KLASIFIKASI BERITA BERBAHASA INDONESIA MENGGUNAKAN METODE MULTINOMIAL NAÏVE BAYES

SKRIPSI

Oleh **:**

Ni'am Shofi Nurdwianto
0710962007-96

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

2010

LEMBAR PENGESAHAN SKRIPSI

KLASIFIKASI BERITA BERBAHASA INDONESIA MENGGUNAKAN METODE MULTINOMIAL NAÏVE BAYES

Oleh:
Ni'am Shofi Nurdwianto
0710962007-96

Setelah dipertahankan di depan Majelis Penguji
pada tanggal 15 Desember 2010
dan dinyatakan memenuhi syarat untuk memperoleh gelar
Sarjana Komputer dalam bidang Ilmu Komputer

Pembimbing I,	Pembimbing II,
Drs. A. Ridok, M.Kom	**Nurul Hidayat, SPd.,MSc**
NIP. 196808251994031002	**NIP. 196804302002121001**

Mengetahui,
Ketua Jurusan Matematika
Fakultas MIPA Universitas Brawijaya

Dr.Agus Suryanto, MSc
NIP. 196908071994121001

LEMBAR PERNYATAAN

Saya yang bertanda tangan di bawah ini :

Nama	**:**	Ni'am Shofi Nurdwianto
NIM	**:**	0710962007-96
Jurusan	**:**	Matematika
Program Studi	**:**	Ilmu Komputer
Penulis Skripsi berjudul	**:**	Klasifikasi Berita Berbahasa Indonesia Menggunakan Metode Multinomial Naïve 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 dikemudian 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, 15 Desember 2010
Yang menyatakan,

Ni'am Shofi Nurdwianto
NIM. 0710962007-96

KLASIFIKASI BERITA BERBAHASA INDONESIA MENGGUNAKAN METODE MULTINOMIAL NAÏVE BAYES

ABSTRAK

Klasifikasi dokumen berita adalah suatu proses untuk mengklasifikasikan suatu artikel dokumen berita berbahasa Indonesia ke dalam suatu kategori tertentu. Tujuannya adalah untuk membantu pembaca dalam mencari dokumen sehingga dapat menghemat waktu pembaca dengan menghindari pencarian dokumen yang tidak relevan.

Sistem ini mengimplementasikan metode *MNB* (*Multinomial Naïve Bayes*) untuk mengklasifikasikan sebuah dokumen, dimana dokumen yang digunakan berasal dari situs media *online* surat kabar Kompas.com dan Detik.com. Tahapan-tahapan yang dilakukan dalam sistem ini adalah, pertama dilakukan proses *case folding* yaitu mengubah semua huruf menjadi huruf kecil, tahap kedua yaitu penguraian kata (*tokenizing*), tahap ketiga yaitu mengambil kata-kata yang penting dan penghilangan *stopword* (*filtering*), tahap keempat yaitu mereduksi kata ke bentuk dasarnya (*stemming*), tahap kelima yaitu perhitungan frekuensi dari masing-masing kata, dan tahap terakhir yaitu klasifikasi menggunakan metode *MNB* (*Multinomial Naïve Bayes*). Untuk mengevaluasi efektifitas sistem pengklasifikasian dokumen, digunakan standar pengukuran *precision*, *recall*, dan F_1 *Measure*.

Hasil pengujian dan evaluasi menunjukkan bahwa sistem ini menghasilkan nilai rata-rata *precision* sebesar 0.860, rata-rata *recall* sebesar 0.864 dan rata-rata F_1 *measure* sebesar 0.860. Evaluasi dilakukan untuk mengetahui pengaruh jumlah data latih terhadap efektifitas dari sistem.

INDONESIAN NEWS CLASSIFICATION
USING MULTINOMIAL NAÏVE BAYES METHOD

ABSTRACT

The documents classification is a process for classifying a Indonesian news articles document into a particular category. The objective is to assist the reader in search of documents so that readers can save time by avoiding the search for documents that are not relevant.

This system implements the method MNB (Multinomial Naïve Bayes) to classify a document, where documents were obtained from online newspaper Kompas.com and Detik.com.There are some steps in this system, first step is the case folding that changes all letters to lowercase, the second step is parsing the word (tokenizing), the third step is to take the unique words and stopword removal (filtering), the fourth step is reduce the word to its basic form (stemming), the fifth step is counting each term frequency, and the last step is classification using MNB (Multinomial Naïve Bayes) method. To evaluate the effectiveness of document classification system, using standard measurement precision, recall, and F_1 Measure.

Test and evaluation result show that this system produces an average precision 0.860, average recall 0.864 and average F_1 measure 0.860. The evaluation used to determine the effect of training data on the effectiveness of the system.

KATA PENGANTAR

Alhamdulillahi rabbil 'alamin. Puji syukur penulis panjatkan kehadirat Allah SWT, karena atas segala rahmat dan limpahan hidayahnya, Skripsi yang berjudul "Klasifikasi Berita Berbahasa Indonesia menggunakan Metode *Multinomial Naïve Bayes*" ini dapat diselesaikan. Skripsi ini disusun dan diajukan sebagai syarat untuk memperoleh gelar sarjana pada program studi Ilmu Komputer, jurusan Matematika, fakultas MIPA, universitas Brawijaya.

Semoga Allah melimpahkan rahmat atas Nabi Muhammad SAW yang senantiasa memberikan cahaya petunjuk, dan atas keluarganya yang baik dan suci dengan rahmat yang berkah-Nya menyelamatkan kita pada hari akhirat.

Dalam penyelesaian Skripsi ini, penulis telah mendapat begitu banyak bantuan baik moral maupun materiil dari banyak pihak. Atas bantuan yang telah diberikan, penulis ingin menyampaikan penghargaan dan ucapan terima kasih yang sedalam-dalamnya kepada:

1. Drs. A. Ridok, M.Kom., selaku pembimbing utama penulisan skripsi ini.
2. Nurul Hidayat, SPd., MSc., selaku pembimbing pendamping dalam penulisan skripsi ini
3. Drs Marji MT., selaku Ketua Program Studi Ilmu Komputer.
4. Dr. Agus Suryanto, Msc., selaku Ketua Jurusan Matematika.
5. Segenap bapak dan ibu dosen yang telah mendidik dan mengajarkan ilmunya kepada Penulis selama menempuh pendidikan di Program Studi Ilmu Komputer Jurusan Matematika FMIPA Universitas Brawijaya.
6. Segenap staf dan karyawan di Jurusan Matematika FMIPA Universitas Brawijaya yang telah banyak membantu Penulis dalam pelaksanaan penyusunan proposal skripsi ini.
7. Orang tua Penulis atas dukungan materi dan doa restu yang tak henti – hentinya kepada Penulis.
8. Rekan - rekan di Program Studi Ilmu komputer FMIPA Universitas Brawijaya yang telah banyak memberikan bantuannya demi kelancaran pelaksanaan penyusunan Skripsi ini.
9. Dan semua pihak yang telah membantu dalam penyusunan skripsi ini yang tidak dapat Penulis sebutkan satu persatu.

Semoga penulisan laporan Skripsi ini bermanfaat bagi pembaca sekalian. Akhirnya, penulis menyadari bahwa Skripsi ini masih jauh dari kesempurnaan, dan mengandung banyak kekurangan, sehingga dengan segala kerendahan hati penulis mengharapkan kritik dan saran yang membangun dari pembaca.

Malang, Desember 2010

Penulis

DAFTAR ISI

Halaman

DAFTAR GAMBAR

Halaman

DAFTAR TABEL

Halaman

DAFTAR KODE PROGRAM

Halaman

BAB I
PENDAHULUAN

1.1 Latar Belakang

Seiring dengan semakin pesatnya perkembangan teknologi informasi maka kebutuhan konsumen terhadap informasi dari berbagai media cetak maupun elektronik akan semakin meningkat. Ketersediaan informasi yang semakin banyak ini, menuntut adanya sebuah alat yang mampu mengklasifikasikan informasi atau secara otomatis.

Dengan adanya pengklasifikasian dokumen berita secara otomatis maka akan semakin mudah dalam pencarian informasi mengenai suatu kejadian tertentu. Hal ini dapat menghemat waktu pembaca karena dapat menghindari pencarian berita yang tidak relevan. Pengaruh komputer sebagai alat untuk membantu mempermudah pekerjaan manusia pada saat ini sangat perpengaruh terhadap kinerja manusia dalam menyelesaikan sebuah pekerjaan.

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*). Klasifikasi dokumen akan mengelompokkan data yang dimilikinya sesuai dengan dokumen yang tergantung pada data tersebut (Even dan Zohar, 2002).

Pengklasifikasian berita berbahasa Indonesia sebelumnya sudah pernah dilakukan. Yaitu pengklasifikasian berita dengan algoritma *K-Means clustering*, dokumen berita dimasukkan kedalam *cluster* yang paling cocok berdasarkan ukuran kedekatan dengan *centroid*. *Centroid* adalah vektor term yang dianggap sebagai titik tengah *cluster* (Wibisono dan Khodra, 2005). Dan juga penelitian dengan algoritma *Single pass clustering*, yaitu dengan menggunakan penghitungan tingkat kemiripan (*Similarity*) dengan *standard cosine Similarity*. *Similarity* yang telah dihasilkan selanjutnya dievaluasi untuk menentukan pasangan-pasangan dokumen yang dinyatakan mirip berdasarkan nilai *treshold* tertentu (Arifin dan Setiono, 2002).

Pada penelitian ini, digunakan suatu metode lain dalam menyelesaikan permasalahan pengklasifikasian dokumen berita 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).

Pada metode *naïve bayes* kemunculan kata pada dokumen diperhitungkan sedangkan pada metode *MNB* jumlah frekuensi kemunculan pada dokumen juga diperhitungkan. Sehingga di lain pihak model *MNB* dianggap sebagai model klasifikasi yang lebih akurat untuk set data yang mempunyai variasi besar pada panjang dokumen (McCallum dan Nigam, 1998).

Berdasarkan pada latar belakang yang telah dipaparkan, maka judul yang diambil pada Skripsi ini adalah "**KLASIFIKASI BERITA BERBAHASA INDONESIA MENGGUNAKAN METODE MULTINOMIAL NAÏVE BAYES".**

1.2 Rumusan Masalah

Berdasarkan uraian pada latar belakang masalah, maka dalam Skripsi ini dapat dirumuskan permasalahan sebagai berikut:

1. Bagaimana membuat pengklasifikasian dokumen berita berbahasa Indonesia menggunakan metode *Multinomial Naïve Bayes*.
2. Bagaimana hasil evaluasi dari sistem pengklasifikasian dengan metode tersebut pada beberapa sampel yang diberikan.

1.3 Batasan Masalah

Dari permasalahan yang telah dipaparkan dalam subbab 1.2, batasan masalah untuk menghindari melebarnya masalah yang akan diselesaikan adalah :

1. Dokumen yang digunakan hanya dokumen berita berbahasa Indonesia dan file dokumen berupa file teks (*.txt).
2. Dokumen berita yang digunakan dalam Skripsi ini hanya bersumber dari www.detik.com dan www.kompas.com.

3. Pada proses *stemming* tidak memperhitungkan adanya infiks (sisipan). Proses *stemming* yang dibangun hanya melakukan penghilangan prefiks dan sufiks.

1.4 Tujuan

Tujuan yang ingin dicapai dalam pelaksanaan Skripsi ini adalah :

1. Membuat sebuah perangkat lunak yang mampu mengimplementasikan metode *MNB* untuk mengklasifikasikan dokumen berbahasa Indonesia.
2. Mengevaluasi tingkat akurasi dari sistem pengklasifikasian dengan metode MNB.

1.5 Manfaat

Manfaat yang dapat diambil dari penulisan Skripsi ini adalah menyediakan suatu sistem yang dapat membantu dalam melakukan pencarian kategori dokumen berita.

1.6 Metodologi Pemecahan Masalah

Untuk mencapai tujuan yang dirumuskan sebelumnya, maka metodologi yang digunakan dalam penulisan Skripsi ini adalah :

1. Studi Literatur
   Mempelajari teori-teori yang berhubungan dengan konsep pengklasifikasian suatu dokumen menggunakan metode *MNB* dari berbagai referensi.
2. Pendefinisian dan Analisis Masalah
   Mendefinisikan dan menganalisis masalah untuk mencari solusi yang tepat.
3. Perancangan dan Implementasi Sistem
   Membuat perancangan perangkat lunak dengan analisis terstruktur dan mengimplementasikan konsep pengklasifikasian suatu dokumen berbahasa Indonesia menggunakan metode *MNB*.

4. Uji Coba dan Analisa Hasil Implementasi
   Menguji perangkat lunak dan menganalisa hasil dari implementasi tersebut apakah sudah sesuai dengan tujuan yang dirumuskan sebelumnya, untuk kemudian dievaluasi dan disempurnakan.

1.7 Sistematika Penulisan

Skripsi ini disusun berdasarkan sistematika penulisan sebagai berikut:

BAB I PENDAHULUAN

Berisi latar belakang masalah, perumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi pemecahan masalah, dan sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Menguraikan teori-teori yang berhubungan dengan pengklasifikasian suatu dokumen dengan metode *Multinomial Naïve Bayes (MNB)* dan pemrosesan teks.

BAB III METODOLOGI DAN PERANCANGAN

Pada bab ini akan dijelaskan mengenai langkah-langkah yang digunakan dalam pemrosesan pengklasifikasian dokumen berbahasa Indonesia dengan menggunakan metode *Multinomial Naïve Bayes (MNB).*

BAB IV PEMBAHASAN

Pada bab ini akan dilakukan implementasi sistem, pengujian dan analisa sistem perangkat lunak yang dibangun.

BAB V PENUTUP

Berisi kesimpulan dari seluruh rangkaian penelitian serta saran kemungkinan pengembangannya.

BAB II
TINJAUAN PUSTAKA

2.1 Pengertian Berita

Sesungguhnya 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. Bagaimanapun, tidak ada seorang pun yang sanggup merekonstruksikan realitas sosial memiliki empat muka, maka yang sering diungkap para wartawan hanya dua muka. 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).

2.1.1 Unsur-Unsur Berita

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. Ada pula yang terdiri atas nama media massa, tempat kejadian dan tanggal kejadian. Tujuannya adalah untuk menunjukkan tempat kejadian dan inisial media.

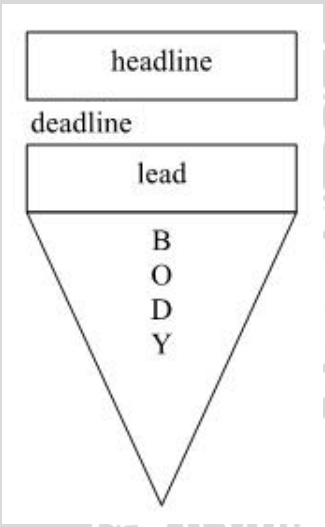
3. *Lead* (Teras Berita)
   Biasanya ditulis pada paragraph pertama sebuah berita. Ia merupakan unsur yang paling penting dari sebuah berita, yang menentukan apakah isi berita akan dibaca atau tidak. Ia merupakan sari pati sebuah berita, yang melukiskan seluruh berita secara singkat.

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

2.1.2 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.



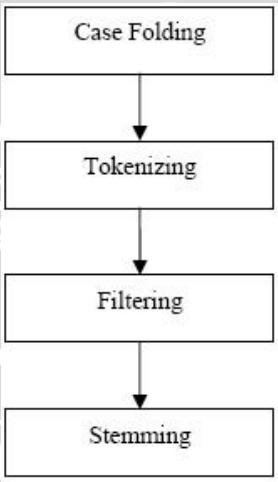
Gambar 2.1 Struktur Berita
(Sumber: Ditjen Pendidikan Tinggi Dep P dan K, 1978)

Struktur-struktur berita di atas 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 sembarang. 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 Ekstraksi Dokumen

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 *token*, 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* Pada 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. Untuk aturan-aturan selengkapnya dapat dilihat pada lampiran 2.

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 seperti pada tabel 2.1.

Tabel 2.1 Pasangan Konfiks yang tidak diperbolehkan
(Tala, 2003)

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 seperti pada tabel 2.2.

Tabel 2.2 Urutan prefiks ganda (Tala, 2003)

Prefiks 1	Prefiks 2
meng	per
di	ber
ter	
ke	

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 :
[...] menunjukkan opsi/pilihan.

2.3.2 Proses S*temming* Bahasa Indonesia

Berdasarkan analisa morfologi yang telah dibahas sebelumya, maka terdapat 5 aturan tahapan pada proses *stemming* dalam bahasa Indonesia (Tala, 2003). Aturan tersebut adalah :

- Aturan tahap pertama menangani partikel infleksional seperti pada tabel 2.3.

Tabel 2.3 Menangani partikel infleksional

Sufiks	Pengganti	Kondisi Ukuran	Kondisi Tambahan	Contoh
kah	NULL	2	NULL	diakah? dia
lah	NULL	2	NULL	adalah ? ada
tah*	NULL	2	NULL	apatah ? apa
pun**	NULL	2	NULL	Buku pun? buku

* tah ? improduktif

** pun ? menurut EYD terpisah dengan kata mengikutinya

- Aturan tahap kedua menangani kata ganti infleksional seperti pada tabel 2.4.

Tabel 2.4 Menangani kata ganti infleksional

Sufiks	Pengganti	Kondisi Ukuran	Kondisi Tambahan	Contoh
ku	NULL	2	NULL	bukuku? buku
mu	NULL	2	NULL	bukumu? buku
nya	NULL	2	NULL	bukunya? buku

- Aturan tahap ketiga menangani urutan prefiks derivasional pertama seperti pada tabel 2.5.

Tabel 2.5 Menangani urutan prefiks derivasional pertama

Prefiks	Pengganti	Kondisi Ukuran	Kondisi Tambahan	Contoh
meng	NULL	2	NULL	mengukur? ukur
meny	s	2	V...*	menyapu? sapu
men	t	2	V...	menuduh? tuduh
men	NULL	2	NULL	menduga? duga
mem	p	2	V...	memukul? pukul
mem	NULL	2	NULL	membakar? bakar
me	NULL	2	NULL	merusak? rusak

peng	NULL	2	NULL	pengukur? ukur
peny	s	2	V...	penyelam? selam
pen	t	2	V...	penari? tari
pen	NULL	2	NULL	Penduga? duga
pem	P	2	V...	Pemandu? pandu
pem	NULL	2	NULL	Pembaca? baca
di	NULL	2	NULL	diukur? ukur
ter	NULL	2	NULL	tersipu? sipu
ke	NULL	2	NULL	kekasih? kasih

* kata dasar dimulai huruf vokal

Ø Aturan tahap keempat menangani urutan prefiks derivasional kedua seperti pada tabel 2.6.

Tabel 2.6 Menangani urutan prefiks derivasional kedua

Prefiks	Pengganti	Kondisi Ukuran	Kondisi Tambahan	Contoh
ber	NULL	2	NULL	berlari? lari
bel	NULL	2	ajar	belajar? ajar
be	NULL	2	kerja	bekerja? kerja
per	NULL	2	NULL	perjelas? jelas
pel	NULL	2	ajar	pelajar? ajar
pe	NULL	2	NULL	pekerja? kerja

Ø Aturan tahap kelima menangani sufiks derivasional seperti pada tabel 2.7.

Tabel 2.7 Menangani urutan sufiks derivasional

Sufiks	Pengganti	Kondisi Ukuran	Kondisi Tambahan	Contoh
kan	NULL	2	prefiks¢ {ke,peng}	tarikkan? tarik (meng)ambilkan? ambil
an	NULL	2	prefiks¢ {di,meng,ter}	makanan? makan (per)janjian? janji
i	NULL	2	V|K...c1c2,c1?s, c2?i dan prefiks¢{ber,ke,peng}	tandai? tanda (men)dapati? dapat

Kondisi ukuran adalah jumlah mininum suku kata dalam sebuah kata. Karena dalam bahasa Indonesia, kata dasar setidaknya

mempunyai 2 suku kata. Maka kondisi ukuran dalam proses *stemming* bahasa Indonesia adalah dua. Adapun suku kata didefinisikan memiliki satu vokal.

2.4 Klasifikasi

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. Sehingga 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*. 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 *Naïve Bayes Classifier*

Naïve bayes classifier merupakan salah satu metode *machine learning* yang dapat digunakan untuk klasifikasi suatu dokumen.

Teorema *Bayes* sendiri berawal dari persamaan 2.1.

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

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

$$P(B \cap A) = P(B \mid A).P(A) \quad (2.2)$$

Sehingga didapatkan teorema *Bayes* pada persamaan 2.3.

$$P(A \mid B) = \frac{P(B \mid A).P(A)}{P(B)} \quad (2.3)$$

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.

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

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

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

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

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

Tingkat kesulitan menghitung $P(a_1, a_2, a_3, ..., a_n \mid v_j) P(v_j)$ menjadi tinggi karena jumlah *term* $P(a_1, a_2, a_3, ..., a_n \mid v_j) P(v_j)$ 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 kata *independent* satu sama lain.

Sehingga menjadi persamaan 2.7.

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

Subsitusi persamaan 2.7 dengan persamaan 2.6 menjadi persamaan 2.8.

$$V_{NB} = \arg\max_{vj \in V} P(v_j) \prod_i P(a_1 \mid v_j) \quad (2.8)$$

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 $P(a_1 \mid v_j)$. 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:

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

$$P(w_k \mid v_j) = \frac{n_k + 1}{n + |kata|} \quad (2.10)$$

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 W_k pada 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  v_j  lakukan:
      a. docs_j
      b. Hitung P(v_j) dengan persamaan 2.9
      c. Untuk setiap kata w_k pada |kata| lakukan:
         i. Hitung P(w_k|v_j ) dengan persamaan 2.10

B. Proses klasifikasi.

   Input adalah dokumen yang belum diketahui
   kategorinya:

   1. Hasilkan V_NB sesuai dengan persamaan 2.8
      dengan menggunakan P(v_j) dan P(w_k|v_j ) 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.

$$\Pr(c \mid t_i) = \frac{\Pr(c)\Pr(t_i \mid c)}{\Pr(t_i)}, c \in C \qquad (2.11)$$

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

$$\Pr(t_i \mid c) = (\sum_n f_{ni})! \prod_n \frac{\Pr(w_n \mid c)^{f_{ni}}}{f_{ni}!} \qquad (2.12)$$

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.

$$\Pr(w_n \mid c) = \frac{1 + F_{nc}}{N + \sum_{x=1}^{N} F_{xc}} \quad (2.13)$$

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

$$\Pr(t_i) = \sum_{k=1}^{|C|} \Pr(k) \Pr(t_i \mid k) \quad (2.14)$$

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.

$$\Pr(t_i \mid c) = \alpha \prod_n \Pr(w_n \mid c)^{f_{ni}} \quad (2.15)$$

Perbedaan metode *Multinomial Naïve Bayes* dengan *Naïve Bayes* polos yaitu pada proses pembuatan model atau *training*, 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 *term* yang berkelas tertentu terhadap semua dokumen. 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. $docs_j$
   b. Hitung Pr(c) dengan persamaan 2.9
   c. Untuk setiap kata w_k pada |kata| lakukan:
      i. Hitung $P(w_n|c)$ dengan persamaan 2.13

B. Proses klasifikasi.

Input adalah dokumen yang belum diketahui kategorinya:

1. Untuk setiap kategori c hitung $Pr(t_i|c)$ dengan persamaan 2.15
2. Hitung total $Pr(t_i)$ dengan persamaan 2.14 dengan menggunakan $Pr(t_i|c)$ yang telah diperoleh dari semua kategori.
3. Hasilkan $Pr(c|t_i)$ dengan persamaan 2.11 untuk dicari nilai yang paling besar.

Algoritma *Multinomial Naïve Bayes*

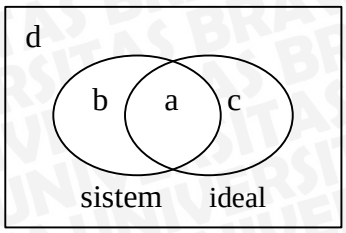
2.6 Evaluasi

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 2.8 menunjukan *Matrix Confusion* (Lewis, 1995).

Tabel 2.8 *Matrix Confusion*

Pengklasifikasian oleh sistem	Pengklasifikasian sebenarnya (ideal)	
	Ya	Bukan
Ya	a	b
Bukan	c	d

Jika *matrix confusion* tersebut digambarkan dalam diagram himpunan bagian dapat dilihat pada Gambar 2.3.



Gambar 2.3 Diagram himpunan *matrix confusion*

- a menunjukan bahwa dokumen yang termasuk dalam hasil klasifikasi oleh sistem memang merupakan anggota klasifikasi. Pengklasifikasi sebenarnya ya dan pengklasifikasi oleh sistem ya.
- b menunjukkan bahwa dokumen yang termasuk dalam hasil klasifikasi oleh sistem ternyata seharusnya bukan merupakan anggota klasifikasi. Pengklasifikasi sebenarnya bukan dan pengklasifikasi oleh sistem ya.
- c menunjukkan bahwa dokumen yang tidak termasuk dalam hasil klasifikasi oleh sistem ternyata seharusnya merupakan anggota klasifikasi. Pengklasifikasi sebenarnya ya dan pengklasifikasi oleh sistem bukan.
- d menunjukkan bahwa dokumen yang tidak termasuk dalam hasil klasifikasi oleh sistem ternyata seharusnya bukan merupakan anggota klasifikasi. Pengklasifikasi sebenarnya bukan dan pengklasifikasi oleh sistem bukan.

Ada beberapa standar pengukuran yang digunakan dalam pengklasifikasian dokumen, diantaranya adalah *recall, precision* dan *F1 Measure*.

Recall adalah ukuran dari jumlah dokumen benar yang berhasil diklasifikasikan oleh sistem (Tala, 2003). Dirumuskan pada persamaan 2.16.

$$recall = \frac{a}{a+c} \tag{2.16}$$

Sedangkan *precision* adalah ukuran dari jumlah dokumen yang diklasifikasikan oleh sistem dan dokumen tersebut benar (Tala, 2003). Dirumuskan pada persamaan 2.17.

$$precision = \frac{a}{a+b} \quad (2.17)$$

F1 measure adalah tingkat ketepatan dan ketelitian sistem yang merupakan gabungan antara *recall* dan *precision* yang didefinisikan dengan persamaan berikut ini (Yang dan Liu, 1999). Dirumuskan pada persamaan 2.18.

$$F_1 = \frac{2 \times recall \times precision}{recall + precision} \quad (2.18)$$

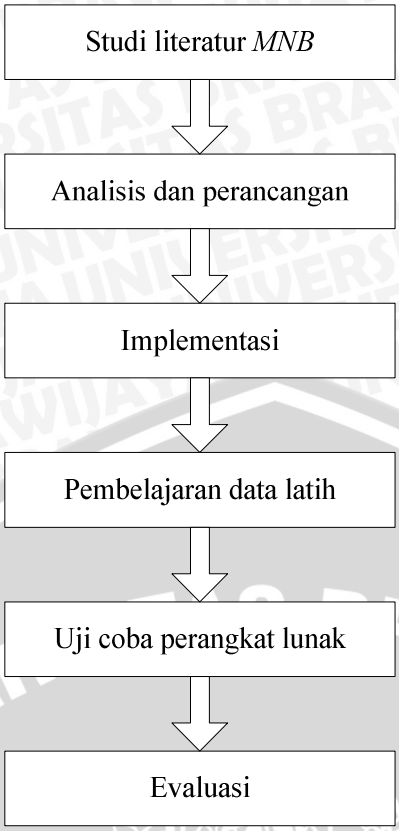
BAB III
METODOLOGI DAN PERANCANGAN

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 dengan menggunakan metode *Multinomial Naïve Bayes* (*MNB*).

Penelitian dilakukan dengan tahapan-tahapan berikut ini :

1. Mempelajari metode yang akan digunakan pada sistem pengklasifikasian dokumen ini, yaitu metode *Multinomial Naïve Bayes* (*MNB*).
2. Menganalisa dan merancang perangkat lunak pengklasifikasian dokumen dengan menggunakan metode *MNB*.
3. Membuat perangkat lunak berdasarkan analisis dan perancangan yang telah dilakukan.
4. Melakukan proses pelatihan (pembelajaran) terhadap perangkat lunak dengan memasukkan sejumlah dokumen berita yang diperoleh dari sumber data tertentu sebagai data latih.
5. Melakukan ujicoba perangkat lunak yang telah dibuat menggunakan dokumen tes berupa data teks berita.
6. Melakukan evaluasi terhadap perangkat lunak yang telah dibuat, berdasarkan besar F_1 *measure*, yang merupakan kombinasi antara *recall* dan *precision*, dari kategori yang dihasilkan oleh sistem dengan kategori asal.

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



Gambar 3.1 Langkah-langkah Pembuatan Perangkat Lunak

3.1 Analisis Data

Pada penelitian ini data yang digunakan berupa dokumen berita yang diambil dari situs berita berbahasa Indonesia yaitu www.detik.com dan www.kompas.com. Situs ini adalah bentuk media *online* yang sering dikunjungi di Indonesia. Koleksi berita yang dikumpulkan diambil dari berita yang diterbitkan dari bulan Mei 2010 sampai dengan November 2010.

Situs ini dipilihnya sebagai sumber berita juga karena situs berita ini telah menjadi situs berbahasa Indonesia yang terpercaya selama bertahun-tahun. Informasi yang disampaikan di-*update* dengan cepat sehingga mempermudah proses pengumpulan data.

Untuk memperoleh data latih (*training set*) yang tepat dan untuk mempermudah pengujian kebenaran dan keakuratan pada data uji (*testing set*) maka dokumen berita yang digunakan dalam penelitian ini adalah berita-berita yang telah dikelompokkan atau dikategorikan oleh situs ini.

3.2 Analisis Sistem

Pada sub-bab analisis sistem ini akan dibahas mengenai semua hal yang diperlukan dalam proses pembuatan perangkat lunak pengklasifikasian dokumen menggunakan metode *MNB*.

3.2.1 Deskripsi Sistem

Dalam penelitian ini, akan dibangun suatu perangkat lunak yang dapat digunakan untuk pengklasifikasian dokumen secara otomatis pada teks dokumen yang berupa file berformat teks (**.txt*) sebagai masukan. Metode yang digunakan dalam sistem ini adalah *Multinomial Naïve Bayes* (*MNB*).

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.

Pada tahap pembelajaran, proses-proses yang dilakukan adalah :

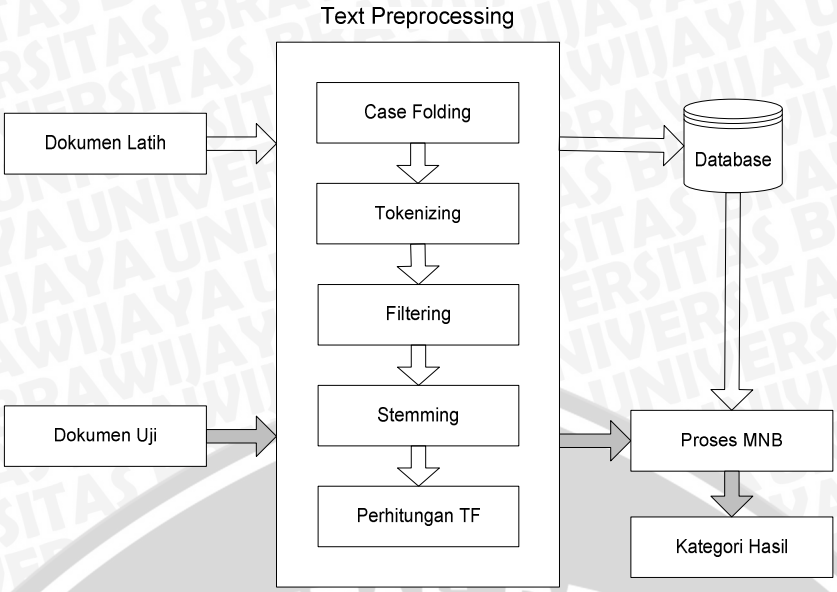
1. *User* memasukkan teks berita yang akan dijadikan data latih.
2. *User* menentukan kategori berita yang telah dimasukkan.
3. Sistem melakukan tahap *preprocessing* terhadap data latih, yang diawali dengan proses *case folding* yaitu mengubah semua huruf di dalam data latih menjadi huruf kecil dan menghilangkan semua karakter angka dan tanda baca.
4. Kemudian proses *tokenizing* yaitu pemotongan (*parsing*) teks dalam data latih menjadi kata tunggal.
5. Proses selanjutnya adalah *filtering*, yaitu menghilangkan *stopword* yang ada dalam data latih.

6. Hasil dari *filtering* kemudian dilakukan *stemming* untuk menghilangkan kata berimbuhan.
7. Dari proses *preprocessing* dihasilkan *term* dan frekuensi dari masing-masing dokumen latih.
8. 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. Sistem melakukan tahap *preprocessing*, yang diawali dengan proses *case folding*, mengubah semua huruf yang ada di dalam dokumen menjadi huruf kecil dan menghilangkan semua karakter angka dan tanda baca.
3. *Tokenizing*, pemotongan (*parsing*) tiap-tiap kata yang menyusun dokumen menjadi kata tunggal.
4. *Filtering*, penghilangan *stopword* yang terdapat pada dokumen.
5. *Stemming* untuk menghilangkan kata berimbuhan.
6. Penghitungan frekuensi tiap kata yang ada pada masing-masing dokumen. Hasil perhitungan dimasukkan ke dalam basis data.
7. Dilakukan perhitungan nilai probabilitas dokumen uji berdasarkan probabilitas masing-masing kategori.
8. Hasil probabilitas yang paling tinggi akan menjadi kategori dari dokumen uji.

Rancangan arsitektur sistem pengklasifikasian dokumen berita ditunjukkan pada Gambar 3.2.



Gambar 3.2 Rancangan Arsitektur Sistem

3.2.2 Batasan Sistem

Sistem yang akan dibuat memiliki batasan-batasan sebagai berikut :

1. Sistem hanya menangani teks dokumen berita berbahasa Indonesia saja.
2. Sumber berita diperoleh dari dua situs berita berbahasa Indonesia, yaitu www.detik.com dan www.kompas.com.
3. Sistem hanya dapat menangani dokumen berupa file berformat teks (**.txt*).
4. Penghilangan *stopword* yang digunakan terkait dengan bahasa, karena dalam penelitian ini yang dipilih hanya dokumen yang berbahasa Indonesia, maka *stopword*-nya juga dalam bahasa Indonesia.
5. Faktor *heading* (bab dan sub-bab) dalam sistem ini tidak dipertimbangkan).

3.3 Perancangan Sistem

Setelah melakukan analisis sistem, deskripsi dan batasan-batasan sistem, yang dilakukan selanjutnya adalah melakukan perancangan terhadap sistem berdasarkan analisis yang telah dilakukan sebelumnya. Perancangan sistem mencakup perancangan proses dan perancangan basis data.

3.3.1 Perancangan Proses

Pada bagian ini menjelaskan secara rinci mengenai proses yang dilakukan sistem pengklasifikasian dokumen otomatis secara beruntun dan sistematis. Pertama-tama sistem melakukan tahap *preprocessing* pada tiap-tiap dokumen. Selanjutnya, sistem melakukan tahap klasifikasi untuk menentukan kategori dokumen dengan menggunakan metode *Multinomial Naïve Bayes* (*MNB*).

3.3.1.1 Text Preprocessing

Tahap *text preprocessing* ini meliputi *case folding*, *tokenizing*, *filtering*, *stemming* dan *term weighting*.

1. *Case Folding*

Pada tahap ini dilakukan proses mengubah semua huruf yang terdapat dalam dokumen menjadi huruf kecil. Tahap selanjutnya adalah menghilangkan semua karakter angka dan tanda baca yang terdapat dalam dokumen tersebut dan menggantinya dengan karakter spasi. Semua karakter selain huruf, termasuk karakter spasi, dianggap sebagai pemisah atau *delimiter*.

Contoh :

Facebook semakin digandrungi para penggunanya. Tua muda, pria wanita, kini seolah tak bisa lepas seharipun tanpa mengakses Facebook. Dari semua kalangan pengakses Facebook, konon wanita yang paling 'kecanduan' dengan situs jejaring sosial ini. Setidaknya begitulah kesimpulan sebuah studi yang dilakukan terhadap 1.605 pengguna Facebook di Amerika Serikat (AS).

Setelah mengalami proses *case folding*, menjadi :

facebook semakin digandrungi para penggunanya tua muda pria wanita kini seolah tak bisa lepas seharipun tanpa mengakses facebook dari semua kalangan pengakses facebook konon wanita yang paling kecanduan dengan situs jejaring sosial ini setidaknya begitulah kesimpulan sebuah studi yang dilakukan terhadap pengguna facebook di amerika serikat as

2. *Tokenizing*

Dalam tahap ini dilakukan pemecahan/pemotongan dokumen menurut tiap-tiap kata yang menyusun dokumen tersebut setelah mengalami proses *case folding*. Hasil pemotongan (*parsing*) terhadap kata-kata tunggal tersebut dijadikan kumpulan *token* dan membentuknya menjadi sebuah daftar atau *list*.

Contoh hasil *parsing* setelah mengalami proses *case folding* :

1	:	facebook	25	:	wanita
2	:	semakin	26	:	yang
3	:	digandrungi	27	:	paling
4	:	para	28	:	kecanduan
5	:	penggunanya	29	:	dengan
6	:	tua	30	:	situs
7	:	muda	31	:	jejaring
8	:	pria	32	:	sosial
9	:	wanita	33	:	ini
10	:	kini	34	:	setidaknya
11	:	seolah	35	:	begitulah
12	:	tak	36	:	kesimpulan
13	:	bisa	37	:	sebuah
14	:	lepas	38	:	studi
15	:	seharipun	39	:	yang
16	:	tanpa	40	:	dilakukan
17	:	mengakses	41	:	terhadap
18	:	facebook	42	:	pengguna
19	:	dari	43	:	facebook
20	:	semua	44	:	di
21	:	kalangan	45	:	amerika
22	:	pengakses	46	:	serikat

23 : facebook
24 : konon
47 : as

3. *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*. Pertama-tama dibentuk sebuah *stoplist*, yaitu sebuah daftar kata yang berisi sekumpulan *stopword*. Selanjutnya proses *stopword removal* dilakukan dengan menghapus kata-kata yang tidak relevan pada hasil *parsing* sebuah dokumen dengan cara membandingkannya dengan *stoplist* yang ada.

Contoh hasil *filtering* dari hasil *parsing* :

1 : amerika
2 : as
3 : digandrungi
4 : facebook
5 : jejaring
6 : kalangan
7 : kecanduan
8 : kesimpulan
9 : konon
10 : lepas
11 : mengakses
12 : muda
13 : pengakses
14 : pengguna
15 : penggunanya
16 : pria
17 : seharipun
18 : serikat
19 : situs
20 : sosial
21 : studi
22 : tua
23 : wanita

4. Stemming

Pada tahap ini akan dicari *root* kata dari tiap kata hasil *filtering*. Dalam bahasa Indonesia, afiks/imbuhan terdiri dari sufiks (akhiran), infiks (sisipan) dan prefiks (awalan). Karena proses penambahan infiks dalam bahasa Indonesia jarang terjadi dan tingkat kesulitan dalam menangani kata yang mengandung infiks maka proses *stemming* yang dibangun hanya menangani kata yang mengalami penambahan prefiks dan sufiks. Setelah melalui proses *stemming*, semua kata-kata yang berimbuhan diubah menjadi kata dasarnya.

Contoh hasil *stemming* dari hasil *filtering* :

1	:	akses	12	:	lepas
2	:	amerika	13	:	muda
3	:	as	14	:	pria
4	:	candu	15	:	serikat
5	:	facebook	16	:	simpul
6	:	gandrung	17	:	situs
7	:	guna	18	:	sosial
8	:	hari	19	:	studi
9	:	jejaring	20	:	tua
10	:	kalang	21	:	wanita
11	:	konon			

5. Perhitungan *TF*

TF adalah *term frequency* atau jumlah kemunculan kata dalam isi dokumen. Dari kata hasil *filtering*, dihitung kemunculan data dalam dokumen tersebut. Nilai kemunculan tersebut menjadi nilai *TF*.

Contoh perhitungan *TF* dari hasil *filtering* :

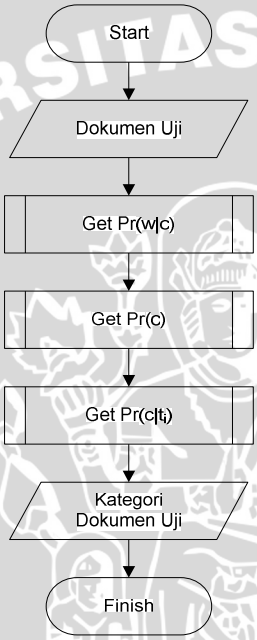
1	:	akses	2	12	:	lepas	1
2	:	amerika	1	13	:	muda	1
3	:	as	1	14	:	pria	1
4	:	candu	1	15	:	serikat	1
5	:	facebook	4	16	:	simpul	1
6	:	gandrung	1	17	:	situs	1
7	:	guna	2	18	:	sosial	1
8	:	hari	1	19	:	studi	1
9	:	jejaring	1	20	:	tua	1
10	:	kalang	1	21	:	wanita	1
11	:	konon	1				

3.3.1.2 Pengklasifikasian Dokumen

Proses pengklasifikasian untuk dokumen tes (dokumen *X*) menggunakan algoritma *MNB* dapat dilakukan dengan langkah-langkah berikut :

1. Menghitung peluang $Pr(w_n|c)$ yaitu peluang setiap kata w_n dalam setiap kategori *c* pada data *corpus* 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.

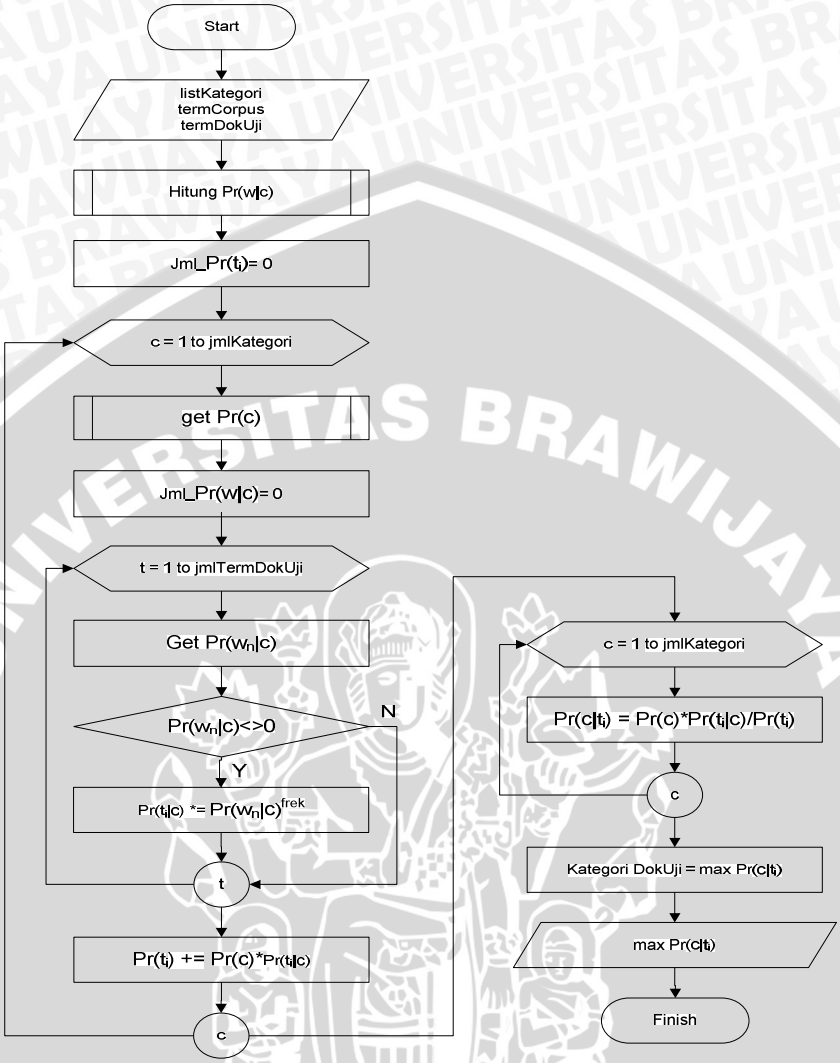
Flowchart proses klasifikasi dengan algoritma *MNB* ditunjukkan pada Gambar 3.3.



Gambar 3.3 Algoritma *MNB*

1. Proses Get Pr(c|t_i)

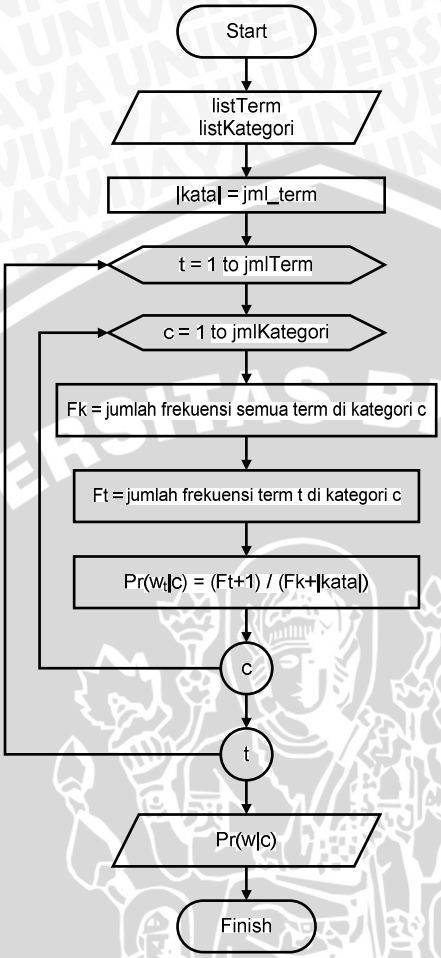
Pada proses get Pr(c|t_i) dilakukan perhitungan peluang setiap kategori terhadap dokumen *corpus*, dengan persaman 2.11. Dari *term* yang ada di setiap dokumen *corpus* dicari peluangnya di setiap kategori. Flowchart proses get p(w|c) ditunjukkan pada Gambar 3.4.



Gambar 3.4 Flowchart Proses Get Pr(c|t_i)

2. Proses Get Pr(w|c)

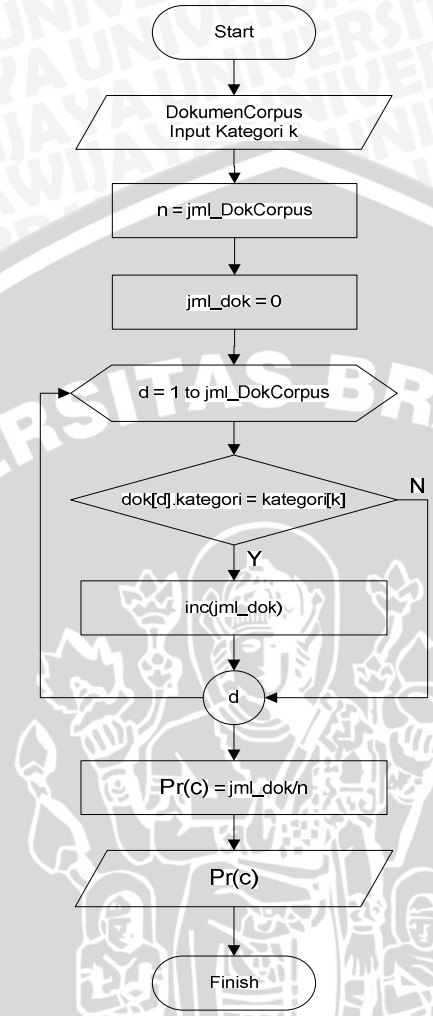
Pada proses get Pr(w|c) dilakukan perhitungan peluang setiap *term* dalam setiap kategori, dengan persaman 2.13. Dari *term* yang ada di setiap dokumen *corpus* dicari peluangnya di setiap kategori. Flowchart proses get p(w|c) ditunjukkan pada Gambar 3.5.



Gambar 3.5 Flowchart Proses Get Pr(w|c)

3. Proses Get Pr(c)

Pada proses get Pr(c) dilakukan perhitungan peluang setiap kategori dalam dokumen *corpus*. 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 3.6.



Gambar 3.6 Flowchart Proses Get Pr(c)

3.3.2 Perancangan Basis Data

Pada bagian ini menjelaskan tentang perancangan basis data yang akan digunakan dalam penelitian ini yang meliputi perancangan tabel dan atributnya, dan perancangan relasi antar tabel yang ditunjukkan pada Gambar 3.7. Adapun tabel-tabel yang akan digunakan ditunjukkan pada tabel 3.1 sampai dengan 3.10.

1. Tabel *Kategori*
   Tabel *Kategori* digunakan untuk menyimpan ID kategori dan nama kategori.

Tabel 3.1 Tabel *Kategori*

Field	**Tipe**	**Keterangan**
ID_Kategori	Varchar	Primary Key
Nama_Kategori	Varchar	

2. Tabel *TermLatih*
   Tabel *TermLatih* digunakan untuk menyimpan kata unik dan frekuensi dalam setiap dokumen latih.

Tabel 3.2 Tabel *TermLatih*

Field	**Tipe**	**Keterangan**
Term	Varchar	Primary Key
Dokumen	Varchar	Foreign Key
Frekuensi	Integer	

3. Tabel *DokLatih*
   Tabel *Latih* digunakan untuk menyimpan data dokumen latih yang dimasukkan oleh *user*.

Tabel 3.3 Tabel *DokLatih*

Field	**Tipe**	**Keterangan**
ID_Corpus	Varchar	Primary Key
Filename	Varchar	
Content	ntext	
Kategori	Varchar	Foreign Key

4. Tabel *PeluangTerm*
   Tabel *PeluangTerm* digunakan untuk menyimpan nilai peluang setiap term di setiap kategori.

Tabel 3.4 Tabel *PeluangTerm*

Field	Tipe	Keterangan
Term	Varchar	Foreign Key
Kategori	Varchar	Foreign Key
Frekuensi	Integer	
Peluang	Double	

5. Tabel *TermUji*
   Tabel *TermUji* digunakan untuk menyimpan kata unik dan frekuensi dalam setiap dokumen uji.

Tabel 3.5 Tabel *TermUji*

Field	Tipe	Keterangan
Term	Varchar	Primary Key
Dokumen	Varchar	Foreign Key
Frekuensi	Integer	

6. Tabel *DokUji*
   Tabel *DokUji* digunakan untuk menyimpan data dokumen uji yang dimasukkan oleh *user*.

Tabel 3.6 Tabel *DokUji*

Field	Tipe	Keterangan
ID_Uji	Varchar	Primary Key
Filename	Varchar	
Content	ntext	

7. Tabel *KategoriUji*
   Tabel *KategoriUji* digunakan untuk menyimpan peluang setiap dokumen dalam setiap kategori.

Tabel 3.7 Tabel *KategoriUji*

Field	Tipe	Keterangan
DokUji	Varchar	Foreign Key
Kategori	Varchar	Foreign Key
Peluang	Double	

8. Tabel *Stopword*
   Tabel *Stopword* digunakan untuk menyimpan semua daftar *stopword*.

Tabel 3.8 Tabel *Stopword*

Field	Tipe	Keterangan
Kata	Varchar	Primary Key

9. Tabel *KataAll*
   Tabel *KataAll* digunakan untuk menyimpan semua kata baik di dokumen latih maupun dokumen uji.

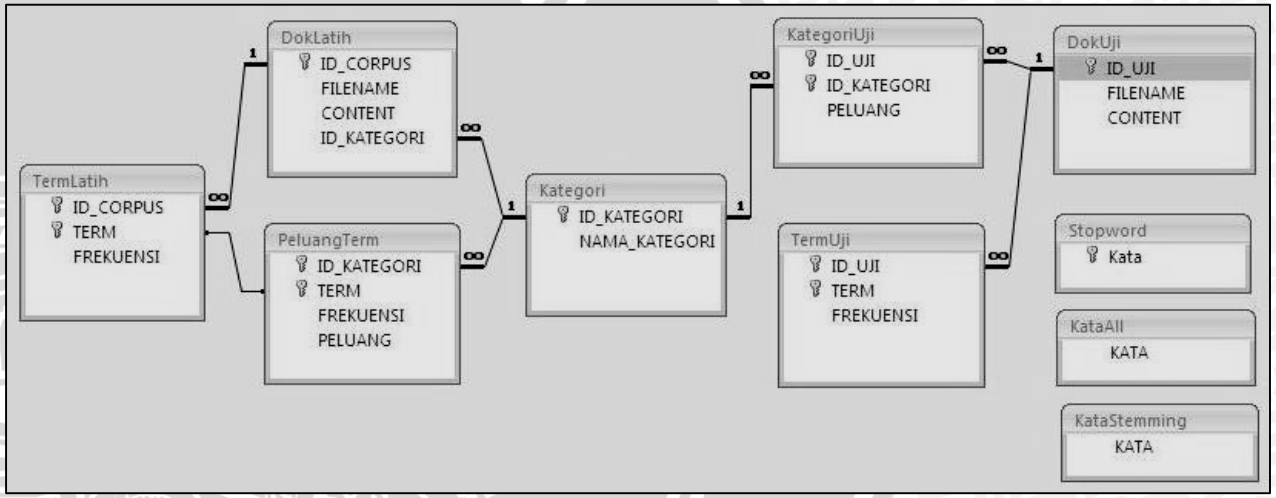
Tabel 3.9 Tabel *KataAll*

Field	Tipe	Keterangan
Kata	Varchar	

10. Tabel *Stemming*
    Tabel *Stemming* digunakan untuk menyimpan semua kata hasil proses *stemming*.

Tabel 3.10 Tabel *Stemming*

Field	Tipe	Keterangan
Kata	Varchar	



Gambar 3.7 Relasi Antar Tabel

3.4 Perancangan Antarmuka

Pada bagian ini akan menjelaskan tentang perancangan antarmuka dari sistem ini. Rancangan antarmuka tersebut dapat dilihat pada Gambar 3.8.

Gambar 3.8 Rancangan Antar Muka

Keterangan :

1. Data Master : berisi daftar stopword dan daftar kategori
2. Data Latih : berisi data latih, untuk disimpan data pembelajaran nya.
3. Data Uji : berisi semua daftar dokumen uji dan hasil uji.
4. Field-field yang akan berisi data dokumen uji.
5. Hasil rekomendasi kategori yang terbentuk.

3.5 Contoh Perhitungan Manual

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

Dokumen Latih 1
Vitamin C punya banyak manfaat kesehatan seperti meningkatkan kekebalan tubuh dan mencegah flu. Tapi terlalu banyak konsumsi vitamin C bukan manfaat yang didapat melainkan efek sampingnya. Oleh karena itu konsumsi vitamin C harus pas, jangan berlebihan untuk menghindari masalah overdosis.Dosis yang berlebihan hanya akan sia-sia masuk ke tubuh dan akan dibuang melalui urine bahkan bisa mengganggu fungsi tubuh.
Kategori : Kesehatan

Dokumen Latih 2
Terapi warna merupakan daerah penyembuhan holistik yang menggunakan warna dalam upaya untuk mempengaruhi suasana hati, emosi bahkan kesehatan. Terapi ini telah ada selama ribuan tahun. Bangsa Mesir kuno, Yunani dan China telah mempelajari dan menggunakannya untuk mengobati penyakit. Setiap warna memiliki frekuensi dan getaran sendiri, warna dipercaya berhubungan dengan bagian-bagian yang berbeda dari tubuh.
Kategori : Kesehatan

Dokumen Latih 3
Peminat iPad di Indonesia tak perlu jauh-jauh lagi mencari iPad ke mancanegara. Sebab tak lama lagi, Apple juga akan membuka titik pemasaran komputer tabletnya di negara tetangga Singapura, bersamaan dengan delapan negara lainnya. Sejak Apple merilis iPad WiFi akhir April 2010 lalu dan kemudian iPad WiFi+3G sebulan setelahnya, animo peminat produk Apple dari seluruh dunia tertuju ke Amerika Serikat. Wajar saja, hanya di negeri Paman Sam itu iPad bisa didapatkan
Kategori : Teknologi

Dokumen Latih 4
Jika Anda ingin tahu aplikasi smartphone apa yang paling digandrungi saat ini, jawabannya adalah Facebook. Sebuah survei di Amerika Serikat mengungkapkan, hampir semua pengguna ponsel pintar memiliki aplikasi Facebook di dalamnya. Maka tak heran jika Facebook menjadi situs yang paling sering dikunjungi pengguna internet, baik melalui komputer maupun piranti mobile. Survei yang dilakukan Nielsen mengenai aplikasi mobile menyebutkan, Facebook menjadi aplikasi yang paling banyak digunakan pada iPhone, iPod Touch, Blackberry dan smartphone lainnya, kecuali yang menggunakan platform Android.
Kategori : Teknologi

Dokumen Latih 5
Setelah sekian lama tidak terdengar gaungnya, serangan virus mancanegara kembali datang mengancam pengguna komputer Indonesia. Kali ini berasal dari varian keluarga W32/Xorer. Dijelaskan analis virus dari Vaksincom, Adi Saputera, virus ini memiliki kemampuan seperti halnya seorang penyusup yang masuk ke dalam komputer, kemudian beraksi dan mengacaukan sistem komputer. Untuk dapat melakukan tersebut, pembuat virus menggunakan teknik social engineering untuk mengelabui korban dan menyebarkan dirinya.
Kategori : Teknologi

Dokumen Uji
Warna kulit ternyata mempengaruhi penyerapan vitamin D alami dari sinar matahari. Orang dengan kulit gelap membutuhkan lebih banyak suplemen vitamin D untuk mencukupi asupan tubuh. Kulit gelap membuat penyerapan sinar matahari ke kulit tidak maksimal. Terlebih jika aktivitasnya orang tersebut lebih banyak di dalam ruangan.Sinar matahari memicu tubuh untuk memproduksi vitamin D. Tapi berdasarkan penelitian, remaja pria dan wanita yang berkulit hitam sebagian besar mengalami kekurangan atau defisiensi vitamin D, meski mereka tinggal di daerah mendapat banyak matahari.
Kategori : ?

Dari dokumen-dokumen tersebut, langkah pertama adalah *preprosesing* yaitu proses *case folding, tokenizing, filtering, stemming dan perhitungan TF (term frequency)*. Hasil dari *preprocessing* dari dokumen-dokumen tersebut ditunjukkan oleh tabel 3.11.

Tabel 3.11 Daftar Token dan Frekuensinya

No	Term	Frekuensi					
		D1	D2	D3	D4	D5	Uji
1	adi	0	0	0	0	1	0
2	ajar	0	1	0	0	0	0
3	aksi	0	0	0	0	1	0
4	amerika	0	0	1	1	0	0
5	analis	0	0	0	0	1	0
6	ancam	0	0	0	0	1	0
7	android	0	0	0	1	0	0
8	animo	0	0	1	0	0	0
9	aplikasi	0	0	0	4	0	0
10	apple	0	0	3	0	0	0
11	april	0	0	1	0	0	0
12	asal	0	0	0	0	1	0
13	bangsa	0	1	0	0	0	0
14	beda	0	1	0	0	0	0
15	blackberry	0	0	0	1	0	0
16	buang	1	0	0	0	0	0
17	buat	0	0	0	0	1	0
18	buka	0	0	1	0	0	0
19	bulan	0	0	1	0	0	0
20	c	3	0	0	0	0	0
21	cari	0	0	1	0	0	0
22	cegah	1	0	0	0	0	0
23	china	0	1	0	0	0	0
24	daerah	0	1	0	0	0	**1**
25	dalam	0	0	0	1	0	0
26	dapat	0	0	1	0	0	0
27	dengar	0	0	0	0	1	0
28	dosis	1	0	0	0	0	0
29	dunia	0	0	1	0	0	0
30	efek	1	0	0	0	0	0

31	emosi	0	1	0	0	0	0
32	engineering	0	0	0	0	1	0
33	facebook	0	0	0	4	0	0
34	flu	1	0	0	0	0	0
35	frekuensi	0	1	0	0	0	0
36	fungsi	1	0	0	0	0	0
37	g	0	0	1	0	0	0
38	gandrung	0	0	0	1	0	0
39	ganggu	1	0	0	0	0	0
40	gaung	0	0	0	0	1	0
41	getar	0	1	0	0	0	0
42	guna	0	1	0	2	1	0
43	hal	0	0	0	0	1	0
44	hati	0	1	0	0	0	0
45	heran	0	0	0	1	0	0
46	hindar	1	0	0	0	0	0
47	holistik	0	1	0	0	0	0
48	hubung	0	1	0	0	0	0
49	indonesia	0	0	1	0	1	0
50	internet	0	0	0	1	0	0
51	ipad	0	0	5	0	0	0
52	iphone	0	0	0	1	0	0
53	ipod	0	0	0	1	0	0
54	jawab	0	0	0	1	0	0
55	kacau	0	0	0	0	1	0
56	kebal	1	0	0	0	0	0
57	kecuali	0	0	0	1	0	0
58	kelabui	0	0	0	0	1	0
59	keluarga	0	0	0	0	1	0
60	komputer	0	0	1	1	3	0
61	konsumsi	2	0	0	0	0	0
62	korban	0	0	0	0	1	0
63	kunjung	0	0	0	1	0	0
64	kuno	0	1	0	0	0	0
65	mampu	0	0	0	0	1	0
66	mancanegara	0	0	1	0	1	0
67	manfaat	2	0	0	0	0	0
68	mesir	0	1	0	0	0	0

69	milik	0	1	0	1	1	**1**
70	minat	0	0	2	0	0	0
71	mobile	0	0	0	2	0	0
72	negara	0	0	2	0	0	0
73	negeri	0	0	1	0	0	0
74	nielsen	0	0	0	1	0	0
75	obat	0	1	0	0	0	0
76	overdosis	1	0	0	0	0	0
77	paman	0	0	1	0	0	0
78	pas	1	0	0	0	0	0
79	pasar	0	0	1	0	0	0
80	pengaruh	0	1	0	0	0	**1**
81	percaya	0	1	0	0	0	0
82	pintar	0	0	0	1	0	0
83	piranti	0	0	0	1	0	0
84	platform	0	0	0	1	0	0
85	ponsel	0	0	0	1	0	0
86	produk	0	0	1	0	0	0
87	ribu	0	1	0	0	0	0
88	rilis	0	0	1	0	0	0
89	sakit	0	1	0	0	0	0
90	sam	0	0	1	0	0	0
91	sama	0	0	1	0	0	0
92	samping	1	0	0	0	0	0
93	saputera	0	0	0	0	1	0
94	sebar	0	0	0	0	1	0
95	sehat	1	1	0	0	0	0
96	sekian	0	0	0	0	1	0
97	sembuh	0	1	0	0	0	0
98	serang	0	0	0	0	1	0
99	serikat	0	0	1	1	0	0
100	sia	2	0	0	0	0	0
101	singapura	0	0	1	0	0	0
102	sistem	0	0	0	0	1	**2**
103	situs	0	0	0	1	0	0
104	smartphone	0	0	0	2	0	0
105	social	0	0	0	0	1	0
106	suasana	0	1	0	0	0	0

107	survei	0	0	0	2	0	**1**
108	susup	0	0	0	0	1	0
109	tablet	0	0	1	0	0	0
110	teknik	0	0	0	0	1	0
111	telah	0	0	1	0	0	0
112	terapi	0	2	0	0	0	0
113	tetangga	0	0	1	0	0	0
114	tingkat	1	0	0	0	0	0
115	titik	0	0	1	0	0	0
116	touch	0	0	0	1	0	0
117	tubuh	3	1	0	0	0	**2**
118	upaya	0	1	0	0	0	0
119	urine	1	0	0	0	0	0
120	vaksincom	0	0	0	0	1	0
121	varian	0	0	0	0	1	0
122	virus	0	0	0	0	4	0
123	vitamin	3	0	0	0	0	**4**
124	w	0	0	0	0	1	0
125	wajar	0	0	1	0	0	0
126	warna	0	4	0	0	0	**1**
127	wifi	0	0	2	0	0	0
128	xorer	0	0	0	0	1	0
129	yunani	0	1	0	0	0	0
Total		**30**	**32**	**40**	**38**	**38**	**13**

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

1. Kategori Kesehatan : $\Pr(v_j) = \frac{f_d(v_j)}{|D|} = \frac{2}{5} = 0.4$

2. Kategori Teknologi : $\Pr(v_j) = \frac{f_d(v_j)}{|D|} = \frac{3}{5} = 0.6$

Setelah $p(v_j)$ dihitung, langkah berikutnya adalah mencari nilai $p(w_j|v_j)$ dari masing-masing *term* di masing-masing kategori dengan menggunakan persamaan 2.10. Hasil perhitungan $p(w_j|v_j)$ terdapat pada tabel 3.12.

Tabel 3.12 Hasil Perhitungan $p(w_j|v_j)$

Term	Frekuensi Categori		Dok Uji	P(w|v)	
	Kes	Tek		Pend	Tek
adi	0	1	0	0.0052	0.0082
ajar	1	0	0	0.0105	0.0041
aksi	0	1	0	0.0052	0.0082
amerika	0	2	0	0.0052	0.0122
analis	0	1	0	0.0052	0.0082
ancam	0	1	0	0.0052	0.0082
android	0	1	0	0.0052	0.0082
animo	0	1	0	0.0052	0.0082
aplikasi	0	4	0	0.0052	0.0204
apple	0	3	0	0.0052	0.0163
april	0	1	0	0.0052	0.0082
asal	0	1	0	0.0052	0.0082
bangsa	1	0	0	0.0105	0.0041
beda	1	0	0	0.0105	0.0041
blackberry	0	1	0	0.0052	0.0082
buang	1	0	0	0.0105	0.0041
buat	0	1	0	0.0052	0.0082
buka	0	1	0	0.0052	0.0082
bulan	0	1	0	0.0052	0.0082
c	3	0	0	0.0209	0.0041
cari	0	1	0	0.0052	0.0082
cegah	1	0	0	0.0105	0.0041
china	1	0	0	0.0105	0.0041
daerah	**1**	**0**	**1**	**0.0105**	**0.0041**
dalam	0	1	0	0.0052	0.0082
dapat	0	1	0	0.0052	0.0082
dengar	0	1	0	0.0052	0.0082
dosis	1	0	0	0.0105	0.0041
dunia	0	1	0	0.0052	0.0082
efek	1	0	0	0.0105	0.0041
emosi	1	0	0	0.0105	0.0041

engineering	0	1	0	0.0052	0.0082
facebook	0	4	0	0.0052	0.0204
flu	1	0	0	0.0105	0.0041
frekuensi	1	0	0	0.0105	0.0041
fungsi	1	0	0	0.0105	0.0041
g	0	1	0	0.0052	0.0082
gandrung	0	1	0	0.0052	0.0082
ganggu	1	0	0	0.0105	0.0041
gaung	0	1	0	0.0052	0.0082
getar	1	0	0	0.0105	0.0041
guna	1	3	0	0.0105	0.0163
hal	0	1	0	0.0052	0.0082
hati	1	0	0	0.0105	0.0041
heran	0	1	0	0.0052	0.0082
hindar	1	0	0	0.0105	0.0041
holistik	1	0	0	0.0105	0.0041
hubung	1	0	0	0.0105	0.0041
indonesia	0	2	0	0.0052	0.0122
internet	0	1	0	0.0052	0.0082
ipad	0	5	0	0.0052	0.0245
iphone	0	1	0	0.0052	0.0082
ipod	0	1	0	0.0052	0.0082
jawab	0	1	0	0.0052	0.0082
kacau	0	1	0	0.0052	0.0082
kebal	1	0	0	0.0105	0.0041
kecuali	0	1	0	0.0052	0.0082
kelabui	0	1	0	0.0052	0.0082
keluarga	0	1	0	0.0052	0.0082
komputer	0	5	0	0.0052	0.0245
konsumsi	2	0	0	0.0157	0.0041
korban	0	1	0	0.0052	0.0082
kunjung	0	1	0	0.0052	0.0082
kuno	1	0	0	0.0105	0.0041
mampu	0	1	0	0.0052	0.0082
mancanegara	0	2	0	0.0052	0.0122
manfaat	2	0	0	0.0157	0.0041
mesir	1	0	0	0.0105	0.0041
milik	**1**	**2**	**1**	**0.0105**	**0.0122**

minat	0	2	0	0.0052	0.0122
mobile	0	2	0	0.0052	0.0122
negara	0	2	0	0.0052	0.0122
negeri	0	1	0	0.0052	0.0082
nielsen	0	1	0	0.0052	0.0082
obat	1	0	0	0.0105	0.0041
overdosis	1	0	0	0.0105	0.0041
paman	0	1	0	0.0052	0.0082
pas	1	0	0	0.0105	0.0041
pasar	0	1	0	0.0052	0.0082
pengaruh	**1**	**0**	**1**	**0.0105**	**0.0041**
percaya	1	0	0	0.0105	0.0041
pintar	0	1	0	0.0052	0.0082
piranti	0	1	0	0.0052	0.0082
platform	0	1	0	0.0052	0.0082
ponsel	0	1	0	0.0052	0.0082
produk	0	1	0	0.0052	0.0082
ribu	1	0	0	0.0105	0.0041
rilis	0	1	0	0.0052	0.0082
sakit	1	0	0	0.0105	0.0041
sam	0	1	0	0.0052	0.0082
sama	0	1	0	0.0052	0.0082
samping	1	0	0	0.0105	0.0041
saputera	0	1	0	0.0052	0.0082
sebar	0	1	0	0.0052	0.0082
sehat	2	0	0	0.0157	0.0041
sekian	0	1	0	0.0052	0.0082
sembuh	1	0	0	0.0105	0.0041
serang	0	1	0	0.0052	0.0082
serikat	0	2	0	0.0052	0.0122
sia	2	0	0	0.0157	0.0041
singapura	0	1	0	0.0052	0.0082
sistem	**0**	**1**	**2**	**0.0052**	**0.0082**
situs	0	1	0	0.0052	0.0082
smartphone	0	2	0	0.0052	0.0122
social	0	1	0	0.0052	0.0082
suasana	1	0	0	0.0105	0.0041
survei	**0**	**2**	**1**	**0.0052**	**0.0122**

susup	0	1	0	0.0052	0.0082
tablet	0	1	0	0.0052	0.0082
teknik	0	1	0	0.0052	0.0082
telah	0	1	0	0.0052	0.0082
terapi	2	0	0	0.0157	0.0041
tetangga	0	1	0	0.0052	0.0082
tingkat	1	0	0	0.0105	0.0041
titik	0	1	0	0.0052	0.0082
touch	0	1	0	0.0052	0.0082
tubuh	**4**	**0**	**2**	**0.0262**	**0.0041**
upaya	1	0	0	0.0105	0.0041
urine	1	0	0	0.0105	0.0041
vaksincom	0	1	0	0.0052	0.0082
varian	0	1	0	0.0052	0.0082
virus	0	4	0	0.0052	0.0204
vitamin	**3**	**0**	**4**	**0.0209**	**0.0041**
w	0	1	0	0.0052	0.0082
wajar	0	1	0	0.0052	0.0082
warna	**4**	**0**	**1**	**0.0262**	**0.0041**
wifi	0	2	0	0.0052	0.0122
xorer	0	1	0	0.0052	0.0082
yunani	1	0	0	0.0105	0.0041
Total	**62**	**116**	**13**		

Dari hasil perhitungan p(v) dan p(w|v), selanjutnya menentukan kategori dari dokumen uji.

a. Metode *Naïve Bayes*

Dengan persamaan 2.8 dicari peluang dari masing-masing kategori.

P(Kesehatan|DokUji) = 0.4*0.0105*0.0105*0.0105*0.0052*0.0052*
0.0262*0.0209*0.0262
= $1.80.10^{-16}$

P(Teknologi|DokUji) = 0.6*0.0041*0.0122*0.0041*0.0082*0.0122*
0.0041*0.0041*0.0041
= $8.48.10^{-19}$

Dari perhitungan di atas, kategori kesehatan mempunyai peluang yang lebih besar dibandingkan dokumen uji. Maka dokumen uji termasuk kategori kesehatan.

b. Metode *Multinomial Naïve Bayes*

Dengan persamaan 2.15 dicari peluang dari dokumen masing-masing kategori.

P(DokUji|Kesehatan)
= $0.0105*0.0105*0.0105*0.0052^2*0.0052*0.0262^2*0.0209^4*0.0262$
= $5.59.10^{-25}$

P(DokUji|Teknologi)
= $0.0041*0.0122*0.0041*0.0082^2*0.0122*0.0041^2*0.0041^4*0.0041$
= $3.28.10^{-30}$

Kemudian dengan persamaan 2.14 dicari peluang total dari dokumen uji.

P(DokUji) = $(0.4*5.59.10^{-25})+(0.6*3.28.10^{-30})$
= $2.24.10^{-25}$

Dengan persamaan 2.11 dicari peluang dari masing-masing kategori.

P(Kesehatan|DokUji) = $0.4*5.59.10^{-25}/ 2.24.10^{-25}$
= 0.998

P(Teknologi|DokUji) = $0.6*3.28.10^{-30}/ 2.24.10^{-25}$
= $8.79.10^{-6}$

Dari perhitungan MNB juga diketahui bahwa kategori kesehatan mempunyai peluang yang lebih besar dibandingkan dokumen uji. Maka dokumen uji termasuk kategori kesehatan.

3.6 Perancangan Uji Coba

Setelah proses pengembangan sistem, tahap selanjutnya adalah melakukan pengujian dan evaluasi terhadap sistem tersebut. Pengujian dilakukan mengevaluasi kategori dokumen yang dihasilkan, dengan membandingkan kategori dari sistem dengan kategori sebenarnya dari dokumen uji.

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 2.6.

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 empat kali uji coba dengan jumlah data latih yang berbeda, dengan proporsi 25 %, 50%, 75% dan 100 % dari data latih.

3.6.2 Hasil Evaluasi

Untuk mengetahui keberhasilan sistem dalam proses pengklasifikasian dokumen, dilakukan evaluasi terhadap sistem.

Tabel 3.13 Rancangan Evaluasi Klasifikasi

Kategori	a	b	c	Precision	Recall	F1
Ekonomi						
Kesehatan						
Pendidikan						
Olahraga						
Teknologi						

Tabel 3.14 adalah tabel hasil evaluasi perhitungan nilai *precision*, *recall*, dan F_1 *measure*.

Tabel 3.14 Tabel Evaluasi

Jumlah Data Latih	Precision	Recall	F_1 Measure
1			
2			
3			
…			
n			

BAB IV
IMPLEMENTASI DAN PEMBAHASAN

Implementasi merupakan proses transformasi representasi rancangan ke bahasa pemrograman yang dapat dimengerti oleh komputer. Pada bab ini akan dibahas hal-hal yang berkaitan dengan implementasi sistem pengklasifikasian dokumen.

4.1 Lingkungan Implemetasi

Lingkungan implementasi yang akan dijelaskan dalam sub-bab ini adalah lingkungan implementasi perangkat keras dan perangkat lunak.

4.1.1 Lingkungan Implementasi Perangkat Keras

Perangkat keras yang digunakan dalam pengembangan sistem pengklasifikasian dokumen ini adalah :

1. Processor Intel Pentium Dual-Core 1.83 GHz
2. RAM 1014 MB
3. Harddisk dengan kapasitas 160 GB

4.1.2 Lingkungan Implementasi Perangkat Lunak

Perangkat lunak yang akan digunakan dalam pengembangan sistem pengklasifikasian dokumen ini adalah :

1. Sistem operasi Microsoft Windows XP Professional
2. Borland Delphi 7
3. Text editor notepad
4. Microsoft Office Access 2007

4.2 Implementasi Program

Berdasarkan analisa dan perancangan proses yang terdapat pada sub-bab 3.3, maka pada sub-bab ini akan dijelaskan implementasi proses-proses tersebut.

4.2.1 Implementasi *Preprocessing*

4.2.1.1 *Case Folding*

Tahap awal dari *Preprocessing* adalah melakukan proses *case folding*. Tujuan dari proses ini adalah mengubah semua karakter huruf menjadi huruf kecil. Dalam proses ini karakter yang diterima adalah karakter huruf, sedangkan karakter selain huruf seperti tanda baca akan diganti dengan karakter spasi. Proses *case folding* ditunjukkan pada Kode Program 4.1.

```
teks := LowerCase(teks);
for i := 0 to 96 do
  teks := StringReplace(teks,char(i),' ',[rfReplaceAll]);
for i := 123 to 254 do
  teks := StringReplace(teks,char(i),' ',[rfReplaceAll]);
```

Kode Program 4.1 Proses *Case Folding*

4.2.1.2 *Tokenizing*

Proses ini merupakan proses penguraian kata (*parsing*) dari dokumen teks yang bertujuan untuk mendapatkan potongan-potongan kata tunggal. Masukan dari proses *parsing* ini adalah *string* yang dihasilkan dari proses *case folding* dan akan memberikan nilai kembalian berupa *list* yang berisi kumpulan *string* kata tunggal. Proses *tokenizing* ditunjukkan pada Kode Program 4.2. Selanjutnya hasil dari proses ini kemudian disimpan di basis data. Proses penyimpan ditunjukan di Kode Program 4.3.

```
procedure Split
(const Delimiter: Char; // delimiter charachter
 Input: string;         // input string
 const Strings: TStrings) ; // list of string result
begin
  Assert(Assigned(Strings)) ;
  Strings.Clear;
  Strings.Delimiter := Delimiter;
  Strings.DelimitedText := Input;
end;
```

Kode Program 4.2 Procedure *Split*

```
sDel := 'DELETE FROM KATAALL';
DM.QDel.Close;
DM.QDel.SQL.Clear;
DM.QDel.SQL.Add(sDel);
```

```
DM.QDel.ExecSQL;
ListKata := TStringList.Create;
Split(' ', teks, ListKata) ;
n := ListKata.Count;
ListKata.Sort;
DM.TKataAll.Open;
for i := 0 to n-1 do
begin
  kata := ListKata[i];
  DM.TKataAll.AppendRecord([kata]);
end;
ListKata.Free;
DM.TKataAll.Close;
```

Kode Program 4.3 Proses *Tokenizing*

4.2.1.3 Filtering

Proses *filtering* bertujuan untuk memperoleh kata-kata penting dan menghilangkan/menghapus kata-kata yang tidak relevan dan tidak merefleksikan isi dokumen (*stopword*). Proses ini diawali dengan membentuk sebuah *stoplist*, yaitu daftar kata yang berisi sekumpulan *stopword*. Kemudian dilanjutkan dengan membandingkan *list* hasil *parsing* dengan *stoplist* yang telah dibuat. Proses *filtering* ditunjukkan pada Kode Program 4.4.

```
s := 'DELETE FROM KATAALL WHERE KATA IN (SELECT KATA FROM
    STOPWORD)';
DM.QFiltering.Close;
DM.QFiltering.SQL.Clear;
DM.QFiltering.SQL.Add(s);
DM.QFiltering.ExecSQL;
```

Kode Program 4.4 Proses *Filtering*

4.2.1.4 Stemming

Proses *stemming* bertujuan mencari *root* kata dari tiap kata hasil *filtering*. Proses ini akan menghilangkan imbuhan dalam bahasa Indonesia yang terdiri dari sufiks (akhiran) dan prefiks (awalan). Untuk keseluruhan proses *stemming* ditunjukkan pada Kode Program 4.5.

```
function Stemming(term:string):String;
var kata: string;
    len : integer;
begin
  kata := term;
  // first step
  if ((AnsiEndsStr('lah', kata)) or
     (AnsiEndsStr('kah', kata)) or
     (AnsiEndsStr('pun', kata))) then
  begin
    len := (StrLen(Pchar(kata)-3));
    if (countVocal(Copy(kata, 0,len))>=2) then
      kata := Copy (kata, 0,len-3);
  end;

  //second step
  kata := StrLower(Pchar(kata));
  if ((AnsiEndsStr('ku', kata)) or
     (AnsiEndsStr('mu', kata)) or
     (AnsiEndsStr('nya', kata))) then
  begin
    if (AnsiEndsStr('nya', kata)) then
      len := StrLen(Pchar(kata))-3
    else
      len := StrLen(Pchar(kata))-2;
    if (countVocal(Copy (kata, 0,len))>=2) then
      kata := Copy (kata, 0,len);
  end;

  //third step
  if (RemFirstPrefix(kata,kata)) then
  begin
    kata := kata;
    if (not (RemSuffix(kata,kata))) then
      kata := kata;
    if (RemSecondPrefix(kata,kata)) then
    begin
      kata := kata;
    end;
  end
  else if (RemSecondPrefix(kata,kata)) then
  begin
    kata := kata;
    if (RemSuffix(kata,kata)) then
      begin
        kata := kata;
      end;
   end;
   Result := kata;
end;
```

Kode Program 4.5 Proses *Stemming*

Pada proses *stemming* terdiri dari beberapa tahapan. Yang pertama adalah proses untuk menghilangkan sufiks (akhiran). Pada proses ini, setiap kata yang berakhiran ‘kan’, ‘an’ atau ‘i’ akan dilakukan penghilangan sufiks. Proses ini ditunjukkan di Kode Program 4.6.

```
function RemSuffix(input : string;var output : string) :
      boolean;
var
  isfired : boolean;
  kata : string;
  len : integer;
begin
  isfired := false;

  kata := StrLower(Pchar(input));
    if (AnsiEndsStr('kan', kata)) then
    begin
      len := StrLen(Pchar(kata))-3;
      if (countVocal(Copy (kata, 0,len))>=2) then
      begin
        output := Copy (kata, 0,len);
        isfired := true;
      end;
    end
    else
    if ((AnsiEndsStr('an', kata)) or
        (AnsiEndsStr('i', kata))) then
    begin
      if (AnsiEndsStr('an', kata)) then
        len := StrLen(Pchar(kata))-2
      else
        len := StrLen(Pchar(kata))-1;
      if (countVocal(Copy (kata, 0,len))>=2) then
      begin
        output := Copy (kata, 0,len);
        isfired := true;
      end;
    end;
    Result := isfired;
end;
```

Kode Program 4.6 Fungsi *RemSuffix*

Proses selanjutnya adalah proses untuk menghilangkan prefiks (awalan). Pada proses ini, setiap kata yang mempunyai awalan akan dilakukan penghilangan prefiks. Proses menghilangkan prefik bentuk pertama ditunjukkan di Kode Program 4.7. Proses menghilangkan prefik bentuk kedua ditunjukkan di Kode Program 4.8.

```
function RemFirstPrefix(input : string; var output : string) :
     boolean;
var
  isfired : boolean;
  kata, temp : string;
  len, index : integer;
begin
  isfired := false;

  //meng peng
  kata := StrLower(Pchar(input));
    if ((AnsiStartsStr('meng', kata)) or
        (AnsiStartsStr('peng', kata))) then
    begin
      len := StrLen(Pchar(kata))-4;
      if (countVocal(Copy (kata, 5,len))>=2) then
      begin
        output := Copy (kata, 5,len);
        isfired := true;
      end;
    end
    else
    //meny peny
    if ((AnsiStartsStr('meny', kata)) or
        (AnsiStartsStr('peny', kata))) then
    begin
      len := StrLen(Pchar(kata))-3;
      if (countVocal(Copy (kata, 4,len))>=2) then
      begin
        kata := Copy (kata, 4,len);
        temp := Pchar(kata);
        temp[1] := 's';
        output := temp;
        isfired := true;
      end;
    end
    else
    //men pen
    if ((AnsiStartsStr('men', kata)) or
        (AnsiStartsStr('pen', kata))) then
    begin
      len := StrLen(Pchar(kata))-2;
      if (countVocal(Copy (kata, 4,len))>=2) then
      begin
        temp := Copy (kata, 3,len);
        if (temp[2] in ['a','i','u','e','o']) then
          temp[1] := 't'
        else
          temp := Copy(temp, 2,strlen(Pchar(temp)));
        output := temp;
        isfired := true;
      end;
    end
    else
```

```
    //mem pem
    if ((AnsiStartsStr('mem', kata)) or
        (AnsiStartsStr('pem', kata))) then
    begin
      len := StrLen(Pchar(kata))-2;
      if (countVocal(Copy (kata, 4,len))>=2) then
      begin
        temp := Copy (kata, 3,len);
        if (temp[2] in ['a','i','u','e','o']) then
          temp[1] := 'p'
        else
          temp := Copy(temp, 2,strlen(Pchar(temp)));
        output := temp;
        isfired := true;
      end;
    end
    else
    //me
    if (AnsiStartsStr('me', kata)) then
    begin
      len := StrLen(Pchar(kata))-2;
      if (countVocal(Copy (kata, 3,len))>=2) then
      begin
        output := Copy (kata, 3,len);
        isfired := true;
      end;
    end
    else
    //di ter ke
    if ((AnsiStartsStr('di', kata)) or
        (AnsiStartsStr('ter', kata)) or
        (AnsiStartsStr('ke', kata))) then
    begin
      if (AnsiStartsStr('ter', kata)) then
      begin
        len := StrLen(Pchar(kata))-3;
        index := 4;
      end
      else
      begin
        len := StrLen(Pchar(kata))-2;
        index := 3;
      end;
      if (countVocal(Copy (kata, index,len))>=2) then
      begin
        output := Copy (kata, index,len);
        isfired := true;
      end;
    end;
    Result := isfired;
end;
```

Kode Program 4.7 Fungsi *RemFirstPrefix*

```
function RemSecondPrefix(input : string;var output : string) :
     boolean;
var
  isfired : boolean;
  kata, temp,ajar : string;
  len : integer;
begin
  isfired := false;
  ajar := 'ajar';
  kata := StrLower(Pchar(input));
    if ((AnsiStartsStr('ber', kata)) or
       (AnsiStartsStr('per', kata))) then
    begin
      len := StrLen(Pchar(kata))-3;
      if (countVocal(Copy (kata, 4,len))>=2) then
      begin
        output := Copy (kata, 4,len);
        isfired := true;
      end;
    end
    else
    //bel pel
    if ((AnsiStartsStr('bel', kata)) or
       (AnsiStartsStr('pel', kata))) then
    begin
      if ((StrPos(Pchar(kata),Pchar(ajar))<> nil)) then
      begin
        output := 'ajar';
      end
      else
      begin
        len := StrLen(Pchar(kata))-2;
        temp := Copy (kata, 3,len);
        if (countVocal(Copy (kata, 2,len))>=2) then
        begin
          output := Copy (kata, 3,len);
        end;
      end;
      isfired := true;
    end
    else
    //be
    if (AnsiStartsStr('be', kata)) then
    begin
      len := StrLen(Pchar(kata))-2;
      if (countVocal(Copy (kata, 3,len))>=2) then
      begin
        temp := Copy (kata, 3,len);
        if ((temp[2]='e') and (temp[3]='r')) then
        begin
          output := temp;
          isfired := true;
        end
      end;
    end
```

```
    else
    //pe
    if (AnsiStartsStr('pe', kata)) then
    begin
      len := StrLen(Pchar(kata))-2;
      if (countVocal(Copy (kata, 3,len))>=2) then
      begin
        output := Copy (kata, 3,len);
        isfired := true;
      end;
    end ;
    result := isfired
end;
```

Kode Program 4.8 Fungsi *RemSecondPrefix*

4.2.1.5 Perhitungan *Term Frequency*

Perhitungan *Term Frequency* yaitu menghitung jumlah kemunculan tiap kata (*term*) yang terkandung dalam dokumen. Daftar kata yang diperoleh dari hasil proses sebelumnya (proses *stemming*) digunakan sebagai parameter *input*. Hasil akhir dari proses ini adalah sebuah daftar kata (*term*) yang menyusun dokumen beserta kemunculannya yang nantinya akan disimpan dalam basis data. Proses perhitungan *TF* ini dilakukan untuk isi dari dokumen latih serta isi dari dokumen uji. Proses perhitungan *TF* ini ditunjukkan di Kode Program 4.9.

```
procedure HitungFrekuensi(tabel,dok,id:string);
var sDel, sTerm, sAppend, term : string;
    n, frek : integer;
begin
  sDel := 'DELETE FROM '+tabel+' WHERE '+
          'ID_'+dok+'='+QuotedStr(id);
  DM.QDel.Close;
  DM.QDel.SQL.Clear;
  DM.QDel.SQL.Add(sDel);
  DM.QDel.ExecSQL;
  sAppend:='INSERT INTO '+tabel+' (ID_'+dok+',TERM,FREKUENSI)'+
           ' SELECT '+QuotedStr(id)+' ,KATA, COUNT(KATA) FROM'+
           ' KATASTEMMING GROUP BY '+QuotedStr(id)+' ,KATA';
  DM.QAppend.Close;
  DM.QAppend.SQL.Clear;
  DM.QAppend.SQL.Add(sAppend);
  DM.QAppend.ExecSQL;
end;
```

Kode Program 4.9 Proses Perhitungan *TF*

Pada proses ini terdapat 3 parameter, yaitu *tabel* yang menunjukkan tabel yang akan diisi (Tabel TermLatih untuk kata isi dokumen latih dan Tabel TermUji untuk kata isi dari dokumen uji). Parameter kedua yaitu *dok* yang digunakan untuk membedakan dokumen latih dan dokumen uji. Parameter ketiga yaitu *id* yang menunjukan id dari dokumen tersebut.

4.2.2 Implementasi Pembelajaran *MNB*

Proses pembelajaran ini diawali dengan mengumpulkan semua *term* di dokumen latih. Selanjutnya menghitung peluang setiap term di setiap kategori. *Term* dan peluang terhadap kategori ini kemudian disimpan di basis data sebagai data pembelajaran. Proses pembelajaran *term* terhadap kategori ini ditunjukan di Kode Program 4.10.

```
procedure TFormLatih.MNB;
var sTerm, sJmlFrek, sFrek, sDel, sPeluang: string;
    idKat, term: string;
    JmlTerm, totalFrek, i : integer;
    Start, Stop : TDateTime;
begin
  Start := now;
  sDel := 'DELETE FROM PELUANGTERM';
  DM.QDel.Close;
  DM.QDel.SQL.Clear;
  DM.QDel.SQL.Add(sDel);
  DM.QDel.ExecSQL;
  // Mengumpulkan semua TERM
  sTerm := 'SELECT DISTINCT TERM FROM TERMLATIH';
  DM.QTerm.Close;
  DM.QTerm.SQL.Clear;
  DM.QTerm.SQL.Add(sTerm);
  DM.QTerm.ExecSQL;
  DM.QTerm.Open;
  JmlTerm := DM.QTerm.RecordCount;
  DM.QTerm.First;
  i := 1;
  DM.TPeluangTerm.Open;
  DM.QTerm.First;
  while NOT DM.QTerm.Eof do
  begin
    FormLatih.Refresh;
    term := DM.QTerm.FieldValues['TERM'];
    // Hitung Frekuensi tiap Term tiap Kategori
    sFrek := 'INSERT INTO PELUANGTERM (ID_KATEGORI, TERM, '+
             'FREKUENSI) '+
             'SELECT  K.ID_KATEGORI, '+QuotedStr(term)+', ' +
             '(IIF(FREK IS NULL,0,FREK)) as FREQ '+
             'FROM KATEGORI K LEFT OUTER JOIN ( '+
```

```
            'SELECT DL.ID_KATEGORI,SUM(TL.FREKUENSI) AS FREK'+
            'FROM DOKLATIH DL, TERMLATIH TL '+
            'WHERE TL.ID_CORPUS=DL.ID_CORPUS AND '+
            'TL.TERM='+QuotedStr(term)+
            ' GROUP BY DL.ID_KATEGORI) AS Q '+
            'ON K.ID_KATEGORI=Q.ID_KATEGORI';
    DM.QFrek.Close;
    DM.QFrek.SQL.Clear;
    DM.QFrek.SQL.Add(sFrek);
    DM.QFrek.ExecSQL;
    DM.QTerm.Next;
    inc(i);
  end;
  DM.TPeluangTerm.Close;
  // Hitung jumlah frekuensi semua Term tiap kategori
  sJmlFrek := 'SELECT DL.ID_KATEGORI, SUM(TL.FREKUENSI) as '+
              'TOTALFREK '+
              'FROM TERMLATIH TL, DOKLATIH DL '+
              'WHERE TL.ID_CORPUS=DL.ID_CORPUS GROUP BY '+
              'DL.ID_KATEGORI';
  DM.QKat.close;
  DM.QKat.SQL.Clear;
  DM.QKat.SQL.Add(sJmlFrek);
  DM.QKat.ExecSQL;
  DM.QKat.Open;
  DM.QKat.First;
  while NOT DM.QKat.Eof do
  begin
    idKat := DM.QKat.FieldValues['ID_KATEGORI'];
    totalFrek := DM.QKat.FieldValues['TOTALFREK'];
    // Hitung Peluang tiap Term tiap Kategori
    sPeluang := 'UPDATE PELUANGTERM SET '+
     'PELUANG=(FREKUENSI+1)/('+IntToStr(JmlTerm+totalFrek)+')'+
     'WHERE ID_KATEGORI='+QuotedStr(idKat);
    DM.QPeluang.Close;
    DM.QPeluang.SQL.Clear;
    DM.QPeluang.SQL.Add(sPeluang);
    DM.QPeluang.ExecSQL;
    DM.QKat.Next;
  end;
  Stop := now;
end;
```

Kode Program 4.10 Proses Pembelajaran *MNB*

4.2.3 Implementasi Pengklasifikasian *MNB*

Proses pengklasifikasian dokumen dengan metode *MNB* ini digunakan dengan menentukan kategori dari dokumen uji. Proses pemilihan kategori ini ditunjukkan di Kode Program 4.11.

```
function TFormUtama.GetClass(id:string):string;
var sKat, sAppend, kat : string;
    n : integer;
    totP, PLama, PBaru : double;
begin
  DM.TDokLatih.Open;
  n := DM.TDokLatih.RecordCount;
  DM.TDokLatih.Close;
  sKat := 'SELECT Q2.ID, K.ID_KATEGORI, (Q1.PV*Q2.PW) AS '+
          'PELUANG '+
          'FROM KATEGORI K, '+
          '(SELECT DL.ID_KATEGORI, '+
          'COUNT(DL.ID_CORPUS)/'+IntToStr(n)+' AS PV '+
          'FROM DOKLATIH DL '+
          'GROUP BY DL.ID_KATEGORI) AS Q1, '+
          '(SELECT PT.ID_KATEGORI, U.ID_UJI AS ID, '+
          'SUM(PT.PELUANG^U.FREKUENSI) AS PW '+
          'FROM PELUANGTERM PT, TERMUJI U '+
          'WHERE PT.TERM=U.TERM AND  U.ID_UJI='+QuotedStr(id)+
          ' GROUP BY PT.ID_KATEGORI, U.ID_UJI ) AS Q2 '+
          'WHERE K.ID_KATEGORI=Q1.ID_KATEGORI AND '+
          'K.ID_KATEGORI=Q2.ID_KATEGORI '+
          'ORDER BY (Q1.PV*Q2.PW) DESC ';
  DM.QKatUji.Close;
  DM.QKatUji.SQL.Clear;
  DM.QKatUji.SQL.Add(sKat);
  DM.QKatUji.ExecSQL;
  DM.QKatUji.Open;
  DM.QKatUji.First;
  totP := 0;
  while NOT DM.QKatUji.Eof do
  begin
     totP := totP+DM.QKatUji.FieldValues['PELUANG'];
     DM.QKatUji.Next;
  end;
  DM.QKatUji.First;
  while NOT DM.QKatUji.Eof do
  begin
    kat := DM.QKatUji.FieldValues['ID_KATEGORI'];
    PLama := DM.QKatUji.FieldValues['PELUANG'];
    PBaru := PLama/totP;
    sAppend := 'INSERT INTO KATEGORIUJI VALUES'+
               '('+QuotedStr(id)+','+QuotedStr(kat)+
               ','+FloatToStr(PBaru)+')';
    DM.QAppend.Close;
    DM.QAppend.SQL.Clear;
    DM.QAppend.SQL.Add(sAppend);
    DM.QAppend.ExecSQL;
    DM.QKatUji.Next;
  end;
  DM.QKatUji.First;
  Result := DM.QKatUji.FieldValues['ID_KATEGORI'];
end;
```

Kode Program 4.11 Proses Pengklasifikasian *MNB*

4.3 Implementasi Basis Data

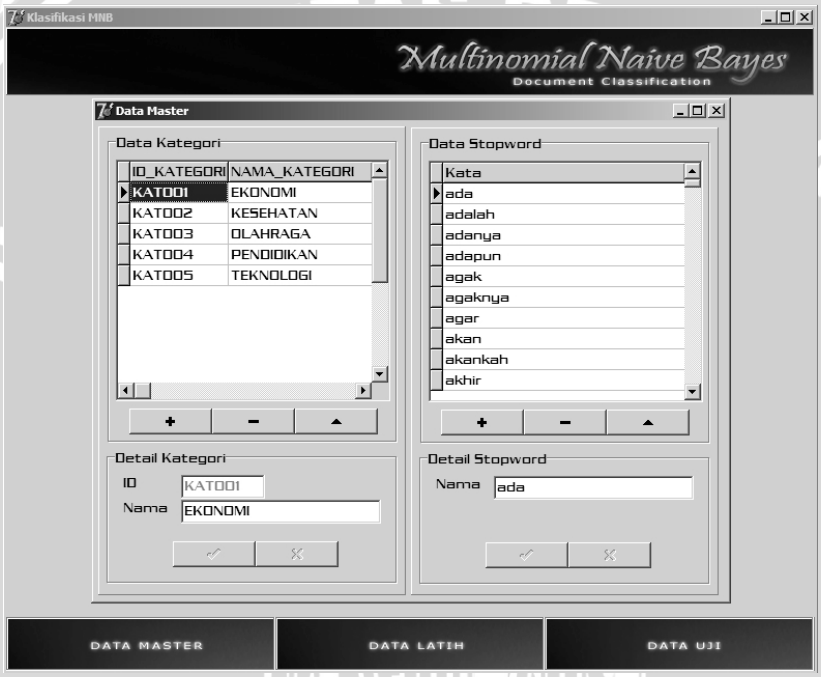
Pada Sub-bab 3.3.2 sebelumnya telah dijelaskan rancangan tabel dan relasi basis data yang terdiri dari sepuluh tabel. Dari tabel-tabel tersebut beserta relasinya diimplementasikan menggunakan Microsoft Office Access 2007.

4.4 Implementasi Antarmuka

Antarmuka sistem pengklasifikasian dokumen ini terdiri dari empat bagian seperti yang ditunjukkan pada Gambar 4.1 sampai dengan Gambar 4.4.

4.4.1 Tampilan Antarmuka Data Master

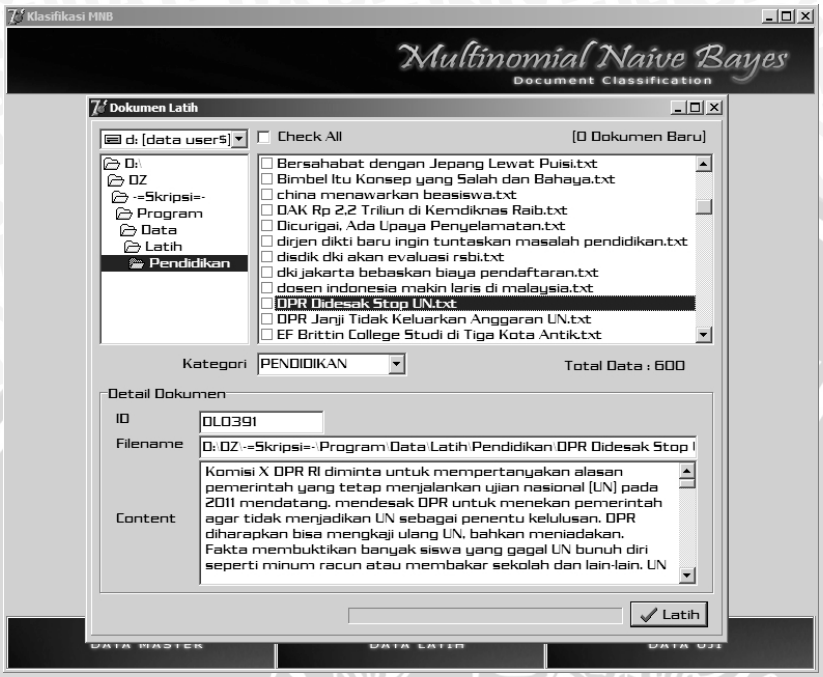
Data Master adalah daftar kategori dan daftar *stopword* yang digunakan dalam aplikasi ini. Tampilan data master ditunjukkan di Gambar 4.1.



Gambar 4.1 Tampilan Antarmuka Data Master

4.4.2 Tampilan Antar Muka Data Latih

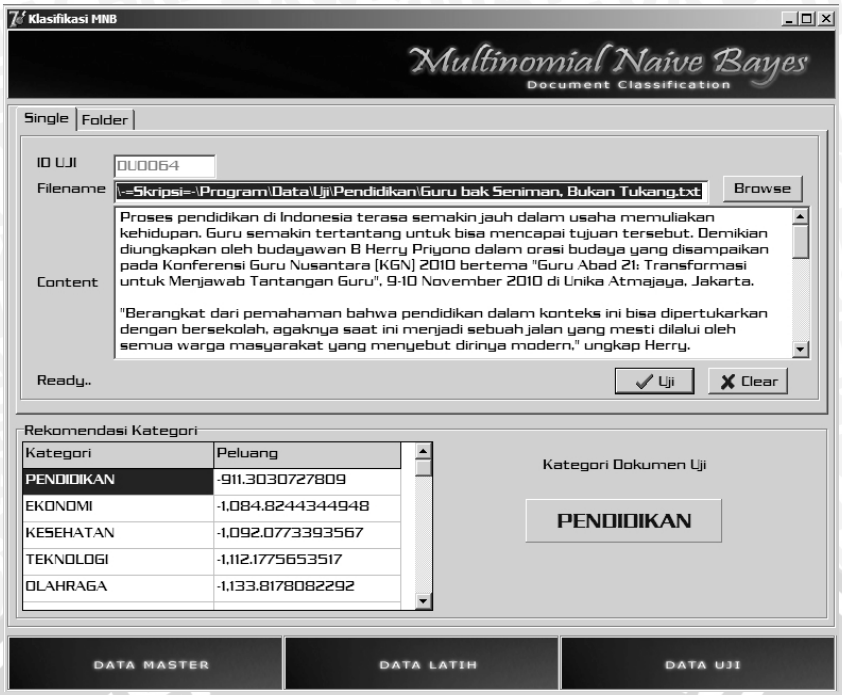
Data latih dalam aplikasi ini berupa file teks (*.txt) yang berada dalam satu *folder*. Data latih tiap kategori diletakan di *folder* yang berbeda. Untuk setiap ada dokumen baru dalam *folder* tersebut, maka aplikasi akan mengidentifikasi dengan tanda *check*. Setiap ada dokumen baru, maka aplikasi akan menghitung ulang peluang *term* terhadap kategori. Tampilan untuk data latih ini ditunjukkan di Gambar 4.2.



Gambar 4.2 Antarmuka Data Latih

4.4.3 Tampilan Antarmuka Pengujian

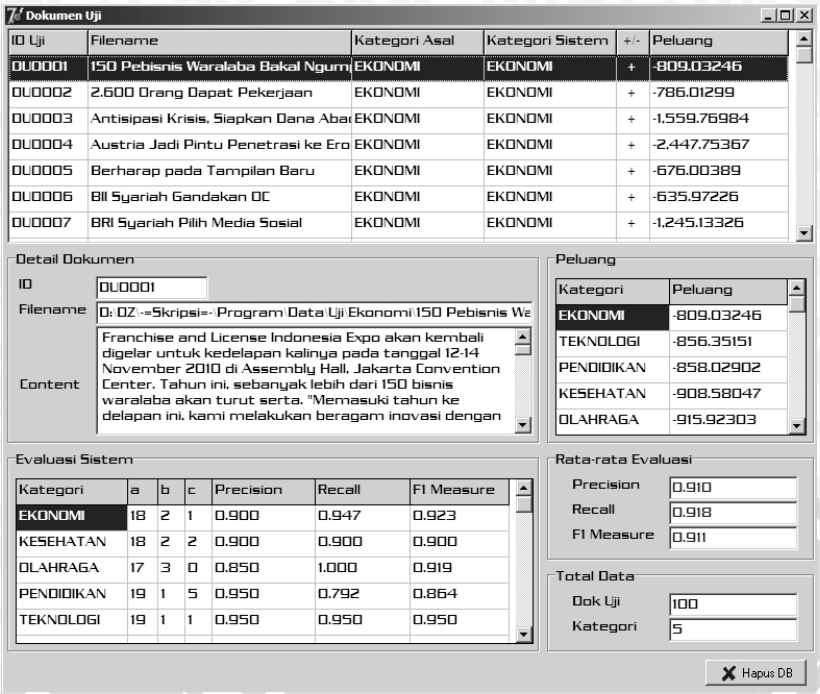
Tampilan antarmuka pengujian merupakan tampilan awal dari aplikasi ini. Proses pengujian ini menggunakan file teks (*.*txt*) dengan cara menekan tombol *browse* untuk memilih dokumen yang di uji. Kemudian tekan tombol uji, maka dokumen tersebut akan dihitung peluang kategorinya. Tampilan antarmuka pengujian ditunjukkan pada Gambar 4.3.



Gambar 4.3 Tampilan Antarmuka Pengujian

4.4.4 Tampilan Antarmuka Data Uji

Antarmuka data uji digunakan untuk menampilkan semua dokumen-dokumen yang telah dilakukan pengujian yang kemudian digunakan sebagai bahan evaluasi sistem. Tampilan antarmuka ini ditunjukkan pada Gambar 4.4.



Gambar 4.4 Tampilan AntarMuka Data Uji

4.5 Implementasi Uji Coba

Pada sub-bab ini akan dilakukan pembahasan mengenai pengujian yang telah dilakukan pada sistem dan hasil evaluasi dari sistem tersebut.

4.5.1 Skenario Evaluasi

Pada pengujian sistem pengklasifikasian dokumen berita, sekumpulan dokumen dibagi menjadi dokumen latih dan dokumen uji. Uji coba dilakukan berdasarkan pada skenario evaluasi pada Sub-bab 3.6.1. Pada dokumen latih menggunakan 600 buah dokumen yang sebelumnya telah diketahui kategorinya dan berasal dari 5 kategori yang berbeda. Sedangkan pada dokumen uji menggunakan 100 buah dokumen yang akan dicari kategorinya oleh sistem.

4.5.2 Hasil Evaluasi

4.5.2 Hasil Evaluasi

4.5.2.1 Evaluasi Klasifikasi

Evaluasi klasifikasi digunakan untuk mengukur keberhasilan sistem dalam mengklasifikasikan dokumen. Pada uji coba pengklasikasian, jumlah dokumen masing-masing kategori sama. Pada uji coba pertama digunakan data latih sebesar 150 buah (25%), uji coba kedua digunakan data latih sebesar 300 buah (50%), uji coba ketiga digunakan data latih sebesar 450 buah (75%) dan uji coba keempat digunakan data latih sebesar 600 buah (100%). Hasil evaluasi klasifikasi ditunjukkan pada tabel 4.1 sampai dengan 4.4.

Tabel 4.1 Evaluasi Klasifikasi Uji Coba Pertama (25%)

Kategori	a	b	c	Precision	Recall	F_1
Ekonomi	16	4	4	0.800	0.800	0.800
Kesehatan	17	3	4	0.850	0.810	0.829
Pendidikan	14	6	4	0.700	0.778	0.737
Olahraga	17	3	6	0.850	0.739	0.791
Teknologi	15	5	3	0.750	0.833	0.789
Rata-rata				**0.790**	**0.792**	**0.789**

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 4.2 Evaluasi Klasifikasi Uji Coba Kedua (50%)

Kategori	a	b	c	Precision	Recall	F_1
Ekonomi	18	2	2	0.900	0.900	0.900
Kesehatan	17	3	3	0.850	0.850	0.850
Pendidikan	16	4	3	0.800	0.842	0.821
Olahraga	18	2	5	0.900	0.783	0.837

Teknologi	16	4	2	0.800	0.889	0.842
Rata-rata				**0.850**	**0.853**	**0.850**

Tabel 4.3 Evaluasi Klasifikasi Uji Coba Ketiga (75%)

Kategori	**a**	**b**	**c**	**Precision**	**Recall**	**F_1**
Ekonomi	18	2	1	0.900	0.947	0.923
Kesehatan	17	3	2	0.850	0.895	0.872
Pendidikan	17	3	2	0.850	0.895	0.872
Olahraga	19	1	5	0.950	0.792	0.864
Teknologi	18	2	1	0.900	0.947	0.923
Rata-rata				**0.890**	**0.895**	**0.891**

Tabel 4.4 Evaluasi Klasifikasi Uji Coba Keempat (100%)

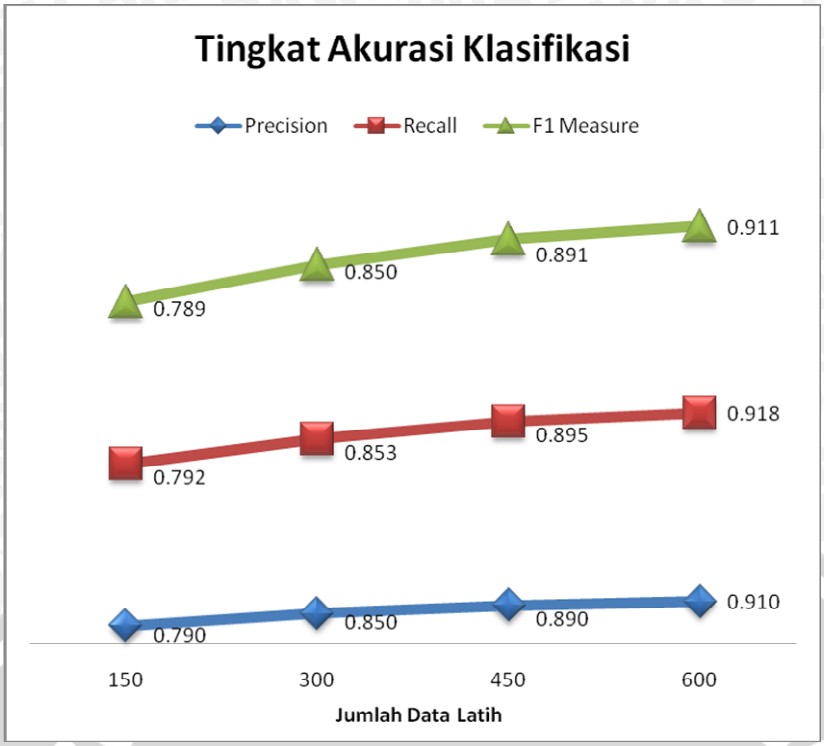
Kategori	**a**	**b**	**c**	**Precision**	**Recall**	**F_1**
Ekonomi	18	2	1	0.900	0.947	0.923
Kesehatan	18	2	2	0.900	0.900	0.900
Pendidikan	17	3	0	0.850	1.000	0.919
Olahraga	19	1	5	0.950	0.792	0.864
Teknologi	19	1	1	0.950	0.950	0.950
Rata-rata				**0.910**	**0.918**	**0.911**

Maka dari tabel evaluasi klasifikasi empat kali uji coba diatas dapat dibuat tabel rata-rata nilai *precision*, *recall* dan *F_1 measure* seperti pada tabel 4.5.

Tabel 4.5 Hasil Evaluasi Klasifikasi

Jumlah Data Latih	**Precision**	**Recall**	**F_1 Measure**
150 (25%)	0.790	0.792	0.789
300 (50%)	0.850	0.853	0.850
450 (75%)	0.890	0.895	0.891
600 (100%)	0.910	0.918	0.911
Rata-rata	**0.860**	**0.864**	**0.860**

Dari hasil tabel 4.5 dapat disajikan dalam bentuk grafik seperti pada Gambar 4.5.



Gambar 4.5 Grafik Hasil Evaluasi Klasifikasi

4.5.3 Analisa Hasil

Dari hasil evaluasi klasifikasi, didapatkan nilai rata-rata *precision*, *recall* dan F_1 *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 F_1 *measure*. Tetapi dibutuhkan kecermatan dalam memilih besarnya jumlah data latih, agar sistem dapat berjalan dengan baik dan seimbang.

Pada percobaan yang telah dilakukan, peningkatan terbesar nilai *precision*, *recall* dan F_1 *measure* terjadi pada saat data latih ditambah dari 150 menjadi 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. Akan tetapi pada penambahan data latih menjadi 450 dan 600 buah, sistem juga mengalami peningkatan akurasi yang cukup signifikan. Dengan demikian dapat dikatakan bahwa pada jumlah data latih sebesar 600 buah, sistem masih dapat bekerja secara optimal. Dari empat kali uji coba didapatkan rata-rata dan nilai *precision* sebesar 0.860 dan nilai *recall* sebesar 0.864 serta nilai F_1 *measure* sebesar 0.860, sehingga dapat disimpulkan bahwa tingkat akurasi sistem secara rata-rata sudah berjalan dengan baik.

.

BAB V
PENUTUP

5.1 Kesimpulan

Sistem pengklasifikasian dokumen berita berbahasa Indonesia dengan metode *Multinomial Naïve Bayes (MNB)* yang datanya bersumber dari www.kompas.com dan www.detik.com, mempunyai spesifikasi sebagai berikut :

1. Sistem ini menghasilkan nilai rata-rata *precision* sebesar 0.860, rata-rata *recall* sebesar 0.864 dan rata-rata F_1 *measure* sebesar 0.860.
2. Sistem ini mengalami peningkatan yang paling signifikan pada saat data latih ditambah dari 150 menjadi 300 buah. Sehingga jumlah data latih sebesar 300 buah sudah cukup untuk dapat mengklasifikasikan dokumen dengan baik.

5.2 Saran

Saran pengembangan lebih lanjut yang dapat diberikan setelah mengerjakan Skripsi ini adalah perlu kecermatan dalam memilih isi dari data latih sehingga sistem dapat berjalan dengan lebih optimal.

DAFTAR PUSTAKA

Arifin, Agus Z dan Setiono Ari N. 2002. *Klasifikasi Dokumen Berita Kejadian Berbahasa Indonesia dengan Algoritma Single Pass Clustering*. Institut Teknologi Sepuluh Nopember(ITS). Surabaya.

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.

Dumais, Susan, John Platt, David Heckerman, dan Mehran Sahami. 1998. *Inductive Learning Algorithms and Representations for Text Categorization*. AAAI 98 Workshop on Text Categorization.

Even,Yahir dan Zohar. 2002. *Introduction to Text Mining*. Automated Learning Group National Center For Supercomputing Applications. University of Illionis.

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

Hamilton, H dan Olive, W. 2003. Confusion Matrix.

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.

Lewis, D, 1995. *Evaluating and Optimizing Autonomous Text Classification Systems*. AT&T Bell Laboratories Murray Hill, NJ 07974. USA. Proceedings of the Eighteenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, July, 1995, pp. 246–254.

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. 1995. *Ilmu Bahasa Indonesia : Morfologi Suatu Tinjauan Deskriptif*. CV. Karyono. Yogyakarta.

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

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.

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

LAMPIRAN 1

Daftar Stopword

1	ada	36	awalnya	71	benarlah
2	adalah	37	bagai	72	berada
3	adanya	38	bagaikan	73	berakhir
4	adapun	39	bagaimana	74	berakhirlah
5	agak	40	bagaimanakah	75	berakhirnya
6	agaknya	41	bagaimanapun	76	berapa
7	agar	42	bagi	77	berapakah
8	akan	43	bagian	78	berapalah
9	akankah	44	bahkan	79	berapapun
10	akhir	45	bahwa	80	berarti
11	akhiri	46	bahwasanya	81	berawal
12	akhirnya	47	baik	82	berbagai
13	aku	48	bakal	83	berdatangan
14	akulah	49	bakalan	84	beri
15	amat	50	balik	85	berikan
16	amatlah	51	banyak	86	berikut
17	anda	52	bapak	87	berikutnya
18	andalah	53	baru	88	berjumlah
19	antar	54	bawah	89	berkali
20	antara	55	beberapa	90	berkata
21	antaranya	56	begini	91	berkehendak
22	apa	57	beginian	92	berkeinginan
23	apaan	58	beginikah	93	berkenaan
24	apabila	59	beginilah	94	berlainan
25	apakah	60	begitu	95	berlalu
26	apalagi	61	begitukah	96	berlangsung
27	apatah	62	begitulah	97	berlebihan
28	artinya	63	begitupun	98	bermaksud
29	asal	64	bekerja	99	bermula
30	asalkan	65	belakang	100	bersama
31	atas	66	belakangan	101	bersiap
32	atau	67	belum	102	bertanya
33	ataukah	68	belumlah	103	berturut
34	ataupun	69	benar	104	bertutur
35	awal	70	benarkah	105	berujar

106	berupa	141	demikian	176	dilalui
107	besar	142	demikianlah	177	dilihat
108	betul	143	dengan	178	dimaksud
109	betulkah	144	depan	179	dimaksudkan
110	biasa	145	di	180	dimaksudkannya
111	biasanya	146	dia	181	dimaksudnya
112	bila	147	diakhiri	182	diminta
113	bilakah	148	diakhirinya	183	dimintai
114	bisa	149	dialah	184	dimisalkan
115	bisakah	150	diantara	185	dimulai
116	boleh	151	diantaranya	186	dimulailah
117	bolehkah	152	diberi	187	dimulainya
118	bolehlah	153	diberikan	188	dimungkinkan
119	buat	154	diberikannya	189	dini
120	bukan	155	dibuat	190	dipastikan
121	bukankah	156	dibuatnya	191	diperbuat
122	bukanlah	157	didapat	192	diperbuatnya
123	bukannya	158	didatangkan	193	dipergunakan
124	bulan	159	digunakan	194	diperkirakan
125	bung	160	diibaratkan	195	diperlihatkan
126	cara	161	diibaratkannya	196	diperlukan
127	caranya	162	diingat	197	diperlukannya
128	cukup	163	diingatkan	198	dipersoalkan
129	cukupkah	164	diinginkan	199	dipertanyakan
130	cukuplah	165	dijawab	200	dipunyai
131	cuma	166	dijelaskan	201	diri
132	dahulu	167	dijelaskannya	202	dirinya
133	dalam	168	dikarenakan	203	disampaikan
134	dan	169	dikatakan	204	disebut
135	dapat	170	dikatakannya	205	disebutkan
136	dari	171	dikerjakan	206	disebutkannya
137	daripada	172	diketahui	207	disini
138	datang	173	diketahuinya	208	disinilah
139	dekat	174	dikira	209	ditambahkan
140	demi	175	dilakukan	210	ditandaskan

211	ditanya	246	hendaknya	281	juga
212	ditanyai	247	hingga	282	jumlah
213	ditanyakan	248	ia	283	jumlahnya
214	ditegaskan	249	ialah	284	justru
215	ditujukan	250	ibarat	285	kala
216	ditunjuk	251	ibaratkan	286	kalau
217	ditunjuki	252	ibaratnya	287	kalaulah
218	ditunjukkan	253	ibu	288	kalaupun
219	ditunjukkannya	254	ikut	289	kalian
220	ditunjuknya	255	ingat	290	kami
221	dituturkan	256	ingin	291	kamilah
222	dituturkannya	257	inginkah	292	kamu
223	diucapkan	258	inginkan	293	kamulah
224	diucapkannya	259	ini	294	kan
225	diungkapkan	260	inikah	295	kapan
226	dong	261	inilah	296	kapankah
227	dua	262	itu	297	kapanpun
228	dulu	263	itukah	298	karena
229	empat	264	itulah	299	karenanya
230	enggak	265	jadi	300	kasus
231	enggaknya	266	jadilah	301	kata
232	entah	267	jadinya	302	katakan
233	entahlah	268	jangan	303	katakanlah
234	guna	269	jangankan	304	katanya
235	gunakan	270	janganlah	305	ke
236	hal	271	jauh	306	keadaan
237	hampir	272	jawab	307	kebetulan
238	hanya	273	jawaban	308	kecil
239	hanyalah	274	jawabnya	309	kedua
240	hari	275	jelas	310	keduanya
241	harus	276	jelaskan	311	keinginan
242	haruslah	277	jelaslah	312	kelamaan
243	harusnya	278	jelasnya	313	kelihatan
244	hendak	279	jika	314	kelihatannya
245	hendaklah	280	jikalau	315	kelima

316	keluar	351	luar	386	memperbuat
317	kembali	352	macam	387	mempergunakan
318	kemudian	353	maka	388	memperkirakan
319	kemungkinan	354	makanya	389	memperlihatkan
320	kemungkinannya	355	makin	390	mempersiapkan
321	kenapa	356	malah	391	mempersoalkan
322	kepada	357	malahan	392	mempertanyakan
323	kepadanya	358	mampu	393	mempunyai
324	kesampaian	359	mampukah	394	memulai
325	keseluruhan	360	mana	395	memungkinkan
326	keseluruhannya	361	manakala	396	menaiki
327	keterlaluan	362	manalagi	397	menambahkan
328	ketika	363	masa	398	menandaskan
329	khususnya	364	masalah	399	menanti
330	kini	365	masalahnya	400	menantikan
331	kinilah	366	masih	401	menanya
332	kira	367	masihkah	402	menanyai
333	kiranya	368	masing	403	menanyakan
334	kita	369	mau	404	mendapat
335	kitalah	370	maupun	405	mendapatkan
336	kok	371	melainkan	406	mendatang
337	kurang	372	melakukan	407	mendatangi
338	lagi	373	melalui	408	mendatangkan
339	lagian	374	melihat	409	menegaskan
340	lah	375	melihatnya	410	mengakhiri
341	lain	376	memang	411	mengapa
342	lainnya	377	memastikan	412	mengatakan
343	lalu	378	memberi	413	mengatakannya
344	lama	379	memberikan	414	mengenai
345	lamanya	380	membuat	415	mengerjakan
346	lanjut	381	memerlukan	416	mengetahui
347	lanjutnya	382	memihak	417	menggunakan
348	lebih	383	meminta	418	menghendaki
349	lewat	384	memintakan	419	mengibaratkan
350	lima	385	memisalkan	420	mengibaratkannya

421	mengingat	456	misalnya	491	pertama
422	mengingatkan	457	mula	492	pertanyaan
423	menginginkan	458	mulai	493	pertanyakan
424	mengira	459	mulailah	494	pihak
425	mengucapkan	460	mulanya	495	pihaknya
426	mengucapkannya	461	mungkin	496	pukul
427	mengungkapkan	462	mungkinkah	497	pula
428	menjadi	463	nah	498	pun
429	menjawab	464	naik	499	punya
430	menjelaskan	465	namun	500	rasa
431	menuju	466	nanti	501	rasanya
432	menunjuk	467	nantinya	502	rata
433	menunjuki	468	nyaris	503	rupanya
434	menunjukkan	469	nyatanya	504	saat
435	menunjuknya	470	oleh	505	saatnya
436	menurut	471	olehnya	506	saja
437	menuturkan	472	pada	507	sajalah
438	menyampaikan	473	padahal	508	saling
439	menyangkut	474	padanya	509	sama
440	menyatakan	475	pak	510	sambil
441	menyebutkan	476	paling	511	sampai
442	menyeluruh	477	panjang	512	sampaikan
443	menyiapkan	478	pantas	513	sana
444	merasa	479	para	514	sangat
445	mereka	480	pasti	515	sangatlah
446	merekalah	481	pastilah	516	satu
447	merupakan	482	penting	517	saya
448	meski	483	pentingnya	518	sayalah
449	meskipun	484	per	519	se
450	meyakini	485	percuma	520	sebab
451	meyakinkan	486	perlu	521	sebabnya
452	minta	487	perlukah	522	sebagai
453	mirip	488	perlunya	523	sebagaimana
454	misal	489	pernah	524	sebagainya
455	misalkan	490	persoalan	525	sebagian

526	sebaik	561	sekadar	596	semua
527	sebaiknya	562	sekadarnya	597	semuanya
528	sebaliknya	563	sekali	598	semula
529	sebanyak	564	sekalian	599	sendiri
530	sebegini	565	sekaligus	600	sendirian
531	sebegitu	566	sekalipun	601	sendirinya
532	sebelum	567	sekarang	602	seolah
533	sebelumnya	568	sekecil	603	seolah-olah
534	sebenarnya	569	seketika	604	seorang
535	seberapa	570	sekiranya	605	sepanjang
536	sebesar	571	sekitar	606	sepantasnya
537	sebetulnya	572	sekitarnya	607	sepantasnyalah
538	sebisanya	573	sekurangnya	608	seperlunya
539	sebuah	574	sela	609	seperti
540	sebut	575	selain	610	sepertinya
541	sebutlah	576	selaku	611	sepihak
542	sebutnya	577	selalu	612	sering
543	secara	578	selama	613	seringnya
544	secukupnya	579	selamanya	614	serta
545	sedang	580	selanjutnya	615	serupa
546	sedangkan	581	seluruh	616	sesaat
547	sedemikian	582	seluruhnya	617	sesama
548	sedikit	583	semacam	618	sesampai
549	sedikitnya	584	semakin	619	sesegera
550	seenaknya	585	semampu	620	sesekali
551	segala	586	semampunya	621	seseorang
552	segalanya	587	semasa	622	sesuatu
553	segera	588	semasih	623	sesuatunya
554	seharusnya	589	semata	624	sesudah
555	sehingga	590	semata-mata	625	sesudahnya
556	seingat	591	semaunya	626	setelah
557	sejak	592	sementara	627	setempat
558	sejauh	593	semisal	628	setengah
559	sejenak	594	semisalnya	629	seterusnya
560	sejumlah	595	sempat	630	setiap

631	setiba	667	tegasnya	703	tiap
632	setibanya	668	telah	704	tiba
633	setidaknya	669	tempat	705	tidak
634	setinggi	670	tengah	706	tidakkah
635	seusai	671	tentang	707	tidaklah
636	sewaktu	672	tentu	708	tiga
637	siap	673	tentulah	709	tinggi
638	siapa	674	tentunya	710	toh
639	siapakah	675	tepat	711	tunjuk
640	siapapun	676	terakhir	712	turut
641	sini	677	terasa	713	tutur
642	sinilah	678	terbanyak	714	tuturnya
643	soal	679	terdahulu	715	ucap
644	soalnya	680	terdapat	716	ucapnya
645	suatu	681	terdiri	717	ujar
646	sudah	682	terhadap	718	ujarnya
647	sudahkah	683	terhadapnya	719	umum
648	sudahlah	684	teringat	720	umumnya
649	supaya	685	terjadi	721	ungkap
650	tadi	686	terjadilah	722	ungkapnya
651	tadinya	687	terjadinya	723	untuk
652	tahu	688	terkira	724	usah
653	tahun	689	terlalu	725	usai
654	tak	690	terlebih	726	waduh
655	tambah	691	terlihat	727	wah
656	tambahnya	692	termasuk	728	wahai
657	tampak	693	ternyata	729	waktu
658	tampaknya	694	tersampaikan	730	waktunya
659	tandas	695	tersebut	731	walau
660	tandasnya	696	tersebutlah	732	walaupun
661	tanpa	697	tertentu	733	wong
662	tanya	698	tertuju	734	yaitu
663	tanyakan	699	terus	735	yakin
664	tanyanya	700	terutama	736	yakni
665	tapi	701	tetap	737	yang
666	tegas	702	tetapi		

LAMPIRAN 2

Aturan derivasional untuk prefiks

Prefiks (alomorf)	Variasi (morf)	Aturan
meng	meng	+ vokal | k | g | h ... , contoh : ambil ? mengambil ikat ? mengikat hilang ? menghilang
	meny	+ s... , contoh : sapu ? menyapu sisir ? menyisir
	mem	+ b | f | p... , contoh : beku ? membeku fitnah ? memfitnah pukul ? memukul
	men	+ c | d | j | t… , contoh : cuci ? mencuci darat ? mendarat jual ? menjual tukar ? menukar
	me	+ l | m | n | r | y |w… , contoh : lintas ? melintas makan ? memakan nikah ? menikah rusak ? merusak wabah ? mewabah yakin ? meyakin(kan)
peng	peng	+ vokal | k | g | h... , contoh : ikat ? pengikat urus ? pengurus ganggu ? penggangu halus ? penghalus
	peny	+ s... , contoh : saring ? penyaring
	pem	+ b | f | p… , contoh : baca ? pembaca fitnah ? pemfitnah pukul ? pemukul

	pen	+ c | d | j | t… , contoh : cuci ? pencuci datang ? pendatang jual ? penjual tukar ? penukar
	pe	+ l | m | n | r | y |w… , contoh : lintas ? pelintas makan ? pemakan rusak ? perusak warna ? pewarna
ber	bel	+ ajar , contoh : ajar ? belajar
	be	+ r | KVr... , contoh : rencana ? berencana kerja ? bekerja
	ber	+ seluruh huruf selain morf bel dan ber , contoh : tamu ? bertamu
per	pel	+ ajar , contoh : ajar ? pelajar
	pe	+ r | KVr... , contoh : ramal ? peramal
	per	+ seluruh huruf selain morf pel dan per , contoh : kaya ? perkaya
ter	te	+ r... , contoh : rasa ? terasa
	ter	+ K | V… , dimana K?r , contoh : atur ? teratur