BAB 2
TINJAUAN PUSTAKA

2.1 Emosi

Emosi merupakan perasaan yang kuat yang ditujukan terhadap seseorang atau sesuatu, dapat dikatakan juga emosi merupakan reaksi terhadap seseorang atau kejadian [11]. Seorang pakar kecerdasan emosional Daniel Goleman (1995) mengartikan istilah emosi sebagai setiap kegiatan atau pergolakan pikiran, perasaan, nafsu, keadaan mental yang hebat dan meluap-luap [11]. Seseorang dapat mengekspresikan emosinya dengan berbagai cara, sering kali sesorang akan mengekspresikan emosinya melalui gerak dan isyarat tubuh [12].

Pada dasarnya emosi dibedakan menjadi beberapa konsep antara lain cinta, senang atau gembira, marah, sedih, dan takut [13]. Kelima konsep emosi tersebut terbagi dalam dua kelompok superordinat, yaitu positif dan negatif. Untuk konsep emosi cinta dan senang atau gembira termasuk ke dalam superordinat positif, sedangkan untuk konsep emosi marah, sedih, dan takut masuk ke dalam superordinat negatif [14].

Berdasarkan dari penelitian yang dilakukan oleh Phillip R. Shaver and Upekkha Murdaya [14] mengenai leksikon emosi pada Bahasa Indonesia dijabarkan subordinat (kosakata) yang terkait dengan kelima konsep emosi. Untuk lebih jelasnya dapat dilihat pada tabel 2.1 di bawah ini.

Tabel 2.1 Subordinat Emosi Shaver

Superordinat	Konsep Emosi Dasar	Subordinat
Positif	Cinta	pemujaan, kasih sayang, cinta, kesukaan, kesukaan, ketertarikan, kepedulian, kelembutan, kasih sayang, sentimentalitas; gairah, keinginan, nafsu, gairah, keberahian, kerinduan
Positif	Senang	hiburan, kebahagiaan, keceriaan, keriangan, keriaan, gembira,

		kegirangan, menyenangkan, kenikmatan, suka hati, suka cita, keringanan hati, kepuasan, ekstasi, euforia, antusiasme, semangat, semangat, kegairahan, kegairahan, kegembiraan, kepuasan, kesenangan, kebanggaan, kemenangan, keinginan, harapan, optimisme, pesona, kegairahan, lega
Negatif	Marah	kejengkelan, jengkel, mengacau, gangguan, kesal, galak, kegusaran, frustrasi, kemarahan, berang, mengamuk, kemurkaan, permusuhan, keganasan, kebencian, benci, jijik, mencaci maki, dendam, kedendaman, enggan, marah, menjijikan, muak, kenistaan, cemburu, kecemburuan, kesengsaraan
Negatif	Sedih	gelisah, terkejut, ketakutan, gecar, kengeriaan, ngeri, panik, histeria, malu, kecemasan, kegugupan, ketegangan, khawatir, keprihatinan, cemas, berbahaya, takut
Negatif	Takut	kesakitan, menderita, terluka, sedih, depresi, putus asa, keputus asaan, murung, kemuraman, kesedihan, ketidakbahagiaan, dukacita, kenestapaan, sengsara, kesengsaraan,

2.2 Text Preprocessing

Text preprocessing merupakan suatu proses untuk membersihkan data teks agar menghasilkan data teks dengan kualitas yang baik dan siap untuk digunakan dalam pembuatan model klasifikasi. *Text preprocessing* diperlukan untuk mengubah format yang tidak terstruktur menjadi format yang terstruktur dan representasi multidimensi karena data teks sering mengandung banyak data asing seperti *tag*, *anchor text*, dan fitur lain yang tidak relevan [15].

2.2.1 Case Folding

Case folding merupakan proses merubah semua huruf pada sebuak dokumen teks menjadi huruf kecil [16]. Proses ini dilakukan karena data teks yang dimiliki tidak selalu terstruktur dan konsisten dalam penggunaan huruf kapital. Contoh dari tahap *case folding* dapat dilihat pada tabel 2.2 di bawah ini. Pada tabel tersebut seluruh huruf kapital pada teks masukkan akan diubah menjadi huruf kecil.

Tabel 2.2 Contoh Tahap Case Folding

Sebelum	Sesudah
Berharap aku ke dokter dksh salep bwt yg lebam2 biru..eehhh malah dkshnya obat nyeri tablet+vitamin pdhl tensi darah aku normal 110/70..ttp aja salep mah gak dksh sm dokternya	**b**erharap aku ke dokter dksh salep bwt yg lebam2 biru..eehhh malah dkshnya obat nyeri tablet+vitamin pdhl tensi darah aku normal 110/70..ttp aja salep mah gak dksh sm dokternya

2.2.2 Data Cleaning

Data cleaning merupkan proses menghilangkan atau mengubah hal-hal yang dianggap sebagai *noise* dari sebuah data teks dan hanya menyisakan karakter berupa huruf alfabet [17]. *Noise* tersebut dapat berupa karakter seperti tanda baca, angka, simbol atau karakter alfabet yang terdiri dari satu hingga beberapa karakter dan dianggap hanya menganggu suatu data teks. Daftar karakter yang biasa dihilangkan pada tahap *data cleaning* dapat dilihat pada tabel 2.3 di bawah ini.

Tabel 2.3 Daftar Karakter Yang Dihilangkan Pada Proses Data Cleaning

Daftar Karakter					
0	7	%	+	@	:
1	8	&	-	{	;
2	9	‘	/	}	`
3	!	(	=	[	~
4	“	)	,	]	|
5	#	^	.	\	<

Contoh dari tahap *data cleaning* dapat dilihat pada tabel 2.4 dibawah ini. Dimana pada tabel tersebut seluruh karakter yang terdaftar pada tabel 2.3 dihilangkan dan diubah menjadi spasi.

Tabel 2.4 Contoh Tahap Data Cleaning

Sebelum	Sesudah
berharap aku ke dokter dksh salep bwt yg lebam2 biru..eehhh malah dkshnya obat nyeri tablet+vitamin pdhl tensi darah aku normal 110/70..ttp aja salep mah gak dksh sm dokternya	berharap saya ke dokter dikasih salep buat yang lebam biru eehhh malah dkshnya obat nyeri tablet vitamin padahal tensi darah saya normal tetap saja salep mah tidak dikasih sama dokternya

2.2.3 Convert Slangword

Slangword atau biasa dikenal sebagai ”kata gaul” merupakan istilah-istilah atau bahasa yang sering digunakan untuk berkomunikasi [3]. *Convert slangword* dilakukan untuk mengubah kata gaul yang dapat berupa singkatan atau kata yang tidak baku menjadi kata utuh atau kata baku. Dalam mengindentifikasi kata *slang* akan digunakan kamus yang berisikan kata *slang* dan kata bakunya sebagai acuan perubahan kata. Contoh dari tahap convert slangword dapat dilihat pada tabel 2.3 di bawah ini. Pada tabel tersebut kata *slang* yang terdapat pada teks masukkan akan diubah menjadi kata yang baku.

Tabel 2.5 Contoh Tahap Convert Slangword

Sebelum	Sesudah
berharap **aku** ke dokter **dksh** salep **bwt yg** lebam biru eehhh malah **dkshnya** obat nyeri tablet vitamin **pdhl** tensi darah **aku** normal **ttp aja** salep **mah gak dksh sm** dokternya	berharap **saya** ke dokter **dikasih** salep **buat yang** lebam biru eehhh malah dkshnya obat nyeri tablet vitamin **padahal** tensi darah **saya** normal **tetap saja** salep mah **tidak dikasih sama** dokternya

2.2.4 Convert Negation

Convert negation merupakan proses penggabungan kata negasi dengan kata selanjutnya. Kata negasi seperti “tidak”, “bukan”, “jangan” dan sebagainya dapat merubah makna sentimen dari sebuah dokumen [18]. Contoh tahap *convert negation* dapat dilihat pada tabel 2.6 di bawah ini. Pada tabel tersebut kata negasi “tidak” pada teks masukkan digabungkan dengan kata selelahnya yaitu kata “dikasih”.

Tabel 2.6 Contoh Tahap Convert Negation

Sebelum	Sesudah
berharap saya ke dokter dikasih salep buat yang lebam biru eehhh malah dkshnya obat nyeri tablet vitamin padahal tensi darah saya normal tetap saja salep mah **tidak** dikasih sama dokternya	berharap saya ke dokter dikasih salep buat yang lebam biru eehhh malah dkshnya obat nyeri tablet vitamin padahal tensi darah saya normal tetap saja salep mah **tidakdikasih** sama dokternya

2.2.5 Tokenization

Tokenization merupakan sebuah proses pemecahan kalimat menjadi beberapa potongan kata atau karakter, dimana hasil dari pemecahan disebut dengan istilah token [19]. Contoh dari tahap tokenization dapat dilihat pada tabel 2.7. Dimana kalimat dari teks masukkan akan dipecah menjadi potongan-potongan kata.

Tabel 2.7 Contoh Tahap Tokenization

Sebelum	Sesudah		
berharap saya ke dokter dikasih salep buat yang lebam biru eehhh malah dkshnya obat nyeri tablet vitamin padahal tensi darah saya normal tetap saja salep mah tidakdikasih sama dokternya	berharap	saya	ke
	dokter	dikasih	salep
	buat	yang	lebam
	biru	eehhh	malah
	dkshnya	obat	nyeri
	tablet	vitamin	padahal
	tensi	darah	saya
	normal	tetap	saja
	salep	mah	tidakdikasih
	sama	dokternya	

2.2.6 Stopword Removal

Stopword Removal merupakan proses menghilangkan kata-kata yang tidak relevan, kata yang tidak memiliki makna tersendiri atau kata yang sering muncul dan menjadi tidak penting [20], seperti kata “dan”, “pada”, “pula”, “saat” serta kata-kata lain yang terdapat dalam *stopword list*. Contoh tahap *stopword removal* dapat dilihat pada tabel 2.8. Pada tabel tersebut kata-kata *stopword* yang terdapat pada *stopword list* dihilangkan dari list kata.

Tabel 2.8 Contoh Tahap Stopword Removal

Sebelum			Sesudah		
berharap	**saya**	**ke**	berharap	dokter	dikasih
dokter	dikasih	salep	salep	lebam	biru
buat	**yang**	lebam	eehhh	dkshnya	obat
biru	eehhh	**malah**	nyeri	tablet	vitamin
dkshnya	obat	nyeri	tensi	darah	normal
tablet	vitamin	**padahal**	salep	mah	tidakdikasih
tensi	darah	**saya**	dokternya		
normal	**tetap**	**saja**			
salep	mah	tidakdikasih			
sama	dokternya				

2.2.7 Stemming

Stemming merupakan proses menghilangkan imbuhan yang terdiri dari awalan, akhiran, awalan dan akhiran, dan sisipan dari sebuah kata untuk mendapatkan kata dasarnya [20]. Tahap *stemming* memiliki fungsi untuk menyederhanakan kata-kata tanpa menghilangkan makna kata tersebut sehingga ukuran data akan berkurang. Contoh dari tahap stemming dapat dilihat pada tabel 2.9. Pada tabel tersebut dapat dilihat kata seperti kata “berharap” dihilangkan imbuhan “ber-“ nya sehingga menjadi kata “harap”.

Tabel 2.9 Contoh Tahap Stemming

Sebelum			**Sesudah**		
berharap	dokter	**dikasih**	**harap**	dokter	**kasih**
salep	lebam	biru	salep	lebam	biru
eehhh	dkshnya	obat	eehhh	dkshnya	obat
nyeri	tablet	vitamin	nyeri	tablet	vitamin
tensi	darah	normal	tensi	darah	normal
salep	mah	tidakdikasih	salep	mah	tidakdikasih
dokternya			**dokter**		

2.3 N-Gram

Dalam melakukan klasifikasi teks diperlukan kata-kata yang unik dalam menentukan kelas dari data uji yang ada. Dengan menggunakan metode n-gram sebuah *string* dapat dipotong menjadi beberapa karakter. Metode n-gram digunakan untuk mengambil potongan karakter berupa huruf atau kata sejumlah *n* dari teks sumber secara terus menerus hingga akhir dokumen [21].

Setiap kata yang dihasilkan oleh n-gram memiliki jumlah kemunculan yang berbeda-beda untuk setiap dokumen yang ada [22]. N-gram dapat dibedakan berdasarkan dari jumlah potongan karakter sebesar *n*. Misalnya jika *n* = 1 maka disebut dengan *unigram*, *n* = 2 maka dengan disebut bigram, n = 3 maka disebut dengan *trigram* dan seterusnya.

Contoh dari tahap metode n-gram dapat dilihat pada tabel 2.10. Pada tabel tersebut dapat dilihat bahwa sebuah kalimat pendek dipotong menggunakan beberapa jumlah potongan n-gram yaitu *unigram*, *bigram*, dan *trigram*.

Tabel 2.10 Contoh Tahap N-Gram

Teks Awal	Jenis N-Gram	Hasil N-Gram
Belanja bahan pokok	Unigram (*n* = 1)	“Belanja”, “bahan”, “pokok”
	Bigram (*n* = 2)	“Belanja bahan”, “bahan pokok”
	Trigram (*n* = 3)	“Belanja bahan pokok”

2.4 Term Frequency

Term Frequency merupakan suatu metode pembobotan yang menentukan bobot dari suatu dokumen berdasarakan kemunculan *term* atau istilah. Banyak atau sedikitnya kemunculan dari *term* atau istilah yang muncul pada suatu dokumen akan menentukan bagaimana bobot dari dokumen tersebut. Ada beberapa jenis dari *term frequency* yang dapat digunakan. Salah satunya adalah *Raw Term Frequency* dimana pembobotan diberikan berdasarkan jumlah kemunculan suatu *term* atau istilah dalam dokumen [23]. Contohnya jika sebuah *term* muncul sebanyak 5 kali maka nilai *term* tersebut adalah 5. Misalkan terdapat beberapa contoh dokumen seperti pada tabel 2.11 di bawah ini.

Tabel 2.11 Contoh Dokumen

No	Dokumen
D1	Dokumen ini adalah dokumen pertama
D2	Ini adalah dokumen kedua
D3	Dan ini dokumen ketiga

Untuk melakukan pembobotan dari contoh dokumen pada tabel 2.11 maka perlu dihitung jumlah kemunculan dari setiap *term* yang ada terhadap dokumen D1, D2, dan D3. Untuk hasil perhitungannya dapat dilihat pada tabel 2.12.

Tabel 2.12 Contoh Perhitungan Kemunculan Term

term	**D1**	**D2**	**D3**
adalah	1	1	0
dan	0	0	1
dokumen	2	1	1
ini	1	1	1
kedua	0	1	0
ketiga	0	0	1
pertama	1	0	0

Sehingga didapatkan bobot dari setiap dokumen seperti pada tabel 2.13 berikut ini.

Tabel 2.13 Contoh Bobot Dokumen Hasil Perhitunga Term Frequency

Dokumen	**Bobot**
D1	(1, 0, 2, 1, 0, 0, 1)
D2	(1, 0, 1, 1, 1, 0, 0)
D3	(0, 1, 1, 1, 0, 1, 0)

2.5 Resampling

Metode *resampling* merupakan sebuah metode untuk memodifikasi dataset dengan tujuan untuk mengurangi perbedaan ukuran data antar kelas [7]. Terdapat dua jenis metode *resampling* yaitu *undersampling* yang menghilangkan data dari kelas mayoritas dan metode *oversampling* yang menambahkan data dari kelas minoritas [7].

2.6 Synthetic Minority Over-sampling Technique

Synthetic Minority Over-sampling Technique (SMOTE) merupakan salah satu perkembangan dari metode *oversampling*. SMOTE menambah jumlah data kelas minoritas dengan cara mengambil sampel dari kelas minoritas dan membangkitkan

data buatan berdasarkan *k-nearest neighbour* dari kelas minoritas [24]. SMOTE bekerja di ruang fitur. Artinya, output dari SMOTE bukanlah data sintetik yang merupakan representasi nyata dari sebuah teks di dalam ruang fiturnya.

Metode SMOTE bekerja dengan melakukan pengelompokan data berdasarkan jarak Ecludian antar data. Jumlah replikasi kelas minoritas disesuaikan berdasar jumlah dari kelas mayoritas, dimana jumlah replikasi harus sesuai dengan jumlah *k* pada *nearest neighbour*, jika jumlah replikasi sebanyak *n* maka jumlah *k* harus sebanyak *n-1* [25].

Untuk persamaan dari jarak Euclidean dapat dilihat pada persamaan 2.1 di bawah ini [25].

$$d(x,y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \cdots + (x_p - y_p)^2} \quad \textbf{(2.1)}$$

Sedangkan, proses replikasi data sintetis atau data buatan dilakukan dengan menggunakan persamaan 2.2 di bawah ini [25].

$$x_{syn} = x_i + (x_{knn} - x_i) \times \tau \quad \textbf{(2.2)}$$

Dimana:

x_{syn} = data hasil replikasi

x_i = data yang akan direplikasi.

x_{knn} = data yang memiliki jarak terdekat dari data yang akan direplikasi.

τ = bilangan random dari 0 sampai 1.

2.7 Naïve Bayes

Naïve Bayes merupakan metode klasifikasi yang berasal dari penerapan teorema bayes. Asumsi yang sangat kuat (naif) terhadap indepedensi dari setiap kondisi atau kejadian merupakan ciri utama dari metode *naïve bayes* [21]. Metode *naive bayes* merupakan metode klasifikasi yang mudah untuk diimplementasikan, yang

memiliki proses klasifikasi singkat dan *learning efficiency* yang tinggi. Persamaan dari posterior dapat dilihat pada persamaan 2.3 berikut [26].

$$P(A_i|B) = P(B|A_i) \times P(A_i) = \left(\prod_{k=1}^{n} P(b_k|A_i)\right) P(A_i) \quad \textbf{(2.3)}$$

Dimana:

$P(A_i|B)$ = Probabilitas dokumen *B* termasuk kelas A_i .

$P(A_i)$ = Probabilitas prior kelas A_i.

b_k = *Term* b_k pada dokumen.

$P(b_k|A_i)$ = Probabilitas *term* b_k dari kelas A_i.

Dengan persamaan dari prior dapat dilihat pada persamaan 2.4 berikut:

$$P(A_i) = \frac{n(A_i)}{N} \quad \textbf{(2.4)}$$

Dimana:

$n(A_i)$ = Jumlah kelas A_i pada seluruh dokumen.

N = Jumlah seluruh dokumen (banyaknya data latih).

2.7.1 Gaussian Naïve Bayes

Gaussian Naïve Bayes (GNB) merupakan salah satu varian model dari algoritma Naïve Bayes. GNB merupakan tipe *Naïve Bayes* yang mengikuti distribusi normal Gaussian dan mendukung data kontinu. GNB merupakan salah satu tipe metode *Naïve Bayes* yang mudah karena hanya perlu memperkirakan rata-rata dan standar deviasi dari data pelatihan. Persamaan dari metode GNB untuk menghitung nilai rata-rata, standar deviasi, dan densitas gaussian dapat dilihat secara berturut-turut pada persamaan 2.5 hingga 2.7 berikut [27].

$$\mu_i = \frac{\sum_{i=1}^{n} x_i}{n} = \frac{x_1 + x_2 + \cdots + x_n}{n} \quad \textbf{(2.5)}$$

Dimana:

μ_i = Rata-rata hitung (*mean*) *term* ke-*i*.

x_i = Nilai *term* ke-*i*.

n = Jumlah *term* ke-*i*.

$$\sigma_i = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \mu)^2}{n}} \quad \textbf{(2.6)}$$

Dimana:

σ_i = Standar deviasi *term* ke-*i*.

x_i = Nilai *term* ke-*i*.

μ = Rata-rata hitung (*mean*) *term* ke-*i*.

n = Jumlah *term* ke-*i*.

$$P(b_k|A_i) = \frac{1}{\sqrt{2\pi\sigma_{ik}}} e^{-\frac{(b_k - \mu_{ik})^2}{2\sigma_{ik}^2}} \quad \textbf{(2.7)}$$

Dimana:

$P(b_k|A_i)$ = Probabilitas *term* b_k terhadap kelas A_i

b_k = Nilai *term* b_k.

σ = Standar deviasi *term* b_k pada kelas A_i.

μ = Rata-rata hitung (*mean*) *term* b_k pada kelas A_i

π = Phi 3.14

2.8 Confusion Matrix

Confusion Matrix merupakan metode yang digunakan untuk menganalisis peformansi dari metode klasifikasi dalam melakukan pengenalan kelas [20]. *Confusion matrix* memiliki beberapa variable yaitu *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN). Dimana TP

merupakan data positif yang terdeteksi dengan benar, TN merupakan data negatif yang terdeteksi dengan benar, FP merupakan data negatif yang terdeteksi sebagai data positif, dan FN merupakan data positif yang terdeteksi sebagai data negatif. Contoh bentuk dari *confusion matrix* dapat dilihat pada tabel 2.14 berikut.

Tabel 2.14 Confusion Matrix

		Kelas Sebenarnya				
		Anger	**Happy**	**Sadness**	**Fear**	**Love**
Kelas Prediksi	**Anger**	TN	FP	TN	TN	TN
	Happy	FN	TP	FN	FN	FN
	Sadness	TN	FP	TN	TN	TN
	Fear	TN	FP	TN	TN	TN
	Love	TN	FP	TN	TN	TN

Dengan menggunakan *confusion matrix* dapat dilakukan evaluasi dari metode klasifikasi terhadap nilai *accuracy*, *precision, recall*, dan *f1- Score*. Untuk melakukan perhitungan nilai *accuracy*, *precision*, *recall*, dan *f1- Score* dapat digunakan rumus pada persamaan 2.8 hingga 2.11 [28].

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad \textbf{(2.8)}$$

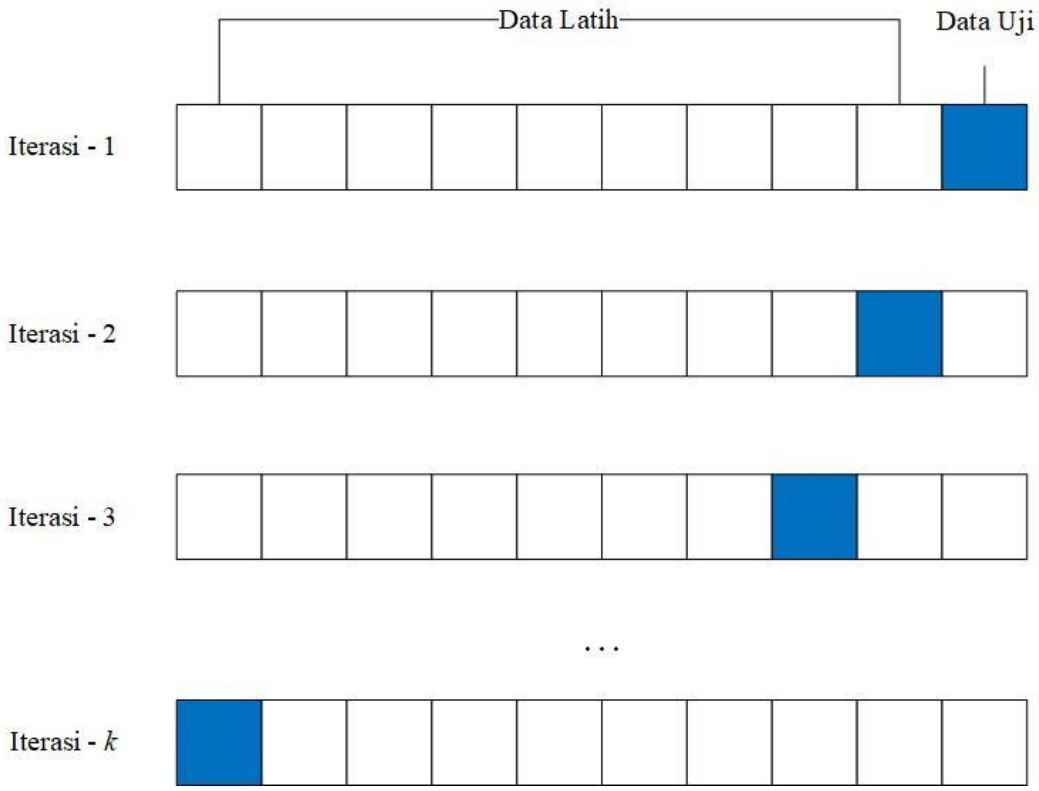
$$Precision = \frac{TP}{TP + FP} \quad \textbf{(2.9)}$$

$$Recall = \frac{TP}{TP + FN} \quad \textbf{(2.10)}$$

$$F1 - Score = 2 \times \frac{recall \times precision}{recall + precision} \quad \textbf{(2.11)}$$

2.9 K-Fold Cross Validation

K-fold Cross Validation (K-fold CV) merupakan salah satu jenis dari pengujian *Cross Validation* (CV). K-fold CV memiliki fungsi untuk menilai kinerja dari sebuah algoritma dengan cara membagi kumpulan data secara acak dan mengelompokka data tersebut sebanyak nilai *k* pada K-fold. Pada metode K-fold CV dataset dibagi menjadi sejumlah partisi secara acak. Dimana data partisi tersebut diolah sejumlah *k* kali iterasi prosedur dengan setiap iterasi prosedur menggunakan data partisi ke-*k* sebagai data testing dan menggunakan sisa data lainya sebagai data training [29]. Untuk contoh skema dari metode K-fold CV dapat dilihat pada gambar 2.1 berikut.



Gambar 2.1 Contoh Skema K-fold Cross Validation