BAB 2 LANDASAN TEORI

Pada bab ini dijelaskan landasan teori dan metode yang digunakan pada tugas akhir ini dalam pengklasifikasian dokumen teks. Pembahasan dimulai dengan penjelasan mengenai klasifikasi dokumen teks. Pada subbab berikutnya dijelaskan metode-metode yang digunakan dalam melakukan klasifikasi dokumen teks.

2.1 Klasifikasi Dokumen Teks

Klasifikasi dokumen teks adalah masalah sederhana namun sangat penting karena manfaatnya cukup besar mengingat jumlah dokumen yang ada setiap hari semakin bertambah. Manfaat dari klasifikasi dokumen adalah untuk pengorganisasian dokumen. Dengan jumlah dokumen yang sangat besar, untuk mencari sebuah dokumen akan lebih mudah apabila kumpulan dokumen yang dimiliki terorganisir dan telah dikelompokkan sesuai kategorinya masing-masing. Contoh aplikasi penggunaan klasifikasi dokumen teks yang banyak digunakan adalah *e-mail spam filtering*. Pada aplikasi *spam filtering* sebuah *e-mail* diklasifikasikan apakah *e-mail* tersebut termasuk *spam* atau tidak dengan memperhatikan kata-kata yang terdapat di dalam *e-mail* tersebut. Aplikasi ini telah digunakan oleh banyak *e-mail provider*.

Sebuah dokumen dapat dikelompokkan ke dalam kategori tertentu berdasarkan kata-kata dan kalimat-kalimat yang ada di dalam dokumen tersebut. Kata atau kalimat yang terdapat di dalam sebuah dokumen memiliki makna tertentu dan dapat digunakan sebagai dasar untuk menentukan kategori sesuai topik dari dokumen tersebut. Perhatikan beberapa kalimat berikut ini:

1. Pemilih yang namanya tercantum dalam daftar pemilih tetap dan daftar pemilih tambahan, apabila sampai dengan 3 (tiga) hari sebelum hari dan tanggal pemungutan suara belum menerima surat pemberitahuan untuk memberikan suara di TPS (Model C4), diberi kesempatan untuk meminta

kepada Ketua KPPS selambat-lambatnya 24 (dua puluh empat) jam sebelum hari dan tanggal pemungutan suara dengan menunjukkan kartu tanda penduduk atau identitas lain yang sah. [PERATURAN KOMISI PEMILIHAN UMUM NOMOR 03 TAHUN 2009]

2. Pemerintah Pusat dan bank sentral berkoordinasi dalam penetapan dan pelaksanaan kebijakan fiskal dan moneter. [UNDANG-UNDANG REPUBLIK INDONESIA NOMOR 17 TAHUN 2003]

3. Pada universitas, institut, dan sekolah tinggi dapat diangkat guru besar atau profesor sesuai dengan peraturan perundang-undangan yang berlaku. [UNDANG-UNDANG REPUBLIK INDONESIA NOMOR 20 TAHUN 2003]

Pada kalimat (1) terdapat kata pemilih, TPS, dan KPPS. Kata-kata tersebut memiliki keterkaitan erat dengan masalah pemilu, sehingga dapat disimpulkan bahwa kalimat (1) membahas masalah pemilu. Kalimat (2) memiliki kata bank, fiskal, dan moneter. Dari kata-kata tersebut akan muncul dugaan bahwa kalimat (2) sedang membahas masalah keuangan. Terakhir, pada kalimat (3) terdapat kata sekolah, guru, dan universitas yang menunjukkan bahwa kalimat tersebut membahas bidang pendidikan.

Kata bank yang terdapat pada dokumen lain belum dapat dijadikan sebagai acuan bahwa dokumen lain tersebut membahas mengenai keuangan. Apabila dokumen lain tersebut memiliki kata-kata lain yang mengarahkan kepada pembahasan keuangan secara bersamaan, maka dapat disimpulkan bahwa dokumen tersebut membahas mengenai keuangan. Untuk dapat menentukan kategori dari sebuah dokumen haruslah dilihat semua kata-kata yang terkait pada dokumen tersebut.

2.2 *Machine Learning* untuk Klasifikasi Dokumen Teks

Teknik *machine learning* mulai banyak digunakan untuk melakukan klasifikasi dokumen teks pada awal tahun 1990-an. Teknik ini dapat dilakukan dengan dua cara

yaitu dengan pendekatan *supervised learning* dan pendekatan *unsupervised learning*. Teknik yang banyak digunakan dalam *unsupervised learning* adalah teknik *clustering*. *Clustering* merupakan teknik mengelompokkan dokumen-dokumen, sehingga dokumen yang memiliki kemiripan dikumpulkan dalam sebuah *cluster* tertentu. Teknik *clustering* umumnya merupakan teknik yang iteratif. Kategori-kategori yang ada untuk setiap dokumen biasanya belum diketahui secara eksplisit. Hal ini berbeda dengan teknik klasifikasi dimana kategori yang ada telah ditentukan sebelumnya.

Pendekatan kedua adalah *supervised learning*. Pendekatan ini dilakukan dengan membangun sebuah *classifier* dari proses pembelajaran mengenai ciri dari tiap-tiap kategori yang ada. Pendekatan ini biasa disebut dengan teknik klasifikasi. Teknik ini membagi kumpulan dokumen yang dimiliki menjadi dokumen *training* dan dokumen *testing*. *Classifier* dibangun dengan mempelajari ciri tiap kategori berdasarkan dokumen *training* yang dimiliki. Pendekatan *supervised learning* dapat dibagi menjadi *fully supervised learning* dan *semi supervised learning*. *Fully supervised learning* adalah teknik klasifikasi dimana semua dokumen *training* telah diketahui kategorinya. Naïve Bayes adalah contoh dari teknik *fully supervised learning*, sedangkan *semi supervised learning* adalah teknik klasifikasi dimana pembelajaran dilakukan dari dokumen *training* yang telah diketahui kategorinya dan dokumen *training* yang belum diketahui kategorinya. Expectation Maximization adalah teknik *semi supervised learning* yang paling populer digunakan untuk klasifikasi dokumen teks.

Metode *machine learning* dapat dipergunakan untuk klasifikasi dokumen teks. Hal ini ditunjukkan dengan penelitian yang telah dilakukan sebelumnya oleh Kamal Nigam (Nigam, 1999). Penelitian tersebut menggunakan Naïve Bayes dan Expectation Maximization untuk melakukan klasifikasi dokumen teks. Akurasi yang diperoleh untuk algoritma Naïve Bayes adalah 76% sedangkan akurasi untuk Expectation Maximization adalah 78%. Hal ini membuktikan bahwa Naïve Bayes dan Expectation Maximization dapat digunakan untuk melakukan klasifikasi dokumen teks.

2.3 Naïve Bayes

Naïve Bayes merupakan salah satu metode *machine learning* yang menggunakan perhitungan probabilitas. Konsep dasar yang digunakan oleh Naïve bayes adalah Teorema Bayes, yaitu melakukan klasifikasi dengan melakukan perhitungan nilai probabilitas $p(C = c_i \mid D = d_j)$, yaitu probabilitas kategori c_i jika diketahui dokumen d_j. Klasifikasi dilakukan untuk menentukan kategori $c \ \varepsilon \ C$ dari suatu dokumen $d \ \varepsilon \ D$ dimana $C = \{c_1, c_2, c_3, \ldots, c_i\}$ dan $D = \{d_1, d_2, d_3, \ldots, d_j\}$. Penentuan dari kategori sebuah dokumen dilakukan dengan mencari nilai maksimum dari $p(C = c_i \mid D = d_j)$ pada $P=\{ p(C = c_i \mid D = d_j) \mid c \ \varepsilon \ C \ dan \ d \ \varepsilon \ D\}$. Nilai probabilitas $p(C = c_i \mid D = d_j)$ dapat dihitung dengan persamaan (Mitchell, 2005):

$$p(C = c_i \mid D = d_j) = \frac{P(C = c_i \bigcap D = d_j)}{P(D = d_j)}$$

$$= \frac{p(D = d_j \mid C = c_i) \times p(C = c_i)}{p(D = d_j)} \qquad \textbf{(2.1)}$$

dengan $p(D = d_j \mid C = c_i)$ merupakan nilai probabilitas dari kemunculan dokumen d_j jika diketahui dokumen tersebut berkategori c_i, $p(C = c_i)$ adalah nilai probabilitas kemunculan kategori c_i, dan $p(D = d_j)$ adalah nilai probabilitas kemunculan dokumen d_j.

Naïve Bayes menganggap sebuah dokumen sebagai kumpulan dari kata-kata yang menyusun dokumen tersebut, dan tidak memperhatikan urutan kemunculan kata pada dokumen, sehingga perhitungan probabilitas $p(D = d_j \mid C = c_i)$ dapat dianggap sebagai hasil perkalian dari probabilitas kemunculan kata-kata pada dokumen d_j. Sebuah dokumen dapat dituliskan sebagai $d_j = \{w_{1j}, w_{2j}, w_{3j}, \ldots, w_{kj}\}$, sehingga probabilitas $p(C = c_i \mid D = d_j)$ dapat dituliskan sebagai berikut:

$$p(C = c_i \mid D = d_j) = \frac{\prod_k p(w_{kj} \mid C = c_i) \times p(C = c_i)}{p(w_{1j}, w_{2j}, w_{3j}, ..., w_{kj})} \quad \textbf{(2.2)}$$

dengan $\prod_k p(w_{kj} \mid C = c_i)$ adalah hasil perkalian dari probabilitas kemunculan semua kata pada dokumen d_j jika diketahui dokumen tersebut berkategori c_i.

Proses klasifikasi dilakukan dengan membuat model probabilistik dari dokumen *training*, yaitu dengan menghitung nilai $p(w_{kj} \mid c_i)$. Karena w_{kj} diskrit, maka $p(w_{kj} \mid c_i)$ dicari untuk seluruh kemungkinan nilai w_{kj} dan didapatkan dengan melakukan perhitungan (Mitchell, 2005):

$$p(w_{kj} \mid c_i) = \frac{f(w_{kj}, c_i)}{f(c_i)} \quad \textbf{(2.3)}$$

dan

$$p(c_i) = \frac{f_d(c_i)}{|D|} \quad \textbf{(2.4)}$$

dengan $f(w_{kj}, c_i)$ adalah fungsi yang mengembalikan nilai kemunculan kata w_{kj} pada kategori c_i, $f(c_i)$ adalah fungsi yang mengembalikan jumlah keseluruhan kata pada kategori c_i, $f_d(c_i)$ adalah fungsi yang mengembalikan jumlah dokumen yang memiliki kategori c_i, dan $|D|$ adalah jumlah seluruh *training* dokumen. Persamaan $p(w_{kj} \mid c_i)$ sering kali dikombinasikan dengan *Laplacian Smoothing* untuk mencegah persamaan mendapatkan nilai 0, yang dapat mengganggu hasil klasifikasi secara keseluruhan. Sehingga persamaan $p(w_{kj} \mid c_i)$ dituliskan sebagai (Mitchell, 2005):

$$p(w_{kj} \mid c_i) = \frac{f(w_{kj}, c_i) + 1}{f(c_i) + |W|} \quad \textbf{(2.5)}$$

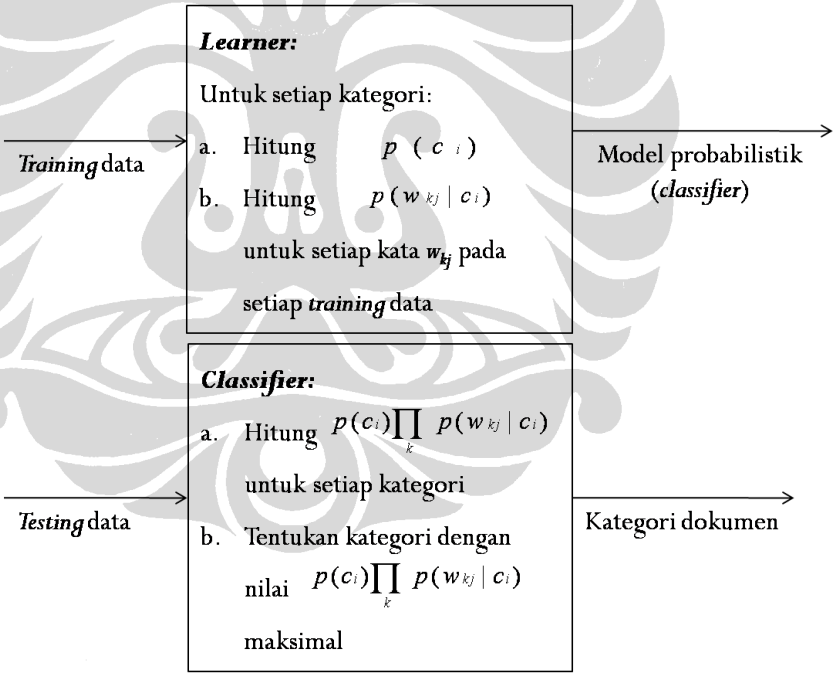
dengan $|W|$ merupakan jumlah keseluruhan kata/fitur yang digunakan.

Pemberian kategori dari sebuah dokumen dilakukan dengan memilih nilai *c* yang memiliki nilai $p(C = c_i \mid D = d_j)$ maksimum, dan dinyatakan dengan:

$$c_* = \arg\max_{c_i \in C} p(c_i \mid d_j) \qquad \textbf{(2.6)}$$

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

Kategori *c** merupakan kategori yang memiliki nilai $p(C = c_i \mid D = d_j)$ maksimum. Nilai $p(D = d_j)$ tidak mempengaruhi perbandingan, karena untuk setiap kategori nilainya akan sama. Berikut ini gambaran proses klasifikasi dengan algoritma Naïve Bayes:



Gambar 2.1 Tahapan Algoritma Klasifikasi Naïve Bayes

2.4 Expectation Maximization

Pada tugas akhir ini digunakan algoritma Expectation Maximization yang dikombinasikan dengan algoritma Naïve Bayes untuk melakukan pengklasifikasian

dokumen teks. Bila Naïve Bayes merupakan algoritma *fully supervised learning*, maka Expectation Maximization merupakan algoritma *semi supervised learning* karena memanfaatkan dokumen-dokumen yang belum diketahui kategorinya atau dapat disebut *unlabeled documents*. Expectation Maximization merupakan algoritma iteratif untuk memperkirakan *maximum likelihood* (kemiripan) pada permasalahan-permasalahan dengan data yang tidak lengkap (Dempster, 1977).

Expectation Maximization memanfaatkan *labeled documents* D_l dan *unlabeled documents* D_u sebagai dokumen *training* $D = D_l \; U \; D_u$. Untuk membangun sebuah model probabilistik θ, diperlukan perhitungan probabilitas kedua himpunan dokumen tersebut. Nilai probabilitas satu buah *unlabeled document* adalah jumlah dari total probabilitas pada semua kategori, sedangkan *labeled documents* telah diketahui kategorinya sehingga nilai probabilitas untuk satu *labeled document* hanya dihitung berdasarkan kategorinya tersebut. Olah karena itu, probabilitas untuk keseluruhan dokumen *training* dihitung dengan persamaan berikut (Nigam, 1999):

$$p(D \mid \vartheta) = \prod_{d_j \in D_u} \sum_{i=1}^{|C|} p(c_i \mid \vartheta)\, p(d_j \mid c_i; \vartheta) \times \prod_{d_j \in D_l} p(C = c_i \mid \vartheta)\, p(d_j \mid C = c_i; \vartheta) \quad \textbf{(2.7)}$$

dengan $|C|$ adalah jumlah semua kategori.

Expectation Maximization bekerja dengan memaksimalkan nilai $p(\vartheta \mid D)$. Biasanya perhitungan untuk memaksimalkan nilai $p(\vartheta \mid D)$ dilakukan dengan menghitung $\log(p(\vartheta \mid D))$, sehingga perhitungan *log likelihood* dapat dilakukan dengan $l(\vartheta \mid D) \equiv \log(p(\vartheta)\, p(D \mid \vartheta))$. Berdasarkan persamaan (2.7), maka perhitungan *log likelihood* dapat dituliskan sebagai berikut (Nigam, 1999):

$$l(\vartheta \mid D) = \log(p(\vartheta)) + \sum_{d_j \in D_u} \log \sum_{i=1}^{|C|} p(c_i \mid \vartheta)\, p(d_j \mid c_i; \vartheta) \quad \textbf{(2.8)}$$

$$+\sum_{d_j \in D_i} \log(p(C=c_i \mid \vartheta)\, p(d_j \mid C=c_i;\vartheta))$$

Pada persamaan (2.8) masih terdapat proses perthitungan *log* dari penjumlahan nilai perkalian *prior* dan *conditional probability*. Hal tersebut akan mempersulit proses komputasi pada saat *maximization step*. Oleh karena itu, kategori dari semua dokumen *training* direpresentasikan sebagai matriks biner *z*, $z_j = \langle z_{j1}, z_{j2}, \ldots, z_{j|C|} \rangle$ dimana $z_{ji} = 1$ jika $C=c_i$ dan 0 jika $C \neq c_i$, sehingga dapat diperoleh persamaan untuk *complete log likelihood* sebagai berikut (Nigam, 1999):

$$l(\vartheta \mid D; z) = \log(p(\vartheta)) + \sum_{d_j \in D} \sum_{i=1}^{|C|} z_{ji} \log(p(c_i \mid \vartheta)\, p(d_j \mid c_i;\vartheta)) \quad \textbf{(2.9)}$$

matriks z_{ji} adalah matriks yang akan coba diestimasi.

Expectation Maximization memiliki dua tahapan dalam melakukan klasifikasi dokumen teks, yang pertama adalah *expectation step* yaitu tahapan memperkirakan kategori dari setiap *unlabeled documents* dengan menentukan probabilitas $p(c_i \mid d_j)$ dengan perhitungan (Bing Liu, 2002):

$$p(c_i \mid d_j) = \frac{p(c_i)\prod_{k=1}^{|dj|} p(w_{kj} \mid c_i)}{\sum_{r=1}^{|C|} p(c_r)\prod_{k=1}^{|dj|} p(w_{kj} \mid c_r)} \quad \textbf{(2.10)}$$

dengan $|d_j|$ merupakan jumlah kata pada dokumen d_j.

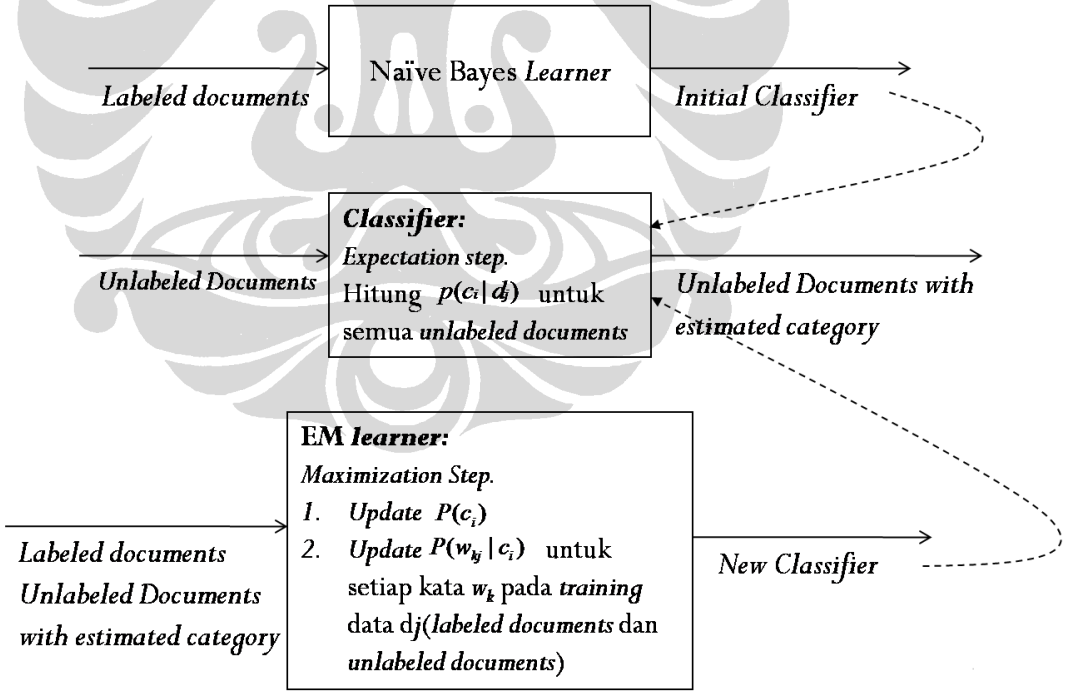
Setelah semua *unlabeled documents* memiliki kategori perkiraan, tahap berikutnya adalah *maximization step* yaitu tahap untuk melakukan *update* terhadap parameter klasifikasi yaitu probabilitas $p(w_{kj} \mid c_i)$ dan probabilitas $p(c_i)$ dengan perhitungan (Bing Liu, 2002):

$$p(w_{kj} \mid c_i) = \frac{1 + \sum_{j=1}^{|D|} N(w_{kj}, d_j) p(c_i \mid d_j)}{|W| + \sum_{s=1}^{|W|} \sum_{j=1}^{|D|} N(w_s, d_j) p(c_i \mid d_j)} \quad \textbf{(2.11)}$$

dan

$$p(c_i) = \frac{1 + \sum_{j=1}^{|D|} p(c_i \mid d_j)}{|C| + |D|} \quad \textbf{(2.12)}$$

dengan $N(w_{kj}, d_j)$ adalah nilai kata atau fitur 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. Proses yang terjadi pada algoritma Expectation Maximization dapat dilihat pada Gambar 2.2.



Gambar 2.2 Tahapan Algoritma Klasifikasi Expectation Maximization

2.5 Klasifikasi Dokumen Bahasa Indonesia

Percobaan klasifikasi dokumen pada tugas akhir ini merupakan kelanjutan dari percobaan klasifikasi topik yang telah dilakukan sebelumnya (Dyta, 2009). Pada penelitian tersebut dilakukan beberapa eksperimen klasifikasi dokumen dengan menggunakan algoritma Naïve Bayes, Naïve Bayes Multinomial, dan Maximum Entropy. Percobaan tersebut dikhususkan untuk melakukan klasifikasi dokumen-dokumen berbahasa Indonesia.

Percobaan klasifikasi topik tersebut menggunakan dua buah kumpulan dokumen, yang pertama adalah kumpulan artikel media massa dari kompas.com yang memiliki lima buah kategori yaitu: ekonomi, olahraga, kesehatan, properti, dan travel dengan jumlah total keseluruhan dokumen mencapai 1240 buah. Kumpulan dokumen yang kedua adalah kumpulan abstrak tulisan ilmiah yang memiliki tiga buah kategori yaitu: *Information Retrieval* (IR), Pengolahan Citra, dan Rekayasa Perangkat Lunak (RPL) dengan jumlah total dokumen mencapai 350 buah.

Selain membandingkan penggunaan metode klasifikasi, percobaan klasifikasi topik dilakukan untuk mengamati beberapa aspek seperti banyaknya jumlah topik yang digunakan, banyaknya jumlah fitur yang digunakan, serta informasi fitur yang digunakan. Pada percobaan menggunakan artikel media massa dan abstrak tulisan ilmiah dilakukan variasi penggunaan jumlah topik sebanyak 2, 3, 4, dan 5 buah topik. Variasi jumlah fitur yang digunakan ada tiga, yaitu 2000, 5000, dan semua fitur dengan variasi informasi fitur *presence*, *frequency*, dan *frequency normalized*.

Selain beberapa beberapa aspek di atas, percobaan klasifikasi topik juga membandingkan hasil klasifikasi topik dari segi keseragaman data yaitu keseragaman jumlah data *training* untuk setiap kategori dan aspek kemiripan data yaitu dengan menggunakan data yang memiliki tingkat kemiripan rendah dari artikel media massa dan data yang memiliki tingkat kemiripan tinggi dari abstrak tulisan ilmiah dalam bidang *computer science*. Namun, dalam percobaan klasifikasi topik tersebut hanya menggunakan *labeled documents* untuk tahap *training*-nya, masalahnya untuk

mendapatkan *labeled documents* dalam jumlah besar akan menghabiskan waktu dan tenaga. Oleh karena itu, tugas akhir ini dilakukan untuk melihat seberapa besar manfaat *unlabeled documents* dalam klasifikasi dokumen teks.