BAB 2
TINJAUAN PUSTAKA

Bab 2 berisi penjelasan mengenai tinjauan pustaka yang digunakan dalam penelitian. Tinjauan pustaka bersumber dari jurnal, artikel, maupun buku yang memiliki keterkaitan dengan penelitian.

2.1 Analisis Sentimen

Analisis sentimen merupakan salah satu cabang penelitian dari *text mining* yang bertujuan untuk mendapatkan informasi terkandung, serta menentukan persepsi atau pendapat suatu golongan terhadap topik pembahasan, kejadian, ataupun permasalahan tertentu. Analisis sentimen merupakan sebuah proses mengklasifikasikan suatu teks ke dalam suatu kelompok, yang pada umumnya bersifat positif, negatif, atau netral (Rachman dan Pramana, 2020).

Secara teknik, analisis sentimen dapat dibagi menjadi empat jenis pendekatan, yaitu pendekatan *Machine Learning*, pendekatan *Lexicon*, *Rule-based approach*, dan *Statistical model approach* (Rachman dan Pramana, 2020).

2.2 *Coronavirus Disease* 2019 (COVID-19)

Coronavirus Disease 2019 (COVID-19) adalah virus yang muncul di akhir tahun 2019 yang pertama kali mewabah di Tiongkok, kota Wuhan. Berdasarkan data dari *World Health Organization* (WHO), terdapat 235 negara yang telah tersebar COVID-19, sekitar 103 juta jiwa di seluruh dunia terinfeksi, dengan angka kematian karena COVID-19 telah mencapai 2.220.000 jiwa. Gejala yang ditimbulkan dari virus COVID-19 antara lain demam yang tinggi, gangguan pernapasan, dan batuk. Gejala COVID-19 tersebut baru dapat dikenali sekitar 2-14 hari setelah terinfeksi (Pustaka dkk, 2020).

Pencegahan COVID-19 dapat dilakukan dengan melakukan pengecekan kesehatan, isolasi, dan melakukan vaksinasi. Vaksin mRNA merupakan vaksin yang memiliki struktur genomik dengan kemampuan *self-amplifying* sehingga dapat menyebabkan replikasi RNA secara ekstrem di dalam sitosol. Vaksin

pengembangan lainnya yaitu *PiCoVacc*, merupakan vaksin konvensional yang menetralisir secara *in-vitro* dan dapat melakukan isolasi terhadap virus SARS-CoV-2. *PiCoVacc* adalah vaksin virus SARS-CoV-2 (*inactivated*) yang telah didistribusikan ke berbagai negara, salah satunya Indonesia (Pustaka dkk, 2020).

2.3 *Naive Bayes Classification*

Klasifikasi *Naive Bayes* merupakan salah satu metode dalam *machine learning* yang menggunakan perhitungan probabilitas. Kelebihan dari penggunaan metode ini adalah memerlukan data pelatihan yang tidak banyak untuk memperhitungkan parameter yang diperlukan dalam proses klasifikasi. Pada klasifikasi ini hanya varian dari variabel masing-masing kelas yang harus ditentukan dan tidak seluruh matriks kovariansi karena variabel independen yang diasumsikan. Sistem klasifikasi *Naive Bayes* bersifat sederhana dengan akurasi yang cukup tinggi. Konsep dasar yang digunakan dalam sistem klasifikasi ini adalah teorema peluang bersyarat oleh *Bayes* berikut (Oktasari dkk, 2016):

$$P(C|x) = P(C) \cdot P(x|C) \quad (1)$$

Peluang kejadian kelas C dengan kondisi x ditentukan dari peluang C dan peluang bersyarat dari C. Persamaan ini kemudian dikembangkan menjadi persamaan *Naive Bayes* berikut (Oktasari dkk., 2016):

$$P(C|x) = \frac{P(x|C) \cdot P(C)}{P(x)} \quad (2)$$

dengan $P(C)$ merupakan probabilitas dari suatu kelas, $P(x|C)$ sebagai frekuensi kondisi data tertentu yang memiliki kelas C, dan $P(x)$ sebagai jumlah seluruh data latih. Persamaan ini dapat diterapkan dalam pengklasifikasian data dengan melakukan perhitungan terhadap nilai probabilitas yang bertujuan untuk menentukan kelompok/golongan dari suatu data (Oktasari dkk, 2016). Klasifikasi dokumen dilakukan dengan terlebih dahulu menentukan kategori masing-masing kelas, kemudian menghitung rata-rata probabilitasnya dan menentukan klasifikasi berdasarkan nilai probabilitas tersebut (Darujati dan Gumelar, 2012).

Pada metode ini setiap variabel dianggap berdiri bebas dan tidak ada keterkaitan antar variabel. Hal ini berarti metode ini tidak memperhatikan urutan

kemunculan kata pada dokumen, dimana sebuah data teks akan dianggap sebagai kumpulan kata yang menyusun data teks tersebut (Oktasari dkk, 2016).

2.4 Twitter API

Twitter, yang diluncurkan pada Juli 2006, merupakan salah satu media sosial yang sampai saat ini banyak digunakan untuk menyampaikan berita atau pendapat secara umum oleh masyarakat di seluruh dunia. Twitter merupakan media sosial bertipe *micro-blogging* dengan pembatasan jumlah karakter dalam satu *tweet* maksimal 140 karakter. Proses *data crawling* pada Twitter dapat menggunakan sistem pencarian dengan nama pengguna (*by user*) dan sistem pencarian dengan kata kunci (*by keyword*). Pencarian data *by keyword* melakukan pencarian menggunakan kata kunci atau tagar (*hashtag*) yang terkait bahasan dengan jumlah maksimal total *tweet* yang dapat diunduh sebanyak 100 *tweet*. Pencarian *by user* dilakukan dengan melakukan input *username* dari akun Twitter yang dicari dengan jumlah maksimal total *tweet* yang dapat diunduh sebanyak 200 *tweet*. *Feature extraction* yang dapat diambil dari indeks Twitter untuk data *user* diantaranya yaitu ID, total *tweet*, lokasi, nama, *bio profile*, total *following* dan *follower*, jumlah *likes*, dan sebagainya (Sembodo dkk, 2016).

2.5 *Text Mining*

Text mining merupakan suatu proses penggalian informasi dari kumpulan data dengan menggunakan komponen yang termasuk dalam *data mining*. Proses ini bertujuan untuk mendapatkan informasi tertentu dari sekumpulan data atau dokumen. Penggalian data menggunakan sumber data berupa kumpulan teks dengan format yang tidak terstruktur atau semi terstruktur. *Text mining* dapat menyelesaikan permasalahan seperti pemrosesan, mengelompokkan dan menganalisis data yang tidak terstruktur dalam jumlah besar (Nurhuda dkk, 2016).

Text mining mengembangkan dan mengadopsi banyak teknik dari bidang-bidang lainnya seperti *Data Mining*, *Machine Learning*, *Natural Language Processing*, *Statistics*, *Linguistics*, dan sebagainya. Kegiatan riset dalam penggalian data diantaranya yaitu penyimpanan teks, ekstraksi, *preprocessing* terhadap teks, *indexing* dan analisis sentimen, serta pengumpulan data statistik

(Nurhuda dkk, 2016).

2.6 *Term Frequency - Inverse Document Frequency* (TF-IDF)

Terdapat beberapa kata yang kemungkinan kemunculannya tinggi di setiap kalimat namun tidak memiliki arti yang signifikan dalam pembuatan kelas atau golongan. Kata-kata yang tidak relevan dengan konteks dan penentuan kelas, atau tidak memberi makna yang signifikan dari data tersebut seharusnya dapat diabaikan atau diberi bobot yang lebih kecil dari kata lainnya yang lebih relevan. Pembobotan (*weighting*) kata bertujuan untuk membedakan tingkat relevan dan kontribusi suatu kata dalam penentuan kelas. Salah satu metode pembobotan yaitu TF-IDF (*Term Frequency - Inverse Document Frequency*), yaitu metode pembobotan yang menggabungkan konsep frekuensi kemunculan suatu kata (*term*) dalam suatu kalimat atau dokumen (*term frequency*), dengan jumlah kalimat atau dokumen yang memiliki frekuensi kemunculan kata tertentu (*document frequency*). Pada TF-IDF, kata-kata yang jarang muncul dalam dokumen akan memiliki bobot kata yang lebih tinggi dibanding dengan kata yang sering muncul. Pembobotan berfungsi untuk mengukur seberapa penting suatu kata terhadap data atau dokumen dan seluruh *dataset* (Hayati dan Alifi, 2021).

TF-IDF adalah metode yang dikenal baik untuk mengevaluasi pentingnya kata dalam dokumen. *Term frequency* dari term tertentu dihitung sebagai berapa kali sebuah *term* muncul dalam sebuah dokumen terhadap jumlah total kata dalam dokumen tersebut. IDF digunakan untuk menghitung pentingnya suatu istilah. IDF dihitung sebagai berikut (Ahuja dkk, 2019):

$$IDF(t) = log(\frac{N}{DF}) \quad (3)$$

dimana N adalah jumlah dokumen dan DF adalah jumlah dokumen yang mengandung *term* t. TF-IDF adalah cara yang lebih baik untuk mengonversi representasi tekstual informasi menjadi *Vector Space Model* (VSM) (Ahuja dkk, 2019).

2.7 *Preprocessing*

Data *preprocessing* adalah salah satu teknik *data mining* yang melibatkan transformasi data mentah menjadi format yang mudah dimengerti. *Preprocessing*

diperlukan dalam klasifikasi data untuk mengurangi masalah seperti nilai data yang kosong, redundansi, *noise* pada data, dan sebagainya. Adapun berbagai macam tahap data *preprocessing* adalah tokenisasi, *filtering*, *stemming*, *case folding*, dan *labeling* (Hayati dan Alifi, 2021).

2.7.1 Tokenisasi

Tokenisasi adalah proses memecah rangkaian teks menjadi kata, frasa, simbol, atau elemen bermakna lainnya yang disebut token. Tujuan tokenisasi adalah eksplorasi kata-kata dalam sebuah kalimat. Daftar token menjadi input untuk diproses lebih lanjut, misalnya dalam *text mining*. Semua proses dalam pencarian informasi membutuhkan kata-kata dari kumpulan data. Kegunaan utama tokenisasi adalah untuk mengidentifikasi kata kunci yang bermakna (Gurusamy dkk, 2014). Tokenisasi adalah proses pemotongan kalimat (*string*) berdasarkan setiap kata yang menyusunnya. Tokenisasi membagi teks input menjadi unit-unit kecil yang disebut token dengan panjang *n* karakter, dimana pembagian token dapat dibagi menjadi *unigram*, *bigram*, *trigram* dan *n-gram* (Saputra dkk, 2015).

2.7.2 *Filtering*

Banyak kata dalam dokumen yang sangat sering muncul tetapi pada dasarnya tidak berarti karena hanya digunakan sebagai kata penghubung dalam sebuah kalimat. Secara umum kata-kata tersebut tidak berkontribusi pada konteks atau isi dokumen secara tekstual. Karena frekuensi kemunculannya yang tinggi, keberadaaan *stopwords* ini dalam *text mining* menjadi kendala dalam memahami isi dokumen. *Stopwords* yang sangat sering digunakan yaitu kata-kata umum seperti ‘dan’, ‘yang’, ‘ini’ dan sebagainya. Pengembangan daftar *stopwords* tersebut sulit dan tidak konsisten antara sumber tekstual. Proses ini berfungsi untuk mengurangi data teks dan meningkatkan kinerja sistem (Gurusamy dkk, 2014).

2.7.3 *Case Folding*

Case folding digunakan untuk mempermudah proses pencarian dengan cara mengubah format kapitalisasi semua huruf dalam kalimat atau dokumen teks

menjadi seragam (Hayati dan Alifi, 2021). *Case folding* adalah proses manipulasi *case-sensitive* yang bertujuan untuk mengubah semua huruf dalam dokumen menjadi huruf kecil. Proses ini berfungsi agar sistem mampu menyamakan setiap karakter dari dokumen satu dengan dokumen lainnya (Pratama dan Pamungkas, 2016).

2.7.4 *Stemming*

Stemming digunakan untuk menyeragamkan kata dengan mengubah kata menjadi kata dasar atau baku sehingga dapat mengurangi daftar kata yang ada pada data latih (Hayati dan Alifi, 2021). *Stemming* adalah proses mengubah bentuk varian dari sebuah kata menjadi representasi umum, misalnya kata-kata: "disajikan", "menyajikan" semuanya dapat direduksi menjadi representasi umum "saji". *Stemming* merupakan proses yang banyak digunakan dalam pemrosesan teks untuk pencarian informasi (Gurusamy dkk, 2014).

2.8 Kurva ROC dan AUC

Kurva ROC (*Receiver Operating Characteristic*) merupakan representasi grafis dari hubungan antara tingkat positif salah dan positif benar/sejati. Tingkat positif yang salah adalah probabilitas data yang negatif memiliki hasil tes positif, dan positif sejati adalah probabilitas data yang positif memiliki hasil tes positif. Dengan menggunakan tingkat positif/negatif benar/salah, dapat menangani probabilitas bersyarat milik kelas prediksi tertentu yang memiliki label, dalam klasifikasi dua kelas (misalnya, sakit dan tidak sakit, pesan email adalah spam atau bukan, transaksi kartu kredit adalah penipuan atau bukan) (Gonçalves dkk, 2014).

Analisis ROC banyak digunakan untuk mengevaluasi kinerja diskriminatif dari variabel kontinyu yang mewakili uji diagnostik, penanda, atau pengklasifikasi. Tujuan lain analisis ROC yaitu untuk: 1.) mengevaluasi kemampuan diskriminatif penanda kontinu untuk menetapkan klasifikasi dua kelompok dengan benar; 2.) menemukan titik batas optimal untuk paling tidak salah mengklasifikasikan subjek dua kelompok; 3.) membandingkan keefektifan dari dua (atau lebih) tes atau penanda diagnostik (Gonçalves dkk, 2014).

Pendekatan berbeda untuk memperkirakan kurva ROC mengarah pada

perkiraan AUC yang berbeda. AUC (*area under the curve*) dapat diartikan sebagai probabilitas bahwa pada pasangan nilai yang dipilih secara acak dari data yang positif dan negatif, nilai tes diagnostik lebih tinggi untuk subjek yang negatif. Nilai AUC yang mendekati 1 menunjukkan akurasi diagnostik tes yang tinggi (Gonçalves dkk, 2014).

2.9 *Overfitting* dan *Underfitting*

Overfitting adalah salah satu masalah terbesar dalam melatih *neural network* dari data pelatihan. Hal ini berarti jaringan saraf selama periode pelatihan tidak dapat meningkatkan kemampuannya untuk memecahkan masalah, namun hanya mempelajari beberapa keteraturan acak yang terkandung dalam kumpulan pola pelatihan. *Overfitting* terjadi ketika model menggambarkan *error* atau *noise* acak daripada mempelajari keterkaitan yang mendasarinya (Jabbar dan Khan, 2015).

Ketika model kurang pas, bias umumnya tinggi dan variansnya rendah. *Overfitting* biasanya ditandai dengan varian tinggi, estimator bias rendah. Dalam banyak kasus, peningkatan bias yang kecil menghasilkan penurunan varians yang besar. *Underfitting* adalah kebalikan dari *Overfitting*. Ini terjadi ketika model tidak mampu menangkap variabilitas data. Pengklasifikasi yang dihasilkan tidak akan memiliki kekuatan predikatif juga tidak akan mampu memetakan data pelatihan dengan benar. Ini adalah hasil dari pemahaman atau upaya untuk menggunakan model yang terlalu sederhana untuk menggambarkan kumpulan data tertentu. Beberapa metode untuk menghindari masalah *Overfitting* dan *underfitting* dalam pembelajaran mesin yaitu *generalization cross-validation* dan *hold out cross-validation* (Jabbar dan Khan, 2015).

2.10 *K-Fold Cross-Validation*

Cross-validation adalah salah satu metode *resampling* data yang paling banyak digunakan untuk menilai kemampuan generalisasi model prediktif dan untuk mencegah *overfitting*. Tujuan dari *cross-validation* pada tahap pembuatan model adalah untuk memberikan perkiraan kinerja model akhir pada data baru (Berrar, 2019).

Feature extraction umumnya merupakan bagian integral dari proses

pembangunan model. Di sini, sangat penting bahwa fitur prediktif dipilih hanya menggunakan set pelatihan, bukan seluruh set pembelajaran; jika tidak, estimasi kesalahan prediksi bisa sangat bias. Misalkan fitur prediktif dipilih berdasarkan seluruh set pembelajaran terlebih dahulu, kemudian set pembelajaran dipartisi menjadi set validasi dan set pelatihan. Ini berarti informasi dari set validasi digunakan untuk pemilihan fitur prediktif. Namun data dalam set validasi hanya berfungsi untuk mengevaluasi model dan tidak diperbolehkan menggunakan data ini untuk apapun; jika tidak, kebocoran informasi akan menyebabkan perkiraan bias ke bawah, yang berarti meremehkan kesalahan prediksi yang sebenarnya. *Cross-validation* sering digunakan untuk menyetel parameter model, misalnya, jumlah optimal tetangga terdekat dalam *k-nearest neighbor classifier*. Di sini, *cross-validation* diterapkan beberapa kali untuk nilai parameter penyetelan yang berbeda, dan parameter yang meminimalkan kesalahan *cross-validation* kemudian digunakan untuk membangun model akhir. Dengan demikian, *cross-validation* mengatasi masalah *overfitting* (Berrar, 2019).

Dalam *k-fold cross-validation*, set pembelajaran yang tersedia dibagi menjadi *ksubset* terpisah dengan ukuran yang seimbang. *Fold* mengacu pada jumlah himpunan bagian yang dihasilkan. Partisi ini dilakukan dengan mengambil sampel kasus secara acak dari set pembelajaran tanpa penggantian. Model dilatih menggunakan himpunan bagian $k - 1$, yang bersama-sama mewakili himpunan latih. Kemudian, model tersebut diterapkan pada subset yang tersisa, yang dinotasikan sebagai set validasi, dan kinerjanya diukur. Prosedur ini diulang sampai masing-masing subset k telah berfungsi sebagai set validasi. Rata-rata pengukuran kinerja k pada set validasi k adalah kinerja yang dilakukan *cross-validation* (Berrar, 2019).

2.11 *Confusion Matrix*

Dalam konsep data mining terdapat metode yang dapat digunakan untuk mengukur keakuratan data sehingga data tersebut dapat digunakan dalam sistem pendukung keputusan, yaitu *confusion matrix*. *Confusion matrix* adalah salah satu metode yang dapat digunakan untuk mengukur kinerja suatu metode klasifikasi. matriks ini berisi informasi yang membandingkan hasil klasifikasi yang dilakukan

sistem dengan nilai aslinya. Ada 4 istilah dalam *confusion matrix* yang menggambarkan klasifikasi hasil pengukuran kinerja, yaitu *True Negative* (TN), *False Positive* (FP), *True Positive* (TP), dan *False Negative* (FN). Nilai *True Positive* adalah jumlah data positif yang diklasifikasikan dengan benar oleh sistem. Nilai *True Negative* adalah jumlah data berbahaya yang diklasifikasikan dengan benar oleh sistem. Nilai *False Positive* adalah jumlah data positif tetapi diklasifikasikan secara tidak benar oleh sistem. Nilai *False Negative* adalah jumlah data negatif tetapi diklasifikasi secara tidak benar oleh sistem (Rahmad dkk, 2020).

Precision adalah data yang diambil berdasarkan kurangnya informasi. Dalam klasifikasi biner, *precision* dapat dibuat sama dengan nilai prediksi positif. Formulasi berikut adalah fungsi *precision* (Rahmad dkk, 2020):

$$Precision = \frac{TP}{TP+FP} \tag{4}$$

Recall adalah penghapusan data yang berhasil diambil dari data yang relevan dengan kueri. Dalam klasifikasi biner, *recall* dikenal dengan istilah sensitivitas. Pembentukan relevan diambil data yang disetujui oleh *query* dapat dilihat dengan *recall*. Berikut ini adalah fungsi *recall* (Rahmad dkk, 2020):

$$Recall = \frac{TP}{TP+FN} \tag{5}$$

dengan TP adalah banyak data bernilai *True Positive*, FP adalah banyak data bernilai *False Positive*, dan FN adalah banyak data bernilai *False Negative* (Rahmad dkk, 2020). Rata-rata dari *recall* dan *precision* disebut *f-score*. Nilai ini merupakan parameter yang lebih penting daripada akurasi ketika memiliki distribusi kelas yang tidak merata dalam data. *F-score* dihitung sebagai berikut (Ahuja dkk, 2019):

$$Fscore = \frac{2 \cdot Recall \cdot Precision}{Recall + Precision} \tag{6}$$

2.12 Penelitian Terdahulu

Pada Tabel 2.1 berikut ini disajikan mengenai rincian penelitian terdahulu yang berkaitan dan menjadi dasar dari penelitian yang akan dilakukan.

Tabel 2.1 Penelitian Terdahulu

No.	Judul Penelitian	Peneliti dan Tahun	Permasalahan	Metode	Hasil Penelitian
1.	Analisis Sentimen Pada *Tweet* Terkait Vaksin Covid-19 Menggunakan Metode *Support Vector Machine*	Hashri Hayati, Muhammad Riza Alifi (2021)	Konfigurasi *dataset* yang kurang tepat untuk analisis sentimen isu vaksinasi	*Support Vector Machine*	Hasil evaluasi yang lebih tinggi diperoleh saat tokenisasi *bigram* ikut digunakan dengan *unigram* dengan pertambahan akurasi sebesar 0,6% - 0,7%.
2.	Analisis Sentimen Pembelajaran Daring Pada Twitter di Masa Pandemi COVID-19 Menggunakan Metode *Naive Bayes*	Samsir, Ambiyar, dkk. (2021)	Analisis sentimen menggunakan data Twitter dengan kata kunci dan tagar yang difilter dengan kata kunci pada tweet dalam bahasa Indonesia	*Naive Bayes*	Tingkat presisi sistem dengan 12,906 data uji memiliki nilai hasil 97,15%.
3.	Analisis Sentimen Pro dan Kontra Masyarakat Indonesia tentang Vaksin COVID-19 pada Media	Fajar Fathur Rachman, Setia Pramana (2020)	Melihat respon & opini masyarakat Indonesia terhadap vaksin COVID-19 dengan menggunakan data yang bersumber dari media sosial twitter	*Lexicon - based*	Pembangunan model LDA untuk menangkap berbagai macam topik pembicaraan masyarakat di media sosial twitter terkait vaksin COVID-19.

No.	Judul Penelitian	Peneliti dan Tahun	Permasalahan	Metode	Hasil Penelitian
	Sosial Twitter				
4.	Analisis Sentimen Terhadap Opini Masyarakat Tentang Vaksin Covid-19 Menggunakan Algoritma *Naive Bayes Classifier*	Winda Yulita, dkk. (2021)	Menganalisis opini masyarakat tentang proses vaksinasi dalam kasus COVID-19 dengan mempertimbangkan *tweet* yang diposting di Twitter.	*Naive Bayes*	Analisis dengan 3780 data *tweet* yang berkaitan vaksinasi dengan nilai akurasi yang dihasilkan sebesar 93 %.
5.	Analisis Sentimen Pada Review Restoran Dengan Teks Bahasa Indonesia Menggunakan Algoritma *Naive Bayes*	Dinda Ayu (2017)	Kekurangan dari *Naive Bayes* yaitu sangat sensitif pada fitur yang terlalu banyak, sehingga akurasi klasifikasi menjadi rendah.	*Naive Bayes*	Digunakan metode pemilihan fitur, yaitu *Genetic algorithm* agar bisa meningkatkan akurasi pengklasifikasi *Naive Bayes*. Hasil penelitian yaitu terdapat peningkatan akurasi *Naive Bayes* dari 86,50% menjadi 90,50%
6.	Implementasi Algoritma *Naive Bayes*	Fajar Ratnawati (2018)	Dibutuhkan pengklasifikasian sentimen agar	*Naive Bayes*	Akurasi yang didapat adalah 90% dengan nilai *recall*

No.	Judul Penelitian	Peneliti dan Tahun	Permasalahan	Metode	Hasil Penelitian
	Terhadap Analisis Sentimen Opini Film Pada Twitter		mudah untuk mendapatkan nilai kecenderungan opini terhadap film.		90%. *precision* 92%, dan *f-measure* 90%.
7.	*Data Crawling* Otomatis pada Twitter	Jaka Eka dkk, (2016)	Sulit dalam data *crawling* dari twitter secara otomatis baik dengan pencarian berupa *tweet* maupun *user*.	-	Membangun aplikasi yang berfungsi untuk *crawling data* dari twitter secara otomatis.
8.	Analisis Sentimen Berbasis Aspek pada Ulasan Pelanggan Restoran Bakso President Malang dengan Metode *Naive Bayes Classifier*	Whita Parasati dkk, (2020)	Pengolahan dan analisis data untuk mengetahui perspektif pelanggan terhadap aspek kepuasan layanan.	*Naive Bayes*	Hasil nilai akurasi sebesar 76% pada aspek Layanan, 88% pada aspek Makanan, dan 84% pada aspek Atmosfer.
9.	Sistem Analisis Sentimen pada Ulasan	Billy Gunawan dkk, (2018)	Pengklasifikasian data ke dalam lima kelas, yaitu sangat negatif, negatif,	*Naive Bayes*	Pengujian tiga kelas (negatif, netral dan positif) menghasilkan nilai

No.	Judul Penelitian	Peneliti dan Tahun	Permasalahan	Metode	Hasil Penelitian
	Produk Menggunakan Metode *Naive Bayes*		netral, positif, dan sangat positif.		akurasi tertinggi 77.78%, *recall* 93.33% dan *precision* 77.78% dengan perbandingan data latih dan dan uji 9:1, dan pengujian seluruh kelas dengan nilai akurasi 59.33%, *recall* 58.33 % dan *precision* 59.33 %.
10.	Text Mining Dalam Analisis Sentimen Asuransi Menggunakan Metode *Naive Bayes Classifier*	Luthfia Oktasari dkk, (2016)	Dibutuhkan informasi sentimen masyarakat terhadap isu asuransi dengan metode NBC.	*Naive Bayes*	Pengujian dengan pendekatan *rule-based*, *preprocessing*, dan klasifikasi dengan metode NBC menghasilkan nilai akurasi sebesar 95%.