BAB 2
TINJAUAN PUSTAKA

2.1 Text Mining

Text Mining adalah bagian dari penelitian ilmu pengetahuan computer yang mencoba memecahkan krisis dari muatan informasi dengan mengkombinasikan teknik dari *data mining, machine learning, natural language processing, information retrieval* dan *knowledge management*. Text mining juga salah satu teknik yang dapat digunakan untuk melakukan klasifikasi dokumen dimana *text mining* merupakan variasi dari data mining yang berusaha menemukan pola yang menarik dari sekumpulan data tekstual yang berjumlah besar[9].

Text mining, mengacu pada proses mengambil informasi berkualitas tinggi dari teks. Informasi berkualitas tinggi biasanya diperoleh melalui peramalan pola dan kecenderungan melalui sarana seperti pembelajaran pola statistik. Proses *text mining* yang khas meliputi :

1. Kategorisasi teks,
2. *Text clustering*, ekstraksi konsep atau entitas,
3. Produksi taksonomi granular,
4. *Sentiment analysis*, penyimpulan dokumen, dan
5. Pemodelan relasi entitas (yaitu, pembelajaran hubungan antara entitas Bernama)

2.2 Analisis Sentimen

Analisis sentimen adalah sebuah bidang studi yang menganalisis pendapat, sentimen, penilaian dan emosi seseorah terhadap suatu barang, orang maupun sebuah peristiwa. Ada beberapa penamaan terhadap studi ini, yaitu analisis sentiment, penambangan opini, ekstraksi opini dan penambangan sentiment dimana semuanya sekarang berada dibawah ranah analisis sentimen. Analisis sentimen bertujuan untuk menentukan suatu polaritas dalam sebuah review terhadap orang maupun peristiwa dalam kelas positif atau negatif dari pendapat yang berbentuk dokumen atau tulisan[1].

2.3 *Preprocessing*

Proses persiapan teks atau dataset mentah disebut juga dengan proses text preprocessing. *Text preprocessing* berfungsi untuk mengubah teks yang tidak terstruktur atau sembarang menjadi teks yang terstruktur, suatu teks tidak dapat diproses langsung oleh algoritma pencarian, oleh karena itu dibutuhkan *text preprocessing*[9]. Berikut beberapa tahapan *text preprocessing* sebagai berikut :

1. *Cleaning*

   Cleaning adalah proses penghilangan karakter symbol yang dianggap sebagai noise. Karakter yang dimaksud seperti koma(,) titik(.) kutip(‘) kutip dua(“) tanda tanya(?) tanda seru(!) dan sebagainya. Proses cleaning dilakukan untuk membersihkan karakter simbol yang tidak memiliki arti. Selain karakter simbol, angka, *hashtag*(#), *at*(@), URL dalam tweet pun dihilangkan karena tidak berhubungan dengan proses analisis sentimen[10].

2. *Case Folding*

   Case folding adalah proses penyeragaman bentuk huruf atau mengubah ukuran semua huruf menjadi sama besar. Dalam proses ini semua kata dari tweet yang terdapat huruf besar akan diubah menjadi huruf kecil (*lower case*)[11]. Proses ini bertujuan untuk memudahkan pencarian, karena tidak semua kata konsisten dalam penggunaan huruf kapital.

3. *Spelling Normalization*

   Spelling Normalization merupakan perbaikan kata-kata yang salah eja atau disingkat dengan bentuk tertentu[12]. Dalam sebuah tweet pasti terdapat kata-kata yang disingkat ataupun typo seperti “tdk” yang seharusnya “tidak” ataupun “iy” yang seharusnya “iya”. Proses ini dilakukan dengan Menyusun kamus kata atau memanipulasi singkatan tersebut. Sederhananya, proses ini dilakukan untuk memperbaiki kata yang salah ketik dalam penulisan.

4. *Convert Negation*

   Convert Negation adalah proses mempertegas kata yang bermakna negasi. Dalam sebuah *tweet* sangat memungkinkan terdapat negasi yang dapat membalikan arti dari suatu kata yang akan mengubah sentiment yang tadinya

positif menjadi negative maupun sebaliknya[12]. Jika pada *tweet* mengandung kata negasi, maka akan digabungkan dengan kata setelahnya.

5. *Filtering*

Filtering merupakan proses penghilangan kata tidak penting pada deskripsi melalui pengecekan kata-kata hasil parsing deskripsi apakah termasuk di dalam daftar kata tidak penting (stoplist) atau tidak[13].

6. *Tokenizing*

Tokenizing adalah proses pemisahan atau pemotongan semua kata yang ada pada suatu kalimat atau dokumen dan mengubahnya menjadi kumpulan token atau *term*[14]. Token yang dihasilkan akan digunakan sebagai pembanding pada proses klasifikasi. Tokenizing juga berfungsi untuk membuang beberapa karakter tertentu yang dianggap sebagai tanda baca.

7. *Stemming*

Stemming adalah proses menghilangkan imbuhan pada kata yang berimbuhan dan mengembalikan kata tersebut menjadi kata dasarnya[12]. Proses ini bertujuan untuk mengoptimasi pencocokan sehingga hasil yang dicari dapat lebih akurat.

2.4 Pembobotan TF

Pembobotan TF atau *term frequency* merupakan salah satu metode pembobotan yang populer. *Term* dapat berupa kata ataupun frasa, pada dokumen frekuensi pada setiap *term* sangat bervariasi[15]. Penjelasan *term frequency* sendiri adalah banyaknya kata pada suatu dokumen. Rumus TF dapat dilihat sebagai berikut:

$$TF(t_i, d_j) = f(t_i, d_j) \quad (2.1)$$

Keterangan:

$f(t_i, d_j)$ = jumlah kemunculan *term* i pada dokumen j

2.5 Pembobotan TF-IDF

Pembobotan TF-IDF merupakan metode untuk menghitung bobot setiap kata yang paling sering digunakan pada information retrieval. Metode tf-idf terkenal efisien, mudah dan memiliki hasil yang akurat[16].

Metode TF-IDF bekerja dengan cara pemberian bobot hubungan suatu kata (term) terhadap dokumen. TF-IDF ini adalah sebuah ukuran statistik yang digunakan untuk mengevaluasi seberapa penting sebuah kata di dalam sebuah dokumen atau dalam sekelompok kata. Untuk dokumen tunggal tiap kalimat dianggal sebagai dokumen. Frekuensi kemunculan kata di dalam dokumen yang diberikan menunjukkan seberapa penting kata itu di dalam dokumen tersebut. Frekuensi dikumen yang mengandung kata tersebut juga menunjukkan seberapa umum kata tersebut. Bobot kata semakin besar jika sering muncul dalam suatu dokumen dan berlaku sebaliknya semakin kecil jika muncul dalam banyak dokumen[17]. Pada metode TF-IDF digunakan rumus untuk menghitung bobot (W) masing masing dokumen terhadap kata kunci dengan rumus yaitu :

$$W_{dt} = tF_{dt} * IDF_t \quad (2.2)$$

Keterangan :

d = dokumen ke-d

t = kata ke-t dari kata kunci

W = bobot dokumen ke-d terhadap kata ke-t

tf = banyaknya kata yang dicari pada sebuah dokumen

IDF = log10(D/df)

D = total dokumen

Df = banyak dokumen yang mengandung kata yang dicari

Setelah bobot (W) masing-masing dokumen diketahui, maka dilakukan proses pengurutan dimana semakin besar nilai W, semakin besar tingkat similaritas dokumen tersebut terhadap kata kunci, begitupun sebaliknya.

Inverse Documen Frequency (IDF) memperhatikan kemunculan term pada kumpulan dokumen. Pada metode ini, *term* yang memiliki nilai adalah term yang jarang muncul pada kumpulan dokumen. Persamaan (2.3) IDF dapat dilihat sebagai berikut.

$$IDF(t) = log \frac{D}{df(t)} \quad (2.3)$$

Dimana *df(t)* merupakan banyaknya dokumen yang mengandung term t.

2.6 Naïve Bayes Classifier

Naïve Bayes merupakan metode klasifikasi dengan menggunakan metode probabilitas dan statistik, Metode yang juga dikenal sebagai Naive Bayes Classifier ini menerapkan teknik supervised klasifikasi objek di masa depan dengan menetapkan label kelas ke instance/catatan menggunakan probabilitas bersyarat. Probabilitas bersyarat adalah ukuran peluang suatu peristiwa yang terjadi berdasarkan peristiwa lain yang telah (dengan asumsi, praduga, pernyataan, atau terbukti) terjadi[18]. Model probabilitas untuk proses klasifikasi atribut X dapat ditulis sebagai persamaan (2.4) berikut.

$$P(C|X) = \frac{\text{P(X|C).P(C)}}{\text{P(X)}} \qquad (2.4)$$

Di dalam model tersebut P(C|X) merupakan peluang tweet yang memiliki atribut X yang akan diklasifikasikan sebagai kelas C. P(C) merupakan probabilitas kelas C dalam data latih, sedangkan P(X) merupakan peluang atribut X pada seluruh data latih dan {(X|C) merupakan peluang atribut x yang terjadi di dalam kelas C.

2.7 Multinomial Naïve Bayes

Salah satu model dari Naïve Bayes yang sering adigunakan dalam klasifikasi teks adalah multinomial Naïve Bayes. Multinomial Naïve Bayes merupakan metode klasifikasi *supervised learning* dengan model probabilistik[18]. Multinomial Naïve Bayes dipengaruhi oleh kumpulan *term*, dengan kata lain jumlah *term* diperhitungkan. Peluang antar term satu dengan yang lainnya adalah independen atau tidak bergantung[19]. Model Multinomial Naïve Bayes memperhitungkan frekuensi setiap kata yang muncul pada dokumen. Probabilitas suatu dokumen *d* berada di kelas *c* dapat dihitung menggunakan Persamaan (2.5) [18].

$$P(c|d) \propto P(c) \prod_{k=1}^{n} P(t_k|c) \qquad (2.5)$$

Dimana :

$P(c|d)$: Probabilitas dokumen d berada di kelas c

$P(c)$: Prior probability suatu dokumen berada di kelas c

$\{t1, t2, t2, \dots, t_n\}$: Token dalam dokumen d yang merupakan bagian dari vocabulary dengan jumlah n

$P(t_k|c)$: Probabilitas bersyarat term t_k berada di dokumen pada kelas c

Klasifikasi dokumen bertujuan untuk menentukan kelas terbaik pada suatu dokumen. Kelas terbaik dalam klasifikasi Naïve Bayes ditentukan dengan mencari *maximum a posteriori* (MAP) kelas c_{map} melalui persamaan (2.6).

$$c_{map} = \arg\max \hat{P}(c) \prod_{k=1}^{n} \hat{P}(t_k|c) \qquad (2.6)$$

P ditulis dengan $\hat{P}$ karena nilai sebenarnya dari $P(c|d)$ dan $P(t_k|c)$ belum diketahui, yang akan dihitung pada saat proses training [18].

$\hat{P}(c)$ dan $\hat{P}(t_k|c)$ didapatkan dengan menghitung *maximum likelihood* yang merupakan frekuensi relative dari parameter. Untuk prior, dapat dilihat pada persamaan (2.7).

$$\hat{P}(c) = \frac{N_c}{N} \qquad (2.7)$$

Dimana :

$\hat{P}(c)$: Prior probability suatu dokumen di kelas c

N_c : Jumlah dokumen dengan kelas c

N : Jumlah seluruh dokumen

Merupakan probabilitas frekuensi relatif term t dalam dokumen berada di kelas c, yang dapat dihitung menggunakan persamaan (2.8)

$$\hat{\mathrm{P}}(\mathrm{t}|\mathrm{c}) = \frac{T_{ct}}{\sum_{t' \epsilon V} T_{ct'}} \qquad (2.8)$$

Dimana :

$\hat{\mathrm{P}}(\mathrm{t}|\mathrm{c})$: Probabilitas bersyarat term t berada di dokumen pada kelas c

T_{ct} : Jumlah kemunculan term t pada dokumen dengan kategori c

$\sum_{t' \epsilon V} T_{ct'}$: Jumlah frekuensi seluruh term pada kelas c

Perhitungan *maximum likelihood* memiliki kelemahan, yaitu suatu kata dalam kelas yang tidak terlihat pada data training akan memiliki nilai 0. Hal ini yang menyebabkan perhitungan $P(c|d)$ menghasilkan nilai 0, karena setiap bilangan yang dikalikan dengan 0 akan menghasilkan 0. Untuk mengatasi masalah ini, diterapkan teknik *add-one* atau *Laplace smoothing*, sehingga persamaan (2.8) berubah menjadi persamaan (2.9) [19] berikut.

$$\widehat{P}(t|c) = \frac{T_{ct}+1}{\sum_{t' \epsilon V}(T_{ct'}+1)} = \frac{T_{ct}+1}{(\sum_{t' \epsilon V} T_{ct'})+B'} \qquad (2.9)$$

Dimana :

B : Jumlah seluruh term pada vocabulary

2.8 Sarkasme

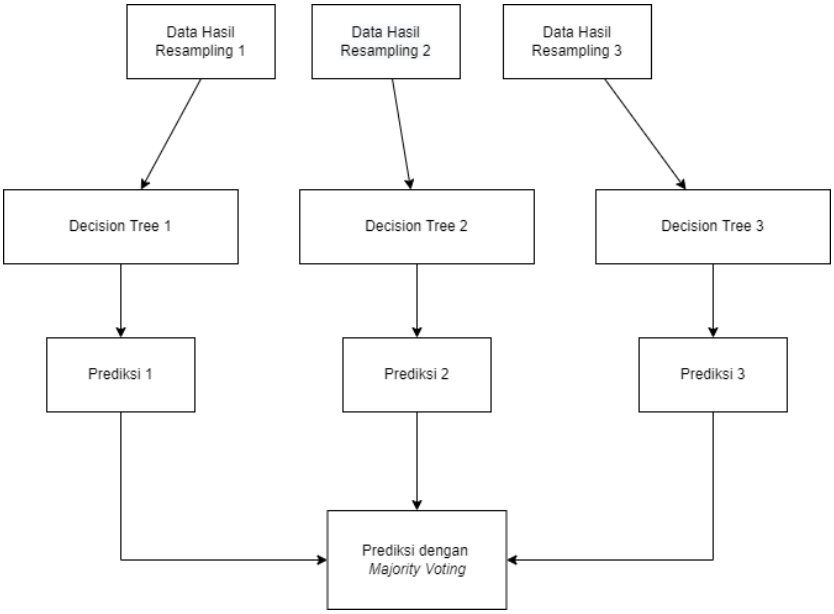
Sarkasme adalah jenis sentimen dimana seseorang mengungkapkan emosi atau tanggapan negatif menggunakan kata-kata yang positif. Penyampaian sarkasme dalam bentuk teks seringkali kompleks sehingga bisa saja menyebabkan pembaca salah memahami polaritas dari suatu teks[2]. Di dalam Bahasa Indonesia, sarkasme merupakan gaya Bahasa yang rumit dan juga seringkali disampaikan dengan kata-kata yang tidak formal. Sarkasme biasanya digunakan oleh masyarakat di Indonesia untuk mengkritik seseorang, peristiwa maupun aturan atau kebijakan yang berlaku.

2.9 *Random Forest Classifier*

Random forest pertama kali dikenalkan oleh Breiman pada tahun 2001. Pada penelitiannya, breiman menunjukkan beberapa kelebihan random forest diantaranya adalah dapat menghasilkan error yang lebih rendah, memberikan hasil yang bagus dalam klasifikasi, dapat mengatasi data training dalam jumlah sangat besar secara efisien, dan metode yang efektif untuk mengestimasi missing data[3].

Random forest merupakan salah satu metode yang digunakan untuk klasifikasi dan regresi. Metode ini merupakan sebuah ensemble (kumpulan) metode pembelajaran menggunakan pohon keputusan (*decision tree*) sebagai base classifier yang dibangun dan dikombinasikan, adapun beberapa aspek penting dalam metode Random Forest diantaranya melakukan bootstrap sampling untuk membangun pohon prediksi, masing-masing pohon keputusan memprediksi dengan prediktor acak dan metode klasifikasi yang terdiri dari kumpulan pohon keputusan ini nantinya akan dijadikan vote untuk mendapatkan hasil terakhir dari pendeteksian sarkasme dengan pendukung berupa data latih dan fitur acak yang independen dengan fitur yang berbeda-beda. Pohon keputusan dibuat dengan menentukan node akar dan berakhir dengan beberapa node daun untuk mendapatkan hasil akhir[3].

Metode ini juga merupakan metode pohon gabungan yang berasal dari metode Classification and Refression Tree (CART) dan didasarkan pada teknik pohon keputusan (*decision tree*). Pada algoritma dari metode Random Forest, algoritma dibagi menjadi dua bagian, bagian pertama adalah pembuatan "n" pohon (*tree*) untuk membentuk hutan (*forest*) yang acak (*random*). Bagian kedua adalah algoritma untuk melakukan prediksi dari random forest yang sudah dibuat[20]. Alur sederhana pada Random Forest ditampilkan pada gambar 2.1.



Gambar 2.1 Alur sederhana Random Forest

2.10 CART

CART (*Classification and Regression Trees*) merupakan salah satu jenis algoritma pada *decision tree*. CART merupakan algoritma prediksi yang dibangun dengan membagi dataset secara rekursif. Algoritma CART dapat digunakan untuk klasifikasi dan regresi[21].

Pohon terbentuk dari tiga hal yaitu *root node* (puncak pohon), *internal node* (cabang pohon), dan *leaf node* (akhir pohon). *Root node* tidak memiliki input sedangkan *leaf node* adalah *node* akhir yang tidak bisa terbagi lagi, *node* ini memiliki 1 input dan tidak memiliki *output*. Setiap node yang bukan node terminal terpecah menjadi dua turunan, sesuai dengan nilai dari salah satu variable prediktor. Untuk variabel prediktor kontinu, pemisahan di tentukan oleh split point. Titik yang nilai prediktornya lebih kecil dari split-point ke kiri, dan sisanya ke kanan[22].

Pembentukan pohon didasarkan pada bagaimana memilih satu *split-point* pada sebuah *node*, sehingga memperoleh *child node* yang paling *pure*. Pemilihan *split point* mempertimbangkan setiap kemungkinan pemisahan pada setiap variable prediktor dan memilih yang terbaik sesuai dengan beberapa kriteria[22]. Pada algoritma CART, pemilihan split terbaik dipilih menggunakan metode gini, dengan mencari gini index dan gini splitting index. Gini indeks menggunakan persamaan yang dapat dilihat pada persamaan .

$$I_G(i) = 1 - \sum P_{i^2}$$

Keterangan :

P_i : probabilitas nilai label yang muncul pada kolom yang bersangkutan.

Sedangkan untuk mendapatkan nilai Gini *Splitting index* dapat dilakukan dengan menggunakan rumus pada persamaan berikut.

$$GINI_{split} = \sum \frac{n_i}{n} * GINI(i)$$

Keterangan :

n_i : banyaknya data partisi pada himpunan

n : banyaknya seluruh himpunan

2.11 *Confusion Matrix*

Evaluasi perfomansi dilakukan untuk menguji hasil klasifikasi dengan mengukur nilai kebenaran dari sistem. Metode yang digunakan untuk evaluasi adalah *confusion matrix*. *Confusion matrix* merupakan salah satu metode yang dapat digunakan untuk mengukur kinerja suatu metode klasifikasi. Pada dasarnya *confusion matrix* mengandung informasi yang membandingkan hasil klasifikasi yang dilakukan oleh sistem dengan hasil klasifikasi yang seharusnya. Dalam pengujian keakuratan hasil pencarian akan dievaluasi nilai *recall*, *precision*, dan akurasi. Dimana *precision* mengevaluasi kemampuan sistem untuk menemukan peringkat yang paling relevan, dan didefinisikan sebagai presentase dokumen yang di-*retrieve* dan benar-benar relevan terhadap *query*. *Recall* mengevaluasi kemampuan sistem untuk menemukan semua item yang relevan dari koleksi

dokumen dan didefinisikan sebagai presentase dokumen yang relevan terhadap query. Sedangkan akurasi sendiri merupakan perbandingan kasus yang diidentifikasi benar dengan jumlah seluruh kasus[23].

Tabel 2.1 Tabel Confusion Matrix

	Kelas Sebenarnya		
Prediksi Kelas		+	-
	+	TP	FP
	-	FN	TN

Nilai *true positive* (TP) dan *true negative* (TN) adalah hasil klasifikasi yang benar. Nilai *false positive* (FP) adalah data yang bernilai negatif namun terdereksi sebagai data positif sedangkan *false negative* (FN) adalah data yang bernilai positif namun terdeteksi sebagai data negatif. Nilai akurasi, *precision, recall* berdasarkan tabel diatas diperoleh sebagai persamaan berikut.

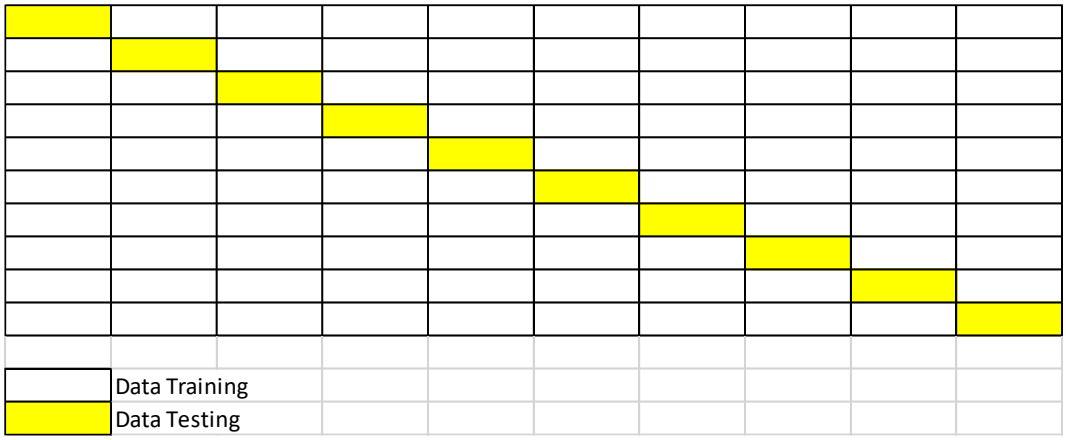
$$Akurasi = \frac{\text{TP+TN}}{\text{TP+FP+FN+TN}} * 100\% \quad (2.11)$$

$$Precision = \frac{TP}{TP+FP} \quad (2.12)$$

$$Recall = \frac{TP}{TP+FN} \quad (2.13)$$

2.12 *K-Fold Cross Validation*

K-Fold Cross Validation adalah salah satu dari jenis pengujian cross validation yang berfungsi untuk menilai kinerja proses sebuah metode algoritma dengan membagi sampel data secara acak dan mengelompokkan data tersebut sebanyak nilai K *k-fold*[24]. Kemudian salah satu kelompok *k-fold* tersebut akan dijadikan sebagai data uji sedangkan sisa kelompok yang lain akan dijadikan sebagai data latih. K-fold digunakan karena dapat mengurangi waktu komputasi dengan tetap menjaga keakuratan estimasi. Model 10-fold cross validation dapat dilihat pada gambar 2.2.



Gambar 2.2 Model 10-fold cross validation

2.13 Twitter

Twitter adalah sebuah situs web yang dimiliki dan dioperasikan oleh Twitter Inc yang dibentuk pada tahun 2006 oleh Jack Dorsey. Twitter berbasis di San Brunomor, California dekat San Francisco, dimana situs ini pertama kali dibuat. Twitter menjadi salah satu media populer yang banyak digunakan pengguna media sosial di Indonesia. Twitter adalah situs yang menyediakan layanan online yang memungkinkan pengguna membagikan konten yang saat ini sudah dapat memuat 280 karakter tulisan[25].

Twitter merupakan salah satu media sosial yang sangat populer dikalangan pengguna internet, hal ini dikarenakan kesederhanaan dan kemudahan dalam penggunaannya, serta pengguna dapat dengan bebas mengeluarkan pendapat atau opini mereka. Terdapat beberapa fitur dan istilah yang ada pada Twitter diantaranya[25]:

1. *Tweet*

   Pesan singkat yang berisi informasi yang dituliskan oleh pengguna yang akan muncul di timeline pengguna tersebut dan timeline pengguna lain.

2. *Retweet*

   Retweet adalah kondisi dimana seorang pengguna akan mengirim kembali sebuah tweet yang mengacu pada tweet pengguna lain sebelumnya.

3. *Reply*

Reply adalah sebuah fitur dimana pengguna dapat membalas pesan tweet pengguna lain

4. *Trending Topic*

Trending Topic adalah istilah yang mengacu kepada suatu topik yang sedang ramai diperbincangkan oleh banyak pengguna, ditandai dengan tanda # (hashtag) diikuti oleh kata atau kalimat tanpa spasi.

5. *Follow*

Follow adalah mengikuti (following) akun lain dalam Twitter untuk berlangganan tweet dari akun tersebut

6. *Follower*

Follower adalah akun lain yang mengikuti dan berlangganan tweet suatu akun Twitter.

7. *Following*

Following adalah jumlah akun lain yang yang diikuti.

8. *Mention*

Mention adalah *tweet* yang memuat tautan ke akun Twitter lain, ditandai dengan adanya tanda @ di depan nama.

9. *Bio*

Bio adalah deskripsi singkat tentang pemilik akun sepanjang 160 karakter atau kurang.

2.14 Penelitian terkait

Penelitian yang dilakukan oleh Debby dan Aulia (2020): Pendeteksian Sarkasme pada Proses Analisis Sentimen Menggunakan Random Forest Classifier [7]. Penelitian ini menggunakan algoritma klasifikasi yaitu *Support Vector Machine (SVM)* untuk mengklasifikasi sentimennya sedangkan untuk mendeteksi sarkasmenya menggunakan *Random Forest Classifier*. Pada penelitian ini juga digunakan fitur TF-IDF, POS *Tagging, Unigram,* TF dan TF-IDF. Sedangkan fitur pada pendeteksian sarkasme terdiri dari fitur *sentiment-relate, punctuatuion-relate, lexical* dan *syntactic,* dan *pattern-relate*.

Penelitian yang dilakukan oleh Aisyah, dkk (2021): *Sentiment Analysis With Sarcasm Detection On Politician's Instagram* [26]. Penelitian ini membandingkan akurasi dari metode analisis sentiment dengan deteksi sarkasme dan tanpa deteksi sarkasme, menggunakan metode Naïve Bayes dan Random Forest untuk analisis sentiment lalu Random Forest untuk deteksi sarkasme. Fitur yang digunakan antara lain adalah TF-IDF, *Sentiment-related Features, Punctuation-related Features,* dan *Lexical Features.*

Penelitian yang dilakukan oleh Edwin dan Ayu (2013): Indonesian Social media Sentiment Analysis with Sarcasm Detection [27]. Penelitian ini memperlihatkan pengaruh pendeteksian sarkasme pada analisis sentiment dengan menggunakan beberapa metode klasifikasi, yaitu *Naïve Bayes, Maximum Entropy,* dan *Support Vector Machine (SVM).* Penelitian ini menggunakan fitur ekstraksi diantaranya adalah Unigram, Negativity, Number of interjection words, dan Question word.