BAB II
TINJAUAN PUSTAKA

Dalam tinjauan pustaka ini dibagi atas pembahasan penelitian terdahulu, logo dan *client brief, text mining*, dan NBC.

2.1 Logo dan Client Brief

Logo adalah sebuah simbol yang mangandung banyak arti dan filosofi. Logo menjadi sebuah pengakuan, kebanggaan, inspirasi kepercayaan, kehormatan, kesuksesan, loyalitas dan keunggulan yang tersirat ke dalam suatu bentuk atau gambar. Logo merupakan bagian yang penting untuk menunjukkan keberadaan sesuatu, sebagai media tertua dalam penyampaian identitas, logo diyakini memberikan efek pengakuan tertentu kepada setiap orang yang melihat atau memakai. *Design* logo biasanya dibuat berdasarkan kesesuaian dengan Visi Misi, standar layanan & sejarah perusahaan, yang intinya sebuah logo dapat menggambarkan karakter pada perusahaan/lembaga/produk/ hal lain yang membutuhkan adanya logo.

Logo dapat diibaratkan dengan wajah. Setiap orang dapat dengan mudah dikenali antara satu dengan yang lain hanya dengan melihat wajah. Begitu juga halnya dengan logo. Logo merupakan sebuah visi penyampaian citra positif melalui sebuah tampilan sederhana dalam bentuk simbol. *Client brief* atau risalah klien ini biasa didapatkan ketika akan mendapatkan pekerjaan dari klien. Dalam dunia desain, *brief* ini sudah tidak asing lagi, karena memang sangat dibutuhkan untuk dapat menentukan desainnya akan dibuat seperti apa. Selain itu *brief* ini pun berfungsi untuk mengurangi sisi subjektif dari pekerjaan itu, karena pekerjaan desain memang ada unsur subjektif seperti nilai estetis, dan keinginan *client*[8].

2.2 Text Mining

Text mining dapat didefinisikan sebagai proses mendapatkan informasi secara intensif dimana pengguna berinteraksi dengan koleksi-koleksi dokumen menggunakan seperangkat *tools* analisis. Secara umum proses-proses pada *textmining* adalah mengadopsi dari proses *data mining*.

Proses-proses utama pada *text mining* diantaranya pemrosesan awal *text* (*text preprocessing*), penemuan pola (*pattern discovery*), transformasi teks (*text transformation*), pemilihan fitur (*feature selection*). Dalam memberikan solusi, *text mining* mengadopsi dan mengembangkan banyak teknik dari bidang lain, seperti *data mining*, *information retrieval*, statistik dan matematik, *machine learning, linguistic*, *natural languange processing*, dan *visualization*.

Salah satu implementasi dari *text mining* adalah tahap *preprocessing text*. Tahap *preprocessing* adalah tahapan dimana sistem melakukan seleksi data yang akan diproses pada setiap dokumen. Kemudian tahap yang selanjutnya adalah melakukan processing. Tahap ini merupakan tahap inti dimana setiap kata akan diolah dengan algoritma tertentu sehingga mempunyai bobot terhadap setiap dokumen yang akan diseleksi. Tahap ini sering disebut juga dengan *analizing*.

Teks yang akan dilakukan proses *text mining*, pada umumnya memiliki beberapa karakteristik diantaranya adalah memiliki dimensi yang tinggi, terdapat *noise* pada data. Cara yang digunakan dalam mempelajari suatu data teks, adalah dengan terlebih dahulu menentukan fitur-fitur yang mewakili setiap kata untuk setiap fitur yang ada pada dokumen. Sebelum menentukan fitur-fitur yang mewakili, diperlukan tahap *preprocessing* yang dilakukan secara umum dalam *text mining* pada dokumen, yaitu *case folding, tokenizing,* dan *filtering*.

Metode yang dapat digunakan pada kasus teks mining cukup beragam, seperti dengan penggunaan tipe *probabilistic* (NBC), *decision trees, decision rules, regresion, VSM*, dan masih banyak lagi. Pemilihan metode dapat dilihat pada permasalahan yang akan diselesaikan, apakah memerlukan data latih atau tidak, kecepatan atau ketepatan yang diutamakan, dan lain sebagainya yang harus dipertimbangkan dalam pemilihan metode.

2.2.1 Case folding

Case folding adalah mengubah semua huruf dalam dokumen menjadi huruf kecil. Hanya huruf "a" sampai dengan "z" yang diterima, karakter selain huruf dihilangkan [9].

2.2.2 Tokenizing

Tahap *tokenizing* adalah tahap pemotongan string *input* berdasarkan tiap kata yang menyusunnya. Tahapan ini tidak harus dilakukan apabila terdapat pengolahan pada bentuk frase yaitu dua kata yang bermakna dalam satu arti [9].

2.2.3 Filtering

Tahap *filtering* adalah tahap mengambil kata-kata penting dari hasil token dapat menggunakan algoritma *stoplist* (membuang kata yang kurang penting) atau *wordlist* (menyimpan kata penting). *Stoplist/stopword* adalah kata-kata yang tidak deskriptif yang dapat dibuang dalam pendekatan *bag-of-words* [9].

2.3 Naïve Bayes Classifier (NBC)

Salah satu tugas *Text Mining* adalah klasifikasi data, yaitu memetakan (mengklasifikasikan) data ke dalam satu atau beberapa kelas yaang sudah didefinisikan sebelumnya. Salah satu metoda dalam klasifikasi data adalah *Naïve Bayes Classifier* (NBC).

Naïve Bayes Classifier merupakan salah satu metoda *machine learning* yang memanfaatkan perhitungan probabilitas dan statistik yang dikemukakan oleh ilmuwan Inggris Thomas Bayes, yaitu memprediksi probabilitas di masa depan berdasarkan pengalaman di masa sebelumnya [1].

Sebuah keuntungan dari NBC adalah bahwa ia memerlukan sejumlah kecildata pelatihan untuk mengestimasi parameter (rata-rata dan varian dari variabel)yang diperlukan untuk klasifikasi. Karena variabel diasumsikan independen, hanya varian dari variabel-variabel untuk setiap kelas yang perlu ditentukan [10].

2.2.3.1 Model Probabilistic *Naïve Bayes*

Model probabilitas untuk classifier adalah model kondisional

$p(C|F_1,\ldots,F_n)$ ..... (2.1)

terhadap variabel kelas dependen C dengan sejumlah kecil hasil atau kelas,tergantung pada beberapa variabel fitur F1 sampai Fn. Masalahnya adalah bahwajika jumlah fitur n besar atau bila fitur bisa mengambil sejumlah besar nilai, makamembuat sebuah model pada tabel probabilitas adalah tidak mungkin. Oleh karenaitu kita mereformulasi model untuk membuatnya lebih fleksibel[10].

Menggunakan teorema Bayes , dapat ditulis

$p(C|F1,\ldots..,Fn) = \frac{p(c)p(F1,\ldots,Fn|C)}{p(F1,\ldots,Fn)}$ ..... (2.2)

Dalam bahasa Inggris persamaan di atas dapat ditulis sebagai

posterior = $\frac{prior \ \ X \ likeli \ hood}{evidence}$ ..... (2.3)

Dalam prakteknya kita hanya tertarik pada pembilang dari persamaan tersebut, karena penyebut tidak tergantung pada C dan nilai-nilai fitur Fi diberikan,sehingga penyebut secara efektif konstan. Pembilang ini setara dengan modelprobabilitas gabungan $p(C|F_1,\ldots,F_n)$ yang dapat ditulis ulang sebagai berikut,menggunakan penggunaan berulang dari definisi probabilitas bersyarat:

$p(C|F_1,\ldots,F_n)$

$= p(C)\ p(F_1,\ldots,F_n|C)$

$= p(C)\ p(F_1|\mathrm{C})\ p(F_2,\ldots,F_n|C,\ F_1)$

$= p(C)\ p(F_1|\mathrm{C})\ p(F_2|C,F_1)\ p(F_3,\ldots,F_n|C,\ F_1,\ F_2)$

$= p(C)\ p(F_1|\mathrm{C})\ p(F_2|C,F_1)\ p(F_3|C,F_1,F_2)\ p(F_4,\ldots,F_n|C,\ F_1,\ F_2,\ F_3)$

$= p(C)\ p(F_1|\mathrm{C})\ p(F_2|C,F_1)\ p(F_3|C,F_1,F_2)\ldots\ p(F_n|C,\ F_1,\ F_2,\ F_3,\ldots,\ F_{n-1})$ ..... (2.4)

Sekarang asumsi kemandirian bersyarat yang "naif" memegang peranan.Menganggap bahwa setiap fitur Fi adalah secara kondisi independen terhadapsetiap fitur lainnya Fj untuk j ≠ i. Ini berarti bahwa

$p(F_i|\mathrm{C},\ F_j) = p(F_i|\mathrm{C})$ ..... (2.5)

untuk i ≠j , sehingga *joint model* dapat dinyatakan sebagai

$p(C|F_1,\ldots,F_n) = p(C)\, p(F_1|C)\, p(F_2|C)\, p(F_3|C)\ldots$

$= p(C)\prod_{i=1}^{n} p(Fi|C)$ ..... (2.6)

Ini berarti bahwa di bawah asumsi independen di atas, distribusi bersyarat dari

variabel kelas C dapat dinyatakan seperti ini :

$p(C|F_1,\ldots,F_n) = \frac{1}{z}\, p(C) \prod_{i=1}^{n} p(Fi|C)$ .....(2.7)

dimana Z (bukti) adalah faktor skala tergantung hanya pada *F1*,…,*Fn* , yaitu,sebuah konstanta jika nilai dari variabel fitur diketahui.

Model dari bentuk ini jauh lebih mudah dikelola, karena mereka memecahmenjadi *class prior* p(C) dan distribusi probabilitas independen *p*(*Fi*|C). Jika adak kelas dan jika model untuk masing-masing *p*(*Fi*|C = c) dapat dinyatakan dalambentuk parameter, maka model naif Bayes yang sesuai memiliki (k - 1) + n r kparameter. Dalam prakteknya, sering k = 2 (klasifikasi biner) dan r = 1 (variabel Bernoulli sebagai fitur) yang umum, sehingga jumlah parameter model Bayes naif
adalah 2n + 1, dimana n adalah jumlah fitur biner yang digunakan untukklasifikasi dan prediksi [1].

2.2.3.2 *Naïve Bayes Classification* untuk klasifikasi dokumen multikelas

Pada NBC setiap record direpresentasikan dalam pasangan atribut < a_1,$a_{2,\ldots,}$ a_n > dimana a_1 adalah kata pertama, a_2 adalah kata kedua dan seterusnya.Sedangkan V adalah himpunan kategori dokumen [1].

Pada saat klasifikasi, pendekatan Bayes akan menghasilkan label kategoriyang paling tinggi probabilitasnya (V_{MAP}) dengan masukan atribut < a_1, $a_{2,\ldots,}$ a_n >

$V_{MAP} = \arg_{vj \epsilon V} \max P(v_j \mid a_1, a_{2,\ldots,} a_n)$ ..... (2.23)

Teorema Bayes menyatakan

$$P(B \mid A) = \frac{P(A|B)P(B)}{P(A)} \quad (2.24)$$

Menggunakan teorema Bayes ini, persamaan (2.23) ini dapat ditulis :

$$V_{MAP} = \arg_{vj \epsilon V} \max \frac{P(a1,a2,\ldots an)P(Vj)}{P(a1,a2,\ldots an)} \quad (2.25)$$

$P(\ a_1, a_{2,\ldots,} a_n\ )$ nilainya konstan untuk semua v_j sehingga persamaan ini dapatditulis sebagai berikut:

$$V_{MAP} = \arg_{vj \epsilon V} \max P(a_1, a_{2,\ldots,} a_n| v_j) P(v_j) \quad (2.26)$$

Tingkat kesulitan menghitung $P(a_1, a_{2,\ldots,} a_n| v_j)$ menjadi tinggi karenajumlah term $P(a_1, a_{2,\ldots,} a_n| v_j)$ bisa jadi akan sangat besar. Ini disebabkan jumlahterm tersebut sama dengan jumlah kombinasi posisi kata dikali dengan jumlahkategori.

Naïve Bayes Classifier menyederhanakan hal ini dengan mengasumsikanbahwa dalam setiap kategori, setiap kata independen satu sama lain. Dengan katalain :

$$P(a_1, a_{2,\ldots,} a_n| v_j) = \Pi_i P(a_i| v_j) \quad (2.27)$$

Substitusi persamaan ini dengan persamaan 3.4 akan menghasilkan :

$$V_{MAP} = \arg_{vj \epsilon V} \max P(v_j) \Pi_i P(a_i| v_j) \quad (2.28)$$

$P(v_j)$ dan probabilitas kata w_k untuk setiap kategori $P(w_k| v_j)$ dihitung pada saat

pelatihan.

$$P(v_j) = \frac{|docs\ j|}{|Conto\ h|} \quad (2.29)$$

$$P(w_k| v_j) = \frac{nk+1}{n+|kosakata\ |} \quad (2.30)$$

Di mana |*docs*$_j$| adalah jumlah dokumen pada kategori j dan |*Contoh*| adalah jumlah dokumen yang digunakan dalam pelatihan. Sedangkan n_k adalah jumlah kemunculan kata w_k pada kategori v_j dan |*kosakata*| adalah jumlah kata yang unik(*distinct*) pada semua data latihan. Jumlah kata dalam tiap kelas dinyatakan sebagai n.

Ringkasan algoritma untuk *Naïve Bayes Classifier* adalah sebagai berikut [1]:

A. Proses pelatihan. Input adalah dokumen-dokumen contoh yangtelah diketahui kategorinya.

1. Kosakata ← himpunan semua kata yang unik dari dokumen-dokumencontoh
2. Untuk setiap kategori vj lakukan :
   a. Docsj ← Himpunan dokumen-dokumen yang berada pada kategori vj
   b. Hitung *P*(vj) dengan persamaan 2.29
   c. Untuk setiap kata wk pada kosakata lakukan :
3. Hitung P(wk |vj) dengan persamaan 2.30

B. Proses klasifikasi. Input adalah dokumen yang belum diketahuikategorinya :

Hasilkan v*MAP* sesuai dengan persamaan 2.28 dengan menggunakan P(vj)dan P(wk |vj) yang telah diperoleh dari pelatihan.

2.4 Alat Bantu Perancangan Sistem

Perancangan sistem sangat perlu dalam pengolahan data dan informasi, karena tanpa dirancang terlebih dahulu maka data yang diolah untuk informasi tidak akan sempurna. Dalam perancangan sistem ini, terdapat dua alat bantu yang digunakan, antara lain DFD, dan pengujian keakuratan.

2.4.1 Data Flow Diagram (DFD)

Data Flow Diagram (DFD) atau diagram alir data adalah sebuah teknik grafis yang menggambarkan aliran informasi dan tranformasi yang

diaplikasikan pada saat data bergerak dari *input* menjadi *output*. DFD dapat digunakan untuk menyajikan sebuah sistem atau perangkat lunak pada setiap tingkat abstraksi.

DFD dapat dipartisi ke dalam tingkat–tingkat yang merepresentasikan aliran informasi yang bertambah dan fungsi ideal. Tingkatan–tingkatan yang ada pada DFD, yaitu [11]:

1. Diagram Konteks

   Diagram konteks menggambarkan ruang lingkup sistem untuk memberikan pandangan umum sistem. Diagram konteks merupakan level tertinggi dari DFD.

2. Diagram level satu

   Tingkat yang lebih bawah dari diagram konteks adalah diagram level satu atau DFD level 1. Diagram level satu menggambarkan proses–proses utama dari sistem.

3. Diagram Level n

   Diagram level n adalah hasil dekomposisi dari diagram level 1. Diagram level n menjelaskan proses secara lebih terperinci. Diagram level 2 merupakan turunan langsung dari diagram level 2, artinya diagram level 2 berada satu tingkat lebih rendah dari diagram level satu. Apabila diagram level 2 ini diuraikan lagi, maka akan terbentuk diagram level 3, dan seterusnya.

2.5 Pengujian Keakuratan

Dalam bidang pengenalan pola (*pattern recognition*) dan temu kembali informasi (*information retrieval*), *precision* dan *recall* adalah dua perhitungan yang banyak digunakan untuk mengukur kinerja dari sistem/metode yang digunakan[7]. *Precision* atau presisi adalah tingkat

ketepatan antara informasi yang diminta oleh pengguna dengan jawaban yang diberikan oleh sistem, sedangkan *recall* adalah tingkat keberhasilan sistem dalam menemukan kembali sebuah informasi. Ukuran *precision* dan *recall* biasanya diberi nilai dalam bentuk persentase, 1 sampai 100%.

Sebuah sistem informasi akan dianggap baik jika tingkat *recall* maupun *precision*-nya tinggi, sebagai contoh ketika dibutuhkan pencarian dokumen tentang "*codeiginter*" pada sebuah sistem, dan jika sistem tersebut memiliki 100 buku tentang *codeiginter*, maka kinerja terbaik adalah jika sistem tersebut berhasil menemukan 100 dokumen tentang *Codeiginter*. Apabila sistem tersebut memberikan 100 temuan, dan di temuan tersebut ada 50 dokumen tentang "*codeiginter*", maka nilai *recall*-nya adalah 0,5 (atau 50%) dan nilai *precision*-nya juga 0,5. Jika sistem tersebut memberikan 1 dokumen saja, dan dokumen tersebut adalah tentang "*codeiginter*", maka *recall*-nya bernilai 0,01 dan *precision*-nya bernilai 100. Nilai *precision* yang tinggi ini sebenarnya terjadi karena sistem memberikan hanya 1 jawaban kepada pencari informasi. Apabila sistem memberikan 100 dokumen, dan hanya 1 yang relevan, maka nilai *recall*-nya tetap 0,01 dan *precision*-nya pun ikut merosot ke 0,01.

Pada bidang lain seperti statistika dikenal juga istilah *accuracy*. *Accuracy* didefinisikan sebagai tingkat kedekatan antara nilai prediksi dengan nilai aktual. Secara umum pengujian yang dilakukan pada suatu sistem temu kembali informasi adalah ketiga hal pokok tersebut, sehingga dapat menunjukkan keakuratan kinerja sistem lebih baik. Untuk menghitung nilai presisi, *recall*, dan akurasi dapat menggunakan *confusion matrix* seperti pada Tabel 2.2 [7].

Tabel 2.1 Confusion Matrix

	Relevant	*Nonrelevant*
Retrieved	*True positive (tp)*	*False positive (fp)*
Non retrieved	*False negatif (fn)*	*True negatives (tn)*

$$Precision = \frac{tp}{(tp+fp)} \quad \text{.....} \quad (2.31)$$

$$Recall = \frac{tp}{(tp+fn)} \text{.....} \quad (2.32)$$

$$Accuracy = \frac{(tp+tn)}{(tp+fp+tn+fn)} \text{.....} \quad (2.33)$$

Dimana,

tp = Dokumen yang ditemukan dan relevan.

fp = Dokumen yang ditemukan dan tidak relevan.

fn = Dokumen yang tidak ditemukan dan relevan.

tn = Dokumen yang tidak ditemukan dan tidak relevan