

### 1.1 Naive Bayes Classifier

*Naive bayes classifier* merupakan salah satu metode *machine learning* yang dapat digunakan untuk klasifikasi suatu dokumen.

Teorema *bayes* berawal dari persamaan 2.1, yaitu:

$$P(A|B) = \frac{P(B \cap A)}{P(B)} \quad (2.1)$$

dimana  $P(A|B)$  artinya peluang A jika diketahui keadaan B. Kemudian dari persamaan 2.1 didapatkan persamaan 2.2.

$$P(B \cap A) = P(B|A) \cdot P(A) \quad (2.2)$$

Sehingga didapatkan teorema bayes seperti persamaan yang ditunjukkan pada persamaan 2.3.

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)} \quad (2.3)$$

*Naive bayes classifier* termasuk dalam algoritma pembelajaran *bayes*. Algoritma pembelajaran *bayes* menghitung probabilitas eksplisit untuk menggambarkan hipotesa yang dicari. Sistem dilatih menggunakan data latih lengkap berupa pasangan nilai-nilai atribut dan nilai target kemudian sistem akan diberikan sebuah data baru dalam bentuk  $\langle a_1, a_2, a_3, \dots, a_n \rangle$  dan sistem diberi tugas untuk menebak nilai fungsi target dari data tersebut (Mitchell, 1997).

*Naive bayes classifier* memberi nilai target kepada data baru menggunakan nilai  $V_{\text{map}}$ , yaitu nilai kemungkinan tertinggi dari seluruh anggota himpunan set domain V yang ditunjukkan pada persamaan 2.4.

$$V_{\text{map}} = \operatorname{argmax}_{v_j \in V} P(V_j | a_1, a_2, a_3, \dots, a_n) \quad (2.4)$$

Teorema *bayes* kemudian digunakan untuk menulis ulang persamaan 2.4 menjadi persamaan 2.5.

$$V_{\text{map}} = \operatorname{argmax}_{v_j \in V} \frac{P(a_1, a_2, a_3, \dots, a_n | V_j) P(V_j)}{P(a_1, a_2, a_3, \dots, a_n)} \quad (2.5)$$

Karena  $P(a_1, a_2, a_3, \dots, a_n)$  nilainya konstan untuk semua  $V_j$  sehingga persamaan 2.5 dapat ditulis dengan persamaan 2.6.

$$V_{\text{map}} = \operatorname{argmax}_{v_j \in V} P(a_1, a_2, a_3, \dots, a_n | v_j) P(v_j) \quad (2.6)$$

Tingkat kesulitan menghitung  $P(a_1, a_2, a_3, \dots, a_n | v_j) P(v_j)$  menjadi tinggi karena jumlah *term*  $P(a_1, a_2, a_3, \dots, a_n | v_j) P(v_j)$  bisa menjadi sangat besar. Ini disebabkan jumlah *term* tersebut sama dengan jumlah kombinasi posisi kata dikali dengan jumlah kategori. Metode klasifikasi *Naïve Bayes* menyederhanakan hal ini dengan bekerja dengan dasar asumsi bahwa atribut-atribut yang digunakan bersifat *conditionally independent* antara satu dan yang lainnya, dengan kata lain dalam setiap kategori, setiap kata *independent* satu sama lain.

Sehingga menjadi persamaan 2.7.

$$P(a_1, a_2, a_3, \dots, a_n | v_j) P(v_j) = \prod_i P(a_i | v_j) \quad (2.7)$$

Substitusi persamaan 2.7 dengan persamaan 2.6 menjadi persamaan 2.8.

$$V_{\text{NB}} = \operatorname{argmax}_{v_j \in V} P(v_j) \prod_i P(a_i | v_j) \quad (2.8)$$

$V_{\text{NB}}$  adalah nilai probabilitas hasil perhitungan *naïve bayes*. Untuk nilai fungsi target yang bersangkutan. Frekuensi kemunculan kata menjadi dasar perhitungan nilai dari  $P(v_j)$  dan  $P(a_i | v_j)$ . Himpunan *set* dari nilai-nilai probabilitas ini berkorespondensi dengan hipotesa yang ingin dipelajari. Hipotesa kemudian digunakan untuk mengklasifikasikan data-data baru. Pada pengklasifikasian teks, perhitungan persamaan 2.7 dapat didefinisikan :

$$P(v_j) = \frac{\text{docs}_j}{|D|} \quad (2.9)$$

$$P(w_k | v_j) = \frac{n_k + 1}{n + |kata|} \quad (2.10)$$

Keterangan :

1.  $\text{Docs}_j$  : kumpulan dokumen yang memiliki kategori  $v_j$ .
2.  $|D|$  : jumlah dokumen yang digunakan dalam pelatihan (kumpulan data latih).
3.  $n$  : jumlah total kata yang terdapat di dalam kata tekstual yang memiliki nilai fungsi target yang sesuai.

4.  $n_k$  : jumlah kemunculan kata  $w_k$  pada semua data tekstual yang memiliki nilai fungsi target yang sesuai.
5.  $| \text{kata} |$  : jumlah kata yang berbeda yang muncul dalam seluruh data tekstual yang digunakan.

### 3.2.1.1 Teks *Preprocessing*

Tahap text *preprocessing* ini meliputi *case folding*, *filtering*, *tokenizing*, *stemming* dan *term weighting*.

#### 1. *Case Folding*

Pada tahap ini dilakukan proses mengubah semua huruf yang terdapat dalam dokumen menjadi huruf kecil. Tahap selanjutnya adalah menghilangkan semua karakter angka dan tanda baca yang terdapat dalam dokumen tersebut dan menggantinya dengan karakter spasi. Semua karakter selain huruf, termasuk karakter spasi, dianggap sebagai pemisah atau *delimiter*.

Contoh :

Jaman sekarang begitu banyak orang memakai batik tetapi mereka tidak tahu apakah batik yang dipakai sesuai dengan kepribadian atau tidak. Termasuk saya, saya juga kurang tahu dengan tipe motif batik yang sesuai dengan kepribadian saya. Saya sendiri memiliki sifat jujur dalam semua hal, teratur dan disiplin. Banyak orang yang bilang saya bijaksana dalam mengambil keputusan. Sebagian orang menilai saya kaku tetapi saya orang yang bertanggungjawab dan setia.

Setelah mengalami proses *case folding*, menjadi :

jaman sekarang begitu banyak orang memakai batik tetapi mereka tidak tahu apakah batik yang dipakai sesuai dengan kepribadian atau tidak termasuk saya saya juga kurang tahu dengan tipe motif batik yang sesuai dengan kepribadian saya saya sendiri memiliki sifat jujur dalam semua hal teratur dan disiplin banyak orang yang bilang saya bijaksana dalam mengambil keputusan sebagian orang menilai saya kaku tetapi saya orang yang bertanggungjawab dan setia

#### 2. *Filtering*

*Filtering* yang dilakukan dalam penelitian ini adalah dengan melakukan penghapusan terhadap kata-kata yang tidak relevan

(*stopword*) dan mengambil kata-kata penting saja. Oleh karena itu, tahap ini disebut juga dengan *stopword removal* yang sudah terdapat daftar *stoplist*, yaitu sebuah daftar kata yang berisi sekumpulan *stopword*.

Karena penggunaan kata ‘tidak’ termasuk dalam *stoplist* yang ada. Untuk penggunaan kata ‘tidak’ pada sifat atau tipe, pengguna disarankan untuk menggabungkan kata tidak dengan kata berikutnya (tanpa spasi).

Contoh hasil *filtering* dari hasil *case folding* menjadi :

jaman memakai batik dipakai kepribadian tipe motif sifat disiplin  
bilang bijaksana keputusan menilai kaku bertanggungjawab

### 3. Tokenizing

Dalam tahap ini dilakukan pemecahan/pemotongan dokumen menurut tiap-tiap kata yang menyusun dokumen tersebut setelah mengalami proses *filtering*. Hasil pemotongan (*parsing*) terhadap kata-kata tunggal tersebut dijadikan kumpulan token dan membentuknya menjadi sebuah daftar atau list.

Contoh hasil *parsing* setelah mengalami proses *filtering* :

1 : jaman	10 : disiplin
2 : memakai	11 : bilang
3 : batik	12 : bijaksana
4 : dipakai	13 : keputusan
5 : kepribadian	14 : menilai
6 : tipe	15 : kaku
7 : motif	16 : bertanggungjawab
8 : sifat	17 : setia
9 : teratur	

### 4. Stemming dengan Algoritma Porter

Pada tahap ini akan dicari *root* kata dari tiap kata hasil *filtering*. Dalam bahasa Indonesia, afiks/imbuhan terdiri dari sufiks (akhiran), infiks (sisipan), dan prefiks (awalan). Karena proses penambahan infiks dalam bahasa Indonesia jarang terjadi dan tingkat kesulitan dalam menangani kata yang mengandung infiks maka proses *stemming* yang dibangun hanya menangani kata yang mengalami

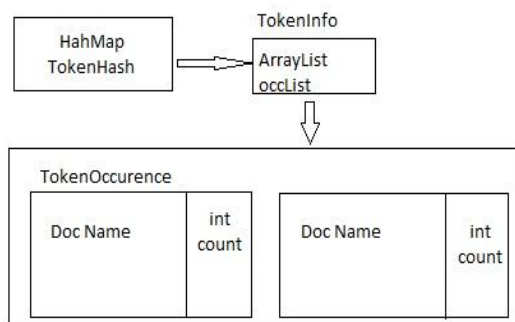
penambahan prefiks dan sufiks. Setelah melalui proses *stemming*, semua kata-kata yang berimbuhan diubah menjadi kata dasarnya.

Contoh hasil *stemming* dari hasil *parsing* :

1 : jaman	9 : disiplin
2 : pakai	10 : bilang
3 : batik	11 : bijaksana
4 : pribadi	12 : putus
5 : tipe	13 : nilai
6 : motif	14 : kaku
7 : sifat	15 : tanggungjawab
8 : atur	16 : setia

## 5. Proses Inverted Index

Untuk membuat proses lebih efisien dibuat suatu struktur data dimana struktur data tersebut dapat menampung semua term pada data corpus beserta jumlah dari masing-masing term untuk tiap dokumen corpus. Berikut ini adalah gambaran umum dari struktur inverted index yang digunakan pada skripsi rekomendasi motif batik berdasarkan identifikasi karakter pada dokumen teks menggunakan *Naive Bayes Classifier* yang ditunjukkan pada gambar 3.2.



Gambar 3.1 Struktur Inverted Index

Dari gambar diatas tokenHash berisi daftar semua *term* untuk dokumen *corpus*. Setiap *term* pada tabel tokenHash memiliki pointer

pada struktur data tokenInfo dimana tokenInfo adalah struktur data yang memiliki data arrayList yang berisi list dokumen yang mengandung *term* tersebut beserta nilai *tf*.

**6. Perhitungan TF**

*TF* adalah *Term Frequency* atau jumlah kemunculan kata dalam isi dokumen. Dari hasil *filtering*, dihitung kemunculan data dalam dokumen tersebut. Nilai kemunculan tersebut menjadi nilai *TF*.

Contoh perhitungan TF dari hasil *filtering* :

1 :	jaman	1	9 :	disiplin	1
2 :	pakai	2	10 :	bilang	1
3 :	batik	3	11 :	bijaksana	1
4 :	pribadi	2	12 :	putus	1
5 :	tipe	1	13 :	nilai	1
6 :	motif	1	14 :	kaku	1
7 :	sifat	1	15 :	tanggungjawab	1
8 :	atur	1	16 :	setia	1

**3.2.1.2 Pengklasifikasian Dokumen**

Proses pengklasifikasian untuk dokumen tes (dokumen X) menggunakan algoritma *Naive Bayes Classifier* dapat dilakukan dengan langkah-langkah berikut :

1. Menghitung peluang  $p(W_i|V_j)$  yaitu peluang setiap kata  $W_i$  dalam setiap kategori  $V_j$  pada data *corpus* sebagai pembelajaran klasifikasi, dengan persamaan 2.10.
2. Menghitung  $p(V_j)$  untuk kategori  $j$  dengan persamaan 2.9.
3. Menghitung  $V_{NB}$  dari dokumen uji dengan persamaan 2.8.
4. Nilai  $V_{NB}$  di setiap kategori yang merupakan kategori dari dokumen uji tersebut.

Proses secara umum rekomendasi motif batik berdasarkan identifikasi karakter pada dokumen teks menggunakan *naive bayes classifier* ditunjukkan pada gambar 3.3.

### 3.3 Contoh Perhitungan Manual

Berikut ini diberikan contoh, yaitu terdapat 5 dokumen latihan yang berasal dari 2 kategori yang berbeda dan sebuah dokumen uji yang kategorinya belum diketahui. Seperti yang sudah dijelaskan pada sub-bab 2.4 terdapat 9 karakter berdasarkan ilmu terapan psikologi *Eneagram* yang akan menjadi kategori dalam penentuan motif batik.

Kategori untuk identifikasi karakter pada dokumen teks adalah perfeksionis, penolong, pengejar prestasi, romantis, pengamat, pencemas, petualang, pejuang, dan pendamai. Pada contoh perhitungan ini diambil 2 contoh kategori untuk 5 dokumen latihan yang digunakan untuk mengetahui kategori 1 dokumen uji.

Berikut ini adalah daftar dokumen latihan dan dokumen uji yang dapat dilihat pada tabel 3.1.

Tabel 3.1 Daftar Dokumen Latihan dan Dokumen Uji

<b>Dokumen latihan 1</b>	Kategori : Perfeksionis	Saya orang yang teratur, siap berkomitmen tinggi, suka kerapian, optimis, suka menuntut dan saya tidak punya waktu untuk bermain.
<b>Dokumen latihan 2</b>	Kategori : Perfeksionis	Saya punya banyak wawasan, agak ambisius, tidak mau ada kesalahan dalam mengerjakan sesuatu, saya suka membuat semua sempurna dan biasanya lebih mudah mendeskripsikan yang ada di pikiran.
<b>Dokumen latihan 3</b>	Kategori : Petualang	Saya adalah orang yang suka mencari kepuasan hati, menyukai banyak ilmu, suka

berkelana, tangguh selalu energik. Jika saya ingin mengungkapkan sesuatu maka saya akan berusaha untuk memperolehnya, saya suka berpergian jika sedang murung.

**Dokumen  
latih 4**      Kategori :    saya tipe orang yang berjiwa  
Petualang      ulet, berani bertarung, dan  
rasa daya juang tinggi.

**Dokumen  
latih 5**      Kategori :    Saya orang yang biasanya suka  
Perfeksionis    segala sesuatu tampak  
sempurna, hal sekecil apapun  
yang timbul diperhatikan,  
kadang-kadang hal yang  
sepele dan beresiko menurut  
orang lain bagi saya itu hal  
penting, saya juga harus  
mengerjakan pekerjaan tepat  
waktu dan saya tidak perlu  
waktu banyak untuk  
bersantai.

**Dokumen  
uji**          Kategori :    Saya tidak suka diatur, saya  
orang yang suka menjalani  
hidup ini dengan optimis, saya  
punya jiwa petarung, tangguh,  
berani, siap mengambil resiko  
dan saya suka  
mengungkapkan apa yang  
saya rasakan dan ini terkadang  
menimbulkan masalah untuk  
saya.





Dari dokumen-dokumen tersebut, langkah pertama adalah melakukan *preprocessing* yaitu proses *case folding*, *filtering*, *tokenizing*, *stemming* dan perhitungan TF (*term frequency*). Dari dokumen tersebut ditunjukkan hasil preprocessing pada tabel 3.2.

Tabel 3.2 Daftar Token dan Frekuensi

No	Term	Frekuensi					
		D1	D2	D3	D4	D5	Uji
1	kecil	0	0	0	0	1	0
2	energik	0	0	1	0	0	0
3	siap	1	0	0	0	0	1
4	timbul	0	0	0	0	1	1
5	kesalahan	0	1	0	0	0	0
6	hati	0	0	1	0	1	0
7	tarung	0	0	0	1	0	1
8	rasa	0	0	0	1	0	1
9	wawas	0	1	0	0	0	0
10	tepat	0	0	0	0	1	0
11	pikir	0	1	0	0	0	0
12	kelana	0	0	1	0	0	0
13	ambisius	0	1	0	0	0	0
14	tampak	0	0	0	0	1	0
15	berani	0	0	0	1	0	1
16	pergi	0	0	1	0	0	0
17	deskripsi	0	1	0	0	0	0
18	mudah	0	1	0	0	0	0
19	kerja	0	0	0	0	1	0
20	resiko	0	0	0	0	1	1
21	optimis	1	0	0	0	0	1

22	juang	0	0	0	1	0	0
23	murung	0	0	1	0	0	0
24	ungkap	0	0	1	0	0	1
25	tuntut	1	0	0	0	0	0
26	suka	2	1	4	0	1	3
27	kerja	0	1	0	0	1	0
28	sepele	0	0	0	0	1	0
29	komitmen	1	0	0	0	0	0
30	ulet	0	0	0	1	0	0
31	santai	0	0	0	0	1	0
32	usaha	0	0	1	0	0	0
33	daya	0	0	0	1	0	0
34	tipe	0	0	0	1	0	0
35	oleh	0	0	1	0	0	0
36	puas	0	0	1	0	0	0
37	rapi	1	0	0	0	0	0
38	main	1	0	0	0	0	0
39	atur	1	0	0	0	0	1
40	sempurna	0	1	0	0	1	0
41	jiwa	0	0	0	1	0	1
42	ilmu	0	0	1	0	0	0
43	tangguh	0	0	1	0	0	1
	<b>Total</b>	<b>9</b>	<b>9</b>	<b>15</b>	<b>8</b>	<b>12</b>	<b>14</b>

Ket :  = Kategori Perfeksionis  
 = Kategori Petualang

Dari hasil *preprocessing* tersebut, selanjutnya adalah menghitung nilai probabilitas dari setiap kategori. Langkah pertama yaitu mencari nilai  $p(v_j)$  dengan persamaan 2.9. berikut adalah perhitungan manual dari dokumen-dokumen latih yang ada :

1. Kategori Perfeksionis :

$$Pr(v_{perfek}) = \frac{Fd(V_{perfek})}{|D|} = \frac{3}{5} = 0.6$$

2. Kategori Petualang :

$$Pr(v_{petualang}) = \frac{Fd(V_{petualang})}{|D|} = \frac{2}{5} = 0.4$$

Setelah  $p(v_j)$  dihitung, maka langkah berikutnya adalah mencari nilai  $p(w_j|v_j)$  dari masing-masing *term* pada masing-masing kategori yang dihitung dengan persamaan 2.10.

Perhitungan  $p(w_j|v_j)$  sebagai berikut :

1. *Term* "kecil"

$$P(w_{kecil} | v_{perfek}) = \frac{n_k + 1}{n + |kata|} = \frac{1 + 1}{53 + 30} = 0.0273$$

$$P(w_{kecil} | v_{petualang}) = \frac{n_k + 1}{n + |kata|} = \frac{0 + 1}{53 + 23} = 0.01515$$

2. *Term* "energik"

$$P(w_{energik} | v_{perfek}) = \frac{n_k + 1}{n + |kata|} = \frac{0 + 1}{53 + 30} = 0.02325$$

$$P(w_{energik} | v_{petualang}) = \frac{n_k + 1}{n + |kata|} = \frac{1 + 1}{53 + 23} = 0.04655$$

·  
·  
·  
·

Dan seterusnya.

Tabel 3.3 adalah hasil perhitungan  $p(w_j|v_j)$

Tabel 3.3 Hasil Perhitungan  $p(w_j|v_j)$ 

No	Term	Frekuensi Kategori		Dok Uji	P (w v)	
		Perfek	Petualang		Perfek	Petualang
1	kecil	1	0	0	0.02739726	0.015152
2	energik	0	1	0	0.02325581	0.046512
3	siap	1	0	1	0.04651163	0.023256
4	timbul	1	0	1	0.04651163	0.023256
5	kesalahan	1	0	0	0.04651163	0.023256
6	hati	1	1	0	0.04651163	0.046512
7	tarung	0	1	1	0.02325581	0.046512
8	rasa	0	1	1	0.02325581	0.046512
9	wawas	1	0	0	0.04651163	0.023256
10	tepat	1	0	0	0.04651163	0.023256
11	pikir	1	0	0	0.04651163	0.023256
12	kelana	0	1	0	0.02325581	0.046512
13	ambisius	1	0	0	0.04651163	0.023256
14	tampak	1	0	0	0.04651163	0.023256
15	berani	0	1	1	0.02325581	0.046512
16	pergi	0	1	0	0.02325581	0.046512
17	deskripsi	1	0	0	0.04651163	0.023256
18	mudah	1	0	0	0.04651163	0.023256
19	kerja	1	0	0	0.04651163	0.023256
20	resiko	1	0	1	0.04651163	0.023256
21	optimis	1	0	1	0.04651163	0.023256
22	juang	0	1	0	0.02325581	0.046512
23	murung	0	1	0	0.02325581	0.046512
24	ungkap	0	1	1	0.02325581	0.046512
25	tuntut	1	0	0	0.04651163	0.023256
26	suka	4	4	3	0.11627907	0.116279
27	kerja	2	0	0	0.06976744	0.023256

28	sepele	1	0	0	0.04651163	0.023256
29	komitmen	1	0	0	0.04651163	0.023256
30	ulet	0	1	0	0.02325581	0.046512
31	santai	1	0	0	0.04651163	0.023256
32	usaha	0	1	0	0.02325581	0.046512
33	daya	0	1	0	0.02325581	0.046512
34	tipe	0	1	0	0.02325581	0.046512
35	oleh	0	1	0	0.02325581	0.046512
36	puas	0	1	0	0.02325581	0.046512
37	rapi	1	0	0	0.04651163	0.023256
38	main	1	0	0	0.04651163	0.023256
39	atur	1	0	1	0.04651163	0.023256
40	sempurna	2	0	0	0.06976744	0.023256
41	jiwa	0	1	1	0.02325581	0.046512
42	ilmu	0	1	0	0.02325581	0.046512
43	tangguh	0	1	1	0.02325581	0.046512
	<b>Total</b>	<b>30</b>	<b>23</b>	<b>14</b>		

Ket :  = Kategori Perfeksionis  
 = Kategori Petualang

Dari hasil perhitungan  $p(v)$  dan  $p(w|v)$ , selanjutnya adalah menentukan kategori dari dokumen uji dengan persamaan 2.8 dicari peluang dari masing-masing kategori.

$$\begin{aligned}
 P(\text{Perfeksionis}|\text{DokUji}) &= 0.4 * 0.0465 * 0.0465 * 0.0232 * 0.0232 \\
 &\quad * 0.0232 * 0.0465 * 0.0465 * 0.0232 * \\
 &\quad 0.1162 * 0.0465 * 0.0232 * 0.0232 \\
 &= \mathbf{1.6246.10^{-18}}
 \end{aligned}$$

$$\begin{aligned}
 P(\text{Petualang}|\text{DokUji}) &= 0.6 * 0.0232 * 0.0232 * 0.0465 * 0.0465 \\
 &\quad * 0.0465 * 0.0232 * 0.0232 * 0.0465 * \\
 &\quad 0.1162 * 0.0232 * 0.0465 * 0.0465 \\
 &= \mathbf{4.8844.10^{-18}}
 \end{aligned}$$

Setelah peluang kedua kategori dihitung, diketahui bahwa kategori petualang mempunyai peluang yang lebih besar. Maka dokumen teks uji termasuk pada kategori petualang. Dari sini dapat diketahui identifikasi karakter pada dokumen teks adalah petualang. Untuk perekomendasi motif batik berdasarkan identifikasi karakter pada dokumen teks yaitu petualang, ditunjukkan dari tabel 2.7 yang cocok untuk karakter petualang adalah menggunakan motif batik parang lereng yang memiliki filosofi parang berarti senjata yang menggambarkan kekuasaan, kekuatan dan kecepatan dalam bergerak yang berhiaskan perubahan, kedinamisan, dan kelebihan.