



### **Hak cipta dan penggunaan kembali:**

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

### **Copyright and reuse:**

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

## BAB II

### LANDASAN TEORI

#### 2.1 Text Mining

*Text mining* merupakan teknik yang digunakan untuk menangani masalah klasifikasi, *clustering*, *information extraction*, dan *information retrieval* (Berry & Kogan, 2010). Perbedaan *text mining* dan *data mining* adalah sumber data yang digunakan. Sumber data yang digunakan pada *data mining* adalah data yang terstruktur, sedangkan sumber data yang digunakan pada *text mining* adalah data yang tidak terstruktur berupa teks (Surahman, 2013). Salah satu tahap dalam *text mining* adalah *text pre-processing*. *Text pre-processing* adalah proses perubahan bentuk data yang belum terstruktur menjadi data yang terstruktur sesuai dengan kebutuhan. Tahap-tahap pada *text pre-processing* secara umum adalah *case-folding*, *tokenizing*, *filtering*, dan *stemming* (Berry & Kogan, 2010).

##### 2.1.1 Case Folding

*Case folding* adalah proses mengubah semua huruf dalam dokumen menjadi huruf kecil (Triawati, 2009). Hal ini disebabkan karena tidak semua artikel teks konsisten dalam penggunaan huruf kapital. Tabel 2.1 menunjukkan contoh dari proses *case folding*.

Tabel 2.1 Contoh Proses *Case Folding*  
(Indraloka & Santosa, 2017)

Sebelum Case Folding	Setelah Case Folding
Algoritma	algoritma
Self	self
Organizing	organizing
Map	map
SOM	som
diperkenalkan	diperkenalkan
oleh	oleh
Professor	professor

Tabel 2.1 Contoh Proses *Case Folding* (Lanjutan)  
(Indraloka & Santosa, 2017)

Sebelum Case Folding	Setelah Case Folding
Teuvo	teuvo
pada	pada
tahun	tahun

### 2.1.2 Tokenizing

*Tokenizing* adalah tahap pemotongan teks menjadi kata, istilah, simbol, tanda baca, atau elemen lain yang memiliki arti yang disebut token (Janani & Vijayarani, 2016). Pada proses *tokenizing*, token yang merupakan tanda baca yang dianggap tidak perlu dan angka dihapus, sehingga hanya tersisa alfabet. Tabel 2.2 menunjukkan contoh dari proses *tokenizing*.

Tabel 2.2 Contoh Proses *Tokenizing*  
(Indraloka & Santosa, 2017)

Sebelum Tokenizing	Setelah Tokenizing
Algoritma Self Organizing Map (SOM) diperkenalkan oleh Professor Teuvo pada tahun 1981.	Algoritma
	Self
	Organizing
	Map
	SOM
	diperkenalkan
	oleh
	Professor
	Teuvo
	pada
tahun	

### 2.1.3 Filtering

*Filtering* adalah tahap mengambil kata-kata penting dari hasil token menggunakan *stopwords* (Putri, 2017). *Stopwords* adalah kata yang bukan merupakan kata unik dalam suatu artikel atau kata-kata umum yang biasanya selalu ada dalam suatu artikel (Mooney, 2006) Contoh kata yang termasuk *stopwords*

adalah “yang”, “dan”, “di”, “dari”, dan sebagainya (Tala, 2003). Tabel 2.3 menunjukkan contoh proses *filtering*.

Tabel 2.3 Contoh Proses *Filtering*  
(Indraloka & Santosa, 2017)

Sebelum Filtering	Setelah Filtering
algoritma	algoritma
self	self
organizing	organizing
map	map
som	som
diperkenalkan	diperkenalkan
oleh	
professor	professor
teuvo	teuvo
pada	
tahun	tahun

#### 2.1.4 Stemming

*Stemming* adalah proses pemetaan variasi morfologikal kata dalam kata dasar atau kata umumnya (Adhitia & Purwarianti, 2012). Tahap ini biasanya dipakai untuk teks dalam bahasa Inggris, namun lebih sulit diterapkan pada teks berbahasa Indonesia. Hal ini dikarenakan bahasa Indonesia tidak memiliki rumus baku (Hamzah, 2013). Proses *stemming* pada teks berbahasa Indonesia berbeda dengan *stemming* pada teks berbahasa Inggris. Pada teks bahasa Inggris, proses yang diperlukan hanya proses menghilangkan *suffix* (akhiran). Sedangkan pada teks berbahasa Indonesia, selain *suffix*, *prefix* (awalan), *infix* (sisipan), dan *confixes* (gabungan awalan akhiran) juga perlu dihilangkan (Wahyudi, dkk., 2017). Tabel 2.4 menunjukkan contoh proses *stemming*.

Tabel 2.4 Contoh Proses *Stemming*  
(Indraloka & Santosa, 2017)

Sebelum Stemming	Setelah Stemming
diperkenalkan	kenal
memakan	makan
memetakan	peta

Salah satu algoritma *stemming* bahasa Indonesia adalah algoritma Nazief dan Adriani. Algoritma ini diterapkan oleh Sastrawi Stemmer, sebuah library *stemmer* sederhana berbasis Nazief dan Adriani, mengutip dari Hidayatullah, dkk. (2016), langkah-langkahnya adalah sebagai berikut.

1. Kata yang belum di-*stemming* dicari pada kamus, jika ditemukan, kata tersebut dianggap sebagai kata dasar yang benar dan algoritma dihentikan.
2. Hilangkan *inflectional suffixes* (partikel “-lah”, “-kah”, “-tah”, atau “-pun”), kemudian hilangkan *inflectional possessive pronoun suffixes* (partikel “-ku”, “-mu”, “-nya”). Cek kata di dalam kamus kata dasar, jika ditemukan, algoritma dihentikan. Jika tidak, lanjut ke langkah 3.
3. Hapus *derivational suffix* (“-i” atau “-an”). Jika kata ditemukan dalam kamus kata dasar, maka algoritma dihentikan. Jika tidak, maka lanjut ke langkah 3a.
  - a. Jika akhiran “-an” telah dihapus dan huruf terakhir dari kata tersebut adalah “-k”, maka “-k” juga dihapus. Jika kata tersebut ditemukan dalam kamus, maka algoritma berhenti. Jika tidak, maka lakukan langkah 3b.
  - b. Akhiran yang dihapus (“-i”, “-an”, atau “-kan”) dikembalikan lanjut ke langkah 4.
4. Hapus *derivational prefix* (“be-“, “di-“, “ke-“, “me-“, “pe-“, “se-“, dan “te”) dengan melakukan langkah berikut.
  - a. Apabila awalan telah dieliminasi pada langkah ketiga, lakukan pengecekan kombinasi awalan dan akhiran yang tidak diizinkan.

Kombinasi awalan dan akhiran yang tidak diizinkan terdapat pada Tabel 2.5.

Tabel 2.5 Kombinasi Awalan dan Akhiran yang Tidak Diizinkan (Hidayatullah, dkk., 2016)

Awalan	Akhiran
be-	-i
di-	-an
ke-	-i, -kan
me-	-an
te-	-i, -kan
se-	-an

- b. Algoritma diberhentikan apabila awalan yang dideteksi sama dengan awalan yang dihilangkan sebelumnya.
- c. Algoritma dihentikan apabila tiga awalan telah dihilangkan.
- d. Tipe awalan yang dikenali sebagai ketentuan berikut.
  - 1) Apabila awalan adalah “di-”, “ke-”, atau “se-” maka tipe awalan adalah “di”, “ke”, atau “se” secara berurutan
  - 2) Apabila awalan adalah “te-”, “be-”, “me-”, atau “pe-”, diperlukan proses ekstra untuk mengekstraksi set karakter untuk menentukan jenis awalan.
  - 3) Algoritma dikembalikan ketika dua karakter pertama tidak cocok dengan “di-”, “ke-”, “se-”, “te-”, “me-”, atau “pe-”.
- e. Apabila tidak ada tipe awalan, maka algoritma akan dikembalikan.

Untuk tipe awalan seperti yang ditemukan pada Tabel 2.6, awalan akan dihilangkan.

Tabel 2.6 Ketentuan Tipe Awalan yang Dihilangkan (Hidayatullah, dkk., 2016)

Tipe Awalan	Awalan yang dihilangkan
di	di-
ke	ke-

Tabel 2.6 Ketentuan Tipe Awalan yang Dihilangkan (Lanjutan)  
(Hidayatullah, dkk., 2016)

Tipe Awalan	Awalan yang dihilangkan
se	se-
te	te-
ter	ter-
ter-luluh	ter-

- f. Langkah 4 akan dilakukan secara rekursif apabila kata dasar belum ditemukan.
- g. Lakukan *recording* (penyusunan kembali kata-kata yang mengalami proses *stemming* berlebih).
- h. Jika semua langkah telah dilakukan tetapi kata dasar tidak ditemukan pada kamus, maka algoritma ini mengembalikan kata yang asli sebelum dilakukan *stemming*.

## 2.2 Algoritma Multinomial Naïve Bayes

Klasifikasi Bayesian adalah pengklasifikasian yang didasarkan pada teorema Bayes (Tan, dkk., 2006). Naïve Bayes mengasumsikan bahwa pengaruh dari nilai atribut pada kelas tertentu tidak bergantung pada nilai atribut lainnya. Asumsi ini disebut *class conditional independence*. Asumsi ini dibuat untuk menyederhanakan perhitungan yang dianggap naif.

Menurut Raghavan & Schutze (2008) probabilitas sebuah dokumen  $d$  yang berada di kelas  $c$  dihitung dengan Persamaan 2.1.

$$P(c|d) \propto P(c) \prod_{1 \leq k \leq n_d} P(t_k|c) \dots (2.1)$$

$P(t_k|c)$  adalah *conditional probability* dari term  $t_k$  yang terdapat dalam sebuah dokumen dari kelas  $c$ .  $P(t_k|c)$  diinterpretasikan sebagai ukuran dari berapa banyak petunjuk  $t_k$  membantu dalam menentukan bahwa  $c$  adalah kelas yang tepat.



$P(c)$  adalah *prior probability* dari sebuah dokumen yang terdapat dalam kelas  $c$ . Bila *term* dari sebuah dokumen tidak memberikan petunjuk yang jelas untuk satu kelas dibandingkan kelas lainnya, maka dipilih salah satu kelas yang memiliki *prior probability* yang tertinggi.

$t_1, t_2, \dots, t_{n_d}$  adalah kumpulan *token* dalam dokumen  $d$  yang merupakan bagian dari kosa kata yang digunakan untuk mengklasifikasi dan  $n_d$  adalah jumlah *token* tersebut di dalam dokumen  $d$ . Contoh  $t_1, t_2, \dots, t_{n_d}$  untuk dokumen dengan satu kalimat *Beijing and Taipei join the WTO*, menjadi (Beijing, Taipei, join, WTO) dengan  $n_d = 4$ , jika *term and* dan *the* dianggap sebagai *stopwords*.

Untuk memperkirakan *prior probability*  $P(c)$  digunakan Persamaan 2.2 berikut (Raghavan & Schutze, 2008).

$$P(c) = \frac{N_c}{N} \quad \dots(2.2)$$

$N_c$  = jumlah dari dokumen *training* dalam kelas  $c$ .

$N$  = jumlah keseluruhan dokumen *training* dari seluruh kelas

Untuk memperkirakan *conditional probability*  $P(t|c)$  persamaan yang digunakan ditunjukkan pada Persamaan 2.3 (Raghavan & Schutze, 2008).

$$P(t_k|c) = \frac{T_{ct}}{\sum_{t' \in V} T_{ct'}} \quad \dots(2.3)$$

$T_{ct}$  = jumlah kemunculan *term*  $t$  dalam sebuah dokumen *training* dari kelas  $c$ .

$\sum_{t' \in V} T_{ct'}$  = jumlah dari total keseluruhan *term* yang terdapat dalam sebuah dokumen *training* dari kelas  $c$ .

Masalah dari proses perkiraan nilai *conditional probabilities* adalah terdapat nilai nol dari sebuah kombinasi (*term|class*) yang tidak terdapat dalam data



*training*. Berdasarkan contoh di atas, bila *term* WTO dalam data *training* hanya terdapat dalam dokumen *China*, maka perkiraan untuk kelas-kelas lainnya, misalnya UK akan bernilai nol (0) seperti dilihat pada Persamaan 2.4 (Raghavan & Schutze, 2008).

$$P(\text{WTO} | \text{UK}) = 0 \quad \dots(2.4)$$

Untuk menghilangkan nilai nol, digunakan *add-one* atau *Laplace smoothing*. Proses ini menambahkan nilai satu pada setiap nilai  $T_{ct}$  dari perhitungan *conditional probabilities*. Sehingga persamaan untuk *conditional probabilities* menjadi seperti yang ditunjukkan pada Persamaan 2.5 (Raghavan & Schutze, 2008).

$$P(t_k|c) = \frac{T_{ct} + 1}{(\sum_{t' \in V} T_{ct'}) + B'} \quad \dots(2.5)$$

$B'$  = jumlah keseluruhan *term* unik dari seluruh kelas.

### 2.3 Confusion Matrix

Pengujian klasifikasi teks dengan algoritma Multinomial Naïve Bayes dilakukan dengan mengukur *precision*, *recall*, dan *F-measure* yang mengacu pada penelitian McCallum dan Nigam (1998) dan Asch (2013). *F-measure* didefinisikan sebagai rata-rata harmonis dari *precision* (P) dan *recall* (R) (Sasaki, 2007). *Precision* adalah jumlah nilai prediksi benar dibagi dengan total hasil benar (Brownlee, 2014). *Recall* adalah jumlah nilai prediksi benar dibagi dengan seluruh nilai pengujian yang relevan (Brownlee, 2014). Secara intuitif, *F-measure* tidak semata dimengerti sebagai akurasi, namun *F-measure* jauh lebih berguna dibanding akurasi, terutama ketika terdapat pendistribusian kelas yang tidak merata (Shung, 2018). Perhitungan *F-measure* menurut Sasaki (2007) dijelaskan pada rumus berikut.

$$F = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad \dots(2.6)$$

Tabel 2.7 berikut merupakan *confusion matrix* yang digunakan untuk mendeskripsikan performa dari model klasifikasi. *Confusion matrix* di bawah menghasilkan nilai *True Positive*, *False Positive*, *False Negative*, dan *True Negative* yang akan digunakan untuk menghitung nilai *precision* dan *recall*.

*True Positive* (TP) merupakan jumlah dokumen dari kelas A yang benar diklasifikasikan sebagai kelas A. *True Negative* (TN) merupakan jumlah dokumen yang bukan merupakan kelas A yang benar diklasifikasikan sebagai bukan kelas A. *False Positive* (FP) merupakan jumlah dokumen yang bukan merupakan kelas A yang salah diklasifikasikan sebagai kelas A. *False Negative* (FN) merupakan jumlah dokumen dari kelas A yang salah diklasifikasikan sebagai bukan kelas A.

Tabel 2.7 *Confusion Matrix* (Brownlee, 2014)

	ACTUAL		
PREDICTED		Positive	Negative
	Positive	True Positive	False Positive
	Negative	False Negative	True Negative

Dengan menggunakan *confusion matrix*, *precision* dan *recall* dapat didefinisikan seperti pada Persamaan 2.7 dan 2.8.

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \quad \dots(2.7)$$

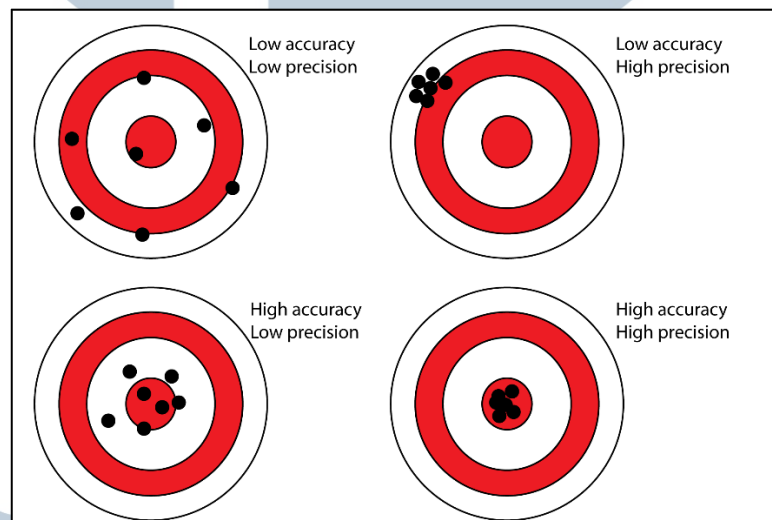
$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad \dots(2.8)$$

Selain *precision* dan *recall*, dapat diperoleh pula nilai akurasi. Nilai akurasi menggambarkan seberapa akurat sistem dapat mengklasifikasikan data secara benar. Dengan kata lain, nilai akurasi merupakan perbandingan antara data yang

terklasifikasi benar dengan keseluruhan data (Achmatim, 2017). Definisi akurasi dapat dilihat pada Persamaan 2.9.

$$Akurasi = \frac{TP+TN}{TP+TN+FP+FN} * 100\% \quad \dots(2.9)$$

Gambar 2.1 menunjukkan ilustrasi perbedaan akurasi dan *precision* pada *dart game* sederhana. Akurasi yang rendah mengakibatkan titik penembakan pada papan target jauh dari target utama, sementara tingkat *precision* yang buruk menambah hasil tembakan tidak terpusat di satu titik (Taylor, 1999). Hasil yang baik dan jelas dalam melakukan pengukuran adalah dengan nilai akurasi dan *precision* yang tinggi sehingga tembakan tepat jatuh di target utama pada papan target serta jatuh pada daerah yang sama jika dilakukan penembakan berulang kali.



Gambar 2.1 Ilustrasi Perbedaan Akurasi dan *Precision* (Taylor, 1999)

## 2.4 MyValue

Loyalitas pelanggan memiliki peran penting dalam sebuah perusahaan, mempertahankan loyalitas pelanggan berarti meningkatkan kinerja keuangan dan mempertahankan kelangsungan hidup perusahaan. Hal ini menjadi alasan utama

bagi perusahaan untuk menarik dan mempertahankan loyalitas pelanggan (Yamin, 2011).

Dikutip dari *website* MyValue (2018), MyValue merupakan program *reward* persembahan Kompas Gramedia untuk para pelanggan. MyValue adalah bentuk transformasi program loyalitas digital dari *membership* berbasis kartu Kompas Gramedia Value Card (KGVC) yang diterbitkan Kompas Gramedia Value Card dan Bank Central Asia (BCA). MyValue ingin memberikan pengalaman berbelanja yang semakin menarik kepada seluruh pelanggan Kompas Gramedia. Aplikasi ini dibangun untuk memudahkan *member* mendapatkan *benefit* poin dari seluruh bisnis Kompas Gramedia, promo menarik dari *merchant* yang bekerja sama dan pengelolaan komunitas berbasis digital. Logo aplikasi MyValue ditunjukkan pada Gambar 2.2.



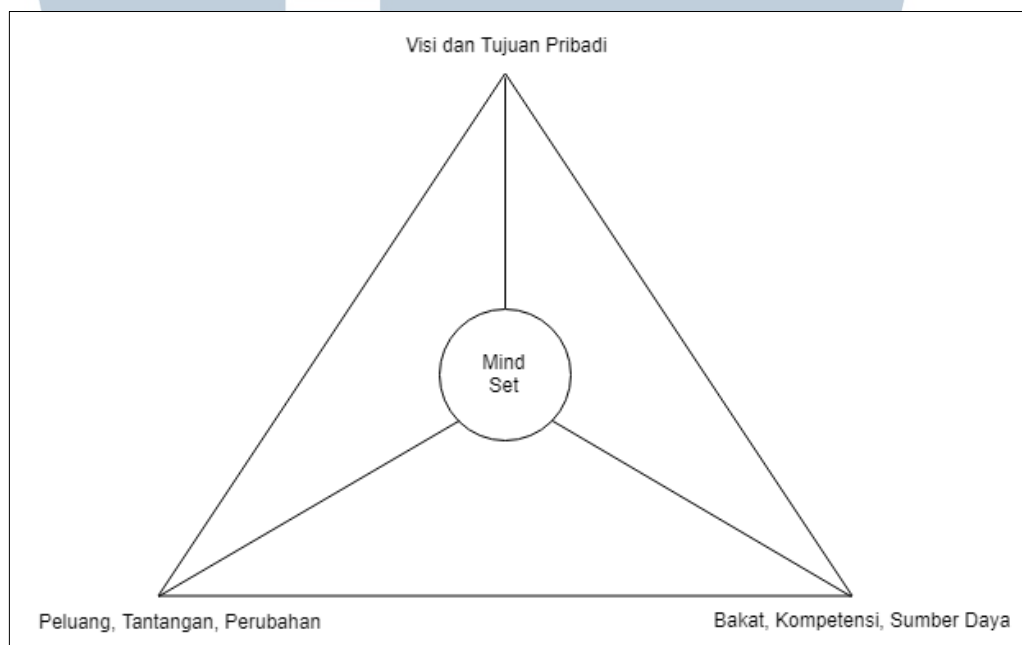
Gambar 2.2 Logo Aplikasi MyValue (MyValue, 2018)

Aplikasi MyValue yang sedang dikembangkan sekarang memiliki beberapa fitur utama selain promo dan *membership*, salah satunya adalah artikel (Danaditya, 2018). Karena Kompas Gramedia adalah perusahaan media massa dan MyValue berfokus pada industri *personal growth*, maka artikel pada aplikasi MyValue dikhususkan untuk konten *personal growth*.

## 2.5 Personal Growth

Secara natur, manusia harus bertumbuh dan berkembang. Kapabilitas dan kapasitasnya harus terus menerus diasah agar mampu memberdayakan kehidupannya dan alam sekitarnya menjadi berkualitas. Pertumbuhan diri (*personal growth*) pada hakekatnya adalah pengembangan kualitas hidup dalam semua aspek yang saling terkait satu dengan yang lainnya (Darmawan, 2015).

Menurut Darmawan (2015), pertumbuhan diri dapat digambarkan melalui tiga unsur dalam segitiga pertumbuhan yang ditunjukkan pada Gambar 2.3.



Gambar 2.3 Segitiga Pertumbuhan Diri (Darmawan, 2015)

Unsur pertama adalah tujuan atau visi hidup. Tanpa visi atau tujuan hidup, tidak mungkin ada upaya pembelajaran, karena tidak ada hal yang dituju dan dikejar dalam kehidupan.

Unsur kedua adalah peluang, tantangan, dan perubahan yang ada di sekitar kita. Perubahan bisa menjadi peluang dan tantangan bagi kita untuk belajar, bertumbuh, dan berkembang. Bagaimana sikap kita dalam berespon

terhadap tantangan dan peluang tersebut sangat menentukan pengembangan diri kita. Unsur ketiga adalah sumber daya yang dimiliki.

Unsur ketiga adalah sumber daya yang dimiliki. Setiap orang punya resources dasar yang dimilikinya: bakat, talenta, kecerdasan, kompetensi dan lain-lain. Ada yang dipercayakan banyak, ada yang sedikit. Tetapi faktor sumber daya ini bukanlah faktor dominan yang menjadi penentu dalam proses pertumbuhan kita. Ada pengikat ketiga unsur tersebut dalam proses pembelajaran diri, yaitu pola pikir (*mind set*).

*Mind set* adalah pola pikir yang terkait dengan kepercayaan dasar yang kita miliki. Ia akan memengaruhi sikap, perilaku, dan tindakan kita. Segala sesuatu dimulai dari proses kita berpikir.

# UMMN

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA