BAB 2 LANDASAN TEORI

Bab ini menjelaskan tentang landasan teori dan metode yang digunakan untuk melakukan klasifikasi dokumen pada penelitian ini. Pembahasan dimulai dari perolehan informasi, klasifikasi dokumen, dan metode yang digunakan untuk melakukan klasifikasi dokumen yaitu Naïve Bayes dan ontologi. Subbab berikutnya menjelaskan tentang persiapan dokumen, pembobotan kata, evaluasi, dan penelitian dalam bidang klasifikasi dokumen.

2.1 Perolehan Informasi

Istilah perolehan informasi memiliki pengertian yang sangat luas, sehingga banyak pakar mendefinisikan istilah perolehan informasi dari berbagai sudut pandang. Baeza-Yates dan rekannya [BYRN99] memberikan definisi tentang perolehan informasi, yaitu "sebuah cabang ilmu dari ilmu komputer yang mempelajari teknik-teknik untuk memperoleh informasi (bukan data) yang relevan berdasarkan kueri yang dimasukkan oleh pencari informasi". Christopher D. Manning dan rekannya [MRS08] memberikan definisi tentang perolehan informasi, yaitu "pencarian informasi, biasanya berupa dokumen, dari sesuatu yang tidak terstruktur, biasanya berupa teks, dalam suatu koleksi yang dapat memenuhi kebutuhan informasi yang diinginkan".

Perolehan informasi berbeda dengan perolehan data. Perolehan informasi merujuk pada representasi, penyimpanan, pengorganisasian sampai ke pengaksesan informasi [BYRN99]. Representasi dan pengorganisasian informasi harus memudahkan pencari informasi dalam mengakses informasi yang terdapat pada koleksi. Sementara itu, perolehan data memiliki lingkup yang lebih sempit. Perolehan data, dalam konteks sistem perolehan informasi, merujuk pada cara untuk menentukan atau mencocokkan antara kata-kata yang terkandung di sebuah dokumen dengan kata-kata yang digunakan seseorang dalam melakukan pencarian informasi [BYRN99].

Informasi dapat berupa teks, gambar, suara, video dan obyek multimedia lainnya. Informasi dalam bentuk teks merupakan fokus utama dalam penelitian Tugas Akhir ini. Informasi merupakan sesuatu yang tidak dapat didefinisikan secara tepat. Informasi berhubungan dengan bahasa alami yang biasanya tidak terstruktur dan secara semantik dapat memiliki makna ganda atau ambigu. Masalah yang muncul kemudian adalah bagaimana caranya untuk memperoleh informasi yang relevan di antara informasi lain dalam suatu koleksi dokumen. Hal inilah yang kemudian mendorong banyaknya penelitian tentang perolehan informasi khususnya informasi dalam bentuk teks. Untuk menanggulangi masalah ini dibutuhkan suatu sistem yang dapat memudahkan pencari informasi untuk mendapatkan informasi yang diinginkan dari suatu koleksi dokumen. Sistem ini kemudian dikenal dengan nama sistem perolehan informasi. Sistem perolehan informasi merupakan sistem yang diharapkan dapat memberikan sebanyak-banyaknya informasi dan serelevan mungkin terhadap kebutuhan pengguna sistem tersebut.

Sebuah sistem perolehan informasi memiliki beberapa operasi dasar, seperti pembuatan indeks koleksi dokumen dan pencarian dokumen berdasarkan kueri yang dimasukkan oleh pencari informasi. Kueri tersebut kemudian dicocokkan dengan koleksi dokumen yang ada di dalam *database*. Dokumen yang ditampilkan berdasarkan kueri masukan belum terurut dari dokumen yang paling relevan sampai yang tidak relevan sehingga dokumen tersebut perlu dilakukan klasifikasi. Klasifikasi dokumen merupakan topik utama yang akan dijelaskan pada penelitian Tugas Akhir ini.

2.2 Klasifikasi Dokumen

Klasifikasi merupakan proses mengidentifikasi obyek ke dalam sebuah kelas, grup, atau kategori berdasarkan prosedur, karakteristik & definisi yang telah ditentukan sebelumnya [UFWS09]. Pada penelitian ini, bentuk klasifikasi yang diterapkan adalah klasifikasi dokumen berupa dokumen teks.

Klasifikasi dokumen adalah bidang penelitian dalam perolehan informasi yang mengembangkan metode untuk menentukan atau mengkategorikan suatu dokumen ke dalam satu atau lebih kelompok yang telah dikenal sebelumnya secara otomatis berdasarkan isi dokumen [TSS08]. Klasifikasi dokumen bertujuan untuk mengelompokkan dokumen yang tidak terstruktur ke dalam kelompok-kelompok yang menggambarkan isi dari dokumen. Dokumen yang sudah diklasifikasikan akan memudahkan pencari informasi dalam mencari suatu dokumen. Dokumen tersebut akan lebih mudah ditemukan dari pada mencarinya dengan melihat satu per satu dokumen yang dimiliki.

Pengklasifikasian dokumen muncul dalam berbagai aplikasi, seperti pada aplikasi *email filtering*, *spam filtering*, dan identifikasi terhadap *genre* dokumen. Aplikasi *spam filtering* pada *email* bertujuan untuk menentukan apakah sebuah *email* termasuk *spam* atau bukan *spam*. Aplikasi ini bekerja dengan memperhatikan kata-kata yang terdapat di dalam *email* tersebut.

Sekarang ini, metode atau teknik yang sangat terkenal dalam klasifikasi dokumen adalah *machine learning* [SEB02]. Secara umum metode *machine learning*, membutuhkan dokumen pembelajaran untuk mengklasifikasikan dokumen baru. Dokumen pembelajaran dibuat dengan melakukan observasi karakteristik fitur pada dokumen yang sudah diklasifikasikan sebelumnya. Dokumen pembelajaran ini disebut *training set*. Dengan *training set* yang ada kemudian digunakan untuk mengklasifikasikan dokumen baru.

Metode *machine learning* dapat dibedakan menjadi dua cara, yaitu metode *supervised document classification* dan *unsupervised document classification*. Metode *Supervised document classification* adalah metode klasifikasi yang melibatkan mekanisme eksternal (*human feedback*) untuk menyediakan informasi klasifikasi yang benar. Metode ini membutuhkan dokumen pembelajaran yang berisi ciri-ciri dari setiap kategori yang ada. Dokumen pembelajaran yang ada digunakan untuk membangun *classifier*. Contoh *supervised document classification* adalah Naïve

Bayes, yang akan dijelaskan lebih lanjut pada laporan Tugas Akhir ini subbab (2.3). Sementara itu, metode *unsupervised document classification* adalah metode klasifikasi yang dilakukan secara mandiri tanpa melibatkan mekanisme eksternal. Metode ini biasa dikenal dengan nama *clustering* (pengelompokkan). Secara umum *clustering,* mengelompokkan dokumen yang memiliki kemiripan dikumpulkan ke dalam sebuah *cluster* secara iteratif. Kategori-kategori yang ada untuk masing-masing dokumen biasanya belum diketahui secara eksplisit.

Metode *machine learning* dapat digunakan untuk melakukan klasifikasi dokumen teks. Selain metode *machine learning*, ontologi juga dapat digunakan untuk melakukan klasifikasi dokumen. Klasifikasi dokumen menggunakan ontologi merupakan klasifikasi dokumen menggunakan pendekatan *knowledge engineering*. *Knowledge engineering* dijelaskan lebih lanjut pada subbab (2.4). Ontologi untuk klasifikasi dokumen dijelaskan lebih lanjut pada subbab (2.5). Lim SooYeon dan rekannya pada tahun 2006 telah membuktikan bahwa ontologi dapat digunakan untuk melakukan klasifikasi dokumen [LSL06]. Penelitian yang dilakukan oleh Lim SooYeon dan rekannya menggunakan dokumen berbahasa Inggris. Dokumen tersebut berupa artikel berita bidang ekonomi yang diperoleh dari http://www.yahoo.com. Penelitian tersebut menggunakan metode ontologi untuk melakukan klasifikasi dokumen dan membandingkan nilai akurasinya dengan nilai akurasi dari metode lain, seperti TF-IDF dan Naïve Bayes. Akurasi yang diperoleh untuk metode ontologi, Naïve Bayes, dan TF-IDF berturut-turut adalah 91,30%, 82,45%, dan 79,87%. Hal ini membuktikan bahwa metode ontologi dapat digunakan untuk melakukan klasifikasi dokumen teks dan tingkat akurasinya lebih tinggi daripada dua metode lainnya.

2.3 Naïve Bayes

Naïve Bayes merupakan salah satu contoh dari metode *supervised document classification.* Metode ini menggunakan perhitungan probabilitas. Naïve Bayes tidak memperhatikan urutan kemunculan kata pada dokumen teks dan menganggap sebuah dokumen teks sebagai kumpulan dari kata-kata yang menyusun dokumen teks

tersebut. Metode ini memiliki tingkat akurasi yang tinggi dengan penghitungan sederhana [KAN06].

Naïve Bayes menggunakan teorema dasar yang dikenal dengan nama Teorema Bayes. Rumus Teorema Bayes dapat dilihat pada Persamaan (2.1) [MIT06].

$$P(C = c_a \mid D = d_b) = \frac{P(C = c_a \cap D = d_b)}{P(D = d_b)} \quad \textbf{(2.1)}$$

dimana $P(C = c_a \mid D = d_b)$, probabilitas kategori c_a jika diketahui dokumen d_b. Kemudian dari Persamaan (2.1) kita dapat membuat Persamaan (2.2).

$$P(C = c_a \cap D = d_b) = P(D = d_b \mid C = c_a) \times P(C = c_a) \quad \textbf{(2.2)}$$

sehingga didapatkan Teorema Bayes seperti pada Persamaan (2.3).

$$P(C = c_a \mid D = d_b) = \frac{P(D = d_b \mid C = c_a) \times P(C = c_a)}{P(D = d_b)} \quad \textbf{(2.3)}$$

dengan $P(D = d_b \mid C = c_a)$ merupakan nilai probabilitas dari kemunculan dokumen d_b jika diketahui dokumen tersebut memiliki kategori c_a, $p(C = c_a)$ adalah nilai probabilitas kemunculan kategori c_a, dan $p(D = d_b)$ adalah nilai probabilitas kemunculan dokumen d_b. Klasifikasi dokumen teks dilakukan dengan terlebih dahulu menentukan kategori $c \in C$ dari suatu dokumen $d \in D$ dimana $C = \{c_1, c_2, c_3, \ldots, c_m\}$ dan $D = \{d_1, d_2, d_3, \ldots, d_n\}$ dan $P(C = c_a \mid D = d_b)$ memiliki nilai maksimum dari suatu distribusi probabilitas $P = \{ P(C = c_a \mid D = d_b) \mid c \in C \text{ dan } d \in D \}$.

Apabila urutan kemunculan kata dalam dokumen teks tidak diperhatikan, maka perhitungan probabilitas $P(D = d_b \mid C = c_a)$ dapat dianggap sebagai hasil perkalian dari probabilitas kemunculan kata-kata dalam dokumen d_b. Sebuah dokumen d_b terdiri dari kata-kata, maka dapat dituliskan sebagai $d_b = \{w_{1b}, w_{2b}, w_{3b}, \ldots, w_{kb}\}$ sehingga probabilitas $P(C = c_a \mid D = d_b)$ dapat dituliskan seperti pada Persamaan (2.4).

$$P(C = c_a \mid D = d_b) = \frac{\prod_k P(w_{kb} \mid C = c_a) \times P(C = c_a)}{P(w_1, w_2, w_3, \ldots, w_n)} \quad \textbf{(2.4)}$$

dengan $\prod_k P(w_{kb} \mid C = c_a)$ adalah hasil perkalian probabilitas kemunculan semua kata pada dokumen teks d_b, jika diketahui dokumen kategorinya adalah c_a.

Tahap klasifikasi dilakukan dengan membuat model probabilistik dari dokumen pembelajaran, yaitu dengan menghitung nilai $P(w_{kb}|c_a)$. Probabilitas yang mungkin dapat dicari untuk seluruh nilai w_{kb} dengan menggunakan Persamaan (2.5) dan (2.6) [MIT06].

$$P(w_{kb} \mid c_a) = \frac{f(w_{kb}, c_a)+1}{f(c_a)+|w|} \quad \textbf{(2.5)}$$

dimana $f(w_{kb,}\ c_a)$ adalah fungsi yang menghasilkan nilai kemunculan kata w_{kb} pada kategori c_a, $f(c_a)$ adalah fungsi yang menghasilkan jumlah keseluruhan kata pada kategori c_a, dan $|W|$ adalah jumlah kemungkinan nilai dari w_{kb} (jumlah keseluruhan kata yang digunakan). Persamaan $f(w_{kb,}\ c_a)$ sering kali dikombinasikan dengan *Laplacian Smoothing* (tambah satu) untuk mencegah persamaan mendapatkan nilai 0. Hal ini dilakukan karena nilai 0 dapat menganggu hasil klasifikasi secara keseluruhan.

$$P(c_a) = \frac{f_d(c_a)}{|D|} \quad \textbf{(2.6)}$$

dimana $fa(c_a)$ adalah fungsi yang menghasilkan jumlah dokumen teks yang memiliki kategori c_a, dan $|D|$ adalah jumlah seluruh dokumen pembelajaran.

Pemberian kategori dari sebuah dokumen teks dilakukan dengan memilih nilai c yang memiliki nilai probabilitas $P(C = c_a \mid D = d_b)$ maksimum, seperti pada Persamaan (2.7).

$$c^* = \arg\max_{ca \in C} P(c_a \mid d_b) \quad \textbf{(2.7)}$$

$$c^* = \arg\max_{ca \in C} \prod_k P(w_{kb} \mid c_a) \times P(c_a)$$

Kategori a^* merupakan kategori yang memiliki nilai probabilitas $p(C = c_a| D = d_b)$ maksimum.

2.4 *Knowledge Engineering*

Pendekatan *knowledge engineering* disebut *rule base* karena pendekatan ini memanfaatkan keahlian manusia (*human expert*) untuk membuat aturan-aturan (*rules*) secara manual melalui proses pemahaman pada sebuah domain penelitian [MIL03]. Dalam penelitian ini, *human expert* atau pakar dituntut untuk bisa memahami sebuah domain yang digunakan dalam pemodelan ontologi untuk klasifikasi dokumen secara otomatis. Dengan pendekatan *rule base* ini, nilai akurasi klasifikasi dokumen menggunakan ontologi sangat tergantung dari pakar yang membuat aturan-aturan yang digunakan dalam klasifikasi dokumen.

Kelebihan dari pendekatan *rule base* adalah dengan menggunakan keahlian manusia untuk mencapai nilai akurasi klasifikasi dokumen yang tinggi. Pendekatan ini tidak terlalu sulit untuk dilakukan selama terdapat pakar yang memahami domain yang digunakan untuk klasifikasi dokumen dengan baik. Akan tetapi, hal inilah yang menjadi kelemahan *rule base*, yaitu metode klasifikasi dokumen menggunakan ontologi sangat bergantung pada adanya pakar. Selain itu, pendekatan ini memiliki kekurangan lain, yaitu membutuhkan waktu yang panjang dan biaya yang tinggi. Biaya yang tinggi ini disebabkan kebutuhan terhadap sumber daya manusia yang banyak terlebih jika domain yang digunakan untuk klasifikasi dokumen memiliki ruang lingkup yang sangat besar. Metode klasifikasi dokumen dengan menggunakan pendekatan *rule base* juga akan mengalami masalah *adaptability*, yaitu ketika pakar yang membuat aturan-aturan dalam sistem sudah tidak ada sehingga pakar yang baru sulit untuk melakukan penyesuaian jika ingin melakukan perubahan pada domain. Oleh karena itu, pendekatan *rule base* cocok untuk digunakan jika terdapat pakar yang memahami domain penelitian.

2.5 Ontologi

Ontologi adalah sebuah deskripsi formal tentang sebuah konsep secara eksplisit dalam sebuah domain, properti dari setiap konsep beserta dengan batasannya

[NM01]. Sebuah konsep di ontologi dapat memiliki objek (*instances*). Secara teknis, ontologi direpresentasikan dalam bentuk *class*, *property*, *facet*, dan *instance* [NM01]:

- *Class*

  Class menerangkan konsep atau makna dari suatu domain. *Class* adalah kumpulan dari elemen dengan sifat yang sama. Sebuah *class* bisa memiliki sub *class* yang menerangkan konsep yang lebih spesifik.

- *Property*

  Property merepresentasikan hubungan diantara dua individu. *Property* menghubungkan individu dari domain tertentu dengan individu dari *range* tertentu. Ada tiga jenis *property*, yaitu *object property*, *data type property* dan *annotation property*. *Object property* menghubungkan suatu individu dengan individu lain. *Object property* terdiri dari empat tipe, yaitu *inverse property*, *functional property*, *transitive property*, dan *symmetric property*. *Data type property* menghubungkan sebuah individu ke sebuah tipe data pada *Resource Description Framework* (RDF) *literal* atau pada *Extensible Markup Language* (XML). *Annotation property* digunakan untuk menambah informasi (*metadata*) ke kelas, individu dan *object/data type property*.

- *Facet*

  Facet digunakan untuk merepresentasikan informasi atau batasan tentang *property*. Ada dua jenis *facet*, yaitu *cardinality* dan *value type*. *Cardinality facet* merepresentasikan nilai eksak yang bisa digunakan untuk *slot* pada suatu kelas tertentu. *Cardinality facet* dapat bernilai *single* dan *multiple cardinality*. *Value type* menggambarkan tipe nilai yang dapat memenuhi *property*, seperti *string*, *number*, *boolean*, dan *enumerated*.

- *Instance*

  Instance adalah objek dari sebuah kelas.

Ada beberapa langkah yang diperlukan untuk mengembangkan ontologi, yaitu [NM01]

1. Tahap penentuan domain

   Tahap ini merupakan tahap awal proses digitalisasi pengetahuan yang dilakukan dengan menjawab beberapa pertanyaan seperti apa yang menjadi domain ontologi.

2. Tahap penggunaan ulang ontologi

   Dalam tahap ini, kita melakukan pengecekan apakah ontologi yang sudah ada dapat digunakan kembali atau kita perlu mengembangkan ontologi dari awal. Apabila kita menggunakan ontologi yang sudah ada kemudian kita melakukan perbaikan dan memperluas ontologi yang sudah ada, maka kita dapat lebih menghemat waktu dari pada mengembangkan ontologi dari awal.

3. Tahap penyebutan istilah-istilah pada ontologi

   Tahap ini menentukan semua istilah penting yang digunakan untuk membuat pernyataan atau menjelaskan hal yang mirip atau sama. Contoh *class* "*wines*" berhubungan dengan istilah *wine*, anggur, lokasi, warna, bentuk, rasa dan kadar gula.

4. Tahap pendefinisian *class* dan hierarki *class*

   Tahap ini membuat definisi dari *class* dalam bentuk hierarki dan kemudian menguraikan *property* dari *class*. Hierarki *class* merepresentasikan sebuah relasi "*is-a*" (sebuah *class* A adalah *subclass* dari B jika setiap *instance* dari B adalah juga sebuah *instance* di A).

5. Tahap pendefinisian *property*

   Tahap ini mendefinisikan *property* dari masing-masing *class* yang ada di ontologi.

6. Tahap pendefinisian *facets*

   Tahap ini mendefinisikan *facets* dari setiap *property* yang ada di *class* pada ontologi.

7. Tahap mendefinisikan *instances*

Tahap ini mendefinisikan sebuah *instance* dari suatu *class* meliputi pemilihan *class*, pembuatan individu *instance* dari *class*, dan pengisian nilai *property*.

Ontologi baru dapat digunakan apabila ontologi tersebut sudah diekspresikan terlebih dahulu dalam notasi yang nyata. Sebuah bahasa ontologi adalah sebuah bahasa formal yang digunakan untuk merepresentasikan ontologi. Beberapa komponen bahasa yang menyusun ontologi, yaitu XML, XML *schema*, RDF, RDF *schema*, dan *Ontology Web Language* (OWL). XML menyediakan sintaksis untuk dokumen keluaran secara terstruktur, tetapi belum menggunakan *semantic constraints*. XML *schema* adalah bahasa untuk pembatasan struktur dari dokumen XML. RDF adalah model data untuk objek dan relasi diantaranya, menyediakan semantic yang sederhana untuk model data, dan disajikan dalam sintaks XML. RDF *schema* adalah kosakata untuk menjelaskan properti dan *class* dari sumber RDF. OWL adalah bahasa ontologi yang baru untuk sebuah web semantik, dikembangkan oleh *World Wide Web Consortium* (W3C) [HOR04]. OWL dapat mendefinisikan relasi antar *class*, kardinalitas, karakteristik dari *properties*, dan *equality*.

Ontologi dapat digunakan untuk melakukan klasifikasi dokumen teks dalam penelitian ini karena ontologi bersifat unik dan memiliki struktur hierarkis. Selain itu, sebuah model ontologi dapat menghilangkan makna ambigu, sehingga dapat menanggulangi masalah yang muncul pada bahasa alami di mana sebuah kata memiliki lebih dari satu makna atau arti bergantung pada konteks kalimatnya. Pengembangan ontologi dalam penelitian ini terdiri dari beberapa komponen utama yaitu:

- Konsep

  Konsep atau *class* merepresentasikan *term* atau kata dalam domain yang spesifik.
- Fitur

  Fitur atau *instance* merepresentasikan individu dari sebuah kelas.
- Relasi

Relasi atau *property* merepresentasikan hubungan diantara konsep. Ada dua relasi yang digunakan dalam penelitian ini yaitu: relasi "*is-a*" dan "*has-a*".

- *Constraint*

  Constraint merepresentasikan kondisi yang harus dipenuhi di sebuah konsep.

2.6 Persiapan Dokumen Teks

Proses persiapan dokumen teks dilakukan terlebih dahulu sebelum melakukan pembobotan kata. Proses persiapan dokumen teks meliputi proses *case folding* (subbab 2.5.1), tokenisasi (subbab 2.5.2), pembuangan *stopwords* (subbab 2.5.3), dan pemotongan imbuhan (subbab 2.5.4) [BYRN99]. Tujuan dari proses persiapan dokumen teks adalah untuk menghilangkan karakter-karakter selain huruf, menyeragamkan kata, mengurangi volume kosakata, dan menghitung bobot kata pada dokumen teks.

2.6.1 *Case Folding*

Sebuah dokumen teks mengandung beragam variasi, seperti huruf dan tanda baca. Variasi huruf harus diseragamkan (menjadi huruf kecil saja). Karakter selain huruf dihilangkan dan dianggap sebagai *delimiter*. Proses ini disebut *case folding* [BYRN99]. Tujuan dari proses ini adalah untuk menghilangkan karakter-karakter selain huruf pada saat pengambilan informasi.

2.6.2 Tokenisasi

Proses persiapan dokumen teks selanjutnya adalah tokenisasi. Tokenisasi adalah proses pemecahan kalimat menjadi kata atau frase [BYRN99]. Koleksi dokumen teks dalam penelitian ini terdiri dari kalimat. Proses tokenisasi dalam penelitian ini memecah kalimat menjadi kata.

2.6.3 Pembuangan *Stopwords*

Proses pembuangan *stopwords* merupakan proses yang dilakukan setelah proses tokenisasi. *Stopwords* adalah kata-kata yang sering muncul dan tidak dipakai di

dalam pemrosesan bahasa alami [BYRN99]. *Stopwords* dapat berupa kata depan, kata penghubung, dan kata pengganti. Contoh *stopwords* dalam bahasa Indonesia adalah "yang", "ini", "dari", dan "di". Ukuran kata dalam sebuah dokumen teks menjadi berkurang setelah dilakukan proses pembuangan *stopwords* sehingga hanya kata-kata yang penting terdapat dalam sebuah dokumen teks dan diharapkan memiliki bobot yang tinggi. Daftar kata-kata *stopwords* berbahasa Indonesia dapat dilihat pada Lampiran 1.

2.6.4 Pemotongan Imbuhan

Pemotongan imbuhan adalah proses pengembalian kata berimbuhan menjadi kata dasar [BYRN99]. Pemotongan imbuhan biasa dikenal dengan nama *stemming*. Sebuah kata memiliki variasi kombinasi imbuhan kata yang beragam. Variasi imbuhan dapat berupa awalan (*prefix*), akhiran (*suffix*), sisipan (*infix*), dan kombinasi antara awalan dan akhiran *(confix)*. Contoh proses pemotongan imbuhan dalam bahasa Indonesia adalah kata "makanan" memiliki bentuk dasar "makan", kata "berbelanja" memiliki bentuk dasar "belanja", dan kata "menjalankan" memiliki bentuk dasar "jalan". Proses pemotongan imbuhan dapat mengurangi variasi kata yang sebenarnya memiliki kata dasar yang sama. Algoritma pemotongan imbuhan bahasa Indonesia yang digunakan dalam penelitian ini adalah algoritma Adriani dan Nazief [AN96].

2.7 Pembobotan Kata

Proses pembobotan kata adalah proses memberikan nilai atau bobot ke sebuah kata berdasarkan kemunculannya pada suatu dokumen teks [BYRN99]. Proses persiapan dokumen teks dalam penelitian ini menghasilkan kumpulan kata atau *term* yang kemudian direpresentasikan dalam sebuah *terms vector*. *Terms vector* dari suatu dokumen teks d adalah *tuple* bobot semua *term* pada d. Nilai bobot sebuah *term* menyatakan tingkat kepentingan *term* tersebut dalam merepresentasikan dokumen teks. Ada tiga metode pembobotan *term* yaitu *binary weighting* (subbab 2.5.5.1), *term*

frequency (subbab 2.5.5.2), dan *term frequency – inverse document frequency* (subbab 2.5.5.3) [BYRN99].

2.7.1 *Binary Weighting*

Binary weighting merupakan proses pembobotan kata yang paling sederhana. Pembobotan dengan metode ini hanya terdiri dari dua nilai yaitu "0" jika *term* i tidak terdapat pada dokumen teks dan 1 jika *term i* terdapat pada dokumen teks [BYRN99]. Rumus *Binary weighting* dapat dilihat pada Persamaan (2.8) [BYRN99].

$$a_{ij} = \begin{cases} 0 : \text{term i tidak terdapat pada dokumen j} \\ 1 : \text{term i terdapat pada dokumen j} \end{cases} \quad \textbf{(2.8)}$$

2.7.2 *Term Frequency*

Term frequency atau biasa sering disebut TF. TF adalah metode pembobotan kata dengan menghitung frekuensi kemunculan kata pada sebuah dokumen teks [BYRN99]. Semakin sering sebuah kata muncul pada suatu dokumen teks, maka bobot kata tersebut semakin besar dan kata tersebut dianggap sebagai kata yang sangat merepresentasikan dokumen teks tersebut. Rumus TF dapat dilihat pada Persamaan (2.9) [BYRN99].

tf(t,d) = frekuensi kemunculan *term* t pada dokumen teks d **(2.9)**

2.7.3 *Term Frequency – Inverse Document Frequency*

Term frequency – inverse document frequency atau biasa sering disebut TF-IDF adalah metode pembobotan kata dengan menghitung nilai TF dan juga menghitung kemunculan sebuah kata pada koleksi dokumen teks secara keseluruhan [BYRN99]. Pada pembobotan ini, jika kemunculan *term pada* sebuah dokumen teks tinggi dan kemunculan *term* tersebut pada dokumen teks lain rendah, maka bobotnya akan semakin besar. Akan tetapi, jika kemunculan *term* tersebut pada dokumen teks lain tinggi, maka bobotnya akan semakin kecil. Tujuan penghitungan IDF adalah untuk

mencari kata-kata yang benar-benar merepresentasikan suatu dokumen teks pada suatu koleksi. Metode pembobotan kata yang digunakan dalam penelitian ini adalah metode TF-IDF. Metode ini digunakan karena metode ini paling baik dalam perolehan informasi [KW05]. Rumus TF-IDF dapat dilihat pada Persamaan (2.10) [SAL83].

$$tfidf(i,j) = tf(i,j) \times \log\left(\frac{N}{df_j}\right) \quad \textbf{(2.10)}$$

dimana tf(i,j) adalah frekuensi kemunculan *term* j pada dokumen teks $d_i \in D^*$, dimana i = 1,2,3,...,N, df(j) adalah frekuensi dokumen yang mengandung *term* j dari semua koleksi dokumen, dan N adalah jumlah seluruh dokumen yang ada di koleksi dokumen.

2.8 Evaluasi

Evaluasi digunakan untuk mengukur kinerja suatu sistem, khusus dalam penelitian ini digunakan untuk mengukur keakuratan metode klasifikasi dokumen teks. Ada beberapa teknik evaluasi yang biasa digunakan untuk mengukur keakuratan metode klasifikasi dokumen teks diantaranya *precision* dan *recall*, *f-measure,* dan *mean average precision* [BYRN99]. Metode evaluasi yang digunakan pada penelitian ini adalah *recall, precision,* dan *f-measure* [LSL06].

Pengevaluasian hasil klasifikasi dokumen teks pada tiap kategori menggunakan standar Tabel 2.1 [LSL06]. Tabel 2.1 berisi setiap kemungkinan hasil klasifikasi pada tiap kategori.

Tabel 2.1 Matriks Keputusan untuk Menghitung Akurasi Klasifikasi Dokumen Teks

Expert	*System*	
	Yes	No
Yes	a	b
No	c	d

Tabel 2.1 menunjukkan bahwa hasil klasifikasi yang dilakukan oleh sistem bisa bernilai benar sesuai dengan keputusan *expert* (a) dan bisa bernilai salah (b). Selain itu, ada dokumen teks yang tidak termasuk dalam hasil klasifikasi suatu kategori karena dokumen teks tersebut bukan anggota dari kategori yang ada (d) dan ada dokumen teks yang diklasifikasikan ke sebuah kategori oleh sistem, namun *expert* tidak setuju dengan hasil klasifikasi tersebut (c). Tiga metode evaluasi dalam penelitian ini dihitung berdasarkan keempat parameter yang ada di Tabel 2.1, yaitu

1. *Recall*

   Recall adalah proporsi jumlah dokumen teks yang relevan terkenali diantara semua dokumen teks relevan yang ada pada koleksi [BYRN99]. Rumus *recall* dapat dilihat pada Persamaan (2.11) [LSL06].

$$Recall = \frac{a}{a+c} \quad \textbf{(2.11)}$$

2. *Precision*

   Precision adalah proporsi jumlah dokumen teks yang relevan terkenali diantara semua dokumen teks yang terpilih oleh sistem [BYRN99]. Rumus *Precision* dapat dilihat pada Persamaan (2.12) [LSL06].

$$Precision = \frac{a}{a+b} \quad \textbf{(2.12)}$$

3. *F-Measure*

   F-Measure adalah nilai yang mewakili seluruh kinerja sistem yang merupakan rata-rata dari nilai *precision* dan *recall* [LSL06]. Rumus *F-Measure* dapat dilihat pada Persamaan (2.13) [LSL06].

$$F\text{ - }Measure = \frac{2}{\frac{1}{precision} + \frac{1}{recall}} \quad \textbf{(2.13)}$$

2.9 Penelitian dalam Bidang Klasifikasi Dokumen Teks

Penelitian dalam bidang klasifikasi dokumen teks berbahasa Indonesia telah banyak dilakukan. Beberapa penelitian pada bidang klasifikasi dokumen teks berbahasa Indonesia adalah penelitian yang dilakukan oleh Agus Zainal Arifin dan Ari Novan Setiono [AS05], Yudi Wibisono [WIB05], Masayu Leylia Khodra dan Yudi Wibisono [KW05], Sylvia Susanto [SUS06], dan Dyta Anggraeni [ANG09].

Penelitian yang dilakukan oleh Agus Zainal Arifin dan Ari Novan Setiono yaitu klasifikasi berita kejadian berbahasa Indonesia dengan algoritma *single pass clustering* [AS05]. Dokumen teks yang digunakan untuk eksperimen diambil dari halaman surat kabar *online* Suara Pembaruan mulai dari tanggal 1 Agustus 2001 sampai dengan 31 Agustus 2001. Metode pembobotan yang digunakan dalam penelitian ini adalah metode TF-IDF. Eksperimen dilakukan mulai dari nilai *threshold* = 0 sampai memperoleh nilai *threshold* di atas keseluruhan *maximal similarity* atau mendapatkan jumlah *cluster* sama dengan jumlah dokumen teks. Kesimpulan yang diperoleh dari penelitian ini adalah algoritma *single pass clustering* dapat diaplikasikan untuk klasifikasi berita berbahasa Indonesia dengan nilai *recall* 79% dan *precision* 88%. Selain itu, pemilihan *threshold* yang tepat akan meningkatkan kualitas klasifikasi dokumen teks.

Penelitian yang dilakukan oleh Yudi Wibisono yaitu klasifikasi berita berbahasa Indonesia menggunakan Naïve Bayes *classifier* [WIB05]. Dokumen teks yang digunakan untuk eksperimen diambil dari halaman surat kabar *online* Kompas dengan jumlah 582 (lima ratus delapan puluh dua) dokumen teks. Dokumen teks tersebut terbagi ke dalam lima kategori, yaitu metro, kesehatan, olahraga, teknologi, dan gaya hidup. Dokumen teks dibagi menjadi dua bagian yaitu dokumen pembelajaran dan dokumen pengujian. Hasil eksperimen penelitian ini adalah metode Naïve Bayes *classifier* memiliki akurasi yang tinggi yaitu 89,47%. Nilai akurasi tetap tinggi terutama jika dokumen pembelajaran yang digunakan besar (lebih besar atau sama dengan 400). Kesimpulan yang diperoleh dari penelitian ini adalah metode Naïve

Bayes *classifier* terbukti dapat digunakan secara efektif untuk mengklasifikasikan berita secara otomatis.

Penelitian yang dilakukan Masayu Leylia Khodra dan Yudi Wibisono yaitu pengklasifikasian dokumen berita berbahasa Indonesia menggunakan algoritma *K-means clustering* [KW05]. Dokumen teks yang digunakan untuk eksperimen diambil dari halaman surat kabar *online* Kompas mulai dari bulan Juli 2005 sampai dengan November 2005. Koleksi dokumen teks ini terdiri atas 8237 (delapan ribu dua ratus tiga puluh tujuh) dokumen teks. Berdasarkan URL-nya, Kompas telah memberikan label kategori pada sebagian berita yaitu sebanyak 4718 (empat ribu tujuh ratut delapan belas) dokumen teks. Kategori yang diberikan yaitu metro, otomotif, kesehatan, olahraga, teknologi, dan gaya hidup. Metode pembobotan yang digunakan dalam penelitian ini adalah *logarithmic term frequency* (log-TF) dan *logarithmic term frequency – inverse document frequency* (log-TFIDF). Eksperimen yang dilakukan dalam penelitian ini meliputi *stemming* dengan TF, *stemming* dengan TFIDF, *non-stemming* TF, dan *non-stemming* TFIDF. Jumlah *cluster* yang dipilih adalah 20 (dua puluh) *cluster*. Inisialisasi *cluster* dilakukan secara acak. Hasil ekperimen dalam penelitian ini menunjukkan bahwa kualitas *cluster* terbaik diperoleh dari kombinasi antara tanpa *stemming* dan TFIDF, lalu diikuti oleh tanpa *stemming* dan TF, kemudian *stemming* dan TF, dan terakhir adalah *stemming* dan TFIDF. Kesimpulan yang diperoleh dari penelitian ini adalah semakin banyak *cluster* yang diinisialisasi, maka semakin spesifik klasifikasi yang dilakukan.

Penelitian yang dilakukan oleh Slyvia Susanto yaitu pengklasifikasian dokumen berita berbahasa Indonesia dengan menggunakan Naïve Bayes *classifier* (*stemming* atau *non-stemming*) [SUS06]. Dokumen teks yang digunakan untuk eksperimen diambil dari halaman surat kabar *online* Suara Pembaruan dengan jumlah 1351 (seribu tiga ratus lima puluh satu) dokumen teks. Dokumen teks tersebut terdiri dari delapan kategori yaitu kesehatan, musik, olahraga, boga, ekonomi, gaya, hukum dan politik, dan iptek.Metode pembobotan kata yang digunakan dalam penelitian ini adalah metode TF. Eksperimen yang dilakukan dalam penelitian ini dengan

menggunakan *stemming* dan *non-stemming*. Hasil eksperimen dalam penelitian ini menunjukkan bahwa jumlah dokumen pembelajaran 90% dan jumlah dokumen pengujian 10% (*stemming*) menghasilkan akurasi yang paling tinggi yaitu dengan *recall* 93,5%, *precision* 90,36%, dan *f-measure* 93,81%. Kesimpulan yang diperoleh dari penelitian ini adalah kinerja Naïve Bayes *classifier* yang menggunakan *stemming* lebih baik dari pada *non-stemming*. Selain itu, kinerja dipengaruhi oleh jumlah dokumen pembelajaran. Semakin banyak dokumen pembelajaran, maka semakin banyak dan beragam kata kunci yang dimiliki oleh sistem. Kata kunci inilah yang berperan dalam proses klasifikasi karena menunjukkan ciri suatu kategori dan menjadi pembeda terhadap kategori lain.

Penelitian yang dilakukan oleh Dyta Anggraeni yaitu pengklasifikasian topik dengan menggunakan metode Naïve Bayes dan Maximum Entropy pada artikel media massa dan abstrak tulisan berbahasa Indonesia [ANG09]. Penelitian ini membandingkan metode pengklasifikasian dokumen teks diantara metode Naïve Bayes dan Maximum Entropy. Dokumen teks yang digunakan untuk eksperimen diambil dari halaman surat kabar *online* Kompas dan kumpulan abstrak tulisan ilmiah. Dokumen teks yang diambil dari Kompas berjumlah 1240 (seribu dua ratus empah puluh) dokumen teks yang terdiri dari lima kategori yaitu ekonomi, olahraga, kesehatan, properti dan travel. Dokumen teks yang berupa abstrak tulisan ilmiah diperoleh dari Sistem Lontar dengan jumlah 350 (tiga ratus lima puluh) dokumen teks. Dokumen teks tersebut terdiri dari tiga kategori yaitu *Information Retrieval* (IR), Pengolahan Citra, dan Rekayasa Perangkat Lunak (RPL). Metode pembobotan kata yang digunakan dalam penelitian ini adalah *presence*, *frequency*, dan *frequency normalized*. Eksperimen yang dilakukan dalam penelitian ini dengan mengamati beberapa aspek seperti jumlah topik yang digunakan, jumlah fitur yang digunakan, dan informasi fitur yang digunakan. Hasil eksperimen dalam penelitian ini menunjukkan bahwa metode Naïve Bayes dan Maximum Entropy memiliki akurasi yang hampir mirip. Kesimpulan yang diperoleh dari penelitian ini adalah kedua metode memiliki tingkat akurasi yang sama baiknya.