelar doktor honoris causa 3. siap ikut temu organisasi buruh internasional ilo Jerman senin ketua umum asosiasi pengusaha indonesia apindoa sofjan wanandi rektor ui sensitif gelar jadi timpa tenaga kerja Indonesia 4. nilai rektor ui gumilar r somantri beri gelar doktor hormat honoris causa hc raja arab saudi abdullah bin abdul aziz al saud. 5. sofjan tokoh international hak terima gelar raja arab saudi buat untung indonesia investasi
Kategori : Edukasi

D4
1. indonesia rebut medali emas nomor regu putra putri ajang juara dunia karate world karate federation istanbul turki minggu 2. juara negara trio putra putri indonesia kalah mesir angka telak 3. trio putri terdiri dewi yulianti sisilia rebut medali emas peraga haiku trio putra faisal zainudiin aswar fidelys lolobua peraga jurus kururunfa. 4. beda nomor nomor kumite karateka indonesia gagal sembah medali 5. karateka indonesia hasil laju babak tiga empat gagal babak repechages 6. evaluasi hasil istanbul open manajer karate sea games indonesia zulkarnaen purba istanbul
Kategori : Olahraga

DU
1. taiwan rebut gelar juara ganda campur turnamen jepang buka superseries pasang chen hung ling cheng wen hsing kalah ganda denmark joachim fischer nielsen christinna pedersen 2. final langsung minggu pasang campur chen cheng butuh waktu jam menit kalah nielsen pedersen rubber game 3. babak semifinal sabtu chen cheng unggul tempat singkir unggul china zhang nan zhao yunlei 4. china rebut gelar juara tunggal putera puteri ganda puteri 5. ganda putera pasangan indonesia bona septano muhammad ahsan menghadapi unggulan china chin cai yun fu haifeng
Kategori : ?

Dari hasil *preprosessing* tersebut, selanjutnya adalah Mengekstrak *multi-word* dengan cara membandingkan setiap kalimat dengan kalimat yang lain dalam satu dokumen untuk mendapatkan pola *multi-word* kategori tertentu. Berikut ini adalah daftar hasil dari ekstaksi *multi-word* untuk semua dokumen :

D1	
Perbandingan Kalimat	Frasa
Kalimat 1- Kalimat 3 Kalimat 2- Kalimat 6	komputer jinjing cair elektrolit
Kategori : Sains	

D4	
Perbandingan Kalimat	Frasa
Kalimat 1- Kalimat 2 Kalimat 1- Kalimat 4	bank indonesia devisa negara
Kategori : Bisnis	

D3	
Perbandingan Kalimat	Frasa
Kalimat 1 - Kalimat 4	gumilar r, r somantri
Kalimat 2 - Kalimat 4	gelar doktor, honoris causa
Kalimat 3 - Kalimat 4	rektor ui
Kalimat 4 - Kalimat 1	gumilar r, r somantri
Kalimat 4 - Kalimat 2	gelar doktor, honoris causa
Kalimat 4 - Kalimat 3	rektor ui
Kalimat 4 - Kalimat 5	raja arab , arab saudi
Kalimat 5 - Kalimat 4	raja arab, arab saudi
Kategori : Edukasi	

D7	
Perbandingan Kalimat	Frasa
Kalimat 1 - Kalimat 2	putra putri
Kalimat 2 - Kalimat 3	trio putra
Kalimat 1 - Kalimat 3	medali emas
Kalimat 4 - Kalimat 5	karateka indonesia
Kategori : Olahraga	

DU	
Perbandingan Kalimat	Frasa
Kalimat 2- Kalimat 3	medali emas, medali emas, lampau
Kalimat 2- Kalimat 7	target
Kalimat 3- Kalimat 4	regu campur, campur dentai, dentai embu, embu grup, grup kyu, kyu kenshi, dentai embu, embu grup
Kalimat 3- Kalimat 5	regu campur, embu grup,
	regu campur, grup yudansaha,
Kalimat 4- Kalimat 5	regu campur, embu grup,
Kategori : ?	

Dari hasil ekstraksi *multi-word* , selanjutnya adalah menghitung TF (*Term Frequency*) dari *multi-word* tersebut. Berikut ini adalah tabel hasil dari perhitungan TF *multi-word* untuk semua dokumen :

Tabel 3. 1 Perhitungan TF *multi-word*

No	Frasa	Frekuensi Frasa				
		D1	D2	D3	D4	DU
1	komputer jinjing	0	0	0	1	0
2	cair elektrolit	0	0	0	1	0
3	bank indonesia	1	0	0	0	0
4	devisa negara	1	0	0	0	0
5	gumilar r	0	1	0	0	0
6	r somantri	0	1	0	0	0
7	gelar doktor	0	1	0	0	0
8	honoris causa	0	1	0	0	0
9	rektor ui	0	1	0	0	0
10	putra putri	0	0	1	0	0
11	trio putra	0	0	1	0	0
12	**medali emas**	**0**	**0**	**1**	**0**	**2**
13	raja arab	0	1	0	0	0
14	arab saudi	0	1	0	0	0
15	karateka indonesia	0	0	1	0	0
16	**lampau target**	**0**	**0**	**0**	**0**	**1**
17	**regu campur**	**0**	**0**	**0**	**0**	**3**
18	**campur dentai**	**0**	**0**	**0**	**0**	**1**
19	**dentai embu**	**0**	**0**	**0**	**0**	**2**
20	**embu grup**	**0**	**0**	**0**	**0**	**4**
21	**grup kyu**	**0**	**0**	**0**	**0**	**1**
22	**kyu kenshi**	**0**	**0**	**0**	**0**	**1**
23	**grup yudansaha**	**0**	**0**	**0**	**0**	**1**

Dari hasil perhitungan TF *multi-word*, selanjutnya adalah menghitung nilai probabilitas dari setiap kategori. Untuk menghitung nilai probabilitas ini, langkah pertama yaitu mencari nilai $p(c_j)$ dengan persamaan 2.9. Berikut ini perhitungan manual dari dokumen-dokumen latih yang ada :

1. Kategori Bisnis : $Pr(c_j) = \frac{f_d(v_j)}{|D|} = \frac{1}{4} = \mathbf{0.25}$
2. Kategori Edukasi : $Pr(c_j) = \frac{f_d(v_j)}{|D|} = \frac{1}{4} = \mathbf{0.25}$
3. Kategori Olahraga : $Pr(c_j) = \frac{f_d(v_j)}{|D|} = \frac{1}{4} = \mathbf{0.25}$
4. Kategori Sains : $Pr(c_j) = \frac{f_d(v_j)}{|D|} = \frac{1}{4} = \mathbf{0.25}$

Setelah $Pr(c_j)$ dihitung, langkah berikutnya adalah mencari nilai $p(w_j|v_j)$ dari masing-masing *multi-word* pada masing-masing kategori dengan menggunakan persamaan 2.13. Hasil perhitungan $Pr(w_n|c_j)$ terdapat pada tabel 3.2

Contoh perhitungan :

1. Pada term *multi-word* "bank indonesia", yaitu terletak pada baris pertama pada kolom hasil ekstraksi *multi-word* pada tabel 3.1 memiliki jumlah term *multi-word* sebanyak dua pada kategori bisnis, untuk perhitungan nilai $Pr(w_n|c_j)$ pada term *multi-word* "bank indonesia" pada kategori bisnis dijelaskan sebagai berikut :

$$Pr(w_n \mid c) = \frac{1 + F_{nc}}{N + \sum_{x=1}^{N} F_{xc}}$$

Dimana :

- F_{xc} : Jumlah *multi-word* pada dokumen latih yang termasuk dalam kategori c
- N : Total *multi-word* yang ditemukan
- F_{xc} : Total *multi-word* yang terdapat pada categori tertentu

Jadi,

$Pr(w_n|c_j)$: (0+ 1) / (23 + 2) = **0.04**

Hasil perhitungan dapat dilihat pada tabel 3.1 yang ditandai dengan warna merah .

Tabel 3. 2 mencari nilai p(wj|vj) dari masing-masing *multi-word*

FrekuensiKategori					P(w|v)			
B	E	O	S	Dok Uji	B	E	O	S
0	0	0	1	0	0.04	0.033	0.037	0.08
0	0	0	1	0	0.04	0.033	0.037	0.08
1	0	0	0	0	0.08	0.033	0.037	0.04
1	0	0	0	0	0.08	0.033	0.037	0.04
0	1	0	0	0	0.04	0.067	0.037	0.04
0	1	0	0	0	0.04	0.067	0.037	0.04
0	1	0	0	0	0.04	0.067	0.037	0.04
0	1	0	0	0	0.04	0.067	0.037	0.04
0	1	0	0	0	0.04	0.067	0.037	0.04
0	0	1	0	0	0.04	0.033	0.074	0.04
0	0	1	0	0	0.04	0.033	0.074	0.04
0	**0**	**1**	**0**	**2**	**0.04**	**0.033**	**0.074**	**0.04**
0	1	0	0	0	0.04	0.067	0.037	0.04
0	1	0	0	0	0.04	0.067	0.037	0.04
0	0	1	0	0	0.04	0.033	0.074	0.04
0	**0**	**0**	**0**	**1**	**0.04**	**0.033**	**0.037**	**0.04**
0	**0**	**0**	**0**	**3**	**0.04**	**0.033**	**0.037**	**0.04**
0	**0**	**0**	**0**	**1**	**0.04**	**0.033**	**0.037**	**0.04**
0	**0**	**0**	**0**	**2**	**0.04**	**0.033**	**0.037**	**0.04**
0	**0**	**0**	**0**	**4**	**0.04**	**0.033**	**0.037**	**0.04**
0	**0**	**0**	**0**	**1**	**0.04**	**0.033**	**0.037**	**0.04**
0	**0**	**0**	**0**	**1**	**0.04**	**0.033**	**0.037**	**0.04**
0	**0**	**0**	**0**	**1**	**0.04**	**0.033**	**0.037**	**0.04**
2	7	4	2					

Ket :
B : Bisnis
E :Edukasi
O :Olahraga
S :Sains

Dari hasil perhitungan p(v) dan p(w|v), selanjutnya menentukan kategori dari dokumen uji. Dengan persamaan 2.15 dicari peluang dari dokumen masing-masing kategori.

- P(DokUji|Bisnis) = 0.04^2 x 0.04^1 x 0.04^3 x 0.04^1 x 0.04^2 x 0.04^4 x 0.04^1 x 0.04^1 x 0.04^1 = **$4{,}29497 \times 10^{-23}$**

- P(DokUji|Edukasi) = 0.033^2 x 0.033^1 x 0.033^3 x 0.033^1 x 0.030^2 x 0.030^4 x 0.03^1 x 0.030^1 x 0.030^1 = **$2{,}32306 \times 10^{-24}$**

- P(DokUji|Olahraga) = 0.074^2 x 0.037^1 x 0.037^3 x 0.037^1 x 0.037^2 x 0.037^4 x 0.037^1 x 0.037^1 x 0.037^1= **$5{,}01464 \times 10^{-23}$**

- P(DokUji|Sains) = 0.04^2 x 0.04^1 x 0.04^3 x 0.04^1 x 0.04^2 x 0.04^4 x 0.04^1 x 0.04^1 x 0.04^1 = **$4{,}29497 \times 10^{-23}$**

Kemudian dengan persamaan 2.14 dicari peluang total dari dokumen uji.

- P(DokUji) = (0.25 x $4{,}29497 \times 10^{-23}$) + (0.25 x $2{,}32306 \times 10^{-24}$) + (0.25 x $5{,}01464 \times 10^{-23}$) + (0.25 x $4{,}29497 \times 10^{-23}$) = **$3{,}45922 \times 10^{-23}$**

Dengan persamaan 2.11 dicari peluang dari masing-masing kategori.

- P(DokUji|Bisnis) = 0.25 x $4{,}29497 \times 10^{-23}$ / $3{,}45922 \times 10^{-23}$ = **0.310399973**

- P(DokUji|Edukasi) = 0.25 x $2{,}32306 \times 10^{-24}$ /$3{,}45922 \times 10^{-23}$ = **0.016788881**

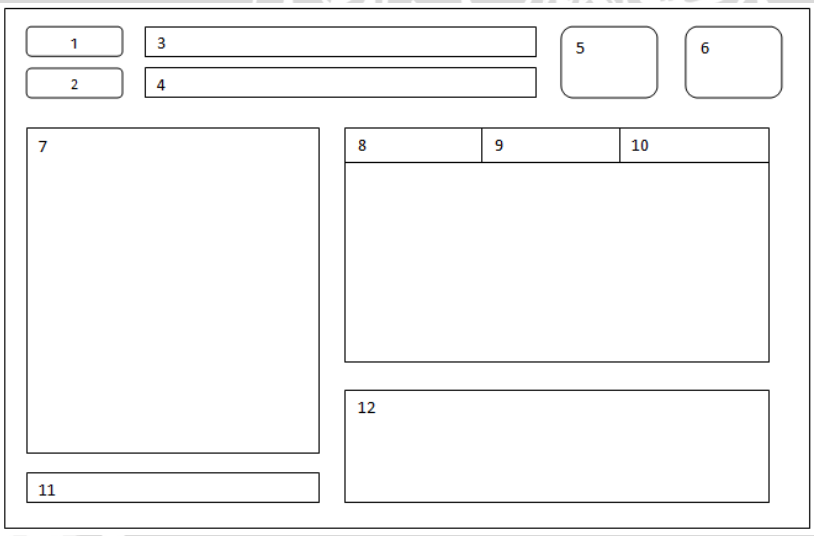
- P(DokUji|Olahraga) = 0.25 x 5,01464 x 10^{-23}/3,45922 x 10^{-23} = **0,362411173**

- P(DokUji|Sains) = 0.25 x 4,29497 x 10^{-23} / 3,45922 x 10^{-23} = **0.310399973**

Dari perhitungan MNB juga diketahui bahwa kategori Kesehatan(hijau) mempunyai peluang yang lebih besar dibandingkan hasil peluang dokumen uji pada ketegori lain. Maka dokumen uji termasuk kategori kesehatan.

3.5. Rancangan Antar Muka

Rancangan antar muka secara umum terdiri dari bagian input data, output hasil, tombol *preprocessing* dan tombol klasifikasi. Inputan berupa dokumen latih dan uji. Output berupa hasil dari pemecahan kalimat, *perpocessing*, ekstraksi *multi-word* dari dokumen uji, hasil klasifikasi dokumen. Gambar 3.9 menunjukkan rancangan antar muka secara umum.



Gambar 3.9 Rancangan Antar Muka Sistem Secara Umum

Adapun penjelasan dari bagian – bagian rancangan antar muka diatas adalah sebagai berikut:

1. *Button* untuk memilih folder dimana dokumen - dokumen latih disimpan.
2. *Button* untuk memilih folder dimana dokumen uji disimpan.
3. *Field* untuk menampilkan folder atau direktori dimana dokumen - dokumen latih disimpan.
4. *Field* untuk menampilkan folder atau direktori dimana dokumen uji disimpan.
5. *Button* untuk menampilkan dokumen latih dan dokumen uji pada datagrid.
6. *Button* untuk proses reset dan menampilkan tampilan ke tampilan awal.
7. Datagrid yang berisi nama dokumen latih dan dokumen uji beserta kategorinya yang merupakan hasil proses dari *button* 5.
8. Tab yang berisi tabel untuk menampilkan isi dokumen berdasarkan pilihan pada datagrid(no. 7).
9. Tab yang berisi tabel untuk menampilkan hasil dari proses perhitungan frekuensi multi word.
10. Tab yang berisi tabel untuk menampilkan hasil dari proses perhitungan probabilitas multi word.
11. *Progres bar* yang digunakan untuk mengetahui sistem sedang berjalan.
12. Tabel untuk menghasilkan hasil dari klasifikasi.

3.6. Rancangan Uji Coba

Setelah melakukan klasifikasi terhadap sejumlah dokumen uji maka sistem dapat dievaluasi untuk mengetahui tingkat akurasi pengklasifikasian. Evaluasi yang dilakukan menggunakan *precision*, *recall* dan *F1-Measure*. Pengujian sejumlah dokumen uji akan dilakukan beberapa kali dengan mengubah – ubah nilai masing – masing parameter.

Tujuan dari uji coba ini adalah untuk mengetahui efektifitas sistem yang dibuat menggunakan standar ukuran evaluasi *precision*, *recall*, dan F_1 *measure* yang telah dijelaskan pada Sub-bab 2.8.

3.6.1.Skenario Evaluasi

Pada pengujian sistem pengklasifikasian dokumen, sekumpulan dokumen akan dibagi menjadi dokumen latih dan dokumen uji. Selain itu juga dilakukan uji coba, yaitu semua dokumen isinya berbeda, baik untuk dokumen latih maupun dokumen uji.

Untuk mempelajari pengaruh jumlah data latih terhadap efektifitas sistem klasifikasi maka dilakukan tiga kali uji coba dengan jumlah data latih yang berbeda, dengan proporsi 100 data latih, 200 data latih, 300 data latih.

3.6.2. Hasil Evaluasi

Untuk mengetahui keberhasilan sistem dalam proses pengklasifikasian dokumen, dilakukan evaluasi terhadap sistem.Tabel 3.3 menunjukan rancangan evaluasi klasifikasi.

Tabel 3. 3 Rancangan Evaluasi Klasifikasi

Kategori	True positive	False positive	False negative	Precision	Recall	F1 Measure
Olahraga						
Bisnis						
Sains						
Edukasi						

Tabel 3.4 adalah tabel hasil evaluasi perhitungan nilai *precision*, *recall*, dan F_1 *measure*.

Tabel 3. 4 Tabel Evaluasi

Jumlah Data Latih	Precision	Recall	F1 Measure
100			
200			
300			

BAB IV
IMPLEMENTASI DAN PEMBAHASAN

Bab implementasi dan pembahasan ini berisi mengenai penerapan, pembahasan dan evaluasi dari sistem yang telah dikembangkan berdasarkan perancangan yang dilakukan pada bab metodologi dan perancangan.

4.1. Lingkungan Implementasi

Sistem dibangun pada lingkungan implementasi perangkat keras dan lingkungan implementasi perangkat lunak. Lingkungan implementasi perangkat keras dan perangkat lunak yang digunakan adalah sebagai berikut :

4.1.1. Lingkungan Implementasi Perangkat Keras

Perangkat keras yang digunakan untuk mengembangkan sistem memiliki beberapa spesifikasi sebagai berikut :

1. Processor Intel(R) Core(TM) i3 M390 @ 2.67 GHz(4CPUs)
2. Memory RAM DDR2 2048MB
3. Harddisk 500GB

4.1.2. Lingkungan Implementasi Perangakat Lunak

Perangkat lunak yang digunakan untuk implementasi sistem ini adalah sebagai berikut :

1. Sistem Operasi Windows 7 Ultimate 32-bit.
2. Microsoft Visual C# 2010 Ultimate sebagai software development dalam implementasi rancangan sistem.
3. Microsoft Office dan Visio sebagai software pengolah data.

4.2. Implementasi Program

Sub bab ini membahas mengenai implementasi proses – proses dari perancangan proses yang telah dijelaskan pada bab metodologi dan perancangan. Terbentuk beberapa proses utama

pada implementasi program ini. Proses – proses tersebut antara lain dapat dilihat pada tabel 4.1.

Tabel 4. 1 Kelas Utama Implementasi Program

No	Proses	Keterangan
1	*Parsing* kalimat dan *case folding*	Memotong dokumen menjadi kalimat-kalimat penyusunnya dan mengubah semua huruf kapital menjadi huruf kecil.
2	Preprocessing	Melakukan proses awal meliputi *tokenizing, filtering dan stemming*
3	Ekstraksi *multi-word*	Malakukan proses untuk mendapatkan multi-word pada suatu dokumen
4	Perhitungan TF (Term Frekuensi) *Multi-word*	Menghitung frekuensi setiap *multi-word* yang telah didapatkan.
5	Perobabilitas *multi-word*	Mendapatkan nilai probabilitas dari setiap *multi-word*.
6	Proses MNB	Melakukan proses klasifikasi dengan menggunakan metode *Multinomial Naive Bayes.*
7	Hasil Klasifikasi	Mendapatkan hasil klasifikasi yang berupa kategori dan nilai probabilitasnya.

4.2.1.Proses *Parsing* Kalimat

Proses *parsing* kalimat berada dikelas PreProcessing , proses ini akan mengubah suatu dokumen menjadi kalimat-kalimat sesuai dengan susunan dokumen tersebut. Pemotongan kalimat-kalimat tersebut akan terjadi ketika menemukan suatu tanda baca akhir kalimat seperti titik(.), tanda tanya(?), tanda seru(!). Dalam proses parsing juga akan dilakukan sub proses *case folding* yaitu perubahan huruf kapital menjadi huruf kecil dengan memanggil method ToLower(). Kemudian menggembalikan dokumen yang telah diproses dalam bentuk *string array*. Proses ini dapat dilihat pada *source code* 4.1.

```
var parsing = dokumen.ToLower().Split(new[] { ".", "?", "!" },
StringSplitOptions.RemoveEmptyEntries);
```

Source code 4. 1 Proses Parsing Kalimat dan *Case Folding*

4.2.2.Proses *Preprocessing*

Proses *preprocessing* terdapat beberapa sub proses yaitu, Tokenizing Filtering dan Stemming. Untuk sub proses Tokenizing dan Filtering berada pada kelas PreProcessing ,sedangkan sub proses Stemming berada pada kelas StemmingNaziefAndriani. Sub proses tersebut akan dijelaskan sebagai berikut.

1. Tokenizing dan Filtering

Tokenizing dan Filtering adalah proses yang menggunakan method prepo(), dimana parameter yang digunakan adalah dokumen yang akan diproses. Pertama method ini akan melakukan proses *Tokenizing* dan penghapusan karakter dan angka. Proses akan dilanjutkan dengan proses *Filtering* yaitu penghapusan terhadap kata-kata yang tidak relevan (*stopword*) dan mengambil kata-kata penting saja. Oleh karena itu, tahap ini disebut juga dengan *stopword removal*. Proses *Tokenizing* dan *Filtering* dapat dilihat pada *source code* 4.2.

```
public string[] prepro(string dokumen){
   var parsing = dokumen.ToLower().Split(new[] { ".", "?",
"!" }, StringSplitOptions.RemoveEmptyEntries);
   string[] hasilPrepro = new string[parsing.Length];
   int i = 0;
   foreach (string item in parsing)
     {
string[] PemisahKata = { " ", "-", ",", "@", "#", "\"",
"$", "%", "^", "&", "_", "+", "=", ";", "/", "\\", "0",
"1", "2", "3", "4", "5", "6", "7", "8", "9", "0", ":", "=",
"'", ">", "<", "(", ")", ":" };
string[] pisah = item.Split(PemisahKata,
StringSplitOptions.RemoveEmptyEntries).Where(x =>
x.Replace(" ", "").Length > 0).ToArray();
    string[] stopword = new string[100000];
    string[] _ceksplit = new string[pisah.Length];
    bool sama = true;
    FileStream sr = new FileStream(@"StopList.txt",
```

```
    FileMode.Open);
    StreamReader str = new StreamReader(sr);
    int a = 0;
    while (!str.EndOfStream)
      {
        stopword[a] = str.ReadLine();
        a++;
      }
    sr.Close();
    str.Close();
    int count = 0;
    for (int j = 0; j < pisah.Length; j++)
     {
      sama = true;
       for (int k = 0; k < stopword.Length; k++)
        {
         if (pisah[j].Equals(stopword[k]))
          {
           sama = false;
           continue;
          }
        }
         if (sama.Equals(true))
           {
             _ceksplit[count] = pisah[j];
             count++;
           }
     }
     foreach (string item2 in _ceksplit)
       {
         hasilPrepro [i] += item2 + " ";
       }
         i += 1;
    }
    return hasilPrepro;
}
```

Source code 4. 2 Proses *Tokenizing* dan *Filtering*

2. Stemming

Proses *stemming* yang digunakan adalah *stemming* dengan algoritma Nazief-Adriani. *Stemming* adalah proses yang digunakan untuk mencari kata dasar. Pada proses *stemming* dilakukan dua tahap, yaitu yang pertama adalah pengecekan kata yang akan di-

stemming dan yang kedua adalah proses *stemming* itu sendiri dengan menggunakan algoritma Nazief-Andriani. Pengecekan kata dilakukan dengan mencocokkan apakah kata yang diinputkan sesuai dengan kata yang terdapat pada kamus. Jika hasilnya sama maka kata tersebut merupakan kata dasar, namun jika belum ,digunakan proses *stemming*.Beberapa fungsi dalam proses *stemming* ditunjukkan pada tabel 4.1.

Tabel 4. 2 Fungsi pada Proses Stemming

Fungsi	Kegunaan
void bacaKamus()	Membaca isi kamus.
void setKata (String kata)	Memasukkan kata yang akan diproses.
cekKamus(String kata)	Mengecek apakah kata yang diinputkan terdapat di dalam kamus atau tidak.
KataDasar(String kata)	Untuk menghasilkan kata dasar
void hapusInfleksionalSuffiks( )	Menghapus *inflectional suffixes*, -lah,-kah, -ku, -mu, -nya
void hapusDerivationSuffiks()	Menghapus *derivation suffixes*
void derivationPrefiksA()	Cek tipe awalan aturan pertama
derivationPrefiksB()	Cek tipe awalan aturan kedua
boolean vowel(char huruf)	Cek huruf *vocal* yang diinputkan.

a. Baca Kamus

Baca kamus digunakan untuk membaca kata-kata dasar yang terdapat pada Kamus Besar Bahasa Indonesia online (KBBI online). Kata-kata dasar tersebut kemudian disimpan pada sebuah list. Proses baca kamus ditunjukkan *Source code* 4.3

```
public void bacaKamus(){
FileStream readDokumen = new FileStream(@"kamus.txt",
FileMode.Open);
  StreamReader str = new StreamReader(readDokumen);
  int a = 0;
```

```
  while (!str.EndOfStream)
   {
     kamusku[a] = str.ReadLine();
     a++;
   }
  foreach (string item in kamusku)
   {
    listKamus.Add(item);
   }
}
```

Source code 4. 3 Baca kamus

b. *Input* Kata

Fungsi setKata digunakan untuk memasukkan kata yang ingin diproses. Kode program *input* kata ditunjukkan pada *Source code* 4.4

```
public void setKata(String kata){
 this.kata = kata;
 this.akarKata = kata;
 bersikan = "";
}
```

Source code 4. 4 *Input* kata

c. Cek Kamus

Cek kamus untuk mengoreksi apakah kata yang diinputkan sesuai yang terdapat di *list* kata dalam kamus, jika iya maka algoritma berhenti. Kode cek kamus ditunjukkan pada *Source code* 4.5

```
public bool cekKamus(String kata){
 if (listKamus.Contains(kata))
  {
   return true;
  }
   else
  {
   return false;
  }
}
```

Source code 4. 5 *Cek kamus*

d. Kata Dasar

Proses mendapatkan kata dasar dilakukan dengan pengecekan kata, jika kata terdapat dalam kamus maka kata dasar telah ditemukan, jika tidak maka akan dilakukan proses *stemming* dengan menggunakan algoritma Nazief Andriani. Proses mendapatkan kata dasar ditunjukkan pada *Source code* 4.6

```
public string KataDasar(string kata){
  setKata(kata);
    if (cekKamus(kata))
       {
          return akarKata;
       }
    else
       {
          hapusInfleksionalSuffiks();
          hapusDerivationSuffiks();
       }
          return akarKata;
     }
```

Source code 4. 6 Kata Dasar

e. Menghapus Infleksional Suffixes

Menghapus *infleksional suffixes* merupakan proses awal pada algoritma *stemming* Nazief-Andriani. Proses hapus *infleksional suffixes* ditunjukkan pada *Source code* 4.7

```
public void hapusInfleksionalSuffiks(){
   if (kata.EndsWith("lah") || kata.EndsWith("kah") ||
   kata.EndsWith("nya") || kata.EndsWith("tah") ||
   kata.EndsWith("pun"))
            {
               kata = kata.Substring(0, kata.Length - 3);
            }
    else if (kata.EndsWith("ku") || kata.EndsWith("mu"))
            {
               kata = kata.Substring(0, kata.Length - 2);
            }
       }
```

Source code 4. 7 Menghapus infleksional suffiks

f. Hapus Derivation Suffixes

Source code hapus *derivation suffixes* ditunjukkan pada *Source code* 4.8

```
public void hapusDerivationSuffiks() {
   isHapusSuffix = false;
    if (kata.EndsWith("i"))
     {
       bersikan = kata.Substring(0, kata.Length - 1);
       isHapusSuffix = true;
     }
       else if (kata.EndsWith("kan"))
     {
       bersikan = kata.Substring(0, kata.Length - 3);
       isHapusSuffix = true;
     }
       else if (kata.EndsWith("an"))
     {
       bersikan = kata.Substring(0, kata.Length - 2);
       isHapusSuffix = true;
            }
            if (cekKamus(bersikan))
            {
                akarKata = bersikan;
            }
            else
            {
                akarKata = kata;
                if (isHapusSuffix == true)
                {
                    derivationPrefiksA();
                }
                else
                {
                   derivationPrefiksB();
                }
            }
     }
```

Source code 4. 8 Hapus *derivation suffixes*

g. Tipe Awalan (derivation prefix A)

Source code cek tipe awalan dari *derivation prefix awal* ditunjukkan pada *Source code* 4.9

```
private void derivationPrefiksA(){
  bool tipe1 = (akarKata.StartsWith("be") &&
akarKata.EndsWith("i"));
  bool tipe2 = (akarKata.StartsWith("di") &&
akarKata.EndsWith("an"));
  bool tipe3 = (akarKata.StartsWith("ke") &&
(akarKata.EndsWith("i") || akarKata.EndsWith("kan")));
  bool tipe4 = (akarKata.StartsWith("me") &&
akarKata.EndsWith("an"));
  bool tipe5 = (akarKata.StartsWith("se") &&
(akarKata.EndsWith("i") || akarKata.EndsWith("kan")))

if (((tipe1) || (tipe2) || (tipe3) || (tipe4) || (tipe5))
== false)
 {
   derivationPrefiksB();
 }
}
```

Source code 4. 9 *derivation prefix* awal

h. Tipe Awalan (derivation prefix B)

Source code cek tipe awalan dari *derivation prefix* kedua ditunjukkan pada *Source code* 4.10

```
private void derivationPrefiksB(){
    if ((akarKata.StartsWith("di") ||
akarKata.StartsWith("ke") || akarKata.StartsWith("se"))
&& akarKata.Length > 2)
     {
                akarKata = akarKata.Substring(2);
     }
    else if ((akarKata.StartsWith("pe")) &&
akarKata.Length > 2)
     {
       string kataku = akarKata.Substring(2);
          if (kataku.Length>1)
            {
               if ((kataku[0] == 'r' && kataku[1] == 'h')
|| (kataku[0] == 'r' && kataku[1] == 'g') || (kataku[0]
```

```
== 'r' && kataku[1] == 'k') || (kataku[0] == 'r' &&
vowel(kataku[1]))|| (kataku[0] == 'n' && kataku[1] ==
'j') || (kataku[0] == 'n' && kataku[1] == 'd')||
(kataku[0] == 'n' && kataku[1] == 'c') || (kataku[0] ==
'n' && kataku[1] == 'z') || (kataku[0] == 'm' &&
kataku[1] == 'b')|| (kataku[0] == 'm' && kataku[1] ==
'f') || (kataku[0] == 'r' && kataku[1] == 'v')||
(kataku[0] == 'l' && (vowel(kataku[1]))) || (kataku[0] ==
'm' && (vowel(kataku[1])))|| (kataku[0] == 'n' &&
(vowel(kataku[1]))) || (kataku[0] == 'r' &&
(vowel(kataku[1])))|| (kataku[0] == 'w' &&
(vowel(kataku[1]))) || (kataku[0] == 'y' &&
(vowel(kataku[1])))
                {
                  akarKata = kataku.Substring(1);
                 }
               else
                 {
                   akarKata = akarKata.Substring(2);
                 }
             }
            }
   else if ((akarKata.StartsWith("me")))
     {
   string kata0 = akarKata.Substring(2);
     if (kata0.Length>1)
       {
           if (((kata0[0] == 'm') && (kata0[1] == 'p'))
|| ((kata0[0] == 'm') && (kata0[1] == 'b')) || ((kata0[0]
== 'n') && (kata0[1] == 'c'))|| ((kata0[0] == 'n') &&
(kata0[1] == 'd')) || ((kata0[0] == 'n') && (kata0[1] ==
'h')) || ((kata0[0] == 'n') && (kata0[1] == 'j')))
 {
   akarKata = kata0.Substring(1);
 }
  else if (((kata0[0] == 'm') && (vowel(kata0[1]))) ||
((kata0[0] == 'l') && (vowel(kata0[1]))) ||((kata0[0] ==
'n') && (vowel(kata0[1]))) || ((kata0[0] == 'r') &&
(vowel(kata0[1]))) ||((kata0[0] == 'w') &&
(vowel(kata0[1]))))
 {
  akarKata = kata0.Substring(0);
 }
  else
 {
  akarKata = akarKata.Substring(2);
```

```
  }
   }
  else if ((akarKata.StartsWith("meng")))
    {
     string kata1 = akarKata.Substring(4);
      if (kata1.Length>1)
       {
        if (kata1[0] == 'k')
         {
           akarKata = kata1;
         }
        else if (kata1[0] == 'g')
         {
           akarKata = kata1;
         }
        else if (vowel(kata1[0]))
         {
          akarKata = kata1;
         }
        else
         {
          akarKata = akarKata.Substring(4);
         }
       }
   }
   else if ((akarKata.StartsWith("te") ))
     {
        string kata2 = akarKata.Substring(2);
         if (kata2.Length > 3)
           {
   if ((kata2[0] == 'r') && (kata2[1] == 'r'))
     {
       akarKata = kata2;
     }
   else if ((kata2[0] == 'r') && (vowel(kata2[1])))
     {
       akarKata = kata2.Substring(1);
     }
   else if ((kata2[0] == 'r') && !((kata2[1] == 'r') ||
(vowel(kata2[1])))&& (kata2[2] == 'e') && (kata2[3] ==
'r') && (vowel(kata2[4])))
     {
       akarKata = kata2.Substring(1);
     }
   else if ((kata2[0] == 'r') && !((kata2[1] == 'r') ||
(vowel(kata2[1]))) && (kata2[2] == 'e') && (kata2[3] ==
```

```
'r') && !(vowel(kata2[4])))
    {
       akarKata = kata2;
    }
  else if ((kata2[0] == 'r') && !((kata2[1] == 'r') ||
(vowel(kata2[1])))&& !(kata2[2] == 'e') && !(kata2[3] ==
'r'))
    {
      akarKata = kata2.Substring(1);
    }
  else if (!((kata2[0] == 'r') || (vowel(kata2[0]))) &&
(kata2[1] == 'e') && (kata2[2] == 'r') &&
(vowel(kata2[3])))
   {
     akarKata = kata2;
   }
 else if (!((kata2[0] == 'r') || (vowel(kata2[0])))&&
(kata2[1] == 'e') && (kata2[2] == 'r') &&
(!vowel(kata2[3])))
   {
     akarKata = kata2;
   }
 else
   {
     akarKata = kata2;
   }
  }
 }
}
```

Source code 4. 10 *derivation prefix* kedua

i. Cek huruf vocal

Source code cek huruf vokal ditunjukkan pada *Source code* 4.11

```
private bool vowel(char huruf){
 if (huruf == 'a' || huruf == 'i' || huruf == 'u'|| huruf ==
'e' || huruf == 'o')
   {
     return true;
   }
    return false;
}
```

Source code 4. 11 cek huruf vokal

4.2.3. Proses Ekstraksi *Multi-word*

Proses ekstraksi *multi-word* adalah proses untuk mendapatkan *multi-word* dari suatu dokumen. Proses ini menggunakan method Ekstraksi() dengan parameter dokumen yang akan diekstrak. Proses ini akan membandingkan setiap kalimat dengan kalimat lain dalam suatu dokumen, ketika ditemukan dua pasang kata yang sama antara dua kalimat maka akan diekstrak sebagai *multi-word* dan akan disimpan kedalam suatu *string array*. Proses ekstraksi *multi-word* ditunjukkan pada *source code* 4.11.

```
public string [] Ekstraksi(string[] dokumenEkstraksi){
   string hasil = "", Kata1Kalimat1 = "", Kata2Kalimat1 =
"", Kata1Kalimat2 = "", Kata2Kalimat2 = "";
    string[] kalimat1 = null, kalimat2 = null;
     for (int i = 0; i < dokumenEkstraksi.Length; i++)
      {
       if(dokumenEkstraksi[i] != null)
            kalimat1 = dokumenEkstraksi[i].Split(new
string[] { " " }, StringSplitOptions.RemoveEmptyEntries);
        for (int j = i+1; j < dokumenEkstraksi.Length; j++)
            {
             if (dokumenEkstraksi[j] != null)
                  kalimat2 = okumenEkstraksi[j].Split(new
string[] { " " }, StringSplitOptions.RemoveEmptyEntries);
              for (int k = 0; k < kalimat1.Length; k++)
               {
                 Kata1Kalimat1 = kalimat1[k];
                  if (k + 1 == kalimat1.Length)
                     Kata2Kalimat1 = null;
                  else
                     Kata2Kalimat1 = kalimat1[k + 1];
                  for (int l = 0; l < kalimat2.Length; l++)
                   {
                     Kata1Kalimat2 = kalimat2[l];
                      if (l + 1 == kalimat2.Length)
                         Kata2Kalimat2 = null;
                      else
                         Kata2Kalimat2 = kalimat2[l + 1];
                       if ((Kata2Kalimat1 != null) ||
(Kata2Kalimat2 != null)){
                         if (Kata1Kalimat1 ==
Kata1Kalimat2 && Kata2Kalimat1 == Kata2Kalimat2)
                           hasil = hasil +
```

```
Convert.ToString(Kata1Kalimat2 + " " + Kata2Kalimat2 +
"\r\n");
                                }
                            }
                        }
                    }
                }
string[] frasa = hasil.Split(new string[] { "\r\n" },
StringSplitOptions.RemoveEmptyEntries);
string[] temuFrasa = getMWUnik(frasa).Split(new string[] {
"\r\n" }, StringSplitOptions.RemoveEmptyEntries);
                        return temuFrasa;          }
```

Source code 4. 12 Ekstraksi *multi-word*

4.2.4. Proses Perhitungan TF(Term Frekuensi) *Multi-word*

Dalam proses ini akan dihitung frekuensi kemunculan multi word dari masing-masing dokumen pada kategori tertentu. Proses perhitungan TF ditunjukkan pada *source code* 4.12.

```
public double [,] getFrekuensiFrasaKategori(string [,]
isiKategori, string [] getFrasa, int pjgKategori, int
pjgIsiKategori) {
  double[,] count = new double[pjgKategori,getFrasa.Length];
    for (int i = 0; i < pjgKategori; i++)
     {
       for (int j = 0; j < pjgIsiKategori; j++)
        {
         for (int k = 0; k < getFrasa.Length; k++)
          {
           if (getFrasa[k].Equals(isiKategori[i, j]))
             count [i,k]+=1;
          }
        }
    }
     return count;
}
```

Source code 4. 13 perhitungan TF

4.2.5. Proses Perhitungan Probabilitas *Multi-word*

Dalam proses ini akan dihitung nilai probabilitas dari setiap *multi-word*. Probabilitas dihitung menggunakan persamaan 2.10. Proses perhitungan Probabilitas ditunjukkan pada *source code* 4.13.

```
public double[,] getProbTerm (int [] totfrasakate, double [,]
frasakate, int kate, int frasa, string [] getFrasa){
  double[,] hasil = new double[kate, frasa];
    for (int i = 0; i < kate-1; i++)
     {
       for (int j = 0; j < frasa; j++)
        {
         if(j<frasakate.Length)
           hasil[i, j] = (frasakate[i, j]+1.0 /
(totfrasakate[i]+getFrasa.Length);
        }
     }       return hasil;   }
```

Source code 4. 14. Perhitungan Probabilitas

4.2.6. Proses Multinomial Naive Bayes(MNB)

Multinomial Naive Bayes merupakan metode yang digunakan untuk melakukan proses klasifikasi. *MNB* merupakan salah satu variasi lain dari metode *naïve bayes*. Model *MNB* mengambil frekuensi jumlah kata(*multi-word*) yang muncul pada sebuah dokumen. Ada beberapa sub proses yang menyusun proses MNB, sub proses tersebut akan ditunjukkan pada tabel 4.2.

Tabel 4. 3 Sub proses *Multinomial Naive Bayes(MNB)*

Sub Proses	**Keterangan**
Probabilitas Kategori	Mendapatkan nilai probabilitas setiap kategori.
Probabilitas Kategori Berdasarkan Uji	Mendapatkan nilai probabilitas kategori berdasarkan multi-word dokumen uji.
Peluang Total Uji	Mendapatkan nilai total peluang dari dokumen uji.
Klasifikasi Kategori	Melakukan proses klasifikasi pada setiap kategori.

1. Probabilitas Kategori

Dalam proses ini kan dihitung nilai peluang dari setiap kategori berdasarkan perbandingan dokumen dalam kategori tersebut

dengan total dokumen latih yang digunakan. Proses perhitungan probabilitas kategori ditunjukkan pada *source code* 4.14.

```
public double[] getProbKate(string[] dapatkategori) {
  int[] probkate = new int[4];
  int jmlDL = dapatkategori.Length-1;
  double [] hasil = new double[probkate.Length];
  for (int i = 0; i < dapatkategori.Length; i++)
   {
     if (dapatkategori[i] == "Bisnis")
       probkate[0] += 1;
     else if (dapatkategori[i] == "Edukasi")
       probkate[1] += 1;
     else if (dapatkategori[i] == "Olahraga")
       probkate[2] += 1;
     else if (dapatkategori[i] == "Sains")
       probkate[3] += 1;
   }
    int j = 0;
     foreach (double item in probkate)
       {
         hasil[j] = item / jmlDL;
         j += 1;
       }
    return hasil;
}
```

Source code 4. 15. Probabilitas Kategori

2. Probabilitas Kategori Berdasarkan Uji

Dalam proses ini akan dihitung nilai peluang kategori berdasarkan kemunculan *multi-word* pada dokumen uji. Frekuensi multi-word yang muncul dari dokumen uji akan dijadikan acuan sebagai nilai pangkat dari perkalian nilai probabilitas *multi-word* pada setiap kategori. Proses perhitungan probabilitas kategori berdasarkan uji ditunjukkan pada *source code* 4.15.

```
public double[] getDUprobDL(double[,] frek, double[,] prob,
int kate, int frasa) {
 double [] hasil= new double [kate-1];
 double [] itemku = new double [frek.Length];
 for (int i = 0; i < hasil.Length; i++)
  {
   hasil[i] = 1;
    for (int k = 0; k < frasa; k++)
     {
```

```
        if (frek[hasil.Length, k] != 0)
         {
       hasil[i] *= Math.Pow(prob[i, k], frek[hasil.Length, k]);
         }
       }
  }
```

Source code 4. 16. Probabilitas Kategori Berdasarkan Uji

3. Peluang Total Uji

Peluang total uji didapatkan dari penjumlahan hasil kali pada proses probabilitas kategori dan proses probabilitas kategori berdasarkan uji. Proses perhitungan peluang total uji ditunjukkan pada *source code* 4.16.

```
public double probDU(double[] probkate, double[]
probDUDL) {
 double hasilprobDU = 0;
  for (int i = 0; i < probkate.Length; i++)
   {
     hasilprobDU += (probkate[i] * probDUDL[i]);
   }
     return hasilprobDU;
 }
```

Source code 4. 17 Peluang total uji

4. Klasifikasi Kategori

Proses akan mendapatkan nilai peluang akhir dari masing-masing kategori. Perhitungan dilakukan dengan mengambil nilai dari tiga proses perhitungan sebelumnya. Proses perhitungan klasifikasi kategori ditunjukkan pada *source code* 4.17.

```
public double[] getKlasifikasi(double hasilprobDU, double[]
probkate, double[] probDUDL) {
  double [] hasilKlasifikasi = new double [probkate.Length];
   for (int i = 0; i < probkate.Length; i++)
    {
      hasilKlasifikasi[i] = (probkate[i] * probDUDL[i]) /
hasilprobDU;
     }
    return hasilKlasifikasi;
  }
```

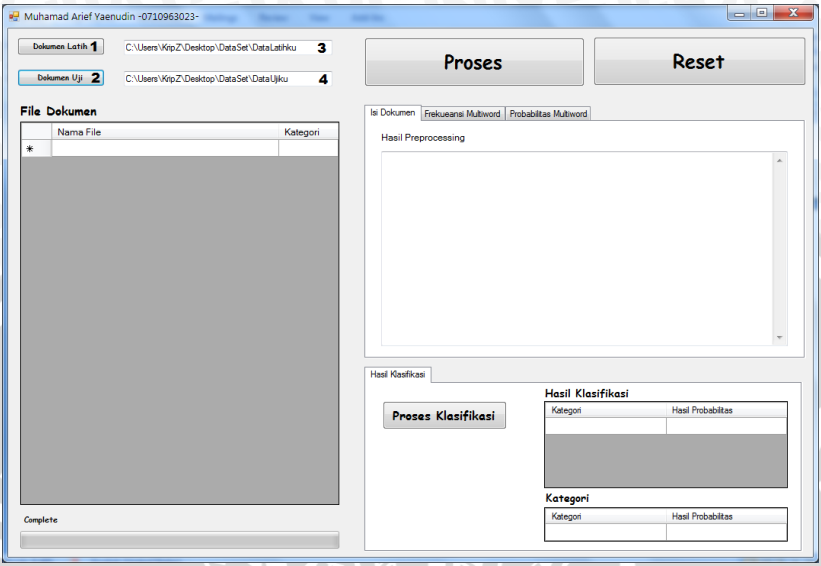
Source code 4. 18 Perhitungan klasifikasi kategori

4.3. Implementasi Antar Muka

Implementasi antar muka sistem klasifikasi dokumen berbahasa Indonesia berbasis *multi-word* menggunakan metode *Multinomial Naive Bayes*(MNB) ini terdiri dari satu *form* utama yang berisi *field* paramater dan direktori serta beberapa tab. Terdapat tiga tombol utama yaitu tombol Proses,tombol Proses Pembelajaran, dan tombol Proses Klasifikasi. Hasil dari tombol Proses ditampilkan pada tabel File Dokumen. Untuk tombol Proses Pembelajaran, hasil akan ditampilkan pada tab Frekuensi Multiword dan tab Probabilitas Multiword. Sedangkan hasil dari tombol Proses Klasifikasi ditampilkan pada tab Hasil Klasifikasi. Proses akan berjalan secara berurutan (sequential) berdasarkan tiga tombol utama tersebut.

Field yang perlu diisi sebelum menekan tombol Proses adalah *field* dokumen latih dan *field* dokumen uji. *Field* dokumen latih dan dokumen uji diisi dengan cara menekan tombol Dokumen Latih dan tombol Dokumen Uji yang berada disebelah masing – masing *field* dan memilih di direktori atau folder mana dokumen – dokumen latih dan dokumen – dokumen uji disimpan. Judul dokumen – dokumen latih dan dokumen uji beserta masing-masing kategorinya ditampilkan pada tabel yang ada pada tabel File Dokumen.

Apabila judul berita pada tabel File Dokumen disorot maka *text area* pada tab Dokumen akan menampilkan isi berita dari dokumen yang disorot tersebut. Tampilan saat penentuan *directory* dokumen uji dan dokumen latih yang akan diproses dapat dilihat pada gambar 4.1.



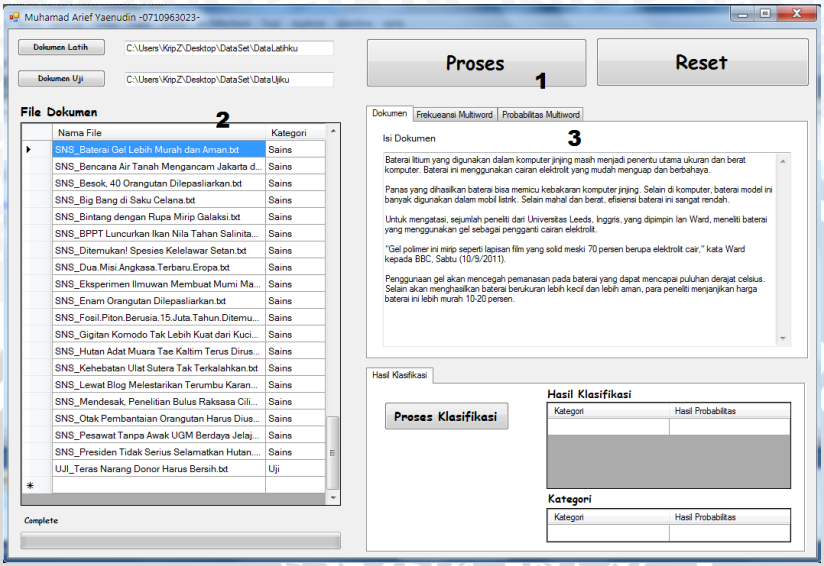
Gambar 4. 1 Tampilan *directory* dokumen uji dan dokumen latih

Keterangan gambar :

1. Tombol untuk mencari letak direktori dokumen latih.
2. Tombol untuk mencari letak direktori dokumen uji.
3. *Field* yang menampilkan direktori dokumen latih.
4. *Field* yang menampilkan direktori dokumen uji.

Setelah proses penentuan *directory* dokumen latih dan dokumen uji , selanjutnya semua file didalam dokumen latih dan dokumen uji akan ditampilkan dalam table File Dokumen. Proses penampilan file dokumen latih dan dokumen uji dilakukan dengan penggunakan tombol Proses. Selanjutnya akan tampil nama dan kategori dari semua dokumen. Untuk dokumen uji akan diberikan kategori "Uji" untuk membedakan dengan dokumen latih.

Apabila judul berita pada tabel File Dokumen disorot maka *text area* pada tab Dokumen akan menampilkan isi berita dari dokumen yang disorot tersebut. Tampilan ketika tombol Proses dijalankan dapat dilihat pada gambar 4.2.



Gambar 4. 2 Tampilan tombol Proses dijalankan

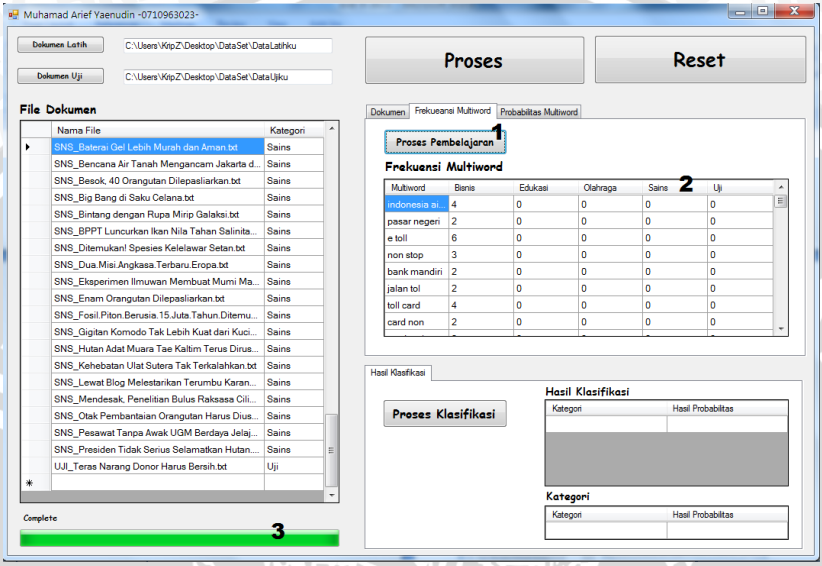
Keterangan gambar :

1. Tombol Proses yang akan digunakan untuk menampilkan dokumen latih dan dokumen uji.
2. Tabel File Dokumen yang berisi nama dokumen beserta kategori dokumen.
3. Tab Dokumen yang digunakan untuk menampilkan isi dokumen yang dipilih.

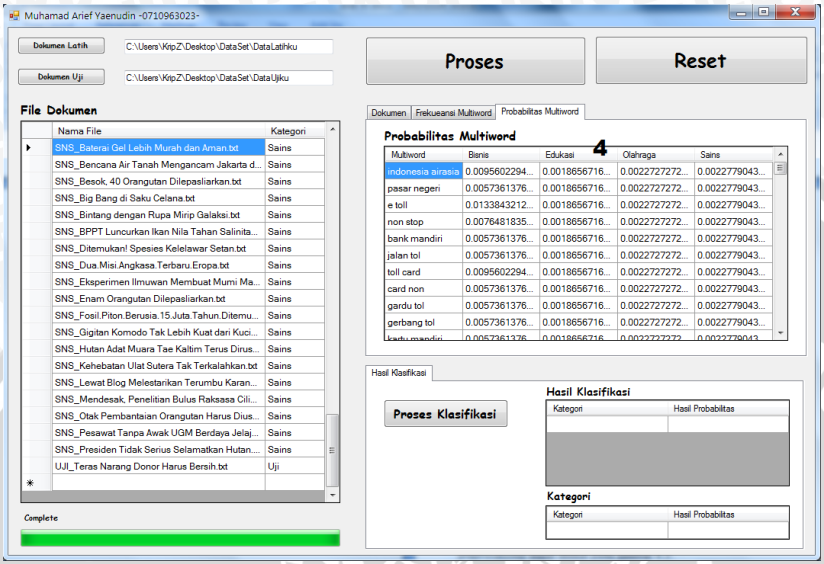
Pada tab Frekuensi Multiword , terdapat tombol Proses Pembelajaran. Tombol ini digunakan untuk melakukan pembelajaran terhadap semua dokumen yang ada didalam tabel File Dokumen. Ada dua hasil yang akan didapatkan dalam proses ini, yang pertama adalah hasil frekuensi *multi-word* yang berhasil ditemukan pada setiap kategori. Kedua adalah hasil probabilitas *multi-word* yang berhasil dihitung pada setiap kategori.

Dalam proses pembelajaran ini terdapat progress bar yang akan berjalan, progress bar ini digunakan untuk mengetahui proses apa saja yang sedang terjadi pada saat proses pembelajaran berlangsung, selain itu manfaat dari progress bar adalah untuk mengetahui bahwa sistem masih tetap melakukan suatu proses.

Tampilan untuk hasil frekuensi *multi-word* dapat dilihat pada gambar 4.3. dan tampilan untuk hasil probabilitas *multi-word* dapat dilihat pada gambar 4.4.



Gambar 4. 3 Tampilan untuk hasil frekuensi *multi-word*

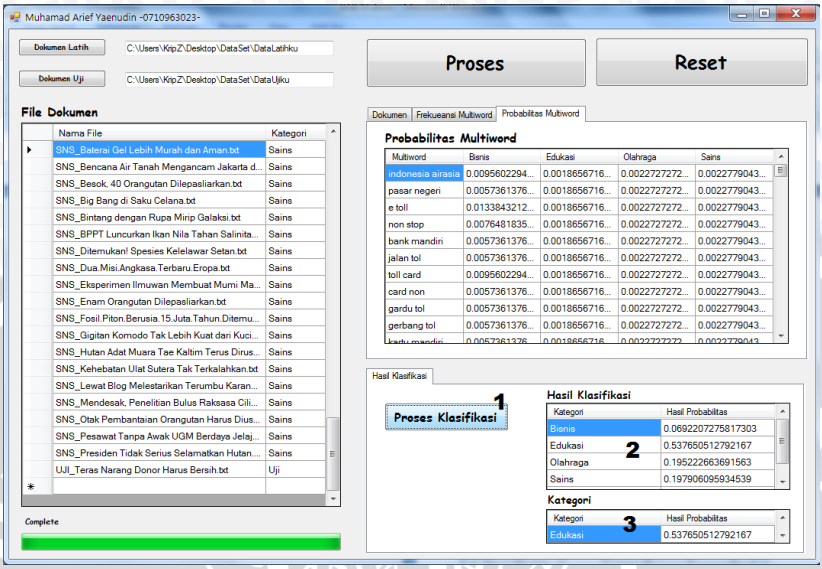


Gambar 4. 4 Tampilan untuk hasil probabilitas *multi-word*

Keterangan gambar :

1. Tombol Proses Pembelajaran yang akan digunakan untuk menjalankan proses pembelajaran pada seluruh dokumen di dalam tabel File Dokumen.
2. Tabel Frekuensi *Multiword* digunakan untuk menampilkan hasil dari frekuensi multiword yang berhasil ditemukan.
3. *Progress Bar* akan menampilkan proses yang berjalan saat proses pembelajaran.
4. Tabel Probabilitas *Multiword* digunakan untuk menampilkan hasil perhitungan probabilitas multiword.

Klasifikasi dokumen akan ditampilkan pada tab Hasil Klasifikasi dengan menjalankan tombol Proses Klasifikasi. Terdapat dua hasil yang akan ditampilkan pada proses klasifikasi. Pertama adalah hasil klasifikasi untuk semua kategori, sehinga setiap kategori memiliki nilai probabilitas klasifikasi dokumen uji berdasarkan kategori tertentu. Kedua adalah menentukan nilai probabilitas terbesar dari semua kategori. Tampilan Proses Klasifikasi dapat dilihat pada gambar 4.5



Gambar 4. 5 Tampilan untuk Proses Klasifikasi

Keterangan gambar :

1. Tombol Proses Klasifikasi yang akan digunakan untuk menjalankan proses klasifikasi pada dokumen uji.
2. Tabel Hasil Klasifikasi digunakan untuk menampilkan hasil klasifikasi untuk semua kategori.
3. Tabel Kategori akan menampilkan nilai klasifikasi terbesar beserta kategorinya.

4.4. Implementasi Uji Coba

Pada sub bab ini akan dibahas mengenai implementasi metode pengujian yang telah dilakukan oleh sistem dan dari hasil pengujian yang berdasarkan pada sistem.

4.4.1. Skenario Evaluasi

Pengujian sistem klasifikasi dokumen berbahasa Indonesia berbasis *multi-word* ini dilakukan dengan menggunakan 360 dokumen berita berbahasa Indonesia yang dibagi menjadi 300

dokumen digunakan sebagai dokumen latih dan 60 dokumen digunakan sebagai dokumen uji. Persebaran dokumen pada setiap kategori dapat dilihat pada tabel 4.2 dan tabel 4.3.

Tabel 4. 4 Jumlah Dokumen Latih

Kategori	**Jumlah Dokumen**
Bisnis	75
Edukasi	75
Olahraga	75
Sains	75
Total	**300**

Tabel 4. 5 Jumlah Dokumen Uji

Kategori	**Jumlah Dokumen**
Bisnis	15
Edukasi	15
Olahraga	15
Sains	15
Total	**60**

Terdapat tiga kali pengujian yang dilakukan. Pengujian yang pertama adalah pengujian dengan menggunakan data latih sebesar 100 buah dimana setiap kategori terdapat 25 buah dokumen latih ,uji coba kedua digunakan data latih sebesar 200 buah dimana setiap kategori terdapat 50 buah dokumen latih dan uji coba ketiga digunakan data latih sebesar 300 buah dimana setiap kategori terdapat 75 buah dokumen latih. Hal ini dilakukan agar hasil pengujian yang dilakukan terbukti dengan baik dan valid.

4.4.2. Hasil Evaluasi dan Analisis

Hasil uji coba terhadap sistem dilakukan untuk mengetahui kinerja dari sistem yang dibangun. Uji coba sistem dilakukan terhadap inputan dokumen uji. Terdapat tiga kali pengujian berdasarkan banyak dokumen latih yang berbeda-beda. Hasil evaluasi klasifikasi ditunjukkan pada tabel 4.5 sampai dengan 4.7.

Tabel 4. 6 Evaluasi Klasifikasi Uji Coba Pertama (100 data latih)

Kategori	a	b	c	Precesion	Recall	F1-Mesure
Bisnis	7	0	8	1	0.466667	0.636364
Edukasi	9	1	6	0.9	0.6	0.72
Olahraga	15	22	0	0.405405	1	0.576923
Sains	6	0	9	1	0.4	0.571429
Rata-rata				**0.826351**	**0.616667**	**0.626179**

Keterangan :

a. Dokumen yang termasuk dalam hasil klasifikasi oleh sistem memang merupakan anggota klasifikasi.
b. Dokumen yang termasuk dalam hasil klasifikasi oleh sistem ternyata seharusnya bukan merupakan anggota klasifikasi.
c. Dokumen yang tidak termasuk dalam hasil klasifikasi oleh sistem ternyata seharusnya merupakan anggota klasifikasi.

Tabel 4. 7 Evaluasi Klasifikasi Uji Coba Pertama (200 data latih)

Kategori	a	b	c	Precesion	Recall	F1-Mesure
Bisnis	6	0	9	1	0.4	0.571429
Edukasi	12	2	3	0.857143	0.8	0.827586
Olahraga	15	16	0	0.483871	1	0.652174
Sains	9	0	6	1	0.6	0.75
Rata-rata				**0.835253**	**0.7**	**0.700297**

Tabel 4. 8 Evaluasi Klasifikasi Uji Coba Pertama (300 data latih)

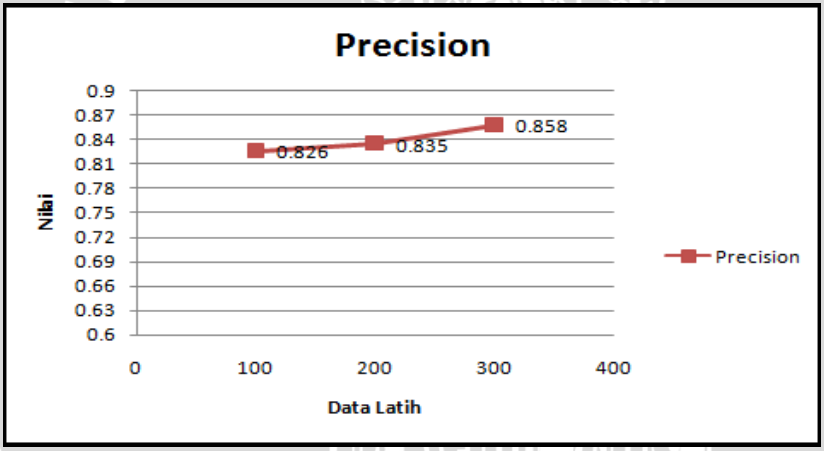
Kategori	a	b	c	Precesion	Recall	F1-Mesure
Bisnis	11	0	4	1	0.733333	0.846154
Edukasi	11	1	4	0.916667	0.733333	0.814815
Olahraga	15	14	0	0.517241	1	0.681818
Sains	8	0	7	1	0.533333	0.695652
Rata-rata				0.858477	0.75	0.75961

Maka dari tabel evaluasi klasifikasi tiga kali uji coba diatas dapat dibuat tabel rata-rata nilai *precision*, *recall* dan *F1 measure* seperti pada tabel 4.8.

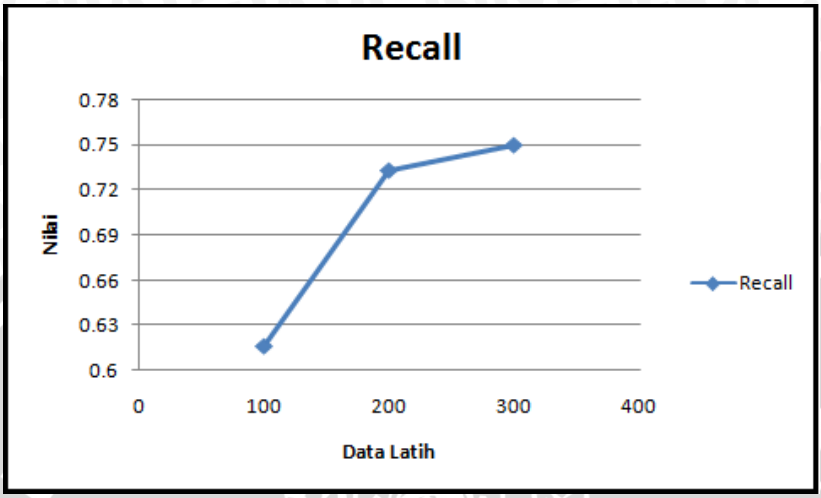
Tabel 4. 9 Hasil Rata-rata Evaluasi Uji Coba

Jumlah Data Uji	Precision	Recall	F1-Mesure
100	0.826351	0.616667	0.626179
200	0.835253	0.733333	0.731353
300	0.858477	0.75	0.75961
Rata-rata	0.840027	0.7	0.705714

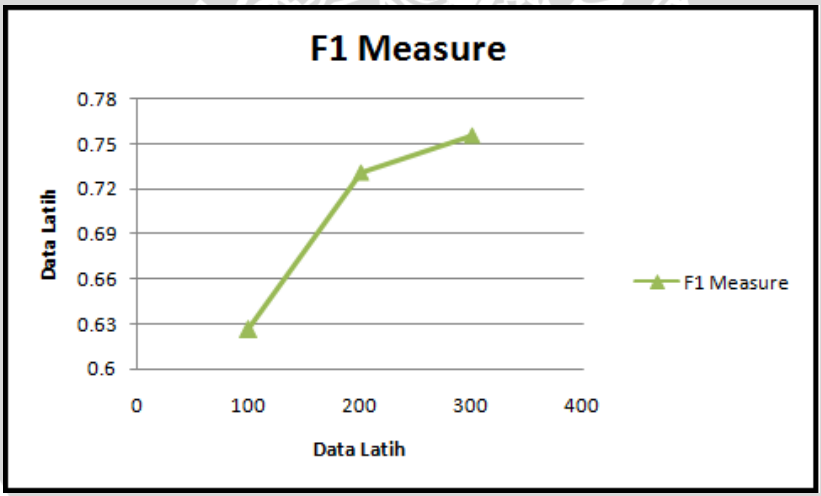
Dari hasil tabel 4.8 dapat disajikan dalam bentuk grafik seperti pada Gambar 4.6.



Gambar 4. 6 Grafik Hasil Evaluasi Klasifikasi Precision



Gambar 4. 7 Grafik Hasil Evaluasi Klasifikasi Recall



Gambar 4. 8 Grafik Hasil Evaluasi Klasifikasi

4.4.3. Analisa Hasil

Dari hasil evaluasi klasifikasi, didapatkan nilai rata-rata *precision, recall* dan *F1 measure* yang berbeda untuk jumlah data latih yang berbeda. Dengan semakin banyak jumlah data latih yang digunakan dalam tahap pembelajaran maka semakin meningkatkan nilai rata-rata *precision, recall* dan *F1 measure*.

Pada percobaan yang telah dilakukan, peningkatan terbesar nilai *precision, recall* dan *F1 measure* terjadi pada saat data latih ditambah dari 100, 200 sampai dengan 300 buah. Hasil pengenalan kembali data latih yang telah dipelajari juga memperlihatkan bahwa pada saat penambahan data latih menjadi 300 buah, jumlah dokumen yang salah diklasifikasikan berkurang secara signifikan. Sehingga dapat disimpulkan bahwa pada jumlah data latih sebesar 300 buah sudah cukup untuk dapat mengklasifikasikan dokumen dengan baik karena nilai akurasi yang didapat sudah mencapai diatas 70%.

Dengan demikian dapat dikatakan bahwa pada jumlah data latih semakin besar, sebesar 300 buah, sistem masih dapat bekerja lebih optimal. Dari tiga kali uji coba didapatkan rata-rata dan nilai *precision* sebesar 0.840027 dan nilai *recall* sebesar 0.7 serta nilai *F1 measure* sebesar 0.705714, sehingga dapat disimpulkan bahwa tingkat akurasi sistem secara rata-rata sudah berjalan dengan baik.

BAB V
PENUTUP

5.1. Kesimpulan

Kesimpulan penelitian klasifikasi dokumen berita berbahasa Indonesia berbasis *multi-word* dengan metode *Multinomial Naïve Bayes (MNB)* yang datanya bersumber dari www.kompas.com, adalah sebagai berikut :

1. Sistem klasifikasi berbahasa Indonesia berbasis *multi-word* menggunakan metode *Multinomial Naive Bayes*(MNB) dilakukan dengan 3 tahapan utama yaitu yang pertama proses penentuan dokumen uji dan dokumen latih. Tahap kedua adalah proses pembelajaran, dalam tahap ini akan dilakukan beberapa sub proses seperti *preprocessing*, ekstraksi *multi-word*, frekuensi *multi-word*, probabilitas *multi-word*. Tahap Ketiga adalah proses klasifikasi dimana dokumen uji diklasifikasikan pada suatu kategori/kelas sesuai hasil yang diperoleh dengan menggunakan metode *Multinomial Naive Bayes*(MNB).
2. Dari hasil 3 kali pengujian, sistem ini menghasilkan nilai rata-rata *precision* sebesar 0.840027, rata-rata *recall* sebesar 0.7 dan rata-rata F_1 *measure* sebesar 0.705714.
3. Sistem ini mengalami peningkatan akurasi yang paling signifikan pada saat data latih sebanyak 300 buah. Sehingga jumlah data latih sebesar 300 buah sudah cukup untuk dapat mengklasifikasikan dokumen dengan baik.

5.2. Saran

Pada penelitian ini masih ada beberapa hal yang dapat dikembangkan dan digunakan untuk penelitian selanjutnya yaitu antara lain :

1. Pembobotan *term* tidak hanya dilakukan pada isi berita saja akan tetapi juga pada judul berita.
2. *Jumlah multi-word* yang digunakan lebih variatif dengan memperhitungkan keterkaitan antar kata.

3. Melakukan *proof reading* untuk membetulkan kata – kata yang mengalami kesalahan pengetikan.
4. Melakukan pengolahan dokumen secara langsung yaitu dengan format dokumen **.html* atau **.htm*.
5. Memperbesar data latih yang digunakan untuk mendapatkan hasil yang lebih baik.
6. Menggunakan daftar kata dasar bahasa Indonesia yang lebih lengkap daripada yang digunakan pada skripsi ini.
7. Menggunakan metode ekstraksi multi-word yang lebih baik dari skripsi ini.
8. Menggunakan algoritma stemming yang lebih baik dari skripsi ini.
9. Memperhitungkan NLP(Natural Language Processing) dalam menghasilkan *multi-word*.
10. Menggunakan metode yang lebih baik untuk melakukan *parsing* kalimat.

DAFTAR PUSTAKA

Arifin, Agus Z dan Setiono Ari N. *Klasifikasi Dokumen Berita Kejadian Berbahasa Indonesia dengan Algoritma Single Pass Clustering*. Institut Teknologi Sepuluh Nopember(ITS). Surabaya. http://mail.itssby.edu/~agusza/SITIAKlasifikas Event.pdf.

Asian,Jelita, Wiliams, Hugh E, dan Tahaghoghi S.M.M. .2005. *Stemming Indonesian*. Australia : School of Computer Science and Information Technology.

Baldi, P, P. Frasconi, P. Smyth. 2003. *Modelling The Internet and The Web.*

Basuki, Maryono. 1983. *Teknik Mencari dan Menulis Berita.* Fakultas Ilmu Komunikasi Universitas Prof. Dr. Moestopo (beragama). Jakarta.

Chaer, Abdu. 2003. *Linguistik Umum*. Jakarta : Rineka Cipta

D. Bourigault, *Surface Grammatical Analysis for the Extraction of Terminological Noun Phrases, in: Proceedings of the 14th International Conference on Computational Linguistics,* Nantes, France, 1992, pp. 977–981.

Ernawati, Sari.,dkk.2009. *KLUSTERISASI DOKUMEN BERITA BERBAHASA INDONESIA MENGGUNAKAN DOCUMENT INDEX GRAPH*.Yogyakarta

Even, Yahir dan Zohar. 2002. *Introduction to Text Mining. Automated Learning Group National Center For Supercomputing Applications.* University of Illionis. http://algdocs.nsca.uiuc.edu/PR-2021116-2.ppt Diakses pada agustus 2011.

F. Smadja, *Retrieving collocations from text: Xtract, Computational Linguistics* 19 (1) (1993) 143–177.

Garcia, Dr. E. 2005. *The Classic Vector Space Model (Description, Advantages and Limitations of the Classic Vector Space Model)*.

I. Fahmi, C . *Value Method for Multi-word Term Extraction, Seminar in Statistics and Methodology, Alfa-informatica, RuG*, May 23, 2005. Available from: <http://odur.let.rug.nl/ fahmi/talks/statistics-c-value.pdf/>.

J.S. Chang et al., *A multiple-corpus approach to recognition of proper names in chinese texts, Computer Processing of Chinese and Oriental Languages* 8 (1) (1994) 75–85.

J. Zhang, J.F. Gao, M. Zhou, *Extraction of Chinese compound words: an experiment study on a very large corpus, in: Proceedings of the Second Chinese Language Processing Workshop*, HongKong, 2000, pp. 132–139.

Keraf, Gorys. 1994. *Tata Bahasa Indonesia/ Jakarta* : Nusa Indah.

Kibriya, A., Eibe Frank, Bernhard Pfahringer, dan Geoffrey Holmes. 2004. *Multinomial Naive Bayes for Text Categorization Revisited.* Department of Computer Science, University of Waikato, Hamilton, New Zealand..

McCallum, A dan Nigam, K. 1998. *A comparison of event models for naive Bayes text classification. Technical report, American Association for Artificial Intelligence Workshop on Learning for Text Categorization..*

Mitchell, Tom. 1997. *Machine Learning.* McGraw-Hill. Singapore..

Rachli, M. 2007. *Email Filtering Menggunakan Naïve Bayesian.* Program Studi Teknik Elektro, Institut Teknologi Bandung: Bandung.

Ramlan, M. 1987. *Ilmu Bahasa Indonesia Sintaksis.* Yogyakarta : CV.Karyono

Sebastiani, F. 2002. *Machine Learning In Automated Text Categorization*. ACM Computing Surveys, Vol34, No.1, March 2002, pages 1-47.

S. Katz, *Distribution of content words and phrases in texts and language modeling, Natural Language Engineering* 2 (1) (1996) 15–59.

Tala, Fadillah Z, 2003, *A Study of Stemming Effects on Information Retrieval in Bahasa Indonesia. Master of Logic Project*. Institute for Logic, Language and Computation. Universiteit van Amsterdam. Amsterdam.

Wahyudi, JB. 2002. *Dasar-dasar jurnalistik radio dan televise.* Perpustakaan Utan Kayu. Jakarta

Wibisono, Y dan Khodra, M. 2005. *Clustering Berita Berbahasa Indonesia*. *FPMIPA Universitas Pendidikan Indonesia*. Bandung.

www.bahasakita.com/2011/05/10/kata-frasa-dan-kalimat/. Diakses pada 29 November 2011

Zhang, Wen., Taketoshi Yoshida, Xijin Tang. 2008. *Text classification based on multi-word with support vector machine*. Institute of Systems Science, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing

LAMPIRAN

LAMPIRAN 1 : DAFTAR STOPWORD

ada	asalkan	beginian	berikan
adalah	asalnya	beginikah	berilah
adakah	atas	beginilah	berikut
adanya	atasi	beginipun	berikutnya
adapun	atasnya	begitu	berisi
agak	atau	begitukah	berjumlah
agaknya	ataukah	begitulah	berkala
agar	ataupun	begitupun	berkali
akan	awal	bekerja	berkata
akankah	awalnya	belakang	berkatalah
akhir	bagai	belakangan	berkehendak
akhiri	bagaikan	belas	berkeinginan
akhirilah	bagaimana	belum	berkenaan
akhirnya	bagaimanakah	belumlah	berlainan
aku	bagaimanapun	benar	berlalu
akulah	bagi	benarkah	berlangsung
akupun	bagian	benarlah	berlebihan
amat	bahkan	benarnya	berlebih
amati	bahwa	berada	bermaksud
amatilah	bahwasanya	berakhir	bermula
amatlah	baik	berakhirlah	bersama
anda	baiklah	berakhirnya	bersiap
andalah	baiknya	berakhirpun	bertanya
antar	bakal	berapa	berturut
antara	bakalan	berapakah	bertutur
antaranya	balik	berapalah	berujar
apa	banyak	berapapun	berupa
apaan	banyaknya	berarti	besar
apabila	bapak	berasal	besarnya
apakah	bapaknya	berawal	betul
apalagi	baru	berbagai	betulkah
apalah	barulah	berbagi	betulnya
apatah	bawah	berdatangan	biasa
arti	bawahan	berdekatan	biasalah
artinya	beberapa	berguna	biasanya
asal	begini	beri	bila

bilakah	dengan	diingatkan	dimulai
bisa	denganku	diinginkan	dimulailah
bisakah	denganmu	diinginkannya	dimulainya
boleh	dengannya	dijawab	dimungkinkan
bolehkah	depan	dijawablah	dini
bolehlah	depanku	dijawabnya	dipastikan
buat	depanmu	dijelaskan	dipastikannya
buatnya	depannya	dijelaskanlah	diperbuat
bukan	di	dijelaskannya	diperbuatnya
bukankah	dia	dikarenakan	dipergunakan
bukanlah	diakhir	dikarenakannya	dipergunakannya
bukannya	diakhiri	dikala	diperkirakan
bulan	diakhirinya	dikatakan	diperkirakannya
bulanan	diakhirilah	dikatakanlah	diperlihatkan
bung	dialah	dikatakannya	diperlihatkanlah
cara	diakah	dikeluarkan	diperlukan
caranya	dianya	dikeluarkannya	diperlukannya
cukup	diapun	dikerjakan	dipersoalkan
cukupkah	diantara	dikerjakannya	dipertanyakan
cukuplah	diantaranya	diketahui	dipunyai
cuma	diberi	diketahuilah	dipunyainya
dahulu	diberikan	diketahuinya	diri
dalam	diberikannya	dikira	dirinya
dalamnya	diberilah	dikiranya	disampaikan
dan	diberinya	dilakukan	disampaikanlah
dapat	dibuat	dilakukannya	disampaikannya
dapatlah	dibuatkan	dilakukanlah	disebut
dapatkah	dibuatkannya	dilalui	disebutkan
dari	dibuatlah	dilaluinya	disebutkannya
darinya	dibuatnya	dilihat	disini
daripada	didapat	dilihatnya	disinilah
datang	didapatkan	dimaksud	disinikah
datanglah	didatangkan	dimaksudkan	ditambahkan
datangnya	digunakan	dimaksudkannya	ditandaskan
dekat	digunakanlah	dimaksudnya	ditanya
delapan	digunakannya	diminta	ditanyai
delapannya	diibaratkan	dimintai	ditanyakan
demi	diibaratkannya	dimintanya	ditanyakanlah
demikian	diingat	dimisalkan	ditanyakannya
demikianlah	diingatnya	dimisalkannya	ditegaskan

ditujukan	hendaklah	jawaban	kebetulan
ditunjuk	hendaknya	jawablah	kebetulankah
ditunjuki	hingga	jawabnya	kecil
ditunjukkan	ia	jelas	kecilnya
ditunjukkannya	ialah	jelaskan	kedelapan
ditunjuknya	ibarat	jelaskanlah	kedua
dituturkan	ibaratkan	jelaslah	keduanya
dituturkannya	ibaratnya	jelasnya	keduluan
diucapkan	ibu	jika	keempat
diucapkannya	ibunya	jikalau	keenam
diungkapkan	ikut	juga	ketujuh
diungkapkannya	ikuti	jumlah	kesembilan
dong	ikutilah	jumlahnya	kesepuluh
dua	ingat	justru	keinginan
duanya	ingatkan	kala	keinginannya
dulu	ingatlah	kalau	kelamaan
dulunya	ingatnya	kalaulah	kelihatan
empat	ingin	kalaupun	kelihatannya
empatnya	inginkah	kalian	kelima
enam	inginkan	kalian	keluar
enamnya	inginnya	kami	kembali
enggak	ini	kamilah	kemudian
enggaknya	inikah	kamu	kemungkinan
entah	inilah	kamulah	kemungkinannya
entahlah	inipun	kan	kenapa
guna	itu	kapan	kepada
gunakan	itukah	kapankah	kepadanya
gunanya	itulah	kapanpun	kesampaian
hal	itupun	karena	keseluruhan
halnya	jadi	karenanya	keseluruhannya
hampir	jadikan	kasus	kesana
hanya	jadilah	kasusnya	kesini
hanyalah	jadinya	kata	kesitu
hari	jangan	katakan	keterlaluan
harian	jangankan	katakanlah	ketika
harus	janganlah	katanya	khusus
haruskah	jauh	ke	khususnya
haruslah	jauhnya	keadaan	kini
harusnya	jauhkah	keadaannya	kinilah
hendak	jawab	keadaannyalah	kira

kiranya	masing	mendatangkan	merekalah
kita	mau	menegaskan	merupakan
kitalah	maupun	mengakhiri	meski
kok	melainkan	mengapa	meskipun
kurang	melakukan	mengatakan	meyakini
kurangnya	melalui	mengatakannya	meyakinkan
lagi	melihat	mengatasi	minta
lagian	melihatnya	mengawali	mirip
lah	memang	mengenai	misal
lain	memastikan	mengerjakan	misalkan
lainnya	memberi	mengetahui	misalnya
lalu	memberikan	menggunakan	mula
lama	membuat	menghendaki	mulai
lamanya	memerlukan	mengibaratkan	mulailah
lanjut	memihak	mengibaratkannya	mulanya
lanjutnya	meminta	mengingat	mungkin
lebih	memintakan	mengingatkan	mungkinkah
lebihnya	memisalkan	menginginkan	nah
lewat	memperbuat	mengira	naik
lima	mempergunakan	mengucapkan	namun
limanya	memperkirakan	mengucapkannya	nanti
luar	memperlihatkan	mengungkapkan	nantinya
luarnya	mempersiapkan	menjadi	nyaris
macam	mempersoalkan	menjawab	nyatanya
macamnya	mempertanyakan	menjelaskan	oleh
maka	mempunyai	menuju	olehnya
makanya	memulai	menunjuk	pada
makin	memungkinkan	menunjuki	padahal
malah	menaiki	menunjukkan	padanya
malahan	menambahkan	menunjuknya	pak
mampu	menandaskan	menurut	paling
mampukah	menanti	menuturkan	panjang
mana	menantikan	menyampaikan	pantas
manakala	menanya	menyangkut	para
manalagi	menanyai	menyatakan	pasti
masa	menanyakan	menyebutkan	pastilah
masalah	mendapat	menyeluruh	penting
masalahnya	mendapatkan	menyiapkan	pentingnya
masih	mendatang	merasa	per
masihkah	mendatangi	mereka	percuma

perlu	sebaik	sekalipun	sepantasnya
perlukah	sebaiknya	sekarang	sepantasnyalah
perlunya	sebaliknya	sekecil	seperlunya
pernah	sebanyak	seketika	seperti
persoalan	sebegini	sekiranya	sepertinya
pertama	sebegitu	sekitar	sepihak
pertanyaan	sebelum	sekitarnya	sepuluh
pertanyakan	sebelumnya	sekurangnya	sering
pihak	sebenarnya	sela	seringnya
pihaknya	seberapa	selain	serta
pukul	sebesar	selaku	serupa
pula	sebetulnya	selalu	sesaat
pun	sebisanya	selama	sesama
punya	sebuah	selamanya	sesampai
rasa	sebut	selanjutnya	sesegera
rasanya	sebutlah	seluruh	sesekali
rata	sebutnya	seluruhnya	seseorang
rupanya	secara	semacam	sesuatu
saat	secukupnya	semakin	sesuatunya
saatnya	sedang	semampu	sesudah
saja	sedangkan	semampunya	sesudahnya
sajalah	sedemikian	semasa	setelah
saling	sedikit	semasih	seterusnya
sama	sedikitnya	semata	setiap
sambil	seenaknya	semata-mata	setempat
sampai	segala	semaunya	setengah
sampaikan	segalanya	sementara	setiba
sana	segera	semisal	setibanya
sangat	seharusnya	semisalnya	setidaknya
sangatlah	sehingga	sempat	setinggi
satu	seingat	semua	seusai
saya	sejak	semuanya	sewaktu
sayalah	sejauh	semula	siap
se	sejenak	sendiri	siapa
sebab	sejumlah	sendirian	siapakah
sebabnya	sekadar	sendirinya	siapapun
sebagai	sekadarnya	seolah	sini
sebagaimana	sekali	seolah-olah	sinilah
sebagainya	sekalian	seorang	soal
sebagian	sekaligus	sepanjang	soalnya

suatu	tempat	ternyata	tuturnya
sudah	tengah	tersampaikan	ucap
sudahkah	tentang	tersebut	ucapnya
sudahlah	tentu	tersebutlah	ujar
supaya	tentulah	tertentu	ujarnya
tadi	tentunya	tertinggi	umum
tadinya	tepat	tertuju	umumnya
tahu	terakhir	terus	ungkap
tahun	terasa	terutama	ungkapnya
tahunan	terbanyak	tetap	untuk
tak	terdahulu	tetapi	usah
tambah	terdapat	tiap	usai
tambahnya	terdiri	tiba	waduh
tampak	terhadap	tidak	wah
tampaknya	terhadapnya	tidakkah	wahai
tandas	teringat	tidaklah	waktu
tandasnya	terjadi	tiga	waktunya
tanpa	terjadilah	tinggi	walau
tanya	terjadinya	toh	walaupun
tanyakan	terkira	tujuh	wong
tanyanya	terlalu	tujuhnya	yaitu
tapi	terlebih	tunjuk	yakin
tegas	terlihat	turut	yakni
tegasnya	termasuk	tutur	yang
telah			

UNIVERSITAS BRAWIJAYA