BAB 3 PERANCANGAN

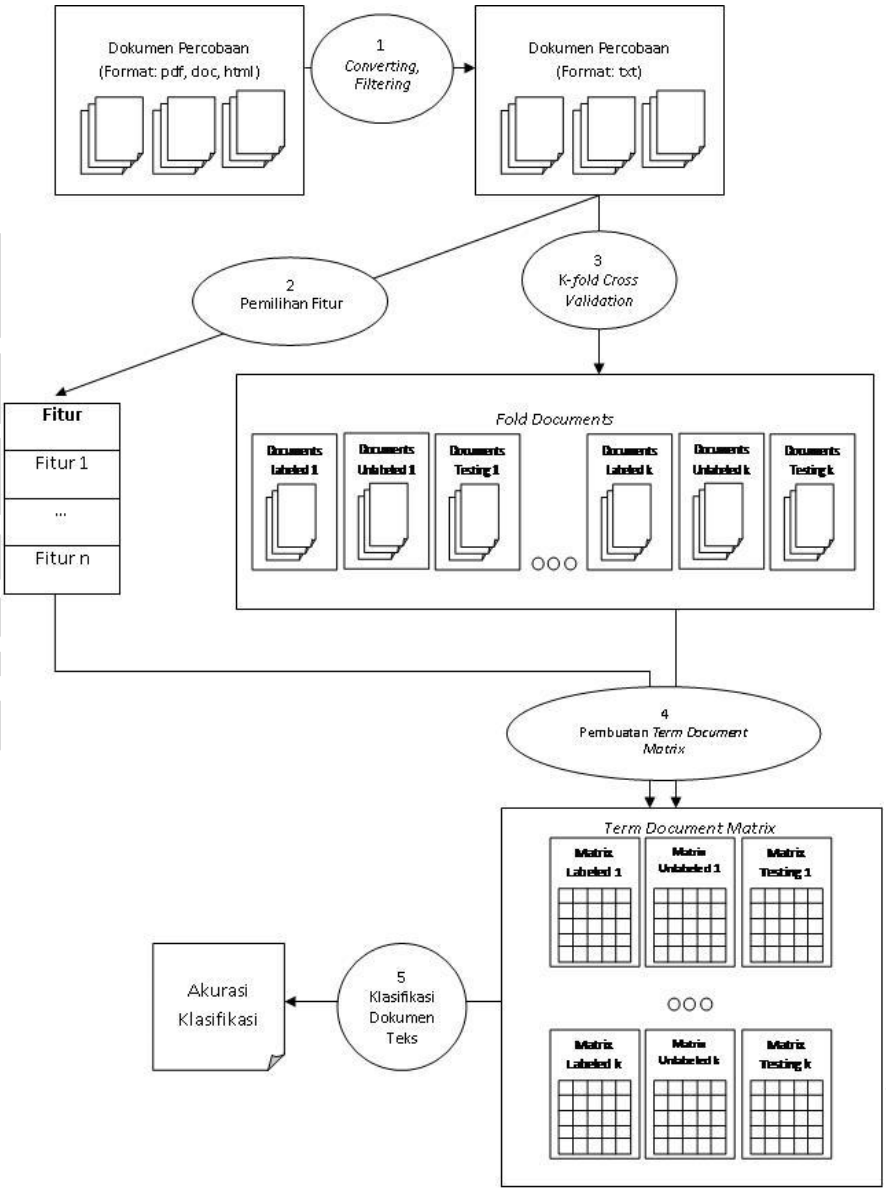
Pada bab ini dijelaskan mengenai perancangan untuk melakukan eksperimen klasifikasi dokumen teks. Klasifikasi dilakukan dengan menentukan kategori dari semua dokumen *testing* yang ada. Perancangan klasifikasi dokumen teks ini meliputi persiapan dokumen (lihat subbab 3.3), pembuatan *term documents matrix* (lihat subbab 3.6) dan klasifikasi dokumen teks menggunakan *machine learning* (lihat subbab 3.7) yaitu dengan algoritma Naïve Bayes dan Expectation Maximization.

3.1 Gambaran Umum Proses Klasifikasi Dokumen Teks

Pada tugas akhir ini klasifikasi dokumen teks dilakukan dengan dua metode, yaitu Naïve Bayes dan Expectation Maximization. Percobaan ini merupakan penelitian lebih lanjut dari percobaan sebelumnya (lihat subbab 2.5) yang melakukan klasifikasi topik menggunakan Naïve Bayes dan Maximum Entropy. Tujuan utama dari penelitian ini adalah untuk melihat manfaat *unlabeled documents* dalam membantu klasifikasi dokumen teks. Dalam hal ini Expectation Maximization tidak dapat dibandingkan secara langsung dengan Naïve Bayes menggunakan parameter yang sama, sehingga untuk menunjukkan hal tersebut, pada penelitian ini dilakukan percobaan tanpa memanfaatkan *unlabeled documents* menggunakan metode Naïve Bayes dan percobaan yang memanfaatkan *unlabeled documents* dengan menggunakan metode Expectation Maximization.

Pendekatan yang digunakan dalam tugas akhir kali ini merupakan pendekatan *supervised learning* yaitu pendekatan yang memerlukan tahap pembelajaran sebelum melakukan *testing*. Data yang digunakan adalah dokumen hukum dari hukumonline.com, 20Newsgroups *dataset,* dan artikel media massa dari kompas.com (lihat subbab 3.2). Beberapa dokumen tersebut pada awalnya memiliki format doc, pdf, maupun html, sehingga perlu dilakukan konversi dari format asal dokumen-dokumen tersebut menjadi format dokumen teks berekstensi txt agar dapat dilakukan proses klasifikasi selanjutnya (lihat subbab 3.3.1). Setelah dokumen-dokumen

tersebut dikonversi, proses selanjutnya adalah menghilangkan *stopwords* dan tanda baca yang dapat mengganggu pemilihan kata atau fitur dari setiap dokumen (lihat subbab 3.3.2). *Stopwords* adalah daftar kata-kata yang tidak dipakai didalam pemrosesan bahasa alami. Sebelum dilakukan pemilihan fitur dan pembuatan *term documents matrix*, dokumen-dokumen tersebut terlebih dahulu dirandomisasi agar tidak terjadi pemusatan dokumen-dokumen yang membahas kategori yang sama.



Gambar 3.1 Perancangan Percobaan Klasifikas Dokumen Teks

Jenis fitur yang digunakan dalam psercobaan ini adalah *presence*, *frequency*, *frequency normalized*, dan pembobotan fitur dengan *tf-idf* (lihat subbab 3.5). Untuk melakukan verifikasi percobaan dilakukan k-*fold cross validation* (lihat subbab 3.4). K-*fold cross validation* dilakukan dengan membagi kumpulan dokumen yang dimiliki menjadi *k* bagian dengan satu bagian untuk dokumen *testing*, *(k-1)/2* bagian untuk *labeled documents* dan *(k-1)/2* bagian lainnya digunakan untuk *unlabeled documents*. Masing-masing bagian akan digunakan secara bergantian.

Nilai tiap fitur dalam sebuah dokumen disimpan di dalam *term documents matrix* (lihat subbab 3.6). *Term documents matrix* ini dibuat untuk setiap *fold* dokumen, baik *fold testing, fold labeled documents*, maupun *fold unlabeled documents*. *Term documents matrix* ini nantinya akan menjadi masukan bagi mesin klasifikasi yang menggunakan *machine learning* baik dengan algoritma Naïve Bayes maupun algoritma Expectation Maximization (lihat subbab 3.7). Nilai yang dihasilkan dari mesin klasifikasi ini adalah hasil akurasi rata-rata untuk setiap percobaan klasifikasi dari tiap *k* bagian yang diujikan. Gambaran keseluruhan proses klasifikasi dokumen teks dapat dilihat pada gambar 3.1.

3.2 Data

Percobaan pada tugas akhir ini menggunakan tiga jenis data yang berbeda. Data pertama adalah dokumen hukum dari hukumonline.com. Dokumen-dokumen tersebut terbagi atas empat kategori, yaitu Perpu, PP, UU, dan UU Darurat. Data kedua adalah 20Newsgroups *dataset* yang merupakan kumpulan *e-mail* yang berjumlah 18828 dokumen. 20Newsgroups *dataset* memiliki 20 kategori dan dapat diunduh pada http://people.csail.mit.edu/jrennie/20Newsgroups/. Dokumen *e-mail* yang terdapat pada 20Newsgroups *dataset* yang digunakan pada percobaan ini merupakan dokumen-dokumen yang telah dihilangkan *tag header*-nya. Data terakhir adalah kumpulan artikel media massa dari kompas.com yang merupakan data yang telah digunakan pada percobaan klasifikasi topik (Dyta, 2009). Dokumen artikel media massa ini memiliki lima kategori, yaitu: ekonomi, olahraga, properti, travel, dan

kesehatan. Contoh dokumen yang digunakan pada tugas akhir ini dapat dilihat pada lampiran 2.

Tabel 3.1 Daftar Kategori dan Jumlah Dokumen yang Digunakan

Kategori	Jumlah Dokumen
Dokumen Hukum	
Perpu	122
PP	149
UU	148
UU Darurat	128
20Newsgroups *Dataset*	
alt.atheism	799
comp.graphics	973
comp.os.ms-windows.misc	985
comp.sys.ibm.pc.hardware	982
comp.sys.mac.hardware	961
comp.windows.x	980
misc.forsale	972
rec.autos	990
rec.motorcycles	994
rec.sport.baseball	994
rec.sport.hockey	999
sci.crypt	991
sci.electronics	981
sci.med	990
sci.space	987
soc.religion.christian	997
talk.politics.guns	910
talk.politics.mideast	940
talk.politics.misc	775
talk.religion.misc	628
Artikel Media Massa	
Ekonomi	365
Kesehatan	314
Olahraga	287
Properti	141
Travel	133

3.3 Persiapan Dokumen

Proses persiapan dokumen meliputi *converting* dan *filtering*. *Converting* adalah proses mengubah dokumen yang tadinya memiliki format doc, pdf, maupun html menjadi dokumen teks berekstensi txt, sedangkan *filtering* adalah proses menghilangkan *stopwords* dan tanda baca. Kedua proses ini perlu dilakukan sebelum proses pemilihan fitur.

3.3.1 *Converting*

Proses *converting* dilakukan untuk semua dokumen percobaan yang memiliki format selain txt. Hal ini dilakukan karena sebagian dokumen yang digunakan untuk melakukan percobaan memiliki format doc, pdf, maupun html. Proses *converting* perlu dilakukan karena dokumen-dokumen yang memiliki format selain txt memiliki format khusus seperti pada dokumen berekstensi pdf dan doc, sedangkan pada dokumen html memiliki *tag-tag* khusus yang harus dihilangkan agar tidak mengganggu proses klasifikasi.

3.3.2 *Filtering*

Proses persiapan dokumen selanjutnya adalah *filtering*, yaitu proses menghilangkan *stopwords* dan tanda baca. *Stopwords* adalah daftar kata-kata yang tidak dipakai didalam pemrosesan bahasa alami (kata depan, kata penghubung, kata pengganti, dll.). Keseluruhan daftar *stopwords* untuk bahasa Indonesia dan bahasa Inggris yang digunakan dapat dilihat pada lampiran 3. Daftar *stopwords* bahasa Indonesia yang digunakan dalam tugas akhir ini didapat dari http://fpmipa.upi.edu/ staff/yudi/stop_words_list.txt, sedangkan daftar *stopwords* untuk bahasa Inggris didapat dari http://members.unine.ch/jacques.savoy/clef/englishST.txt.

Selain menghilangkan *stopwords* dan tanda baca, proses filtering juga menghilangkan karakter-karakter ASCII 0 hingga 31 yang belum hilang setelah proses *converting* dokumen. Karakter-karakter ASCII 0 hingga 31 dapat mengganggu proses pembacaan dokumen pada saat klasifikasi dokumen. Karakter-karakter tersebut akan

menghentikan proses klasifikasi dokumen, karena karakter-karakter tersebut dianggap sebagai akhir dari dokumen pada saat proses pembacaan dokumen.

3.4 K-*fold Cross Validation*

Pada percobaan untuk tugas akhir ini digunakan k-*fold cross validation* untuk menghilangkan bias data. K-*fold cross validation* membagi kumpulan dokumen menjadi *k* bagian. Dalam satu set percobaan akan dilakukan *k* buah percobaan klasifikasi dokumen dengan tiap percobaan menggunakan satu bagian sebagai data *testing*, *(k-1)/2* bagian sebagai *labeled documents*, dan *(k-1)/2* bagian lainnya sebagai *unlabeled documents* yang akan ditukar setiap percobaan sebanyak *k* kali. Kumpulan dokumen yang dimiliki terlebih dahulu diacak urutannya sebelum dimasukkan ke dalam sebuah *fold*. Hal ini dilakukan untuk menghindari pengelompokkan dokumen-dokumen yang berasal dari satu kategori tertentu pada sebuah *fold*.

Data *testing* merupakan kumpulan dokumen yang akan digunakan untuk melakukan pengujian klasifikasi dokumen teks. Sedangkan *labeled documents* dan *unlabeled documents* merupakan data *training* dalam melakukan pembelajaran untuk melakukan klasifikasi dokumen teks. Naïve Bayes hanya akan menggunakan *labeled documents*, sedangkan Expectation Maximization akan menggunakan *labeled documents* dan *unlabeled documents* pada tahap pembelajarannya.

3.5 Pemilihan Fitur

Penelitian ini menggunakan empat buah jenis fitur yaitu *presence*, *frequency*, *frequency normalized*, dan pembobotan *tf-idf*. Jenis fitur *presence* hanya akan memperhatikan apakah sebuah fitur muncul atau tidak, apabila fitur tersebut muncul pada sebuah dokumen maka akan memiliki nilai satu, dan jika yang terjadi adalah sebaliknya, fitur tersebut akan memiliki nilai nol. Jenis fitur *frequency* akan memperhitungkan berapa banyak kemunculan sebuah fitur dalam sebuah dokumen. *Frequency normalized* menyimpan informasi mengenai nilai jumlah kemunculan fitur dalam suatu dokumen dibagi dengan jumlah seluruh fitur yang ada pada dokumen

tersebut, sedangkan pada pembobotan *tf-idf* nilai fitur akan dihitung berdasarkan kemunculan fitur pada sebuah dokumen dibagi dengan jumlah dokumen yang memiliki fitur tersebut. Berikut ini contoh pemberian nilai untuk masing-masing jenis fitur:

a. Nilai sebuah fitur berdasarkan jenis fitur *frequency*.

Dokumen	Fitur (Kemunculan)
dokumen1	pajak (3), cukai (9), uang (2), sistem (1)
dokumen2	java (4), linux (2), sistem (6)
dokumen3	catur (2), menang (1), kalah (1), uang(1)

b. Nilai sebuah fitur berdasarkan jenis fitur *presence*.

Dokumen	Fitur (Kemunculan)
dokumen1	pajak (1), cukai (1), uang (1), sistem (1)
dokumen2	java (1), linux (1), sistem (1)
dokumen3	catur (1), menang (1), kalah (1), uang(1)

c. Nilai sebuah fitur berdasarkan jenis fitur *frequency normalized*.

Dokumen	Fitur (Kemunculan)
dokumen1	pajak ($^{3}/_{15}$), cukai ($^{9}/_{15}$), uang ($^{2}/_{15}$), sistem ($^{1}/_{15}$)
dokumen2	java ($^{4}/_{12}$), linux ($^{2}/_{12}$), sistem ($^{6}/_{12}$)
dokumen3	catur ($^{2}/_{5}$), menang ($^{1}/_{5}$), kalah ($^{1}/_{5}$), uang($^{1}/_{5}$)

d. Nilai sebuah fitur berdasarkan pembobotan *tf-idf*.

Dokumen	Fitur (Kemunculan)
dokumen1	pajak (3), cukai (9), uang ($^{2}/_{2}$), sistem ($^{1}/_{2}$)
dokumen2	java (4), linux (2), sistem ($^{6}/_{2}$)
dokumen3	catur (2), menang (1), kalah (1), uang($^{1}/_{2}$)

Percobaan dilakukan menggunakan *top-n* fitur yaitu *n* buah fitur yang memiliki jumlah frekuensi terbanyak yang dianggap telah cukup baik untuk melakukan satu

rangkaian proses klasifikasi. Oleh karena itu akan dilakukan percobaan klasifikasi dokumen teks pendahuluan menggunakan algoritma Naïve Bayes untuk mencari jumlah fitur yang dirasa telah cukup baik untuk melakukan klasifikasi. Hal ini perlu dilakukan mengingat masalah keterbatasan *memory* dalam melakukan proses klasifikasi dokumen teks menggunakan algoritma Expectation Maximization. Setelah jumlah fitur yang diharapkan telah didapat, maka untuk percobaan selanjutnya akan menggunakan jumlah fitur tersebut untuk menghitung akurasi klasifikasi dokumen teks dengan variabel-variabel lainnya.

3.6 *Term Documents Matrix*

Term documents matrix marupakan representasi kumpulan dokumen yang akan digunakan untuk melakukan proses klasifikasi dokumen teks. Pada *term documents matrix*, sebuah dokumen direpresentasikan sebagai kumpulan fitur dan dapat diilustrasikan sebagai $d_j = [w_{1j}, w_{2j}, ..., w_{kj}]$ dengan d_j merupakan dokumen ke-*j* dan w_{kj} merupakan nilai kemunculan fitur ke-*k* pada dokumen d_j. Matriks ini akan berisi nilai-nilai kemunculan fitur. Nilai kemunculan fitur yang digunakan ada empat buah sesuai jenis fitur yang digunakan, yaitu *presence, frequency, frequency normalized* dan pembobotan *tf-idf*. Baris pada *term documents matrix* merupakan data dokumen, sedangkan kolom dari *term documents matrix* merupakan fitur yang digunakan. Berikut gambaran *term documents matrix*:

	w_1	w_2			w_k
d_1	w_{11}	w_{12}	.	.	w_{k1}
d_2	w_{21}	w_{22}	.	.	w_{k2}
	.	.			.
	.	.			.
	.	.			.
d_j	w_{1j}	w_{2j}	.	.	w_{kj}

Gambar 3.2 *Term Documents Matrix*

Dengan menggunakan contoh dokumen dan fitur yang terdapat pada subbab 3.5, dibawah ini contoh pembuatan *term documents matrix* untuk masing-masing jenis fitur yang digunakan.

a. *Term documents matrix* dengan jenis fitur *frequency*.

	catur	cukai	java	kalah	linux	menang	pajak	sistem	uang
dokumen 1	0	9	0	0	0	0	3	1	2
dokumen 2	0	0	4	0	2	0	0	6	0
dokumen 3	2	0	0	1	0	1	0	0	1

b. *Term documents matrix* dengan jenis fitur *presence*.

	catur	cukai	java	kalah	linux	menang	pajak	sistem	uang
dokumen 1	0	1	0	0	0	0	1	1	1
dokumen 2	0	0	1	0	1	0	0	1	0
dokumen 3	1	0	0	1	0	1	0	0	1

c. *Term documents matrix* dengan jenis fitur *frequency normalized*.

	catur	cukai	java	kalah	linux	menang	pajak	sistem	uang
dokumen 1	0	$^{9}/_{15}$	0	0	0	0	$^{3}/_{15}$	$^{1}/_{15}$	$^{2}/_{15}$
dokumen 2	0	0	$^{4}/_{12}$	0	$^{2}/_{12}$	0	0	$^{6}/_{12}$	0
dokumen 3	2	0	0	1	0	1	0	0	1

d. *Term documents matrix* dengan pembobotan *tf-idf*.

	catur	cukai	java	kalah	linux	menang	pajak	sistem	uang
dokumen 1	0	9	0	0	0	0	3	$^{1}/_{2}$	$^{2}/_{2}$
dokumen 2	0	0	4	0	2	0	0	$^{6}/_{2}$	0
dokumen 3	2	0	0	1	0	1	0	0	$^{1}/_{2}$

Term documents matrix ini akan dibuat untuk setiap *fold* dari data *testing*, *labeled documents* dan *unlabeled documents*. Matriks ini dibentuk dengan menghitung nilai dari masing-masing fitur pada dokumen yang akan diklasifikasi, bergantung pada informasi jenis fitur yang digunakan. Matriks ini nantinya akan digunakan sebagai masukan untuk menjalankan metode klasifikasi Naïve Bayes dan Expectation Maximization.

3.7 Metode Klasifikasi Dokumen Teks

Metode *machine learning* yang akan digunakan pada percobaan tugas akhir ini adalah Naïve Bayes dan Expectation Maximization. Dua metode tersebut merupakan metode *supervised learning* untuk klasifikasi dokumen. *Term documents matrix* yang telah dihasilkan sebelumnya akan menjadi *input* untuk kedua metode ini sehingga menghasilkan model probabilistik yang akan digunakan untuk melakukan klasifikasi dokumen teks.

3.7.1 Naïve Bayes

Naïve Bayes merupakan metode *fully supervised learning* yang memerlukan tahap pembelajaran untuk membangun model probabilistik. Model probabilistik tersebut nantinya akan digunakan untuk melakukan perhitungan *prior* dan *conditional probability* dokumen *testing* dalam menentukan kategori dari dokumen *testing* tersebut. Naïve Bayes membangun model probabilistik dari *term documents matrix* data *labeled*.

Klasifikasi dokumen dilakukan dengan terlebih dahulu menentukan kategori $c \varepsilon C=\{c_1, c_2, c_3, \ldots, c_n\}$ dari suatu dokumen $d \varepsilon D=\{d_1, d_2, d_3, \ldots, d_j\}$ berdasarkan kata-kata yang terkandung dalam dokumen. Kumpulan dokmen *training* dan *testing* yang digunakan direpresentasikan dalam *term documents matrix* seperti telah dijelaskan pada subbab 3.6. Proses penentuan kategori dari sebuah dokumen *testing* dilakukan dengan melakukan perhitungan menggunakan persamaan (2.6) sebagai berikut:

$$c_* = \arg\max_{c_i \in C} p(c_i \mid d_j)$$

$$= \arg\max_{c_i \in C} \prod_k p(w_{kj} \mid c_i) \times p(c_i)$$

dimana w_{kj} merupakan fitur atau kata dari dokumen d_j yang ingin diketahui kategorinya. Nilai $p(w_{kj} \mid c_i)$ dipelajari dari data *training* yang dimiliki dengan menggunakan informasi jenis fitur yang berbeda-beda seperti dijelaskan pada subbab 3.6. Berikut ini adalah contoh penerapan algoritma Naïve Bayes:

Pada contoh ini, akan ditunjukkan bagaimana proses penentuan kategori untuk dokumen3.

Dokumen	Kategori	Fitur (Kemunculan)
dokumen1	olahraga	menang (2), bola (3), gol (2)
dokumen2	politik	partai (3), pemilu (2), capres (4)
dokumen3	?	partai (2), menang (1), tandang (2)

dari kumpulan dokumen diatas akan terbentuk *term documents matrix* sebagai berikut:

	bola	capres	gol	menang	partai	pemilu	tandang
dokumen 1	3	0	2	2	0	0	0
dokumen 2	0	4	0	0	3	2	0
dokumen 3	0	0	0	1	2	0	2

Langkah selanjutnya adalah pembuatan model probabilistik dengan melakukan perhitungan:

$$p(w_{kj} \mid c_i) = \frac{f(w_{kj}, c_i) + 1}{f(c_i) + |W|}$$

$f(w_{kj}, c_i)$ adalah nilai kemunculan kata w_{kj} pada kategori c_i

$f(c_i)$ adalah jumlah keseluruhan kata pada kategori c_i

$|W|$ adalah jumlah keseluruhan kata/fitur yang digunakan

dan

$$p(c_i) = \frac{f_d(c_i)}{|D|}$$

$f_d(c_i)$ adalah jumlah dokumen yang memiliki kategori c_i
$|D|$ adalah jumlah seluruh *training* dokumen

Model probabilistik yang terbentuk adalah sebagai berikut:

Kategori	$p(c_i)$	$p(w_{kj}\|c_i)$						
		bola	capres	gol	menang	partai	pemilu	tandang
olahraga	$^1/_2$	$^4/_{14}$	$^1/_{14}$	$^3/_{14}$	$^3/_{14}$	$^1/_{14}$	$^1/_{14}$	$^1/_{14}$
politik	$^1/_2$	$^1/_{16}$	$^5/_{16}$	$^1/_{16}$	$^1/_{16}$	$^4/_{16}$	$^3/_{16}$	$^1/_{16}$

Setelah pembuatan model probabilistik selesai dilakukan, langkah terakhir yang dilakukan adalah penentuan kategori untuk dokumen3:

$$c^* = \arg\max_{c_i \in C} \prod_k p(w_{kj} \mid c_i) \times p(c_i)$$

$p(\text{“olahraga”} \mid \text{“dokumen3”}) = p(\text{“olahraga”})$ x $p(\text{“partai”} \mid \text{“olahraga”})$ x $p(\text{“menang”} \mid \text{“olahraga”})$ x $p(\text{“tandang”} \mid \text{“olahraga”})$

$= {}^1/_2 \text{ x } {}^1/_{14} \text{ x } {}^3/_{14} \text{ x } {}^1/_{14}$

$= {}^3/_{5488} \approx 0{,}0000594$

$p(\text{“politik”} \mid \text{“dokumen3”}) = p(\text{“politik”})$ x $p(\text{“partai”} \mid \text{“politik”})$ x $p(\text{“menang”} \mid \text{“politik”})$ x $p(\text{“tandang”} \mid \text{“politik”})$

$= {}^1/_2 \text{ x } {}^4/_{16} \text{ x } {}^1/_{16} \text{ x } {}^1/_{16}$

$= {}^1/_{2048} \approx 0{,}0004882$

karena *p("politik"|"dokumen3") > p("olahraga"|"dokumen3")*, maka kategori dari dokumen3 adalah **politik**.

3.7.2 Expectation Maximization

Proses klasifikasi dokumen menggunakan algoritma Eepectation Maximization tidak jauh berbeda dengan Naïve bayes. Perbedaan hanya terletak pada tahap *training*. Expectation Maximization memanfaatkan *labeled documets* dan *unlabeled documents* dalam tahap *training* untuk membangun model probabilistiknya, sehingga metode ini sering disebut algoritma *semi supervised learning*. Langkah awal yang dilakukan pada algoritma Expectation Maximization adalah membangun model probabilistik dari semua *labeled documents* yang ada seperti yang dilakukan pada algoritma Naïve Bayes. Proses tersebut dilakukan dengan mengambil informasi dari *term documents matrix* dari *labeled documents*. Setelah model probabilistik awal terbentuk, dilakukanlah *expectation step* yaitu tahap memperkirakan kategori setiap *unlabeled documents* yang terdapat pada *term documents matrix* dari *unlabeled documents* dengan menggunakan persamaan (2.10) sebagai berikut:

$$p(c_i \mid d_j) = \frac{p(c_i)\prod_{k=1}^{|dj|} p(w_{kj} \mid c_i)}{\sum_{r=1}^{|C|} p(c_r)\prod_{k=1}^{|dj|} p(w_{kj} \mid c_r)}$$

dengan $p(c_i \mid d_j)$ adalah probabililas kemunculan kategori c_i jika diketahui dokumen d_j. Setelah semua *unlabeled documents* memiliki kategori perkiraan, tahap selanjutnya adalah *maximization step*, yaitu tahapan untuk melakukan *update* terhadap parameter klasifikasi yaitu probabilitas $p(w_{kj} \mid c_i)$ dan probabilitas $p(c_i)$ dengan perhitungan sesuai persamaan (2.11) dan (2.12) sebagai berikut:

$$p(w_{kj} \mid c_i) = \frac{1 + \sum_{j=1}^{|D|} N(w_{kj}, d_j)\, p(c_i \mid d_j)}{|W| + \sum_{s=1}^{|W|} \sum_{j=1}^{|D|} N(w_s, d_j)\, p(c_i \mid d_j)}$$

dan

$$p(c_i) = \frac{1 + \sum_{j=1}^{|D|} p(c_i \mid d_j)}{|C| + |D|}$$

dengan $N(w_{kj}, d_j)$ adalah jumlah kata w_k pada dokumen d_j dan $|W|$ merupakan jumlah keseluruhan kata/fitur yang digunakan. Dua tahap tersebut akan terus dilakukan hingga perubahan parameter probabilitas $p(w_{kj} \mid c_i)$ dan $p(c_i)$ tidak melebihi batasan yang ditentukan dari iterasi sebelumnya. Setelah model probabilistik terbentuk dari *labeled* dan *unlabeled documents*, tahap *testing* dapat dilakukan dengan melakukan perhitungan $c^* = \arg\max_{c_i \in C} \prod_k p(w_{kj} \mid c_i) \times p(c_i)$ untuk setiap dokumen pada data *testing* yang dimiliki. Berikut ini adalah contoh penerapan algoritma Expectation Maximization:

Contoh berikut ini menggunakan kumpulan dokumen yang sama seperti contoh penentuan kategori sebuah dokumen menggunakan algoritma Naïve Bayes (lihat subbab 3.7.1) ditambah dengan satu buah *unlabeled document* yaitu dokumen4.

Dokumen	Kategori	Fitur (Kemunculan)
dokumen1	olahraga	menang (2), bola (3), gol (2)
dokumen2	politik	partai (3), pemilu (2), capres (4)
dokumen3	?	partai (2), menang (1), tandang (2)
dokumen4	?	menang (1), bola (2), tandang (3)

dari kumpulan dokumen diatas akan terbentuk *term documents matrix* sebagai berikut:

	bola	capres	gol	menang	partai	pemilu	tandang
dokumen 1	3	0	2	2	0	0	0
dokumen 2	0	4	0	0	3	2	0
dokumen 3	0	0	0	1	2	0	2
dokumen 4	2	0	0	1	0	0	3

Model probabilistik awal yang terbentuk (menggunakan Naïve Bayes *classifier*) adalah sebagai berikut:

Kategori	$p(c_i)$	$p(w_{kj}\|c_i)$						
		bola	capres	gol	menang	partai	pemilu	tandang
olahraga	$^1/_2$	$^4/_{14}$	$^1/_{14}$	$^3/_{14}$	$^3/_{14}$	$^1/_{14}$	$^1/_{14}$	$^1/_{14}$
politik	$^1/_2$	$^1/_{16}$	$^5/_{16}$	$^1/_{16}$	$^1/_{16}$	$^4/_{16}$	$^3/_{16}$	$^1/_{16}$

Tahap selanjutnya adalah *expectation step*, menentukan kategori perkiraan untuk dokumen4:

$$p(c_i \mid d_j) = \frac{p(c_i)\prod_{k=1}^{|dj|} p(w_{kj} \mid c_i)}{\sum_{r=1}^{|C|} p(c_r)\prod_{k=1}^{|dj|} p(w_{kj} \mid c_r)}$$

p(“olahraga”|”dokumen4”)= *(p(“olahraga”)* x *p(“menang”|”olahraga”)* x *p(“bola”|”olahraga”)* x *p(“tandang”|”olahraga”)) : ((p(“olahraga”)* x *p(“menang”|”olahraga”)* x *p(“bola”|”olahraga”)* x *p(“tandang”|”olahraga”)) + (p(“politik”)* x *p(“menang”|”politik”)* x *p(“bola”|”politik”)* x *p(“tandang”|”politik”)))*

= ($^1/_2$ x $^3/_{14}$ x $^4/_{14}$ x $^1/_{14}$) : (($^1/_2$ x $^3/_{14}$ x $^4/_{14}$ x $^1/_{14}$)+ ($^1/_2$ x $^1/_{16}$ x $^1/_{16}$ x $^1/_{16}$))

$= ({}^{3}/_{1372}) : (({}^{3}/_{1372})+({}^{1}/_{8192}))$

$\approx 0{,}947$

$p(\text{“politik”}|\text{”dokumen4”}) =$ *(p(“politik”) x p(“menang”|”politik”) x p(“bola”|”politik”) x p(“tandang”|”pilitik”)) : ((p(“olahraga”) x p(“menang”|”olahraga”) x p(“bola”|”olahraga”) x p(“tandang”|”olahraga”)) + (p(“politik”) x p(“menang”|”politik”) x p(“bola”|”politik”) x p(“tandang”|”politik”)))*

$= ({}^{1}/_{2} \times {}^{3}/_{14} \times {}^{4}/_{14} \times {}^{1}/_{14}) : (({}^{1}/_{2} \times {}^{3}/_{14} \times {}^{4}/_{14} \times {}^{1}/_{14}) + ({}^{1}/_{2} \times {}^{1}/_{16} \times {}^{1}/_{16} \times {}^{1}/_{16}))$

$= ({}^{1}/_{8192})) : (({}^{3}/_{1372}) + ({}^{1}/_{8192}))$

$\approx 0{,}052$

karena $p(\text{“olahraga”}|\text{”dokumen4”}) > p(\text{“politik”}|\text{”dokumen4”})$, maka kategori perkiraan dari dokumen4 adalah **olahraga**.

Setelah *expectation step* selesai, dilakukanlah *maximization step* untuk meng-*update* model probabilistik awal dengan melakukan perhitungan:

$$p(w_{kj} \mid c_i) = \frac{1 + \sum_{j=1}^{|D|} N(w_{kj}, d_j)\, p(c_i \mid d_j)}{|W| + \sum_{s=1}^{|W|} \sum_{j=1}^{|D|} N(w_s, d_j)\, p(c_i \mid d_j)}$$

$N(w_{kj}, d_j)$ adalah jumlah kata w_k pada dokumen d_j
$|W|$ merupakan jumlah keseluruhan kata/fitur yang digunakan
$|D|$ adalah jumlah seluruh *training* dokumen

Karena nilai $|W| + \sum_{s=1}^{|W|} \sum_{j=1}^{|D|} N(w_s, d_j)\, p(c_i \mid d_j)$ akan selalu sama untuk setiap perhitungan, maka nilainya dihitung terlebih dahulu, sebagai pengganti akan diberi nama *f(p)*. Karena masih terdapat beberapa probabilitas yang memiliki nilai 0 yaitu *p(“politik”|”dokumen1”)* dan *p(“olahraga”|”dokumen2”)* serta terdapat beberapa kata yang nilai kemunculannya nol pada beberapa dokumen seperti capres, partai, pemilu, dan

tandang pada dokumen1, bola, gol, menang, dan tandang pada dokumen2, capres, gol, partai, dan pemilu pada dokumen4, sehingga *f(p)* dapat dituliskan sebagai berikut:

f(p) = *7 + N("bola", "dokumen1") p("olahraga"|"dokumen1") +*
N("bola", "dokumen4") p("olahraga"|"dokumen4") +
N("gol", "dokumen1") p("olahraga"|"dokumen1") +
N("menang", "dokumen1") p("olahraga"|"dokumen1") +
N("menang", "dokumen4") p("olahraga"|"dokumen4") +
N("tandang", "dokumen4") p("olahraga"|"dokumen4") +
N("bola", "dokumen4") p("politik"|"dokumen4") +
N("capres", "dokumen2") p("politik"|"dokumen2") +
N("menang", "dokumen4") p("politik"|"dokumen4") +
N("partai", "dokumen2") p("politik"|"dokumen2") +
N("pemilu", "dokumen2") p("politik"|"dokumen2") +
N("tandang", "dokumen4") p("politik"|"dokumen4")

= 7+ 3 x 1 + 2 x 0,947 + 2 x 1 + 2 x 1 + 1 x 0,947 + 3 x 0,947 + 2 x 0,052 + 4 x 1 + 1 x 0,052 + 3 x 1 + 2 x 1 + 3 x 0,052

= 7 + 3 + 1,894 + 2 + 2 + 0,947 + 2,841 + 0,104 + 4 + 0,052 + 3 + 2 + 0,156

= **28,994**

p("bola"|"olahraga") = (1 + N("bola", "dokumen1") p("olahraga"|"dokumen1") +
N("bola", "dokumen2") p("olahraga"|"dokumen2") +
N("bola", "dokumen4") p("olahraga"|"dokumen4")) : f(p)

= (1 + 3 x 1 + 0 x 0 + 2 x 0,947) : 28,994

= (1 + 3 + 0 + 1,894) : 28,994

= 5,895 : 28,994

= **0,203**

p("bola"|"politik") = (1 + N("bola", "dokumen1") p("politik"|"dokumen1") +
N("bola", "dokumen2") p("politik"|"dokumen2") +
N("bola", "dokumen4") p("politik"|"dokumen4")) : f(p)

= (1 + 3 x 0 + 0 x 1 + 2 x 0,052) : 28,994

= (1 + 0 + 0 + 0,104) : 28,994

= 1,104 : 28,994

= **0,038**

p("capres"|"olahraga") = (1 + N("capres", "dokumen1") p("olahraga"|"dokumen1") +
N("capres", "dokumen2") p("olahraga"|"dokumen2") +

N(“capres”, “dokumen4”) p(“olahraga”|”dokumen4”)) : f(p)

= (1 + 0 x 1 + 4 x 0 + 0 x 0,947) : 28,994

= (1 + 0 + 0 + 0) : 28,994

= 1 : 28,994

= **0,034**

p(“capres”|”politik”) = (1 + N(“capres”, “dokumen1”) p(“politik”|”dokumen1”) +
N(“capres”, “dokumen2”) p(“politik”|”dokumen2”) +
N(“capres”, “dokumen4”) p(“politik”|”dokumen4”)) : f(p)

= (1 + 0 x 0 + 4 x 1 + 0 x 0,052) : 28,994

= (1 + 0 + 4 + 0) : 28,994

= 5 : 28,994

= **0,1724**

p(“gol”|”olahraga”) = (1 + N(“gol”, “dokumen1”) p(“olahraga”|”dokumen1”) +
N(“gol”, “dokumen2”) p(“olahraga”|”dokumen2”) +
N(“gol”, “dokumen4”) p(“olahraga”|”dokumen4”)) : f(p)

= (1 + 2 x 1 + 0 x 0 + 0 x 0,947) : 28,994

= (1 + 2 + 0 + 0) : 28,994

= 3 : 28,994

= **0,1034**

p(“gol”|”politik”) = (1 + N(“gol”, “dokumen1”) p(“politik”|”dokumen1”) +
N(“gol”, “dokumen2”) p(“politik”|”dokumen2”) +
N(“gol”, “dokumen4”) p(“politik”|”dokumen4”)) : f(p)

= (1 + 2 x 0 + 0 x 1 + 0 x 0,052) : 28,994

= (1 + 0 + 0 + 0) : 28,994

= 1 : 28,994

= **0,034**

p(“menang”|”olahraga”) = (1 + N(“menang”, “dokumen1”) p(“olahraga”|”dokumen1”) +
N(“menang”, “dokumen2”) p(“olahraga”|”dokumen2”) +
N(“menang”, “dokumen4”) p(“olahraga”|”dokumen4”)) : f(p)

= (1 + 2 x 1 + 0 x 0 + 1 x 0,947) : 28,994

= (1 + 2 + 0 + 0,947) : 28,994

= 3,947 : 28,994

= **0,1361**

p(“menang”|”politik”) = (1 + N(“menang”, “dokumen1”) p(“politik”|”dokumen1”) +
N(“menang”, “dokumen2”) p(“politik”|”dokumen2”) +

N("menang", "dokumen4") p("politik"|"dokumen4")) : f(p)

= (1 + 2 x 0 + 0 x 1 + 1 x 0,052) : 28,994

= (1 + 0 + 0 + 0,052) : 28,994

= 1,052 : 28,994

= **0,0363**

p("partai"|"olahraga") = (1 + N("partai", "dokumen1") p("olahraga"|"dokumen1") +
N("partai", "dokumen2") p("olahraga"|"dokumen2") +
N("partai", "dokumen4") p("olahraga"|"dokumen4")) : f(p)

= (1 + 0 x 1 + 3 x 0 + 0 x 0,947) : 28,994

= (1 + 0 + 0 + 0) : 28,994

= 1 : 28,994

= **0,034**

p("partai"|"politik") = (1 + N("partai", "dokumen1") p("politik"|"dokumen1") +
N("partai", "dokumen2") p("politik"|"dokumen2") +
N("partai", "dokumen4") p("politik"|"dokumen4")) : f(p)

= (1 + 0 x 0 + 3 x 1 + 0 x 0,052) : 28,994

= (1 + 0 + 3 + 0) : 28,994

= 4 : 28,994

= **0,138**

p("pemilu"|"olahraga") = (1 + N("pemilu", "dokumen1") p("olahraga"|"dokumen1") +
N("pemilu", "dokumen2") p("olahraga"|"dokumen2") +
N("pemilu", "dokumen4") p("olahraga"|"dokumen4")) : f(p)

= (1 + 0 x 1 + 2 x 0 + 0 x 0,947) : 28,994

= (1 + 0 + 0 + 0) : 28,994

= 1 : 28,994

= **0,034**

p("pemilu"|"politik") = (1 + N("pemilu", "dokumen1") p("politik"|"dokumen1") +
N("pemilu", "dokumen2") p("politik"|"dokumen2") +
N("pemilu", "dokumen4") p("politik"|"dokumen4")) : f(p)

= (1 + 0 x 0 + 2 x 1 + 0 x 0,052) : 28,994

= (1 + 0 + 2 + 0) : 28,994

= 3 : 28,994

= **0,1035**

p("tandang"|"olahraga") = (1 + N("tandang", "dokumen1") p("olahraga"|"dokumen1") +
N("tandang", "dokumen2") p("olahraga"|"dokumen2") +

N(“tandang”, “dokumen4”) p(“olahraga”|”dokumen4”)) : f(p)

= (1 + 0 x 1 + 0 x 0 + 3 x 0,947) : 28,994

= (1 + 0 + 0 + 2,841) : 28,994

= 1 : 28,994

= **0,0998**

p(“tandang”|”politik”) = (1 + N(“tandang”, “dokumen1”) p(“politik”|”dokumen1”) +
N(“tandang”, “dokumen2”) p(“politik”|”dokumen2”) +
N(“tandang”, “dokumen4”) p(“politik”|”dokumen4”)) : f(p)

= (1 + 0 x 0 + 0 x 1 + 3 x 0,052) : 28,994

= (1 + 0 + 0 + 0,156) : 28,994

= 3 : 28,994

= **0,0381**

Langkah terakhir untuk menyelesaikan *maximization step* adalah meng-*update* nilai probabilitas untuk setiap kategori yang ada.

$$p(c_i) = \frac{1 + \sum_{j=1}^{|D|} p(c_i \mid d_j)}{|C| + |D|}$$

|*C*| adalah jumlah semua kategori
|*D*| adalah jumlah seluruh *training* dokumen

p(“olahraga”) = (1 + p(“olahraga”|”dokumen1”) + p(“olahraga”|”dokumen2”) +
p(“olahraga”|”dokumen4”)) : (2+3)

= (1 + 1 + 0 + 0,947) : 5

= 2,947 : 5

= ***0,589***

p(“politik”) = (1 + p(“politik”|”dokumen1”) + p(“politik”|”dokumen2”) +
p(“politik”|”dokumen4”)) : (2+3)

= (1 + 0 + 1 + 0,052) : 5

= 2,052 : 5

= ***0,41***

Model probabilistik setelah *maximization step* adalah sebagai berikut:

Kategori	$p(c_i)$	$p(w_{kj}\|c_i)$						
		bola	capres	gol	menang	partai	pemilu	tandang
olahraga	*0,589*	0,203	0,034	0,1034	0,1361	0,034	0,034	0,0998
politik	*0,41*	0,038	0,1724	0,034	0,0363	0,138	0,1035	0,0381

Proses *expectation step* dan *maximization step* dilakukan dalam beberapa iterasi hingga perubahan nilai probabilitas $p(w_{kj}|c_i)$ dan $p(c_i)$ tidak melebihi batas yang telah ditentukan dari iterasi sebelumnya. Namun, pada contoh ini, *expectation step* dan *maximization step* hanya dilakukan dalam satu kali iterasi, sehingga langkah selanjutnya adalah penentuan kategori untuk dokumen3:

$$c^* = \arg\max_{c_i \in C} \prod_k p(w_{kj} \mid c_i) \times p(c_i)$$

$p(\text{“olahraga”}|\text{”dokumen3”}) = p(\text{“olahraga”})$ x $p(\text{“partai”}|\text{”olahraga”})$ x $p(\text{“menang”}|\text{”olahraga”})$ x $p(\text{“tandang”}|\text{”olahraga”})$

$= 0,589$ x 0,034 x 0,1361 x 0,0998

$= 2,72 \times 10^{-4}$

$p(\text{“politik”}|\text{”dokumen3”}) = p(\text{“politik”})$ x $p(\text{“partai”}|\text{” politik”})$ x $p(\text{“menang”}|\text{” politik”})$ x $p(\text{“tandang”}|\text{” politik”})$

$= 0,41$ x 0,138 x 0,0363 x 0,0381

$= 7,825 \times 10^{-5}$

karena $p(\text{“olahraga”}|\text{”dokumen3”}) > p(\text{“politik”}|\text{”dokumen3”})$, maka kategori dari dokumen3 adalah **olahraga**.

BAB 4 IMPLEMENTASI

Pada bab ini dijelaskan mengenai implementasi dari perancangan klasifikasi dokumen teks. Penjelasan dimulai dari proses persiapan dokumen yang meliputi *converting* dan *filtering*, hingga modifikasi yang dilakukan pada *framework* yang digunakan untuk melakukan klasifikasi dokumen dengan algoritma Naïve Bayes dan Expectation Maximization. Sebagian besar program dibuat untuk melakukan eksperimen sesuai dengan perancangan dengan menggunakan bahasa pemrograman Java, hanya pada saat *converting* digunakan *tools* yang sudah tersedia untuk sistem operasi Windows.

4.1 Persiapan Dokumen

Persiapan dokumen dilakukan untuk mengubah format dokumen-dokumen yang digunakan untuk klasifikasi dokumen teks agar memiliki format standar. Proses persiapan dokumen terdiri dari dua tahapan, yang pertama adalah tahapan *converting* yang mengubah semua dokumen ke dalam format dokumen teks berekstensi txt. Proses selanjutnya adalah *filtering* yang menghilangkan *stopwords* dan tanda baca.

4.1.1 *Converting*

Proses *converting* dilakukan pertama kali pada rangkaian perancangan percobaan klasifikasi dokumen teks. Proses ini perlu dilakukan karena terdapat beberapa dokumen yang belum memiliki format txt. *Converting* dilakukan pada semua dokumen yang digunakan pada percobaan, baik dokumen yang telah memiliki format txt, maupun dokumen yang belum memiliki format txt.

Proses ini dilakukan dengan bantuan *tools* yang telah tersedia untuk sistem operasi Windows yang bernama Text Mining Tools 1.1.42. *Tools* ini merupakan program yang dapat mengkonversi dokumen pdf, doc, dan html menjadi dokumen txt. Selain telah memiliki antar muka yang cukup baik, *tools* ini juga dapat digunakan melalui *script command prompt* Windows sehingga dapat digunakan untuk mengkonversi dokumen dalam jumlah besar secara otomatis. Fasilitas tersebut memungkinkan

proses konversi dengan membuat *script* untuk perintah menjalankan Text Mining Tools 1.1.42.

```
function Main
       listFile <- listAllFile(root);
       createAndRunBatchFile(listFile);

function listAllFile(root) return listFile
       for each directory in root
              for each categoryDirectory in directory
                     listFile <- categoryDirectory.list;
       return listFile;

function createAndRunBatchFile(listFile)
       for each fileName in listFile
              print batchFile <- minetext, fileName, fileOut
       Runtime.execute(batchFile);
```

Gambar 4.1 *Pseudocode* Proses *Converting* Dokumen

Hasil akhir dari proses *converting* adalah kumpulan dokumen yang telah memiliki format txt. Dokumen-dokumen tersebut akan diproses lagi untuk menghilangkan *stopwords* dan tanda baca yang terdapat didalamnya.

Gambar 4.2 Hasil Keluaran dari Text Mining Tools 1.1.42

4.1.2 *Filtering*

Tahap selanjutnya setelah melakukan proses converting adalah *filtering*, yaitu proses penghilangan *stopwords* dan tanda baca pada setiap dokumen. Masalah yang muncul pada proses *filetring* adalah munculnya karakter-karakter ASCII 0 hingga 31 yang dapat menyebabkan terhentinya proses pembacaan file pada pembuatan *term documents matrix*. Oleh karena itu, selain menghilangkan *stopwords* dan tanda baca, juga perlu dilakukan proses penghilangan karakter-karakter ASCII tersebut.

Pada proses filtering digunakan dua daftar *stopwords* yang berbeda. Daftar *stopwords* pertama berbahasa Indonesia yang digunakan untuk data dokumen hukum dan artikel media massa dari Kompas. Daftar *stopwords* kedua merupakan *stopwords* umum bahasa Inggris yang digunakan untuk data 20Newsgroups *dataset*.

```
function Main
	listStopWordAndPuncMark <- listStopWord(language);
	listStopWordAndPuncMark += listPuctMark();
	listStopWordAndPuncMark += listAscii031();
	for each directory in root
		for each categoryDirectory in directory
			for each file in categoryDirectory
				filter(file, listStopWordAndPuncMark);
```

```
function listStopWord(language) return stopWordList
       if language == indonesia
              stopWordList <- insertFromFile(stopWordIndonesia.txt);
       if language == inggris
              stopWordList <- insertFromFile(stopWordInggris.txt);
       return stopWordList;

function insertFromFile(file) return stopWordList
       for each word in file
              stopWordList += word;
       return stopWordList;

function listPunctMark() return punctMarkList
       for each punctMark
              punctMarkList += punctMark;
       return punctMarkList;

function listAscii031() return ascii031
       for each ascii character in ascii table
              if asci character <= 31
                     asci031 += ascii character;
       return ascii031

function filter(file, listStopWordAndPuncMark)
       for each word in file
              for each stopword in listStopWordAndPuncMark
                     if word != stopword
                            filtered += word;
       print fileOut <- filtered;
```

Gambar 4.3 *Pseudocode* Proses *Filtering* Dokumen

Hasil yang didapatkan dari proses *filtering* ini berupa dokumen yang sudah siap dipakai untuk melakukan klasifikasi dokumen teks dan akan dipergunakan untuk melakukan pemilihan fitur serta pembuatan *term documents matrix*.

4.2 Pemilihan Fitur

Pemilihan fitur dilakukan setelah semua dokumen dihilangkan *stopwords* dan tanda bacanya. Pemilihan fitur ini dilakukan untuk memilih kata-kata apa saja yang akan membentuk model probabilistik dari kumpulan dokumen yang dimiliki. Pada tugas akhir ini percobaan dilakukan dengan jumlah fitur tertentu sesuai hasil pemilihan fitur pada percobaan pendahuluan untuk masing-masing data yang dimiliki.

Pada tugas akhir ini terdapat tiga buah percobaan awal. Percobaan awal pertama dilakukan pada data dokumen hukum dengan variasi jumlah fitur 100, 1000, 2000, 5000, 10000, dan semua fitur. Percobaan kedua dilakukan pada data artikel media massa dengan variasi jumlah fitur 100, 200, 500, 1000, 2000, 5000, 10000, 20000, dan semua fitur. Percobaan pendahuluan terakhir dilakukan pada data 20Newsgroups *dataset* dengan variasi jumlah fitur 100, 200, 500, 100, 2000, 5000, 1000, 20000, dan semua fitur.

Fitur yang digunakan terurut berdasarkan banyaknya jumlah kemunculan fitur tersebut dalam kumpulan dokumen yang ada. Fitur yang memiliki frekuensi kemunculan terbanyak akan berada pada urutan pertama dari kumpulan fitur yang akan digunakan, sebaliknya fitur yang memiliki frekuensi kemunculan paling sedikit akan menempati posisi terakhir.

```
function Main
       listFeature <- pickFeature(root, n);
       print fileFeature <- listFeature;

function pickFeature(root, n) return listFeature
       for each directory in root
              for each categoryDirectory in directory
                     for each file in categoryDirectory
                            for each word in file
                                   hashFeature(word)++;
       arrayListofFeature <- sortDescbyValue(hashFeature);
       maxFeature = 0;
```

```
        for each feature in hashFeature
                if maxFeature < n
                        listFeature[maxFeature]= arrayListofFeature[maxFeature];
                        maxFeature++;
        return listFeature;
```

Gambar 4.4 *Pseudocode* Pemilihan Fitur

4.3 K-*fold Cross Validation*

Pada penelitian klasifikasi dokumen ini digunakan 9-*fold cross validation*. Banyaknya dokumen *testing* pada satu kategori dalam sebuah *fold* berjumlah 1/9 dari jumlah total dokumen pada kategori tersebut. Data *training* pada percobaan ini dibagi menjadi dua yaitu *labeled documents* dan *unlabeled documents* sehingga banyaknya dokumen untuk masing-masing *labeled documents* dan *unlabeled documents* berjumlah 4/9 dari total dokumen pada setiap kategori. Dengan menggunakan 9-*fold cross validation*, maka pada masing-masing metode klasifikasi akan dibuat sembilan data *testing*, *unlabeled documents*, dan *labeled documents*, dengan variasi sebagai berikut:

- Data *testing n* yang digunakan adalah *fold* (*n*-1 modulo 9)+1.
- *Labeled documents n* yang digunakan adalah gabungan *fold* (*n* modulo 9)+1, *fold* (*n*+1 modulo 9)+1, *fold* (*n*+2 modulo 9)+1, dan *fold* (*n*+3 modulo 9)+1.
- *Unlabeled documents n* yang digunakan adalah gabungan dari *fold* (*n*+4 modulo 9)+1, *fold* (*n*+5 modulo 9)+1, *fold* (*n*+6 modulo 9)+1, dan *fold* (*n*+7 modulo 9)+1.

```
function Main
        folding(root, numFold);

function folding(root, numFold)
        for each directory in root
                for each categoryDirectory in directory
                        index = 0;
                        while categoryDirectory not empty
```

```
file <- pickRandomFile();
move file to fold[++index mod numofFold];
```

Gambar 4.5 *Pseudocode Folding* Dokumen

4.4 Pembuatan *Term Documents Matrix*

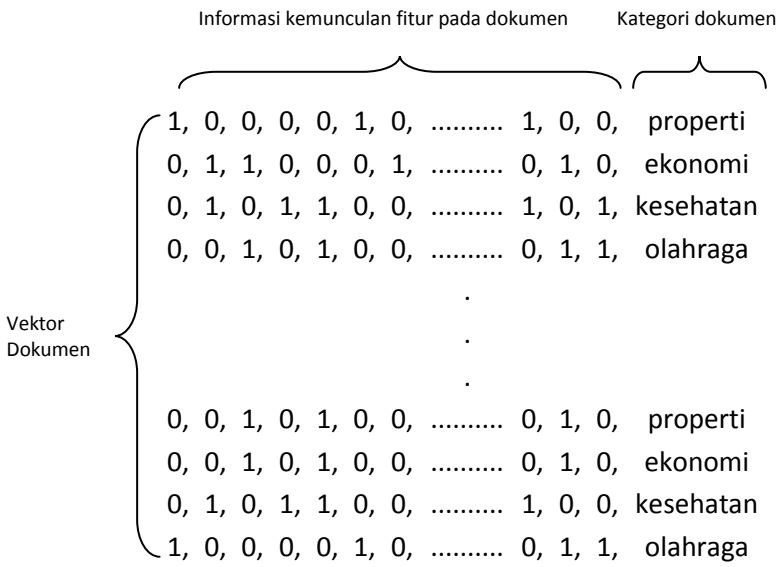
Proses klasifikasi dokumen teks dilakukan dengan merepresentasikan kumpulan dokumen yang dimiliki sebagai sebuah *term documents matrix*. Matriks ini akan menjadi *input* bagi *machine learning classifier* dengan algoritma Naïve Bayes maupun algoritma Expectation Maximization. Matriks ini menyimpan informasi nilai fitur yang dimiliki oleh tiap-tiap dokumen. Pembuatan *term documents matrix* dilakukan untuk setiap variasi jenis fitur. Informasi nilai fitur yang digunakan selama percobaan ada empat buah, yaitu: *presence*, *frequency*, *frequency normalized*, dan pembobotan *tf-idf*.

Tabel 4.1 Variasi Jenis Fitur

Label	Keterangan
Presence	Menyimpan nilai kemunculan fitur, apabila sebuah fitur muncul dalam dokumen, maka fitur tersebut bernilai satu untuk dokumen yang dimaksud, jika tidak muncul fitur tersebut akan bernilai nol.
Frequency	Menyimpan nilai frekuensi kemunculan fitur pada sebuah dokumen.
Frequency Normalized	Menyimpan nilai dari jumlah kemunculan suatu fitur dalam suatu dokumen dibagi dengan jumlah seluruh fitur yang ada pada dokumen tersebut.
TF-IDF	Pembobotan *tf-idf* pada frekuensi kemunculan fitur

Baris pada *term documents matrix* ini merepresentasikan dokumen yang digunakan, sedangkan kolomnya merepresentasikan kata yang terdapat pada dokumen tersebut.

Kolom terakhir dari matriks tersebut merepresentasikan kategori dari dokumen tersebut.



Gambar 4.6 Format Penyimpanan *Term Documents Matrix dengan* Informasi *Presence*

Untuk satu *fold* dokumen akan dibuat tiga buah *term documents matrix*, satu buah untuk matriks dokumen *testing*, satu untuk matriks *labeled documents*, dan terakhir untuk matriks *unlabeled documents*. Jadi untuk satu buah percobaan akan dibuat 27 buah *term documents matrix* karena jumlah *fold* yang digunakan adalah sembilan. Pseudocode pembuatan *term documents matrix* dapat dilihat pada Gambar 4.7.

```
function Main
        createTermDocMatrix(listDocuments, listFeatures, featureType);

function createTermDocMatrix(istDocuments, listFeatures, featureType)
        i <- 0;
        for each documen in listDocuments
                j <- 0;
                for each feature in listFeature
                        if featureType == frequency
                                termDocMatrix[i][++j]<-countFeature(feature, document);
                        else if featureType == presence
                                termDocMatrix[i][++j] <- isExist(feature, document);
```

```
                    else if featureType == frequencyNormalized
                            termDocMatrix[i][++j]<-freqNorm(fetaure, documents);
                    else if featureType == tf-idf
                            termDocMatrix[i][++j]<-tfidf(feature, listDocuments);
            termDocMatrix[i++][j] <- categoryDocument;

function countFeature(feature, document) return numFeature
        for each word in document
                if word == feature
                        numFeature++;
        return numFeature;

function isExist(feature, document) return featureExistance
        featureExistance <- 0;
        for each word in document
                if word == feature
                        featureExistance <- 1;
        return featureExistance;

function freqNorm(fetaure, documents) return freqNorm
        featureFrequency <- countFeature(feature, document);
        for each word in document
                numWords++;
        freqNorm <- featureFrequency/numWords;
        return freqNorm;

function tfidf(feature, listDocuments) return tfidf
        featureFrequency <- countFeature(feature, document);
        for each document in listDocument
                if isExist(feature, document)
                        numDocuments++;
        tfidf <- featureFrequency/numDocuments;
        return tfidf;
```

Gambar 4.7 *Pseudocode* Pembuatan *Term Documents Matrix*

4.5 Klasifikasi Dokumen Teks

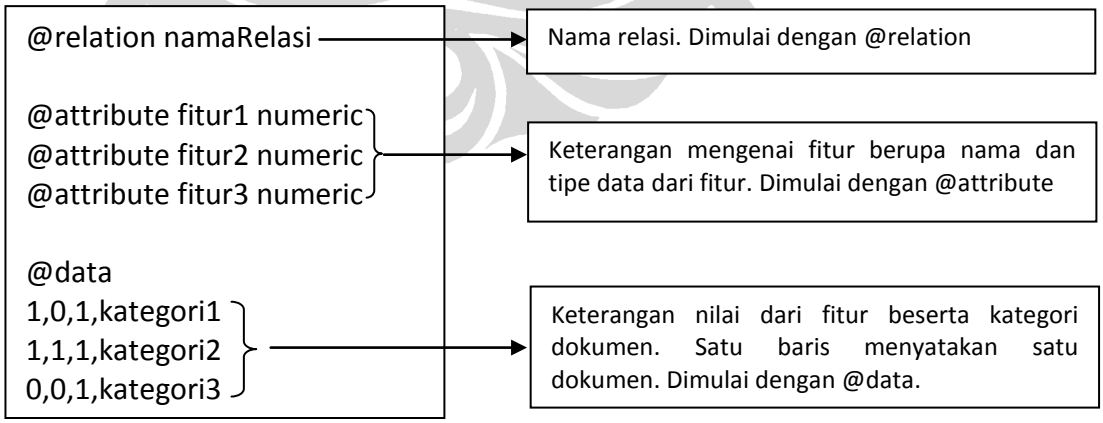
Implementasi program klasifikasi dokumen dilakukan menggunakan *tools* yang berbeda untuk masing-masing metode. Mesin klasifikasi dengan algoritma Naïve Bayes dikembangkan dengan menggunakan WEKA, sedangkan mesin klasifikasi

dengan algoritma Expectation Maximization dikembangkan dengan menggunakan MinorThird.

Proses pembuatan data *training* dan data *testing* pada kedua metode klasifikasi yang digunakan sedikit berbeda. WEKA, *tools* yang digunakan untuk membuat mesin klasifikasi menggunakan algoritma Naïve Bayes memiliki *input* berupa berkas ARFF yang lebih sederhana daripada format *input* pada MinorThird. Oleh karena itu, pada percobaan klasifikasi menggunakan algoritma Expectation Maximization perlu sedikit perubahan agar dapat membaca *input* dari file ARFF yang telah ada.

4.5.1 Naïve Bayes

Implementasi klasifikasi dokumen teks menggunakan algoritma Naïve Bayes dilakukan dengan menggunakan *library* WEKA 3.5.7 yang didapat dari http://www.cs.waikato.ac.nz/~ml/weka/. WEKA merupakan kumpulan algoritma *machine learning* yang ditulis dalam bahasa pemrograman Java. *Input* yang diperlukan WEKA dalam melakukan klasifikasi dokumen teks berupa berkas ARFF (Atribute-Relation File Format).



Gambar 4.8 Format Berkas ARFF

Proses klasifikasi ini dilakukan dengan terlebih dahulu mempersiapkan data *training* dan data *testing*. *Training* data untuk percobaan menggunakan algoritma Naïve Bayes hanyalah *labeled documents* saja. Berkas ARFF dibuat untuk setiap *term documents matrix* pada setiap *fold*. Penyesuaian format *input* ARFF dilakukan dengan menambahkan informasi @relation, @attribute, dan @data. Keterangan namaRelasi diganti dengan klasifikasi_fold_n, dengan n merupakan nomor *fold*, dan fitur diganti dengan daftar fitur yang dimiliki dan diberi tipe data *numeric*. Bagian data tinggal menyalin dari *term documents matrix* yang telah dibuat sebelumnya.

Klasifikasi dokumen teks dengan menggunakan Naïve Bayes dilakukan dengan membuat program dalam bahasa Java dan menggunakan *library* WEKA. Pada implementasinya program yang diciptakan membuat *dataset* dengan membaca *file* ARFF. Pembuatan model probabilistik pada awalnya dilakukan dengan memberikan informasi kepada *classifier* mengenai data *training* yang digunakan. Setelah *classifier* terbentuk, penghitungan akurasi dilakukan dengan mencocokkan kategori data *testing* yang dihasilkan *classifier* dengan kategori dokumen sebenarnya yang telah disimpan sebelumnya.

```
function Main
       classify(testingDocuments, trainingDocuments, listFeature);

function classify(testingDocuments, trainingDocuments, listFeature)
       classifier <- buildClassifier(listFeature);
       for each row in trainingDocuments
              instance <- getInstance(row);
              classifier.update(instance);
       datasetTesting <- createDataset(testingDocuments);
       righClassification <- 0;
       for each instance in datasetTesting
              probDist <- classifier.getProbDistForInstance(instance);
              category <-  searchMax(probDist);
              realCategory <- instance.getRealCategory();
              if(category == realCategory)
                     rightClassification++;
              else
```

```
                    hash-wrongClassfication(realCategory, category);
      numberofTestingData <- datasetTesting.size();
      accuracy <- rightClassification/numberofTestingData;

//penjelasan classifier.update(instance)
function update(instance)
      category <- instance.getCategory();
      countPriorProb(category);
      for each word in instance
            countCondProb(word, category);
```

Gambar 4.9 *Pseudocode* Klasifikasi Dokumen Teks Menggunakan Naïve Bayes

Hasil yang dicatat dari setiap percobaan adalah jumlah data *training*, jumlah kategori, jumlah fitur, jumlah data *testing*, akurasi klasifikasi, serta kesalahan klasifikasi yang terjadi. Data-data tersebut yang akan dianalisis pada tugas akhir ini. Berikut contoh hasil keluaran dari klasifikasi dokumen teks dengan Naïve Bayes.

```
Percobaan ke :1
Jumlah Data Training: 210
Jumlah Fitur: 5000
Jumlah Kategori: 5
Jumlah Data Testing: 55
ekonomi
Ke: ekonomi Salah: 0
Ke: kesehatan Salah: 0
Ke: olahraga Salah: 0
Ke: properti Salah: 0
Ke: travel Salah: 1
kesehatan
Ke: ekonomi Salah: 2
Ke: kesehatan Salah: 0
Ke: olahraga Salah: 0
Ke: properti Salah: 0
Ke: travel Salah: 0
olahraga
Ke: ekonomi Salah: 0
Ke: kesehatan Salah: 0
Ke: olahraga Salah: 0
Ke: properti Salah: 0
```

```
Ke: travel Salah: 0
properti
Ke: ekonomi Salah: 0
Ke: kesehatan Salah: 0
Ke: olahraga Salah: 0
Ke: properti Salah: 0
Ke: travel Salah: 1
travel
Ke: ekonomi Salah: 0
Ke: kesehatan Salah: 1
Ke: olahraga Salah: 0
Ke: properti Salah: 1
Ke: travel Salah: 0
Akurasi 89.0909090909091%

Percobaan ke :2
...

Percobaan ke :9
...

Akurasi Rata-rata: 91.31313131313131
```

Gambar 4.10 Hasil Keluaran Klasifikasi Dokumen Teks dengan Naïve Bayes

4.5.2 Expectation Maximization

Implementasi klasifikasi dokumen teks menggunakan algoritma Expectation Maximization dilakukan dengan menggunakan *library* MinorThird yang didapat dari http://minorthird.sourceforge.net/. MinorThird merupakan kumpulan *library* Java yang dapat digunakan untuk klasifikasi teks. *Input* yang diperlukan MinorThird dalam melakukan klasifikasi dokumen teks pada percobaan ini dibuat dengan mengkonversi berkas ARFF. Selain dapat menghemat penyimpanan berkas, hal ini juga dapat menghemat waktu karena proses pembuatan *dataset* dapat dilakukan sekali saja.

```
function arffToMinorthird
        wekaInstance <- (arffFile);
        SemiSupervisedDataset dsetM3rd <- new SemiSupervisedDataset();
        String subpopID <- dsetWeka.relationName();
        dataNum <- 0;
        for each instance in wekaInstance
                ++dataNum;
                m3Instance = MutableInstance("::data[" + dataNum + "]",subpopID);
                for each feature in instance
                        instM3rd.addNumeric(instance.name(),instance.value());
                dsetM3rd.add(new Example(minorthirdInstance));
```

Gambar 4.11 *Pseudocode* Konversi ARFF ke *Dataset* MinorThird

Klasifikasi dokumen teks dengan menggunakan Expectation Maximization dilakukan dengan membuat program dalam bahasa Java dan menggunakan *library* MinorThird. Keterbatasan *memory* komputer menjadi kendala pada percobaan dengan Expectation Maximization. Beberapa modifikasi telah dilakukan untuk menghilangkan masalah tersebut. Perubahan pertama yang dilakukan dengan mengubah proses pada saat meng-*update* model probabilistik melalui *file* secara langsung urung dilakukan, karena Expectation Maximization merupakan algoritma iteratif yang men-g*update* model probabilistik dalam beberapa iterasi, maka proses pembacaan *dataset* melalui *file* secara langsung akan memakan banyak waktu. Modifikasi proses pembentukan model probabilistik dilakukan dengan menyederhanakan representasi *term documents matrix*

untuk algoritma Expectation Maximization dengan membuang fitur-fitur yang memiliki nilai nol, sehingga hanya fitur yang memiliki nilai lebih besar dari nol yang disimpan dalam matriks tersebut. Selain itu untuk menyederhanakan proses *update*, pada setiap iterasi, hanya kategori hasil *expectation step* dari setiap *instance* saja yang disimpan untuk perhitungan iterasi selanjutnya.

```
function Main
      classify(testingDocuments, labeledDocuments, unlabeledDocuments);
function classify(testingDocuments, labeledDocuments, unlabeledDocuments)
      classifier <- buildClassifier(labeledDocuments);
      datasetUnlabeled(labeledDocuments);
      while classifier not stable
            for each instance in datasetUnlabeled
                  probDist <- classifier.getProbDistForInstance(instance);
                  category <-  searchMax(probDist);
                  newDatasetUnlabeled.add(instance, category);
            for each instance in newDatasetUnlabeled
                  classifier.update(instance);
      datasetTesting <- createDataset(testingDocuments);
      righClassification <- 0;
      for each instance in datasetTesting
            probDist <- classifier.getProbDistForInstance(instance);
            category <-  searchMax(probDist);
            if(category == realCategory)
                  rightClassification++;
            else
                  hash-wrongClassfication(realCategory, category);
      numberofTestingData <- datasetTesting.size();
      accuracy <- rightClassification/numberofTestingData;
//penjelasan classifier.update(instance)
function update(instance)
      category <- instance.getCategory();
      countPriorProb(category);
      for each word in instance
            countCondProb(word, category);
```

Gambar 4.12 *Pseudocode* Klasifikasi Dokumen Teks Menggunakan Expectation Maximization

Hasil yang dicatat dari setiap percobaan adalah jumlah *labeled documents*, *jumlah unlabeled documents*, jumlah kategori, jumlah fitur, jumlah data *testing*, akurasi klasifikasi, serta kesalahan klasifikasi yang terjadi. Data-data tersebut yang akan dianalisis pada tugas akhir ini. Berikut contoh hasil keluaran dari klasifikasi dokumen teks dengan Expectation Maximization.

```
Percobaan ke :1
Jumlah Labeled: 210
Jumlah Unlabeled: 200
Jumlah Data Testing: 55
Jumlah Fitur: 5000
Jumlah Kelas: 5
ekonomi
Ke: ekonomi Salah: 0
Ke: kesehatan Salah: 0
Ke: olahraga Salah: 0
Ke: properti Salah: 0
Ke: travel Salah: 0
kesehatan
Ke: ekonomi Salah: 1
Ke: kesehatan Salah: 0
Ke: olahraga Salah: 0
Ke: properti Salah: 0
Ke: travel Salah: 0
olahraga
Ke: ekonomi Salah: 0
Ke: kesehatan Salah: 0
Ke: olahraga Salah: 0
Ke: properti Salah: 0
Ke: travel Salah: 0
properti
Ke: ekonomi Salah: 0
Ke: kesehatan Salah: 0
Ke: olahraga Salah: 0
Ke: properti Salah: 0
Ke: travel Salah: 0
travel
Ke: ekonomi Salah: 2
Ke: kesehatan Salah: 0
Ke: olahraga Salah: 0
Ke: properti Salah: 1
Ke: travel Salah: 0
Akurasi 92.72727272727272%
Percobaan ke :2
...

Percobaan ke :9
...



Akurasi      Rata-rata:      95.13131313131312
```

Gambar 4.13 Hasil Keluaran Klasifikasi Dokumen Teks dengan Expectation Maximization

BAB 5 HASIL DAN PEMBAHASAN

Pada bab ini ditampilkan hasil penelitian klasifikasi dokumen teks dengan menggunakan metode Naïve Bayes dan Expectation Maximization. Pembahasan dibedakan berdasarkan variabel yang diujikan pada penelitian ini. Pembahasan diawali dengan penjelasan mengenai variabel eksperimen yang digunakan (lihat subbab 5.1), hasil eksperimen terhadap jumlah fitur dan penghilangan *stopwords* (lihat subbab 5.2), dilanjutkan dengan pembahasan mengenai hasil klasifikasi dokumen teks untuk masing-masing variabel percobaan yaitu hasil klasifikasi berdasarkan aspek penggunaan jenis fitur (lihat subbab 5.3), jumlah kategori (lihat subbab 5.4), dan pengaruh penggunaan *unlabeled documents* terhadap hasil klasifikasi dokumen teks (lihat subbab 5.5). Pada setiap subbab dibahas klasifikasi untuk setiap jenis data yang digunakan yaitu data dokumen hukum, data 20Newsgroups *dataset*, dan data artikel media massa.

5.1 Variabel Eksperimen

Pembahasan mengenai hasil klasifikasi dikelompokkan pada beberapa subbab. Pengelompokkan ini dilakukan berdasarkan aspek-aspek yang ingin dilihat dari hasil percobaan klasifikasi dokumen teks. Hal ini dilakukan untuk memusatkan perhatian mengenai pengaruh aspek-aspek tersebut terhadap tingkat akurasi klasifikasi. Ada tiga buah aspek utama yang ingin dibahas pada percobaan ini, diantaranya adalah: penggunaan jenis fitur, jumlah kategori, dan pengaruh *unlabeled documents* terhadap hasil klasifikasi dokumen teks. Adapun variabel-variabel yang digunakan sebagai variasi *input* dalam percobaan untuk tugas akhir ini dapat dilihat pada Tabel 5.1.

Tabel 5.1 Variabel Percobaan

Variabel	Nilai
Metode	- Naïve Bayes - Expectation Maximization

Data	- Dokumen hukum - 20Newsgroups *dataset* - Artikel media massa
Jenis fitur	- *Presence* - *Frequency* - *Frequency Normalized* - Pembobotan *tf-idf*
Stopwords	- *Stopwords* bahasa Indonesia pada data dokumen hukum dan artikel media massa - *Stopwords* bahasa Inggris pada data 20Newsgroups *dataset*
Jumlah *labeled documents*	a. Dokumen hukum: 5, 10, 20, 50, 100, 200. b. 20Newsgroups *dataset*: 20, 100, 750, 1200, 2000, 4000, 6000, 8000. c. Artikel media massa: 5, 10, 20, 30, 40, 50, 100, 200, 300, 400, 500, 550.
Jumlah *unlabeled documents*	a. Dokumen hukum: 0, 50, 100, 200. b. 20Newsgroups *dataset*: 0, 500, 1000, 2000, 4000 6000, 8000. c. Artikel media massa: 0, 200, 300, 400, 500.
Jumlah kategori	a. Dokumen hukum: 3, 4. b. 20Newsgroups *dataset*: 3, 5,10, 15, 20. c. Artikel media massa: 3, 4, 5.

Pada tugas akhir ini jumlah fitur yang akan digunakan pada setiap percobaan untuk melihat pengaruh aspek-aspek yang telah disebutkan diatas, ditentukan dengan melakukan percobaan awal untuk mencari jumlah fitur yang dirasa telah memiliki akurasi cukup baik untuk melakukan klasifikasi dokumen pada masing-masing data baik data dokumen hukum, 20Newsgroups *dataset*, dan artikel media massa. Percobaan ini dilakukan dengan memvariasikan jumlah fitur. Jumlah dokumen yang digunakan pada percobaan awal ini adalah semua dokumen yang dimiliki untuk masing-masing data, sedangkan jenis fitur yang digunakan adalah *presence*, *frequency*, *frequency normalized* dan pembobotan *tf-idf*.

Nilai akurasi yang didapatkan untuk setiap percobaan adalah nilai rata-rata akurasi percobaan untuk k-*fold*. Pada percobaan ini digunakan sembilan *fold*. Nilai akurasi merupakan hasil pembagian antara jumlah dokumen yang terklasifikasi dengan benar dengan jumlah keseluruhan dokumen *testing* yang digunakan.

5.2 Hasil Eksperimen terhadap Jumlah Fitur dan Penghilangan *Stopwords*

Percobaan yang dilakukan pada subbab ini bertujuan untuk menentukan jumlah fitur yang akan digunakan pada percobaan-percobaan selanjutnya. Hal tersebut dilakukan dengan memilih *top-n* fitur yaitu *n* buah fitur yang memiliki jumlah frekuensi terbanyak yang dianggap telah cukup baik untuk melakukan satu rangkaian proses klasifikasi. Pertimbangan yang digunakan untuk menentukan pemilihan fitur tersebut adalah penelitian (Margaretha, 2008). Pada penelitian tersebut telah dilakukan variasi pemilihan fitur yaitu dengan pemilihan fitur secara random, dan pemilihan *top-n* fitur. Pada pemilihan fitur secara random dihasilkan *precision* 0.0181 dan *recall* 0.0623, sedangkan pada pemilihan *top-n* fitur dihasilkan *precision* 0.1009 dan *recall* 0.2285. Berdasarkan hasil tersebut, maka pada tugas akhir ini digunakan *top-n* fitur untuk melakukan klasifikasi dokumen teks. Pemilihan fitur ini dilakukan untuk semua data yang digunakan yaitu data dokumen hukum, artikel media massa, dan 20Newsgroups *dataset*.

Percobaan pada subbab ini dilakukan dengan menggunakan algoritma Naïve Bayes saja. Hal ini dilakukan mengingat adanya keterbatasan *memory* komputer pada percobaan menggunakan algoritma Expectation Maximization dengan jumlah fitur dan dokumen yang sangat besar. Selain itu, algoritma Expectation Maximization yang digunakan pada eksperimen ini adalah algoritma yang berbasis Naïve Bayes, jadi hasil yang diperoleh pada percobaan ini cukup mewakili untuk kedua metode yang digunakan (Naïve Bayes dan Expectation Maximization). Pada percobaan ini juga ditunjukkan pengaruh penggunaan *stopwords* terhadap hasil klasifikasi dokumen teks. Hasil pemilihan fitur ini akan digunakan sebagai variabel tetap pada percobaan-percobaan yang akan dilakukan berikutnya.

5.2.1 Hasil Klasifikasi untuk Data Dokumen Hukum

Pada bagian ini diperlihatkan hasil pemilihan fitur untuk data dokumen hukum. Percobaan pada dokumen hukum untuk pemilihan fitur ini dilakukan dengan menggunakan empat buah kategori dokumen hukum, yaitu kategori Perpu, PP, UU, dan UU Darurat. Hasil yang ditunjukkan adalah hasil akurasi rata-rata klasifikasi dokumen dengan menggunakan jenis fitur *presence*, *frequency*, *frequency normalized*, dan pembobotan *tf-idf*. Jumlah dokumen yang digunakan adalah 400 dokumen hukum. Variasi jumlah fitur yang digunakan adalah 100, 1000, 2000, 5000, 10000, dan semua fitur.

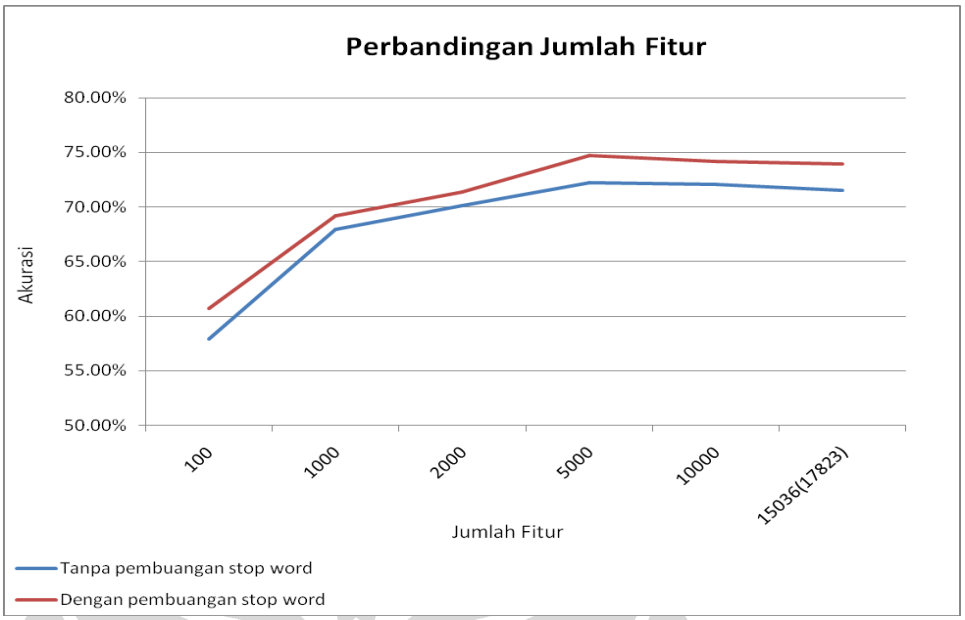
Hasil pemilihan fitur ini nantinya akan digunakan untuk standar penggunaan jumlah fitur untuk percobaan-percobaan selanjutnya pada dokumen hukum. Percobaan ini juga akan membandingkan pengaruh penggunaan *stopwords*. *Stopwords* yang digunakan pada percobaan ini adalah *stopwords* bahasa Indonesia. *Stopwords* tersebut merupakan *stopwords* umum bahasa Indonesia, bukan *stopwords* khusus untuk dokumen hukum (lihat subbab 3.3.2).

Tabel 5.2 Hasil Pemilihan Fitur pada Dokumen Hukum

Dengan pembuangan *stopwords*			Tanpa pembuangan *stopwords*	
Jumlah Fitur	Akurasi		Jumlah Fitur	Akurasi
100	60.74%		100	57.88%
1000	69.19%		1000	67.93%
2000	71.38%		2000	70.11%
5000	74.67%		5000	72.22%
10000	74.15%		10000	72.06%
15036	73.90%		17823	71.55%

Dari hasil yang diperoleh, akurasi yang didapatkan berkisar antara 60,74% hingga 74,67% pada percobaan dengan melakukan pembuangan *stopwords*. Hasil akurasi yang diperoleh tersebut tidak terlalu tinggi, hal ini dikarenakan pembagian kategori pada data dokumen hukum bukan pembagian berdasarkan topik, namun lebih berdasarkan jenis dokumen. Pembahasan mengenai pengaruh pembagian kategori ini akan dibahas lebih lanjut pada subbab 5.4.

Dari tabel diatas terlihat perbedaan hasil akurasi untuk masing-masing jumlah fitur tidak terlalu lebar. Hanya pada saat awal, yaitu dengan menggunakan 100 fitur, hasil akurasi yang dihasilkan sangat rendah dan memiliki perbedaan cukup jauh dengan hasil akurasi lain. Hasil akurasi tertinggi yakni 74,67% diperoleh pada saat menggunakan 5000 fitur. Penurunan akurasi terjadi saat jumlah fitur yang digunakan ditambah menjadi 10000 fitur. Akurasi yang didapat pada kondisi tersebut turun 0,52% dari hasil akurasi tertinggi yang didapat. Hal ini menunjukkan bahwa kemunculan fitur pada urutan di atas 5000 jumlahnya tidak terlalu signifikan, hanya akan mempengaruhi proses klasifikasi dokumen-dokumen tertentu saja.



Gambar 5.1 Hasil Pemilihan Fitur pada Dokumen Hukum

Percobaan berikutnya dilakukan tanpa menggunakan tahap pembuangan *stopwords*, sehingga tidak ada kata-kata yang dihilangkan dari daftar fitur yang ada. Hasil akurasi yang diperoleh berkisar antara 57,88% hingga 72,22%. Hasil ini lebih rendah apabila dibandingkan dengan hasil klasifikasi dengan menggunakan tahapan pembuangan *stopwords*. Hal ini menunjukkan bahwa pembuangan *stopwords* mampu meningkatkan hasil akurasi klasifikasi dokumen teks.

Sama halnya seperti hasil klasifikasi dengan pembuangan *stopwords*, perbedaan hasil akurasi antara masing-masing jumlah fitur tidak terlalu lebar, hanya pada penggunaan 100 fitur, hasil akurasi yang diperoleh sangat rendah dan terpaut cukup jauh hingga mencapai 10,05% dari hasil klasifikasi dengan jumlah fitur 1000. Hasil akurasi tertinggi 72,22% didapat pada saat penggunaan 5000 fitur. Hal ini semakin memperkuat alasan untuk menggunakan jumlah fitur sebanyak 5000 pada percobaan-percobaan selanjutnya untuk data dokumen hukum.

5.2.2 Hasil Klasifikasi untuk Data Artikel Media Massa

Percobaan pada bagian ini dilakukan dengan menggunakan data artikel media massa dari kompas.com. Percobaan ini masih dilakukan untuk mencari jumlah fitur yang akan digunakan untuk melakukan percobaan-percobaan lanjutan pada data artikel media massa. Percobaan ini dilakukan dengan menggunakan lima buah kategori, yaitu kategori keuangan, kesehatan, olahraga, properti dan travel. Kategori-kategori tersebut memiliki kemiripan yang kecil. Hasil percobaan yang dihasilkan merupakan akurasi rata-rata dengan menggunakan empat buah jenis fitur. Jumlah dokumen yang digunakan pada percobaan ini adalah 1100 dokumen untuk data *training* dan 140 untuk data *testing*. Variasi jumlah fitur yang digunakan pada percobaan data artikel media massa lebih banyak dibanding variasi jumlah fitur pada percobaan dokumen hukum, dengan variasi 100, 200, 500, 1000, 2000, 5000, 10000, 20000, dan semua fitur.

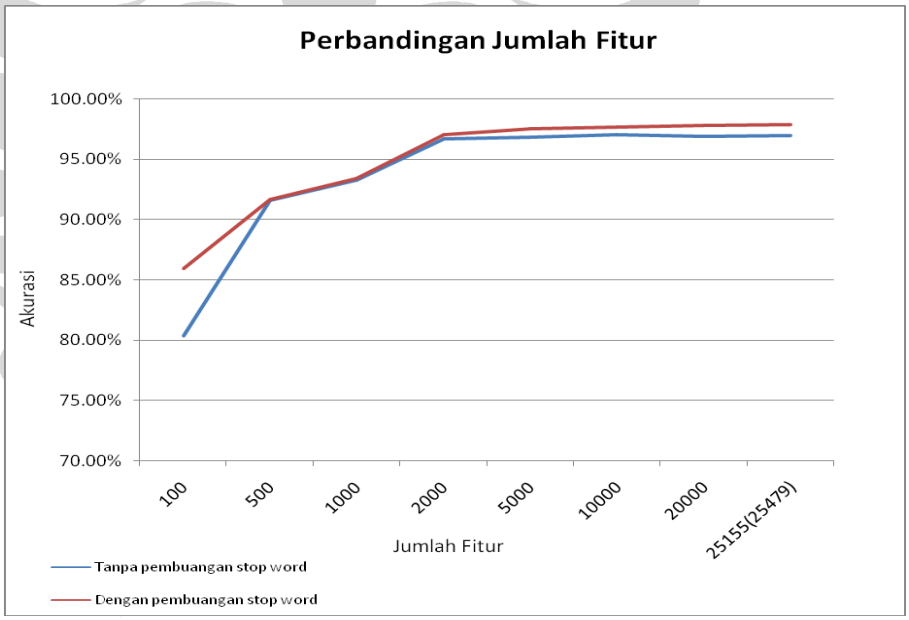
Tabel 5.3 Hasil Pemilihan Fitur pada Artikel Media Massa

Dengan pembuangan *stopwords*			Tanpa pembuangan *stopwords*	
Jumlah Fitur	Akurasi		Jumlah Fitur	Akurasi
100	85.91%		100	80.36%
500	91.67%		500	91.61%
1000	93.39%		1000	93.30%
2000	97.05%		2000	96.73%
5000	97.54%		5000	96.83%
10000	97.67%		10000	97.06%
20000	97.81%		20000	96.88%
25155	97.86%		25479	96.99%

Hasil akurasi klasifikasi dengan penghilangan *stopwords* yang didapat sangat tinggi, yaitu berkisar antara 85,91% hingga 97,86%. Akurasi yang cukup tinggi ini bahkan telah terlihat ketika penggunaan 500 fitur. Peningkatan akurasi ini terlihat pada Tabel 5.3, dimana akurasi terus meningkat dan mencapai akurasi tertinggi 97,86% pada penggunaan 25155 (semua) fitur. Hal ini memang didukung dengan tingkat kemiripan dokumen antar kategori yang cukup rendah. Percobaan kedua pada data

artikel media massa dilakukan tanpa melakukan penghilangan *stopwords* dari daftar fitur yang ada. *Stopwords* yang digunakan sama dengan *stopwords* pada percobaan pada data dokumen hukum (lihat subbab 3.3.2).

Hasil yang diperoleh pada percobaan klasifikasi artikel media massa tanpa melakukan penghilangan *stopwords* secara keseluruhan lebih rendah dibandingkan dengan hasil klasifikasi artikel media massa dengan melakukan penghilangan *stopwords*. Hasil akurasi terendah yang diperoleh adalah 80,36% yang diperoleh dengan menggunakan 100 fitur, sedangkan akurasi tertinggi 97.06% diperoleh pada penggunaan 10000 fitur. Dari tabel diatas dapat terlihat pengaruh penggunaan *stopwords* membantu meningkatkan akurasi terutama pada jumlah fitur besar diatas 10000 fitur yang ditunjukkan dengan terus menaiknya tingkat akurasi pada penggunaan lebih dari 10000 fitur pada percobaan dengan melakukan penghilangan *stopwords*.



Gambar 5.2 Hasil Pemilihan Fitur pada Artikel Media Massa

Dari kedua hasil diatas penggunaan 5000 fitur dan penghilangan *stopwords* akan dilakukan untuk percobaan-percobaan berikutnya pada data artikel media massa. Hal ini didasarkan pada hasil percobaan data artikel media massa dengan penghilangan

stopwords dan penggunaan 5000 fitur telah mencapai akurasi 97,54% yang dirasa telah cukup baik untuk melakukan klasifikasi.

5.2.3 Hasil Klasifikasi untuk Data 20Newsgroups *Dataset*

Percobaan pendahuluan lain untuk mencari jumlah fitur yang akan digunakan adalah percobaan dengan menggunakan 20Newsgroups *datatset*. Data ini memiliki 20 kategori dengan jumlah total data yang digunakan adalah 16000 dokumen. Beberapa kategori yang ada pada 20Newsgroups *dataset* ini memiliki tingkat kemiripan yang cukup tinggi, sebagi contoh kategori *Computer* IBM PC *Hardware* dan *Computer* Mac *Hardware* memiliki tingkat kemiripan yang cukup tinggi. Variasi jumlah fitur yang digunakan pada percobaan ini mirip dengan variasi jumlah fitur yang digunakan pada percobaan dengan data artikel media massa yakni dengan jumlah 100, 200, 500, 100, 2000, 5000, 1000, 20000, dan semua fitur.

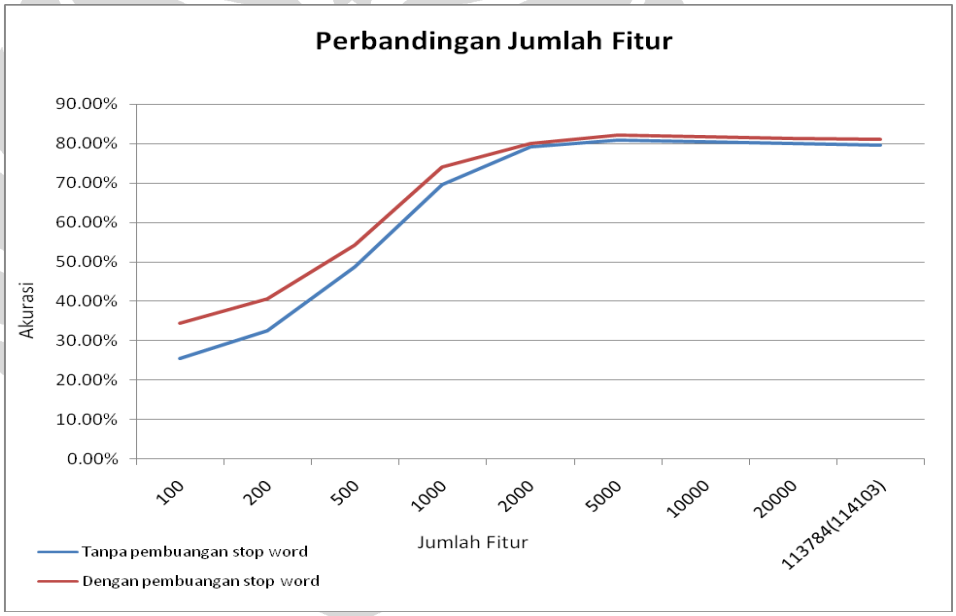
Tabel 5.4 Hasil Pemilihan Fitur pada 20Newsgroups *Dataset*

Dengan pembuangan *stopwords*			Tanpa pembuangan *stopwords*	
Jumlah Fitur	Akurasi		Jumlah Fitur	Akurasi
100	34.53%		100	25.52%
200	40.66%		200	32.57%
500	54.33%		500	48.77%
1000	74.09%		1000	69.65%
2000	80.12%		2000	79.25%
5000	82.17%		5000	80.98%
10000	81.71%		10000	80.62%
20000	81.40%		20000	80.20%
113784	81.16%		114103	79.66%

Hasil akurasi yang diperoleh pada percobaan dengan pembuangan *stopwords* ini cukup rendah, yakni hanya berkisar antara 34,53% hingga akurasi tertinggi 82,17%. Akurasi tertinggi tersebut diperoleh pada penggunaan 5000 fitur. Kenaikan tingkat akurasi hanya terjadi pada jumlah fitur yang kecil dibawah 5000 fitur. Setelah penggunaan fitur diatas 5000 justru terjadi penurunan akurasi hasil klasifikasi. Hal

tersebut menunjukkan bahwa fitur-fitur yang berada pada urutan 5000 keatas berpengaruh negatif terhadap hasil klasifikasi dokumen teks. Hal ini disebabkan oleh nilai kemunculan fitur pada urutan 5000 keatas yang rendah sehingga nilainya kurang merepresentasikan kategori yang ada.

Selain jumlah kategori yang besar yaitu mencapai 20 buah kategori (lihat subbab 5.4), hasil klasifikasi yang rendah ini dipengaruhi oleh banyaknya dokumen yang memiliki *footer* yang tidak mencerminkan isi dokumen tersebut. 20Newsgroups *dataset* adalah kumpulan *e-mail* yang membahas topik-topik tertentu, sehingga cukup banyak *e-mail* yang terdapat dalam 20Newsgroups *dataset* memiliki *footer* yang berisi penjelasan mengenai pengirim *e-mail*.



Gambar 5.3 Hasil Pemilihan Fitur pada 20Newsgroups *Dataset*

Hasil akurasi percobaan klasifikasi 20Newsgroups *dataset* tanpa penghilangan *stopwords* secara keseluruhan lebih rendah apabila dibandingkan dengan percobaan dengan penghilangan *stopwords*. Akurasi terendah adalah 25,52% atau lebih rendah 9,01% dari hasil terendah pada percobaan dengan penghilangan *stopwords*, sedangkan hasil tertinggi mencapai 80,98% yang diperoleh pada saat penggunaan

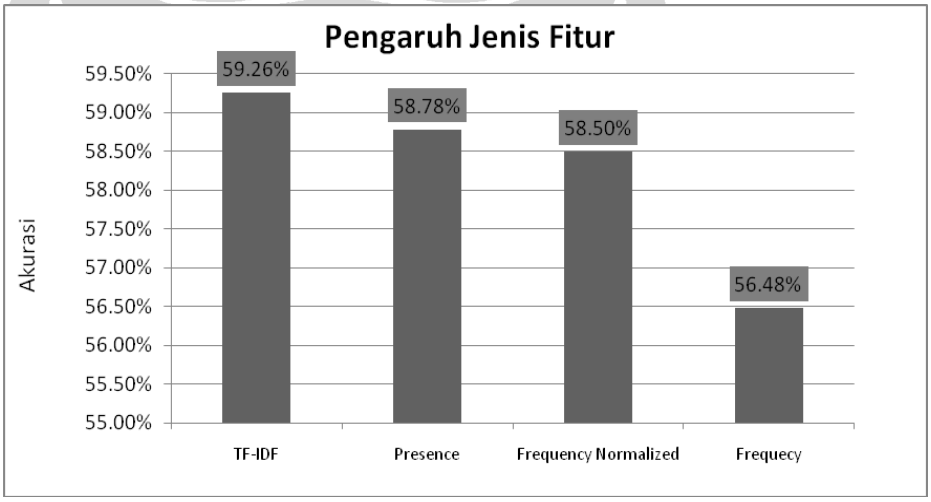
5000 fitur. Dari kedua hasil percobaan diatas pada penggunaan 5000 fitur diperoleh hasil akurasi tertinggi. Dengan hasil tersebut pada percobaan-percobaan dengan 20Newsgroups *dataset* selanjutnya akan digunakan jumlah fitur sebanyak 5000.

5.3 Hasil Klasifikasi dari Aspek Jenis Fitur

Percobaan selanjutnya dilakukan untuk melihat pengaruh jenis fitur yang digunakan terhadap hasil akurasi klasifikasi dokumen teks. Jenis fitur yang digunakan ada empat buah, yaitu *presence*, *frequency*, *frequency normalized*, dan pembobotan *tf-idf*. Hasil akurasi yang disajikan merupakan hasil akurasi klasifikasi dengan jenis fitur tertentu dengan merata-ratakan aspek jumlah kategori dan penggunaan *labeled documents* dan *unlabeled documents*.

5.3.1 Hasil Klasifikasi untuk Data Dokumen Hukum

Percobaan ini dilakukan untuk mengetahui pengaruh pemilihan jenis fitur terhadap klasifikasi dokumen teks. Jenis fitur yang digunakan adalah *presence*, *frequency*, *frequency normalized*, dan pembobotan *tf-idf*. Percobaan ini dilakukan dengan menggunakan jumlah fitur sebanyak 5000 buah dimana *stopwords* pada kumpulan fitur tersebut telah dihilangkan. Data *training* yang digunakan secara keseluruhan berjumlah 400 data dengan data *testing* berjumlah 73 buah.

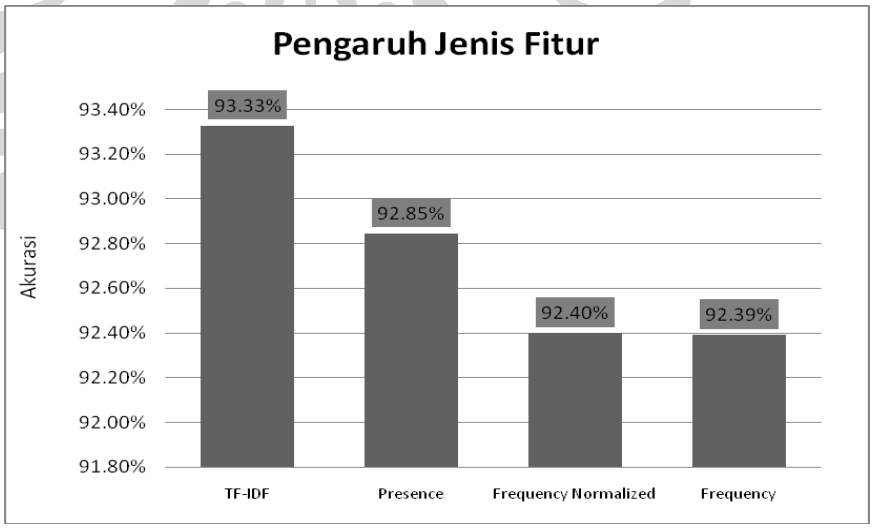


Gambar 5.4 Pengaruh Penggunaan Jenis Fitur pada Dokumen Hukum

Dari hasil yang diperoleh terlihat bahwa pembobotan *tf-idf* memberikan hasil akurasi terbaik dari semua jenis fitur yang digunakan. Pembobotan tf-idf memperoleh nilai akurasi tertinggi yaitu 59,26%, sedangkan penggunaan jenis fitur *presence* menghasilkan akurasi 58,78% dan hanya unggul tipis 0,28% dari penggunaan jenis fitur *frequency normalized*. Hasil terendah diperoleh pada penggunaan fitur *frequency* dengan nilai akurasi hanya 56,48%.

5.3.2 Hasil Klasifikasi untuk Data Artikel Media Massa

Percobaan kedua yang dilakukan untuk mengetahui pengaruh jenis fitur terhadap hasil akurasi klasifikasi dokumen teks adalah percobaan menggunakan data artikel media massa. Percobaan ini dilakukan dengan menggunakan total data *training* sebesar 1100 dokumen, dan data *testing* yang digunakan sebesar 140 dokumen. Hasil akurasi yang diperoleh merupakan hasil akurasi rata-rata pada percobaan dengan menggunakan jumlah kategori sebanyak tiga, empat, dan lima buah kategori.

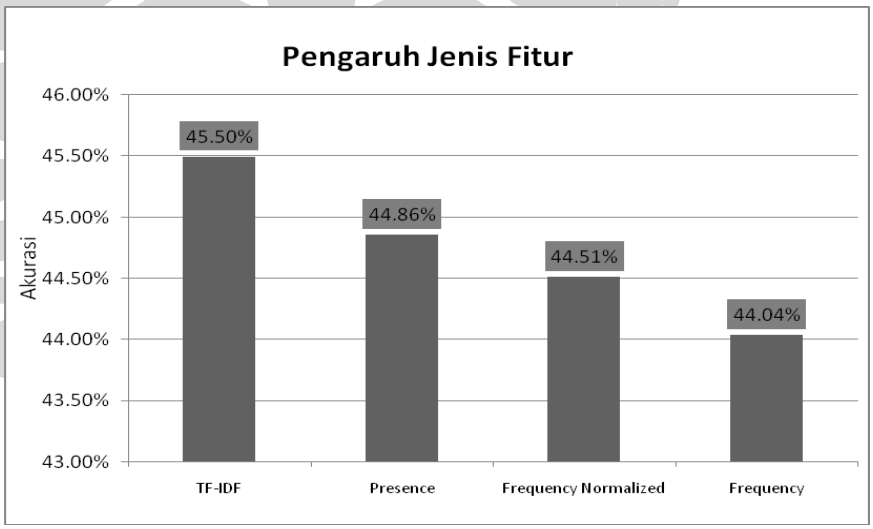


Gambar 5.5 Pengaruh Penggunaan Jenis Fitur pada Artikel Media Massa

Akurasi klasifikasi yang dihasilkan pada percobaan menggunakan data artikel media massa sangat tinggi, bahkan menembus angka 90%. Hasil yang diperoleh pada percobaan ini menunjukkan perbedaan yang cukup kecil sekitar 0,5% pada masing-masing jenis fitur yang digunakan. Pembobotan *tf-idf* menghasilkan akurasi tertinggi, sedangkan jenis fitur *frequency* menghasilkan akurasi terendah.

5.3.3 Hasil Klasifikasi untuk Data 20Newsgroups *Dataset*

Percobaan terakhir yang dilakukan untuk mengetahui pengaruh jenis fitur terhadap hasil akurasi klasifikasi dokumen teks adalah percobaan menggunakan data 20Newsgroups *dataset*. Percobaan ini dilakukan dengan menggunakan total data *training* sebesar 16000 dokumen, dan data *testing* yang digunakan sebesar 2098 dokumen. Hasil akurasi yang diperoleh merupakan hasil akurasi dengan merata-ratakan jumlah kategori, dan jumlah dokumen yang digunakan.



Gambar 5.6 Pengaruh Penggunaan Jenis Fitur pada 20Newsgroups *Dataset*

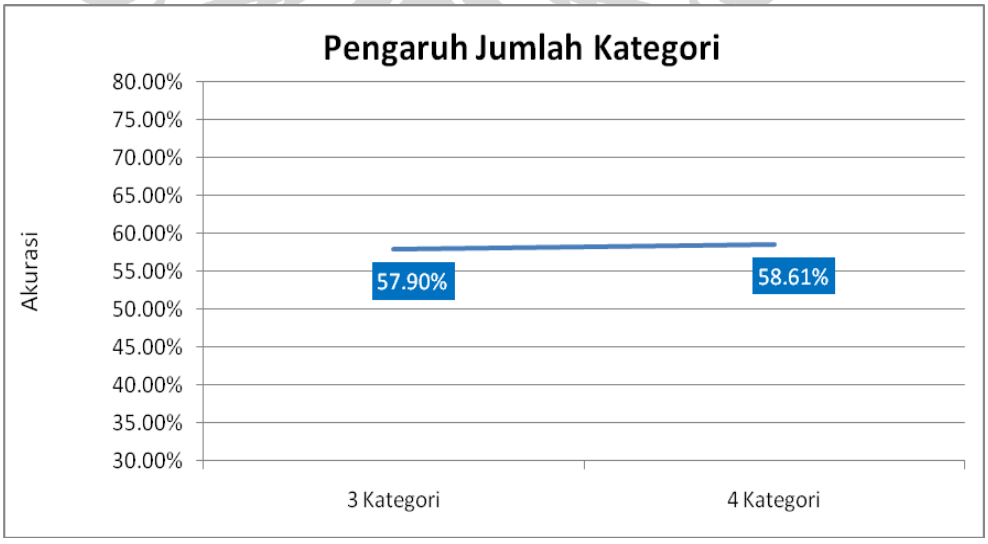
Perbedaan akurasi diantara keempat jenis fitur yang digunakan sangat kecil. Akurasi tertinggi diperoleh pada penggunaan jenis fitur dengan pembobotan *tf-idf*, sedangkan hasil terendah diperoleh pada penggunaan jenis fitur *frequency*. Hasil ini mirip dengan dua percobaan sebelumnya dimana pembobotan *tf-idf* memberikan hasil terbaik diikuti oleh jenis fitur *presence*, *frequency normalized*, dan *frequency*.

5.4 Hasil Klasifikasi dari Aspek Jumlah Kategori

Percobaan pada subbab ini bertujuan untuk mengetahui pengaruh jumlah kategori terhadap hasil akurasi klasifikasi dokumen teks. Pada bagian ini juga akan dibahas kesalahan-kesalahan klasifikasi yang terjadi. Pembahasan akan dibagi kedalam tiga bagian sesuai data yang digunakan. Hasil akurasi yang disajikan merupakan hasil akurasi klasifikasi dengan jumlah kategori tertentu dengan merata-ratakan aspek jenis fitur dan penggunaan *labeled documents* dan *unlabeled documents*.

5.4.1 Hasil Klasifikasi untuk Data Dokumen Hukum

Percobaan klasifikasi pada dokumen hukum ini dilakukan untuk mengetahui pengaruh jumlah kategori dengan membandingkan hasil akurasi klasifikasi dengan tiga buah kategori dan hasil klasifikasi dengan empat buah kategori. Jumlah total kategori yang terdapat pada data dokumen hukum ini ada empat yakni PP, Perpu, UU, dan UU Darurat. Kategori-kategori tersebut dibagi berdasarkan jenis dokumen, bukan berdasarkan jenis topik yang ada. Total dokumen yang digunakan untuk data *training* berjumlah 400 dokumen, sedangkan untuk data *testing* berjumlah 73 dokumen.



Gambar 5.7 Pengaruh Jumlah Kategori pada Dokumen Hukum

Hasil yang diperoleh pada percobaan ini menunjukkan peningkatan akurasi seiring penambahan jumlah kategori walaupun peningkatan yang terjadi tidak terlalu signifikan yaitu 0,71% dari 57,90% menjadi 58,61%. Hal ini berbeda dengan perkiraan awal bahwa dengan penambahan jumlah kategori akan menurunkan akurasi klasifikasi, namun yang terjadi pada percobaan ini justru sebaliknya. Penyebab utama terjadinya kesalahan perkiraan adalah penentuan kategori yang didasarkan atas jenis dokumen bukan berdasarkan topik.

Tabel 5.5 Kesalahan Klasifikasi pada Dokumen Hukum

	Perpu	UU	PP	UU Darurat
Perpu		2	3	2
UU	1		2	5
PP	4	3		2
UU Darurat	2	5	2	

Tabel diatas menunjukkan rata-rata kesalahan klasifikasi yang terjadi pada percobaan menggunakan empat buah kategori dengan jenis fitur *presence*, *frequency*, *frequency normalized*, dan pembobotan *tf-idf*. Kesalahan klasifikasi pada dokumen hukum cukup tinggi sekitar 41% dari total data *testing* yang digunakan. Dari 73 data *testing* yang digunakan rata-rata terdapat 30 dokumen yang salah diklasifikasikan.

Dari tabel di atas dapat dilihat bahwa terdapat dua kelompok dokumen yang meiliki tingkat kemiripan yang cukup tinggi, yakni dokumen-dokumen PP dan Perpu, serta dokumen-dokumen UU dan UU Darurat. Hal tersebut terlihat dari banyaknya jumlah dokumen PP yang salah diklasifikasikan ke dalam kategori Perpu dan sebaliknya, serta banyaknya jumlah dokumen UU yang salah diklasifikasikan ke dalam kategori UU Darurat begitu pula sebaliknya.

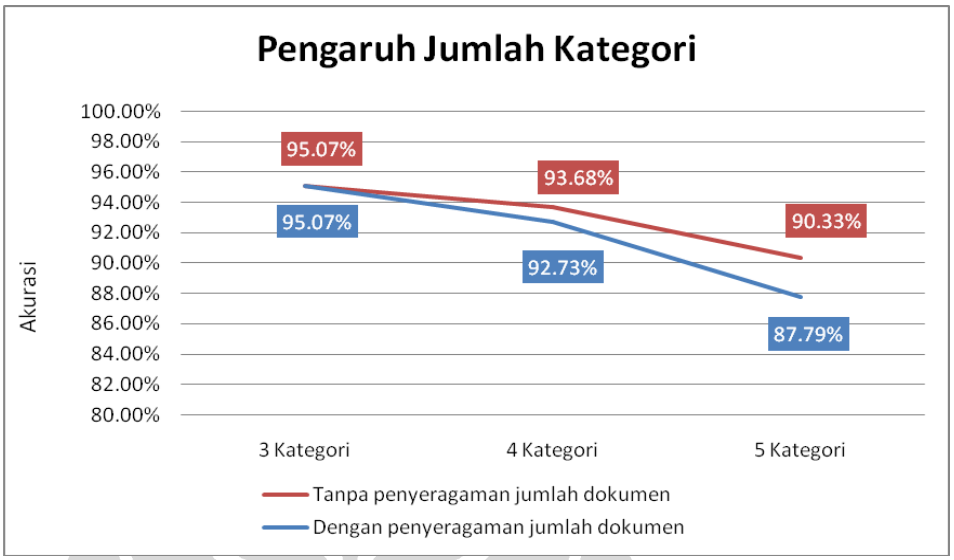
5.4.2 Hasil Klasifikasi untuk Data Artikel Media Massa

Percobaan selanjutnya yang dilakukan untuk mengetahui pengaruh jumlah kategori terhadap akurasi hasil klasifikasi adalah percobaan dengan menggunakan data artikel media massa. Jumlah total kategori yang terdapat pada artikel media massa ada lima

buah, yaitu keuangan, kesehatan, olahraga, properti, dan travel. Pada percobaan ini variabel jumlah kategori dibagi menjadi tiga, yaitu dengan tiga, empat, dan lima buah kategori.

Karena jumlah dokumen *training* maksimum pada masing-masing percobaan dengan tiga, empat, dan lima buah kategori berbeda-beda, hasil eksperimen yang ditampilkan akan dibagi menjadi dua bagian. Bagian pertama akan ditunjukkan hasil eksperimen tanpa penyeragaman jumlah dokumen *training* yang digunakan pada tiap kategori yaitu hasil eksperimen dengan merata-ratakan nilai akurasi untuk semua percobaan pada masing-masing konfigurasi penggunaan kategori dengan rincian: jumlah maksimum dokumen yang digunakan pada konfigurasi tiga kategori adalah 400 *labeled documents* dan 400 *unlabeled documents*, pada konfigurasi empat kategori adalah 500 *labeled documents* dan 500 *unlabeled documents*, pada konfigurasi lima kategori adalah 550 *labeled documents* dan 500 *unlabeled documents*. Bagian kedua akan ditunjukkan hasil eksperimen dengan penyeragaman jumlah dokumen *training* pada tiap kategori yaitu hasil eksperimen dengan merata-ratakan nilai akurasi dengan jumlah maksimum dokumen yang digunakan untuk setiap konfigurasi kategori adalah 400 *labeled documens* dan 400 *unlabeled documents*. Hal ini dilakukan untuk melihat pengaruh penambahan jumlah kategori ketika jumlah dokumen *training* yang digunakan disamakan. Dokumen *testing* yang digunakan berjumlah 140 dokumen.

Hasil yang diperoleh dari percobaan pada data artikel media massa tanpa penyeragaman jumlah dokumen *training* ini menunjukkan penurunan akurasi. Akurasi klasifikasi menunjukkan penurunan hampir 2% hingga 3% setiap terjadi penambahan satu buah kategori. Dari gambar 5.8 dapat terlihat penurunan dari 95,07% pada penggunaan tiga buah kategori menjadi 93,68% pada penggunaan empat buah kategori, kemudian pada penggunaan 5 buah kategori hasil akurasi kembali menurun menjadi 90,33%. Hasil ini sesuai dengan perkiraan bahwa penambahan jumlah kategori akan menurunkan akurasi klasifikasi dokumen teks.



Gambar 5.8 Pengaruh Jumlah Kategori pada Artikel Media Massa

Hasil yang diperoleh dari percobaan pada data artikel media massa dengan penyeragaman jumlah dokumen *training* ini menunjukkan penurunan akurasi. Dari gambar 5.8 dapat terlihat penurunan dari 95,07% pada penggunaan tiga buah kategori menjadi 93,73% pada penggunaan empat buah kategori, kemudian pada penggunaan 5 buah kategori hasil akurasi kembali menurun menjadi 87,79%.

Tabel 5.6 Kesalahan Klasifikasi pada Artikel Media Massa

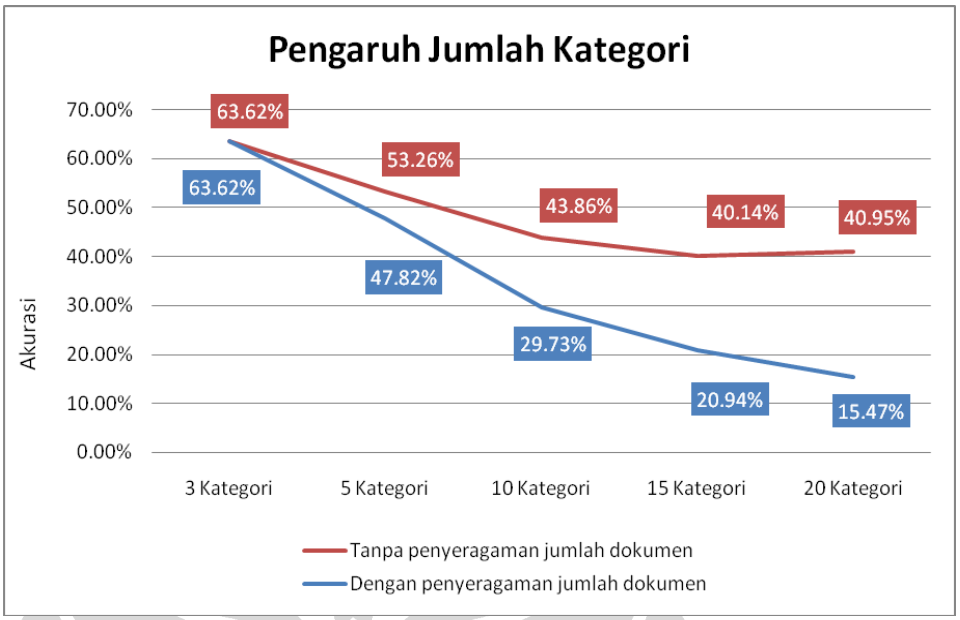
	Ekonomi	Kesehatan	Olahraga	Properti	Travel
Ekonomi		2	0	1	1
Kesehatan	2		1	0	1
Olahraga	0	1		0	1
Properti	1	1	0		1
Travel	0	1	0	0	

Tabel 5.6 diatas menunjukkan rata-rata kesalahan klasifikasi yang terjadi pada penggunaan lima buah kategori dengan jenis fitur *presence*, *frequency*, *frequency normalized*, dan pembobotan *tf-idf*. Tingkat kesalahan pada percobaan menggunakan data artikel media massa ini cukup rendah. Dari 140 data *testing*, rata-rata hanya terjadi 14 kali kesalahan klasifikasi.

5.4.3 Hasil Klasifikasi untuk Data 20Newsgroups *Dataset*

Percobaan selanjutnya yang dilakukan untuk mengetahui pengaruh jumlah kategori terhadap akurasi hasil klasifikasi adalah percobaan dengan menggunakan data 20Newsgroups *dataset*. Pada percobaan ini variabel jumlah kategori dibagi menjadi lima, yaitu dengan 3, 5, 10, 15, dan 20 buah kategori.

Karena jumlah dokumen *training* maksimum pada masing-masing percobaan dengan 3, 5, 10, 15, dan 20 buah kategori berbeda-beda, hasil eksperimen yang ditampilkan akan dibagi menjadi dua bagian. Bagian pertama akan ditunjukkan hasil eksperimen tanpa penyeragaman jumlah dokumen *training* yang digunakan pada tiap kategori yaitu hasil eksperimen dengan merata-ratakan nilai akurasi untuk semua percobaan pada masing-masing konfigurasi penggunaan kategori dengan rincian: jumlah maksimum dokumen yang digunakan pada konfigurasi tiga kategori adalah 1200 *labeled documents* dan 1000 *unlabeled documents*, pada konfigurasi lima kategori adalah 2000 *labeled documents* dan 2000 *unlabeled documents*, pada konfigurasi 10 kategori adalah 4000 *labeled documents* dan 4000 *unlabeled documents*, pada konfigurasi 15 kategori adalah 6000 *labeled documents* dan 6000 *unlabeled documents*, pada konfigurasi 20 kategori adalah 8000 *labeled documents* dan 8000 *unlabeled documents*. Bagian kedua akan ditunjukkan hasil eksperimen dengan penyeragaman jumlah dokumen *training* pada tiap kategori yaitu hasil eksperimen dengan merata-ratakan nilai akurasi dengan jumlah maksimum dokumen yang digunakan untuk setiap konfigurasi kategori adalah 1200 *labeled documens* dan 1000 *unlabeled documents*. Hal ini dilakukan untuk melihat pengaruh penambahan jumlah kategori ketika jumlah dokumen *training* yang digunakan disamakan. Dokumen *testing* yang digunakan berjumlah 2098 dokumen.



Gambar 5.9 Pengaruh Jumlah Kategori pada 20Newsgroups *Dataset*

Hasil yang diperoleh dari percobaan pada data 20Newsgroups *dataset* tanpa penyeragaman jumlah dokumen *training* ini menunjukkan penurunan akurasi. Hal ini mirip dengan hasil yang diperoleh dari percobaan dengan artikel media massa. Akurasi klasifikasi terus menurun dari 63,62% pada percobaan dengan 3 kategori hingga 40,95% pada percobaan dengan 20 kategori.

Hasil yang diperoleh dari percobaan pada data 20Newsgroups *dataset* dengan penyeragaman jumlah dokumen *training* juga menunjukkan penurunan akurasi. Akurasi klasifikasi terus menurun dari 63,62% pada percobaan dengan 3 kategori hingga 15,47% pada percobaan dengan 20 kategori.

Tabel 5.7 Kesalahan Klasifikasi dengan Metode Naïve Bayes pada 20Newsgroups *Dataset*

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1		3	4	39	5	8	1	1	1	1	2	0	1	2	2	2	1	1	28	1
2	30		76	4	3	1	2	1	1	1	1	1	1	1	1	3	1	1	1	1
3	14	3		2	8	1	7	0	0	0	1	0	0	1	1	1	1	1	1	1
4	11	6	31		3	4	3	1	1	0	2	0	1	0	1	1	2	1	1	2
5	22	14	12	3		0	0	0	2	0	0	0	2	1	2	1	1	1	1	3
6	8	1	13	1	0		6	0	1	2	0	0	2	1	1	2	1	0	2	2
7	1	1	2	0	1	3		24	10	7	0	0	2	1	1	1	1	0	1	0
8	1	1	0	0	1	1	20		13	14	0	1	1	0	2	1	1	0	3	0
9	2	1	2	0	0	4	13	32		15	1	0	2	0	3	2	0	0	1	0
10	1	2	1	1	1	0	0	9	11		0	0	2	0	2	1	0	0	0	1
11	4	1	1	1	0	0	1	1	1	0		4	11	6	1	0	0	1	1	0
12	2	2	13	3	2	4	3	1	1	0	17		0	14	1	0	0	1	1	0
13	7	1	3	2	1	2	7	1	1	1	9	8		0	0	0	1	0	2	0
14	7	1	3	2	1	1	6	0	1	2	18	2	4		0	3	0	1	12	0
15	1	1	1	1	1	1	3	1	1	1	1	2	1	0		12	12	3	2	0
16	1	2	0	2	0	1	0	4	2	1	1	1	1	0	14		21	1	0	0
17	0	1	1	1	1	1	1	1	2	2	1	1	2	8	15	11		0	0	0
18	3	1	0	1	1	1	2	1	1	1	1	1	1	0	0	0	0		55	25
19	0	2	0	1	1	0	0	2	1	1	1	0	0	0	1	2	1	41		18
20	0	1	0	0	0	0	1	0	1	1	1	1	0	0	0	1	7	15	60	

Keterangan kategori:

1. comp.graphics
2. comp.os.ms-windows.misc
3. comp.sys.ibm.pc.hardware
4. comp.sys.mac.hardware
5. comp.windows.x
6. misc.forsale
7. rec.autos
8. rec.motorcycles
9. rec.sport.baseball
10. rec.sport.hockey
11. sci.crypt
12. sci.electronics
13. sci.med
14. sci.space
15. talk.politics.guns
16. talk.politics.mideast
17. talk.politics.misc
18. alt.atheism
19. soc.religion.christian
20. talk.religion.misc

Tabel 5.7 diatas menunjukkan rata-rata kesalahan klasifikasi yang terjadi pada penggunaan 20 buah kategori dengan jenis fitur *presence*, *frequency*, *frequency*

normalized, dan pembobotan *tf-idf*. Dari tabel diatas dapat dilihat bahwa pada 20Newsgroups *dataset* terdapat beberapa kelompok kategori yang lebih umum, yaitu: *computer, recreation, science, politics,* dan *religion.* Hal tersebut terlihat jelas pada tabel 5.7 dimana kesalahan klasifikasi dari sebuah kategori ke kategori lain yang memiliki kemiripan tinggi jumlahnya sangat besar.

5.5 Pengaruh *Unlabeled Documents* terhadap Akurasi Klasifikasi

Aspek terakhir yang ingin diamati adalah pengaruh penggunaan *unlabeled documents* terhadap hasil akurasi klasifikasi dokumen teks. Aspek ini pula yang menjadi perhatian utama pada tugas akhir ini. Percobaan ini memvariasikan penggunaan jumlah *labeled documents* dan jumlah *unlabeled documents*. Pada data dokumen hukum dan data artikel media massa jumlah dokumen yang digunakan tidak terlalu besar, yakni hanya berjumlah 400 dokumen untuk data dokumen hukum dan 1050 dokumen untuk data artikel media massa. Jumlah data yang besar akan digunakan pada percobaan dengan data 20Newsgroups *dataset,* yakni mencapai jumlah dokumen sebesar 16000 buah.

5.5.1 Hasil Klasifikasi untuk Data Dokumen Hukum

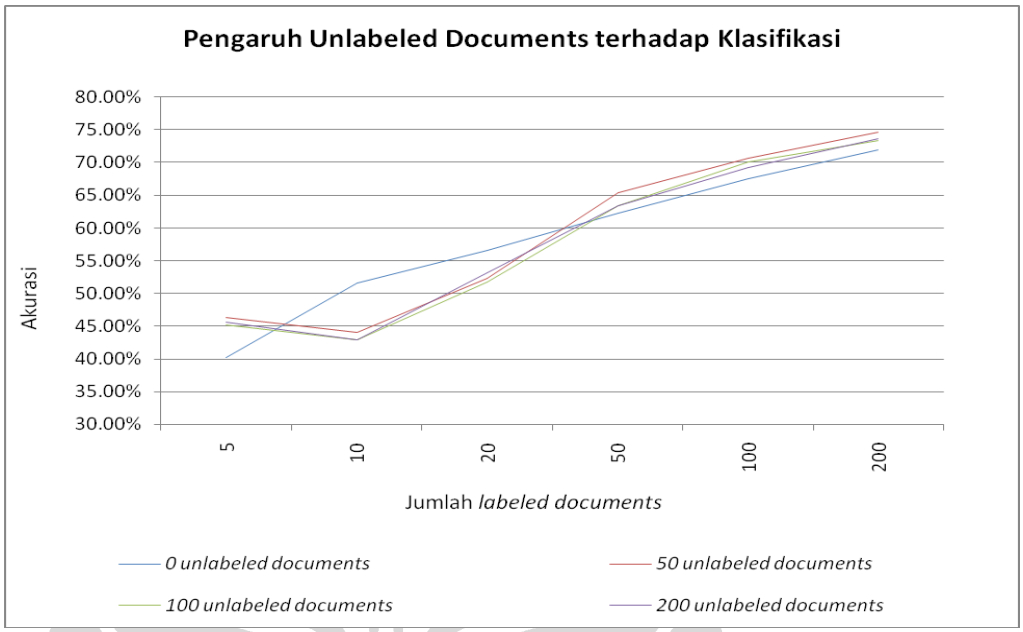
Percobaan pertama untuk mengetahui pengaruh *unlabeled documents* terhadap hasil klasifikasi dokumen teks dilakukan pada dokumen hukum. Percobaan ini menggunakan jumlah fitur 5000, total data *training* sebanyak 400 dokumen, dan jumlah data *testing* sebanyak 78 dokumen. Hasil akurasi yang diperoleh merupakan rata-rata dari akurasi pada percobaan dengan menggunakan tiga dan empat kategori serta penggunaan fitur *presence, frequency, frequency normalized* dan pembobotan *tf-idf.*

Tabel 5.8 Pengaruh *Unlabeled Documents* pada Dokumen Hukum

Labeled documents / *Unlabeled documents*	5	10	20	50	100	200
0	40.21%	51.63%	56.59%	62.31%	67.45%	**71.88%**
50	46.35%	44.03%	52.32%	65.36%	70.68%	**74.57%**
100	45.17%	42.91%	51.72%	63.39%	70.15%	**73.33%**
200	45.59%	42.96%	53.22%	63.42%	69.29%	**73.58%**

Hasil akurasi tertinggi 74,57% diperoleh pada penggunaan 200 *labeled documents* dan 50 *unlabeled documents*. Dari tabel 5.8 terlihat bahwa akurasi tertinggi tanpa menggunakan *unlabeled document* hanya mencapai 71,88% yang dicapai pada saat penggunaan 200 *labeled documents*. Hasil akurasi tanpa menggunakan *unlabeled documents* mengalami peningkatan seiring bertambahnya jumlah *labeled documents* yang digunakan. Hal serupa juga terjadi pada percobaan dengan menggunakan *unlabeled documents* berjumlah 50, 100, dan 200 buah dokumen, hasil akurasi yang diperoleh cenderung meningkat seiring peningkatan jumlah *labeled documents* yang digunakan.

Penggunaan *n* buah *labeled documents* dengan penambahan *n* buah *unlabeled documents* memberikan rata-rata peningkatan akurasi sekitar 2,45% dari hasil klasifikasi menggunakan *n* buah *labeled documents* tanpa menggunakan *unlabeled documents*. Hasil ini diperoleh dari peningkatan 3,05% pada penggunaan 50 *labeled documents* dan 50 *unlabeled documents*, 2,6% pada penggunaan penggunaan 100 *labeled documents* dan 100 *unlabeled documents*, 1,7% pada penggunaan penggunaan 200 *labeled documents* dan 200 *unlabeled documents*.



Gambar 5.10 Pengaruh *Unlabeled Documents* pada Dokumen Hukum

Pemanfaatan *unlabeled documents* dengan penggunaan jumlah *labeled documents* kecil menyebabkan hasil klasifikasi yang diperoleh tidak stabil. Hal tersebut dapat dilihat pada gambar 5.10, dimana hasil yang diperoleh pada percobaan menggunakan *unlabeled documents* justru lebih rendah dibandingkan percobaan tanpa menggunakan *unlabeled documents* pada penggunaan 10 dan 20 *labeled documents*, namun setelah penggunaan 50 *labeled documents*, proses klasifikasi dengan memanfaatkan *unlabeled documents* mulai menampakkan hasil yang stabil dan menghasilkan akurasi yang lebih baik bila dibandingkan dengan proses klasifikasi tanpa memanfaatkan *unlabeled documents*.

5.5.2 Hasil Klasifikasi untuk Data Artikel Media Massa

Percobaan kedua untuk mengetahui pengaruh *unlabeled documents* dilakukan dengan menggunakan artikel media massa. Percobaan ini dilakukan dengan total jumlah data *training* sebanyak 1050 dokumen dan data *testing* berjumlah 140 dokumen. Jumlah fitur yang digunakan sesuai dengan hasil percobaan pemilihan fitur yaitu sebanyak 5000 fitur. Hasil akurasi yang diperoleh merupakan hasil akurasi rata-rata klasifikasi

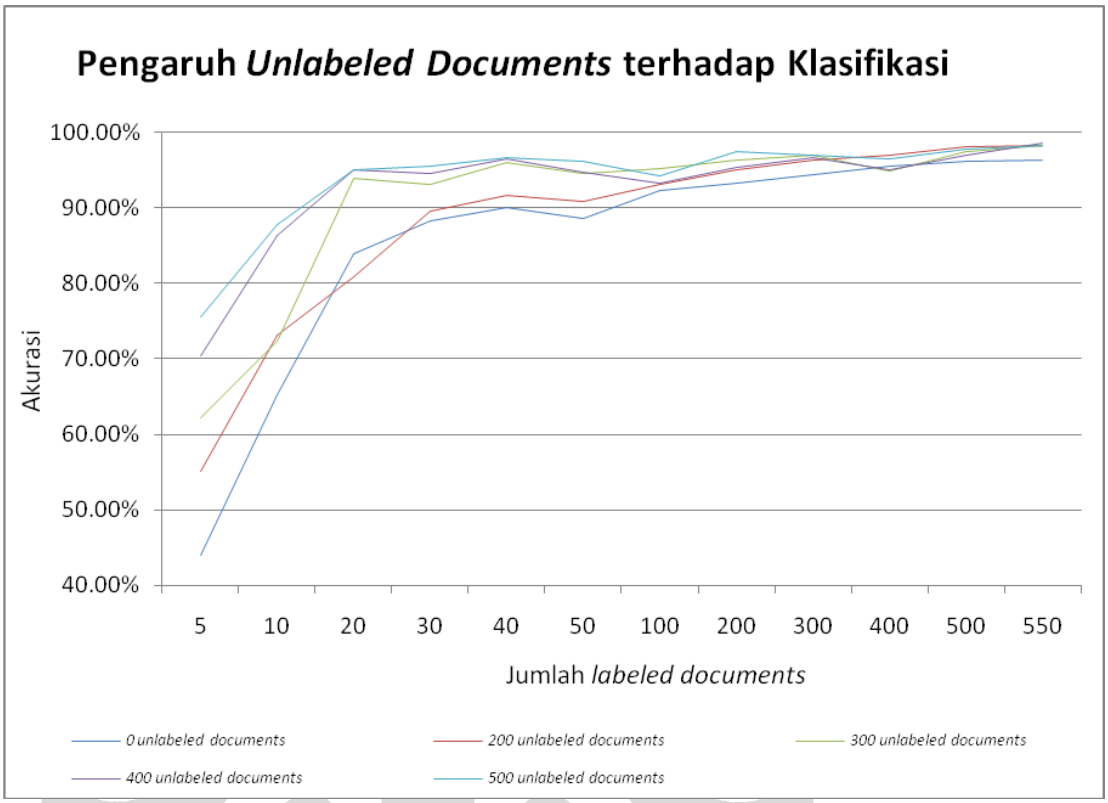
dengan menggunakan lima buah kategori dengan jenis fitur *presence*, *frequency*, *frequency normalized* dan pembobotan *tf-idf*.

Tabel 5.9 Pengaruh *Unlabeled Documents* pada Artikel Media Massa

Labeled documents / *Unlabeled documents*	5	10	20	30	40	50	100	200	300	400	500	550
0	43.92%	65.23%	83.92%	88.36%	90.06%	88.54%	92.34%	93.29%	94.35%	95.58%	96.23%	**96.29%**
200	55.04%	73.18%	80.91%	89.53%	91.74%	90.90%	93.05%	95.09%	96.36%	96.92%	98.10%	**98.23%**
300	62.22%	72.32%	93.91%	93.11%	96.06%	94.61%	95.28%	96.30%	97.04%	95.88%	97.51%	**98.33%**
400	70.38%	86.34%	95.06%	94.54%	96.43%	94.72%	93.22%	95.41%	96.60%	96.00%	97.06%	**98.54%**
500	75.50%	87.78%	95.07%	95.50%	96.70%	96.16%	94.25%	97.45%	96.94%	96.49%	97.74%	**98.34%**

Hasil tertinggi yang diperoleh tanpa menggunakan *unlabeled documents* dicapai pada saat penggunaan 550 *labeled documents*. Hal yang sama juga terjadi pada saat penggunaan *unlabeled document* sebanyak 200, 300, 400, dan 500 buah, hasil tertinggi diperoleh dicapai pada penggunaan 550 *labeled documents* dengan nilai akurasi tertinggi secara keseluruhan mencapai nilai 98,54% dengan menggunakan 550 *labeled documents* dan 400 *unlabeled documents*.

Penggunaan *n* buah *labeled documents* dengan penambahan *n* buah *unlabeled documents* memberikan rata-rata peningkatan akurasi sekitar 1,6% dari hasil klasifikasi menggunakan *n* buah *labeled documents* tanpa menggunakan *unlabeled documents*. Hasil ini diperoleh dari peningkatan 1,8% pada penggunaan 200 *labeled documents* dan 200 *unlabeled documents*, 2,69% pada penggunaan penggunaan 300 *labeled documents* dan 300 *unlabeled documents*, 0,42% pada penggunaan penggunaan 400 *labeled documents* dan 400 *unlabeled documents*, 1,51% pada penggunaan penggunaan 500 *labeled documents* dan 500 *unlabeled documents*.



Gambar 5.11 Pengaruh *Unlabeled Documents* pada Artikel Media Massa

Pada percobaan ini terlihat bagaimana penggunaan *unlabeled documents* secara konsisten memberikan manfaat yang cukup berarti dalam meningkatkan akurasi klasifikasi dokumen teks. Dari gambar 5.11 terlihat bahwa percobaan dengan menggunakan *unlabeled documents* hampir selalu menghasilkan akurasi yang lebih tinggi dari percobaan tanpa menggunakan *unlabeled documents*. Penggunaan jumlah *unlabeled documents* tidak terlalu memberikan dampak yang cukup besar apabila jumlah *labeled documents* yang digunakan telah memberikan akurasi yang tinggi. Pada percobaan ini perbedaan hasil akurasi dengan menggunakan *unlabeled documents* tidak memberikan perbedaan yang berarti setelah penggunaan *labeled documents* sebanyak 200 buah.

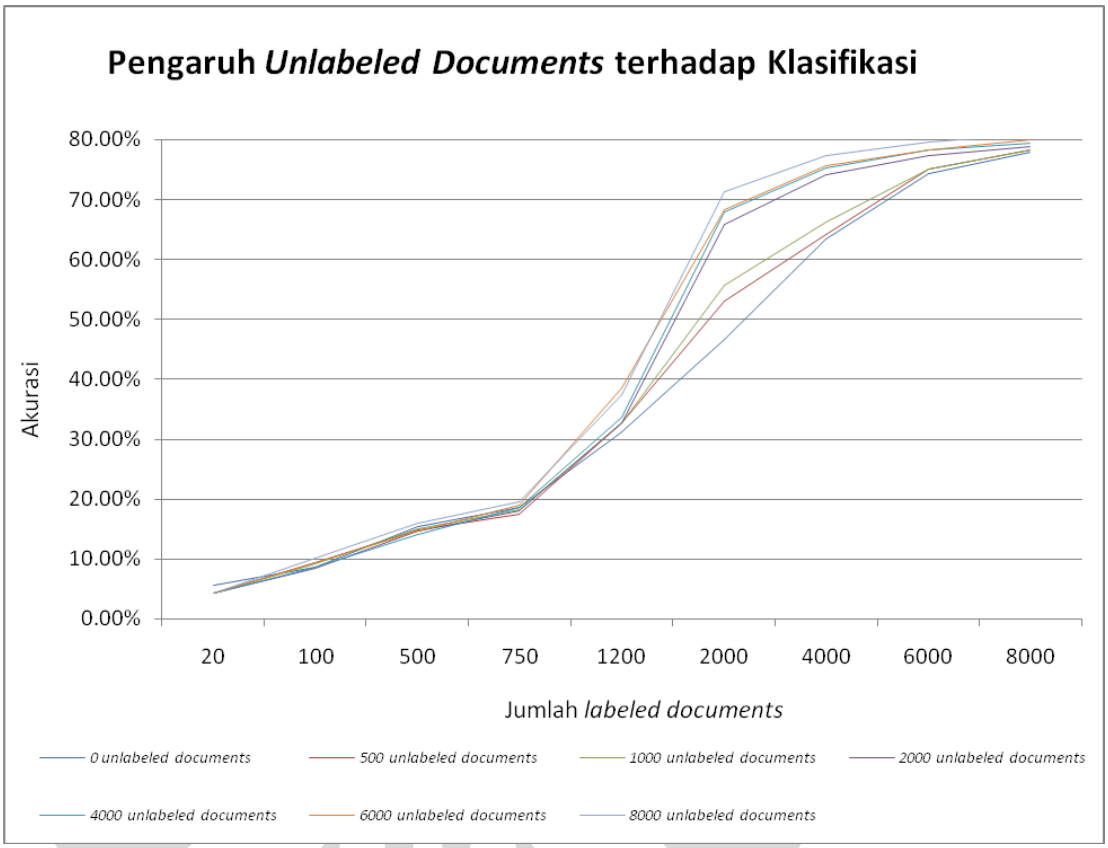
5.5.3 Hasil Klasifikasi untuk Data 20Newsgroups *Dataset*

Percobaan terakhir untuk mengetahui pengaruh *unlabeled documents* dilakukan dengan menggunakan data 20Newsgroups *dataset*. Percobaan ini dilakukan dengan total jumlah data *training* sebanyak 16000 dokumen dan data *testing* berjumlah 2098 dokumen. Jumlah fitur yang digunakan sesuai dengan hasil percobaan pemilihan fitur yaitu sebanyak 5000 fitur. Hasil akurasi yang diperoleh merupakan hasil akurasi dengan merata-ratakan variabel jenis fitur yaitu *presence*, *frequency*, *frequency normalized* dan pembobotan *tf-idf*, dan jumlah kategori yang digunakan adalah 20 buah.

Tabel 5.10 Pengaruh *Unlabeled Documents* pada 20Newsgroups *Dataset*

Labeled documents / *Unlabeled documents*	20	100	500	750	1200	2000	4000	6000	8000
0	5.56%	8.61%	15.36%	18.61%	31.16%	46.69%	63.47%	74.30%	**77.90%**
500	4.25%	9.36%	14.75%	17.39%	32.64%	53.14%	64.12%	75.16%	**78.25%**
1000	4.27%	9.11%	15.00%	18.20%	32.76%	55.75%	66.17%	75.08%	**78.36%**
2000	4.24%	8.45%	14.64%	18.05%	32.77%	65.91%	74.22%	77.30%	**78.88%**
4000	4.22%	8.66%	14.10%	18.56%	33.64%	67.98%	75.19%	78.23%	**79.49%**
6000	4.24%	9.42%	14.65%	18.98%	38.56%	68.24%	75.67%	78.28%	**79.97%**
8000	4.27%	10.20%	15.93%	19.51%	37.49%	71.37%	77.34%	79.69%	**81.00%**

Hasil akurasi tertinggi yang diperoleh pada percobaan ini mencapai nilai 81,00% dengan menggunakan 8000 *labeled documents* dan 8000 *unlabeled documents*. Pada tabel diatas terlihat bahwa penambahan *unlabeled documents* justru akan menurunkan hasil akurasi klasifikasi pada penggunaan jumlah *labeled documents* yang terlalu kecil. Hal ini dapat disebabkan buruknya hasil perkiraan kategori untuk semua *unlabeled documents* yang mengakibatkan turunnya hasil klasifikasi secara keseluruhan. Namun, dengan penggunaan jumlah *labeled documents* yang cukup besar, penggunaan *unlabeled documents* memberikan dampak yang cukup signifikan.



Gambar 5.12 Pengaruh *Unlabeled Documents* pada 20Newsgroups *Dataset*

Penggunaan *unlabeled documents* pada percobaan dengan jumlah *labeled documents* kecil menyebabkan hasil yang diperoleh tidak stabil, bahkan cenderung menurun. Pada grafik dan tabel diatas terlihat bagaimana penggunaan *unlabeled documents* baru memberikan dampak terhadap hasil akurasi setelah penggunaan *labeled documents* melebihi 1200 buah.

Penggunaan *n* buah *labeled documents* dengan penambahan *n* buah *unlabeled documents* memberikan rata-rata peningkatan akurasi sekitar 9,5% dari hasil klasifikasi menggunakan *n* buah *labeled documents* tanpa menggunakan *unlabeled documents*. Hasil ini diperoleh dari peningkatan 19.22% pada penggunaan 2000 *labeled documents* dan 2000 *unlabeled documents*, 11.72% pada penggunaan penggunaan 4000 *labeled documents* dan 4000 *unlabeled documents*, 3.98% pada

penggunaan penggunaan 6000 *labeled documents* dan 6000 *unlabeled documents*, 3.10% pada penggunaan penggunaan 8000 *labeled documents* dan 8000 *unlabeled documents.*

5.6 Rangkuman Hasil Percobaan

Pembahasan secara mendetail mengenai hasil klasifikasi dokumen teks menggunakan *machine learning* telah dilakukan pada subbab 5.2 hingga 5.5. Subbab ini akan memberikan rangkuman singkat dari hasil-hasil yang telah diperoleh pada keempat subbab tersebut. Beberapa hal yang dapat dirangkum, antara lain:

1. Dari aspek metode klasifikasi yang digunakan, secara umum Expectation Maximization memberikan hasil akurasi yang lebih baik dari pada Naïve Bayes. Hasil ini ditunjukkan dengan dampak positif penggunaan *unlabeled documents* pada klasifikasi dokumen teks yang meningkatkan akurasi hasil klasifikasi. Pada percobaan menggunakan dokumen hukum rata-rata peningkatan akurasi yang terjadi sekitar 2,45% (lihat subbab 5.5.1), pada percobaan menggunakan artikel media massa rata-rata peningkatan yang terjadi sekitar 1,8% (lihat sibbab 5.5.2), dan pada percobaan menggunakan 20Newsgroups *dataset* rata-rata peningkatan akurasi yang terjadi sekitar 9,5% (lihat subbab 5.5.3).

2. Penggunaan *unlabeled documents* memberikan manfaat yang cukup berarti bagi peningkatan akurasi hasil klasifikasi. Namun penggunaan *unlabeled documents* ini harus didukung oleh penggunaan *labeled documents* dalam jumlah yang tepat. Penggunaan *unlabeled documents* akan bermanfaat apabila jumlah *labeled documents* yang digunakan telah melewati batas tertentu. Penggunaan jumlah *labeled documents* yang terlalu sedikit justru akan menurunkan hasil klasifikasi dokumen secara keseluruhan. Pada percobaan menggunakan artikel media massa (lihat subbab 5.5.2), pengaruh penggunaan *unlabeled documents* telah terlihat pada penggunaan lima buah *labeled documents*. Hasil positif tersebut dipengaruhi oleh kinerja *initial* (Naïve

Bayes) *classifier* yang dapat memberikan hasil akurasi diatas 40% dengan memanfaatkan lima buah *labeled documents*. Pada percobaan menggunakan dokumen hukum (lihat subbab 5.5.1), penggunaan *unlabeled documents* baru memberikan dampak positif saat jumlah *labeled documents* yang digunakan mencapai 50 buah. Hal serupa juga terjadi pada percobaan menggunakan 20Newsgroups *dataset* (lihat subbab 5.5.3). Pada percobaan dengan menggunaan total 20 kategori tersebut, penggunaan *unlabeled documents* baru memberikan manfaat saat jumlah *labeled documents* yang digunakan mencapai 1200 buah. Dari hasil tersebut dapat ditarik kesimpulan bahwa diperlukan sekitar 30 hingga 60 *labeled documents* tiap kategorinya untuk membangun *initial classifier* yang dapat memanfaatkan *unlabeled documents* secara maksimal.

3. Pembobotan *tf-idf* memberikan nilai akurasi tertinggi dibandingkan tiga jenis fitur lain yang digunakan. Pembobotan *tf-idf* secara konsisten menunjukkan dominasinya pada setiap metode yang digunakan, pembobotan *tf-idf* selalu memberikan hasil yang terbaik. Pada percobaan menggunakan artikel media massa (lihat subbab 5.3.2), pembobotan *tf-idf* memberikan nilai akurasi rata-rata tertinggi yakni 93.33%, sedangkan pada percobaan menggunakan dokumen hukum dan 20Newsgroup *dataset* (lihat subbab 5.3.1 dan 5.3.3), pembobotan *tf-idf* mencapai nilai akurasi 59,26% dan 45,50%.

4. Penambahan jumlah kategori akan cenderung menurunkan nilai akurasi. Pada percobaan menggunakan artikel media massa tanpa penyeragaman jumlah dokumen *training* (lihat subbab 5.4.2) akurasi menurun dari 95,07% pada penggunaan 3 buah kategori menjadi 90,33% pada penggunaan 5 buah kategori, sedangkan pada percobaan menggunakan artikel media massa dengan penyeragaman jumlah dokumen *training* (lihat subbab 5.4.2) akurasi menurun dari 92,73% pada penggunaan 3 buah kategori menjadi 87,79% pada penggunaan 5 buah kategori. Hal serupa juga terjadi pada percobaan menggunakan 20Newsgroups *dataset* (lihat subbab 5.4.1), pada percobaan

tanpa penyeragaman jumlah dokumen *training* penurunan akurasi terjadi dari 63,62% pada penggunaan 3 buah kategori menjadi 40,95% pada penggunaan 20 kategori, sedangkan pada percobaan dengan penyeragaman jumlah dokumen *training* penurunan akurasi terjadi dari 63,62% pada penggunaan 3 buah kategori menjadi 15,47% pada penggunaan 20 kategori. Pada data dokumen hukum terjadi sedikit perbedaan (lihat subbab 5.4.1), nilai akurasi justru meningkat seiring penambahan jumlah kategori yaitu dari 57,90% saat menggunakan tiga buah kategori naik menjadi 58,61% saat penggunaan empat kategori.

5. Penghilangan *stopwords* dari daftar fitur yang ada dapat meningkatkan nilai akurasi klasifikasi dokumen. Hal ini terlihat pada ketiga kumpulan dokumen yang digunakan. Pada percobaan dengan tahapan penghilangan *stopwords*, akurasi tertinggi yang diperoleh adalah: 82,17% pada percobaan dengan 20Newsgroups *dataset* (lihat subbab 5.2.3), 74,67% pada percobaan dengan dokumen hukum (lihat subbab 5.2.1), dan 97,86% pada percobaan dengan artikel media massa (lihat subbab 5.2.2). Akurasi tertinggi yang diperoleh pada percobaan tanpa tahapan penghilangan *stopwords* adalah: 80,98% pada percobaan dengan 20Newsgroups *dataset* (lihat subbab 5.2.3), 72,22% pada percobaan dengan dokumen hukum (lihat subbab 5.2.1), dan 97,06% pada percobaan dengan artikel media massa (lihat subbab 5.2.2).