BAB II

LANDASAN TEORI

2.1 Penelitian Terkait

Penelitian terkait yang menjadi acuan adalah sebagai berikut,

Berdasarkan *literatur riview* yang dilakukan dapat kesimpulan bahwa *sentiment analysis* dilakukan pada berbagai topik yang diperbincangkan di internet melalui twitter, ulasan pada *website* maupun *google playstore* dan sudah banyak peneliti yang menggunakan metode *Naive Bayes* dan *Particle Swarm Optimization* untuk metode klasifikasi. Pada penelitian ini dilakukan implementasi *sentiment analysis* terkait opini masyarakat di twitter tentang pembelajaran jarak jauh menggunakan metode *Naive Bayes* dan PSO dengan melakukan pengukuran performa model klasifikasi dengan penggunaan PSO. Pada beberapa penelitian *Naïve Bayes* mengklasifikasikan sentiment dari sebuah dataset untuk membuktikan bahwa *Naïve Bayes* lebih akurat, maka pada penelitian ini dilakukan implementasi analisis sentiment opini pada *twitter* terhadap pembelajaran jarak jauh menggunakan algoritma *Naive Bayes.* Beberapa penelitian juga menggunakan fitur tambahan seperti *preposesing* teks dan pembobotan kata yang dapat dibuktikan bahwa penambahan fitur tersebut dapat menambah tingkat akurasi pada hasil klasifikasi, maka pada penelitian ini diterapkan *fitur Particle Swarm Optimizaion.*

State of the art penelitian terkait dijelaskan pada table 2.1 berikut

Tabel 2.1 *penelitian terkait*

No	Peneliti (Tahun)	Judul	Metodologi	Teknik Pengujian	Kesimpulan
1	Nuzuliarini Nuris, Eka Rini Yulia, Kusmayanti Solecha (2021)	Implementasi Particle Swarm Optimization (PSO) Pada Analysis Sentiment Review Aplikasi Halodoc Menggunakan Algoritma Naïve Bayes	• Preprocessing : *Tokenization, stem (snow ball), Stopword, Stemming* • Penambahan fitur PSO • Fitur N-Gram • Klasifikasi Algoritma Naïve Bayes	*10-fod Validation* dan *Confusion matrix*	Dataset didapat dari playstore Halodoc terdiri dari 100 data positif dan 100 data negatif. Hasil akurasi yang didapat naïve bayes dengan fitur n-gram menghasilkan 88.50% setelah dioptimalkan dengan PSO menghasilkan 90.50% dengan peningkatan akurasi mencapai 2%
2	Ni Putu Gita Naraswati, Delvira Cindy Rosmilda, Dinda Desinta, Fadhilatul Khairi, Riska Damaiyanti, Rani Nooraeni (2021)	Analisis Sentimen Publik dari Twitter Tentang Kebijakan Penanganan Covid-19 di Indonesia dengan Naive Bayes Classification	• Preprocessing : *tokenization, cleansing, case folding, filtering, stemming* • Klasifikasi *Naïve Bayes*	*Confusion matrix*	Dataset yang digunakan sebanyak 4292 tweets. Hasilnya menunjukan masyarakat banyak memberikan sentiment negative terhadap PSBB dengan persentase 72.27% dan positif 27.23%
3	Tri Ngudi Wiyatno, Effendi	Naive Bayes Algorithm	• Preprocessing : *Cleaning,*		Dataset sebanyak 800 dari PT Mane Indonesia dengan 6 atribut.

Table 2.1 *penelitian terkait (Lanjutan 1)*

	bin Mohamad (2021)	Implementation Based on Particle Swarm Optimization in Analyzing the Defect Product	*Replacing, Reduction, And Transformation* • PSO • Klasifikasi *Naïve Bayes*		akurasi yang didapat dari data training antara NB + PSO sebesar 88.62% (800 data) dan akurasi dari data testing sebesar 92.22% (372 data)
4	Rizki Aulianita, Achmad Maezar Bayu Aji, Yuni Eka Achyani (2021)	Text Mining Menggunakan Naive Bayes Berbasis Particle Swarm Optimization Untuk Sentiment Restaurant	• *Preprocessing : tokenize, stopword removal* dan *stemming* • Klasifikasi Naïve Bayes • Particle Swarm Optimization (PSO)	*Confusion Matrix Model*	data diambil dari restaurant.com dan yelp.com dengan 500 data training, yang terdiri dari 250 review negatif dan 250 review positif). hasil akurasi pada algoritma Naive Bayes sebesar 81.00% kemudian dibandingkan dengan berbasis Particle Swarm Optimmization dengan akurasi 83.80%. PSO dapat memberikan solusi terhadap permasalahan klasifikasi review restoran lebih akurat
5	Annisa Elfina Augustia, Resi Taufan, Yuris Alkhalifi, Windu Gata (2021)	Analisis Sentimen Omnibus Law Pada Twitter Dengan Algoritma Klasifikasi Berbasis Particle Swarm Optimizatin	• Preprocessing : *Transform cases, Remove http, RemoveAnnotation, tokenize* • Klasifikasi *naïve bayes* • PSO • Penggunaan SMOTE	*10 - fold cross validation* dan *Confusion Matrix*	Dataset sebanyak 1.332 data tweet. Hasil penelitian bahwa penggunaan algoritma NB+PSO menghasilkan nilai akurasi lebih baik dibandingkan dengan algoritma SVM, NB dan SVM+PSO. Sehingga algoritma NB yang dioptimasi dengan PSO menjadi solusi untuk

					melakukan klasifikasi.

Table 2.1 *penelitian terkait (Lanjutan 2)*

6	Betesda(2020)	Peningkatan Optimasi Sentimen Dalam Pelaksanaan Proses Pemilihan Presiden Berdasarkan Opini Publik Dengan Menggunakan Algoritma Naïve Bayes Dan PSO	• Preprocessing : *Tokenization, transform cases* • PSO • TF-IDF • Klasifikasi *Naïve Bayes*	*10 - fold cross validation* dan *Confusion Matrix*	Data 130 review positif dan 130 review negative yang diambil dari pemilu.com. PSO terbukti dapat meningkatkan akurasi pada klasifikasi review opini publik berita pilpres untuk mengidentifikasi antara review positif dan review negative, dengan hasil Naive Bayes (NB) memiliki Accuracy sebesar 63.85% dan berbasis PSO Accuracy sebesar 71.15% bahwa pso dapat meningkatkan nilai akurasi
7	Karsito, Ahmad Taufiq (2020)	Analisis Sentimen Terhadap Pemindahan Ibu Kota Pada Media Sosial Twitter Menggunakan Algoritma Naive Babasis Particle Swarm Optimization	• *Preprocessing* : *remove duplicate, cleansing*, normalisasi kata, pemberial label, pembagian data, dan proses dokumen. • PSO • Klasifikasi dengan algoritma Naïve bayes	*10 - fold cross validation* dan *Confusion Matrix*	1000 data (Tweet) yang terdiri dari 522 data (Tweet) dengan respon positif dan 478 data (Tweet) dengan respon negative. Hasilnya bahwa 35% pengguna twitter setuju dengan rencana pemindahan Ibu Kota dan 65% menolak rencana tersebut. Dengan akurasi NB 78,88% dan NB+PSO sebesar 91,50% bahwa penerapan pso dapat meningkatkan akurasi

Table 2.1 *penelitian terkait (Lanjutan 3)*

8	Risa Wati (2020)	Penerapan Algoritma *Naive Bayes* Dan *Particle Swarm Optimization* Untuk Klasifikasi Berita *Hoax* Pada Media Sosial	• *Preprocessing : tokenization, stopwords removal, stemming* • Klasifikasi *Naïve Bayes* • PSO • Memasukan nilai parameter population size dan inertia weight	*Confusion Matrix*	Dataset yang digunakan adalah berita dari turnbackhoax.id terdiri 75 berita Hoax dan 75 berita non Hoax. Hasil yang didapat bahwa penerapan PSO dapat meningkatkan tingkat akurasi.
9	Alvie Delia Cahyani, Tati Mardiana, Laela Kurniawati (2020)	Sentiment Analysis Of Digital Wallet Service Users Using Naïve Bayes Classifier And Particle Swarm Optimization	• *preprocessing : transform case, tokenize, stopwords* • *removal, stemming* • klasifikasi *Naïve Bayes* • penambahan fitur PSO	*10 - fold cross validation* dan *Confusion Matrix* *T-Test and ANOVA*	Data sebanyak 272 tweet dana wallet dan 218 iSaku. Hasil akurasi Naïve Bayes dompet digital Dana sebesar 60.00% setelah penambahan dengan PSO menjadi 91.67% dan untuk iSaku akurasi NBC 53.23% setelah penambahan PSOmenjadi 85.00%
10	Kuncahyo Setyo Nugroho, Istiadi, Fitri Marisa(2020)	Optimasi Naive Bayes Classifier Untuk Klasifikasi Teks Pada E-	• *Preprocessing : case folding, tokenizing, stemming, filtering*	*10 - fold cross validation* dan *Confusion Matrix*	Dataset berasal dari portal Sambat Online Kota Malang terdiri dari 200 data dengan 7 kategori sebagai label. PSO dapat diterapkan sebagai seleksi

Table 2.1 *penelitian terkait (Lanjutan 4)*

		Government Menggunakan Particle Swarm Optimization	• Klasifikasi *Naïve Bayes* dan KNN • PSO		fitur yang dapat meningkatkan hasil akurasi algoritma NBC sebesar 87,44 % dalam melakukan klasifikasi teks dengan memberikan nilai akurasi terbaik dibandingkan NBC tanpa PSO dan k-NN.
11	Yoga Dwi Pambudi1 Wing Wahyu Winarno, Andi Sunyoto (2020**)**	Analisa Sentimen Twitter Menggunakan Algoritma Klasifikasi Pada Promosi Wisata Museum Sangiran Kabupaten Sragen	• Preprocessing : *case folding, tokenize, stopword removal* • Klasifikasi dengan *naïve bayes* • PSO • Menambah parameter *Term Frequency*		pengambilan data melalui studi lapangan ke Museum Sangiran. Data yang diperoleh dari seleksi fitur PSO menghasilkan dataset yang terseleksi sebanyak 173 kata dan hasil dari pengujian menggunakan parameter Term Frecuency (TF) pada klasifikasi algoritma Naive Bayes sebesar 87,91%
12	Nur Hayatin, Gita Indah Marthasari, Lia Nuraini (2020)	Optimization of Sentiment Analysis for Indonesian Presidential Election using Naïve Bayes and Particle Swarm	• Preprocessing: *Normalization, case folding, tokenize, stopword removal, stemming* • PSO • Klasifikasi dengan *naïve bayes*		1200 data set terdiri dari 800 data train dan 400 data test. Hasil dari pengujian didapatkan akurasi Naïve Bayes sebesar 86.62% dan akurasi dari penambahan dengan PSO sebesar 90.74%

		Optimization			

Table 2.1 *penelitian terkait (Lanjutan 5)*

13	Ratih Yulia Hayuningtyas, Retno Sari (2019)	Analisis Sentimen Opini Publik Bahasa Indonesia Terhadap Wisata TMII Menggunakan *Naïve Bayes* Dan PSO	• Preprocessing : *tokenization, stopword removal, stemming* • PSO • Klasifikasi dengan *Naïve Bayes*	*10 - fold cross validation* dan *Confusion Matrix*	data sebanyak 50 ulasan positif dan 50 ulasan negative dari web travidsor. Hasil yang didapat menggunakan Naive Bayes akurasi sebesar 70%. Sedangkan hasil eksperimen menggunakan Naive Bayes dengan PSO didapati akurasinya sebesar 94.02%
14	Suwanda Aditya Saputra, Didi Rosiyadi, Windu Gata, Syepry Maulana Husain (2019)	Analisis Sentimen *E-Wallet* Pada *Google Play* Menggunakan Algoritma *Naive Bayes* Berbasis Particle *Swarm Optimization*	• Preprocessing: *tokenization, stopword removal, stemming, Transform Case* • PSO • Klasifikasi menggunakan *Naïve Bayes*	*fold cross validation*	Dataset review pengguna aplikasi OVO pada google play. Penggunaan feature PSO dapat meningkatkan akurasi yang sangat signifikan yang sebelumnya model Naive Bayes sebesar 82.30% setelah ditambahkan feature selection menjadi 83.60%, penerapan PSO berpengaruh besar yang dapat memberikan kenaikan sekitar 1.3%
15	Fajar Ratnawati (2018)	Implementasi Algoritma Naive Bayes Terhadap Analisis Sentimen Opini Film Pada	• Preprocessing : *Case folding* normalisasi fitur • Klasifikasi dengan *naïve bayes*	*5 - fold cross validation*	Dataset sebanyak 200 tweet positif dan 200 negative. Pembagian dataset menggunakan 5-fold cross validation dengan hasil akurasi tertinggi pada fold-2 yaitu 90%, precision 92%,

		Twitter			Recall 90% dan f-measure 90%.

1.2 Landasan Teori

2.2.1 Pembelajaran Jarak Jauh

Pembelajaran Jarak Jauh (PJJ) adalah pembelajaran dengan menggunakan suatu media yang memungkinkan terjadi interaksi antara pengajar dan pembelajar. Dalam PJJ antara pengajar dan pembelajar tidak bertatap muka secara langsung, dengan kata lain melalui PJJ dimungkinkan antara pengajar dan pembelajar berbeda tempat, bahkan bisa dipisahkan oleh jarak yang sangat jauh. Pembelajaran jarak jauh (juga disebut juga pendidikan jarak jauh) merupakan pelatihan yang diberikan kepada peserta atau siswa yang tidak berkumpul bersama di satu tempat secara rutin untuk menerima pelajaran secara langsung dari instruktur. Bahan-bahan dan instruksi- instruksi detail yang bersifat khusus dikirimkan atau disediakan untuk para peserta yang selanjutnya melaksanakan tugas-tugas yang akan dievaluasi oleh instruktur. Dalam kenyataannya dapat dimungkinkan instruktur dan peserta tersebut terpisah tidak hanya secara geografis namun juga waktu (Prawiyogi et al., 2020)

2.2.2 Twitter

Twitter adalah jejaring sosial yang membatasi penggunanya untuk mengirim sebuah tweet dengan batas 140 kata. *Twitter* dengan *facebook* mempunyai kesamaan dan perbedaan. Kesamaannya ialah *twitter* dan *facebook* sama-sama layanan jejaring sosial yang berguna untuk saling menghubungkan antara pengguna satu dengan pengguna lainnya. *Twitter* sangat memudahkan penggunanya untuk saling menjalin pertemanan dengan pengguna lainnya, di

twitter juga ada fitur top trending yaitu fitur yang memudahkan penggunanya untuk melihat kicauan burung

apa yang paling populer dan paling sering dikicaukan oleh pengguna. (Basri, 2017).

2.2.3 RapidMiner

RapidMiner merupakan perangkat lunak yang dibuat oleh Dr. Markus Hofmann dari *Institute of Technologi Blanchardstown* dan *Ralf Klinkenberg* dari rapid-i.com dengan tampilan GUI (*Graphical User Interface*) sehingga memudahkan pengguna dalam menggunakan perangkat lunak ini. Perangkat lunak ini bersifat terbuka (*open source)* dan dibuat dengan menggunakan program Java di bawah lisensi GNU *Public Licence* dan rapidminer dapat dijalankan di sistem operasi manapun. Dengan menggunakan rapidminer, tidak dibutuhkan kemampuan koding khusus, karena semua fasilitas sudah disediakan. Rapidminer dikhususkan untuk penggunaan data mining. Model yang disediakan juga cukup banyak dan lengkap, seperti model *Bayesian, Modelling, Tree Induction, Neural Network* dan lain-lain. (Haryati et al., 2015).

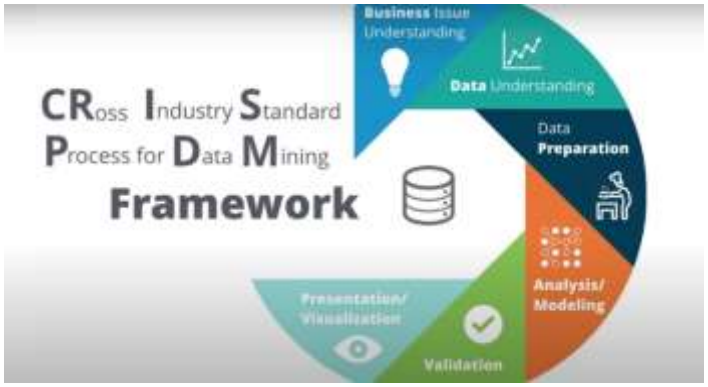
RapidMiner adalah sebuah solusi untuk melakukan analisis terhadap *data mining*, *text mining* dan analisis prediksi. RapidMiner menggunakan berbagai teknik deskriptif dan prediksi dalam memberikan wawasan kepada pengguna sehingga dapat membuat keputusan yang paling baik (Aprilla, dkk, 2013).

2.2.4 Metode Data Mining Standard Process (CRISP-DM)

CRISP-DM (Cross Industry Standard Process for Data Mining) suatu standarisasi pemrosesan data mining yang telah dikembangkan dimana data yang ada akan melewati setiap fase terstruktur dan terdefinisi dengan jelas dan efisien.

Selain menerapkan suatu model dalam proses penambangan data, pemilihan algoritma sangat mempengaruhi terhadap komparasi kinerja metode data mining

(Hasanah et al., 2021). Proses metodologi ini terdiri dari 6 tahapan yang dapat dijelaskan sebagai berikut (Hidayat et al., 2020):



Gambar 2.I *Metodologi Crisp-DM*

1. *Business Understanding* (Pemahaman Bisnis)

   Fase pertama pada metode CRISP-DM ini dibutuhkan pemahaman tentang substansi dari kegiatan data mining yang akan dilakukan seperti menentukan sasaran atau tujuan dan batasan menjadi formula dari permasalahan data mining dan membuat perencanaan strategi serta jadwal penelitian

2. *Data Understanding* (Pemahaman Data)

   Fase mengumpulkan data awal, mempelajari data untuk bisa mengenal data yang akan dipakai untuk mendapatkan pemahaman yang mendalam tentang data, mengidentifikasi masalah kualitas data, atau untuk mendeteksi adanya bagian yang menarik dari data yang dapat digunakan untuk hipotesa untuk informasi yang tersembunyi.

3. *Data Preparation* (Persiapan Data)

   Fase yang meliputi semua kegiatan untuk membangun dataset akhir (data yang akan diproses padatahap pemodelan/modeling) dari data mentah. Pada tahap ini yaitu mempersiapkandata untuk melakukan langkah-langkah yang

disebut dengan text preprocesing

4. *Modelling* (Pemodelan)

   Fase menentukan teknik data mining yang digunakan, menentukan tools data mining,teknik data mining, algoritma data mining, menentukan parameter dengan nilai yang optimal.

5. *Evaluation*

   Fase ini dilakukan evaluasi terhadap model yang telah dibentuk pada fase sebelumnya. Fase ini dilakukan secara mendalam dengan tujuan untuk menentukan apakah model dapat mencapai tujuan yang ditetapkan pada fase awal

6. *Deployment* (Penyebaran)

   Semua pengetahuan dan informasi yang telah diperoleh dari fase sebelumnya akan disampaikan dan dipresentasikan dalam bentuk khusus sehingga dapat digunakan oleh pengguna. Tahap deployment dapat berupa pembuatan laporan sederhana atau mengimplementasikan proses data mining yang berulang dalam perusahaan

2.2.5 Data Mining

Data mining adalah proses menemukan korelasi, pola, dan tren baru yang berarti dengan memilah-milah sejumlah besar data yang disimpan dalam repositori, menggunakan teknologi pengena;an pola serta teknik statistic dan matematika. Data mining dibagi 6 kelompok berdasarkan tugas yang biasanya dilakukan. Larose & Larose, 2014), yaitu:

1. Dekripsi

   Mendeskripsikan pola dan tren dalam data dengan hasil model *data mining* harus menggambarkan pola yang jelas dan dapat diterima untuk interpretasi dan penjelasan intuitif.

2. Estimasi

   Mirip dengan klasifikasi tetapi variabel target lebih kearah numerik dari pada kategorikal

3. Prediksi

   Prediksi mirip dengan klasifikasi dan estimasi, hanya saja untuk prediksi hasilnya ada pada masa yang akan datang. Contoh : memprediksi harga saham tiga bulan kedepan.

4. Klasifikasi

   Mengelompokan data berdasarkan keterikatan data terhadap data sampel. Memilih sampel yang mempresentasikan kelas khusus atau proses menemukan defenisi kesamaan dalam suatu kelas

5. Pengklasteran

   Proses untuk membagi semua data kedalam kelompok atau mengelompokan data yang memiliki karakteristik tertentu yang sama

6. Asosiasi

   Menemukan atribut mana yang "*go to-gether*" atau yang muncul bersamaan.

2.2.6 *Sentiment Analysis*

Sentiment analysis atau opinion mining mengacu pada bidang yang luas dari pengolahan bahasa alami, komputasi linguistic dan text mining yang bertujuan

menganlisa pendapat, sentimen, evaluasi, sikap, penilaian dan emosi seseorang apakah pembicara atau penulis berkenaan dengan suatu topik , produk, layanan, organisasi, individu, ataupun kegiatan tertentu (Liu, 2012). Tugas dasar dalam analisis sentimen adalah mengelompokkan teks yang ada dalam sebuah kalimat atau dokumen kemudian menentukan pendapat yang dikemukakan dalam kaliamat atau dokumen tersebut apakah bersifat positif, negatif atau netral (Dehaff, M., 2010). Sentiment analysis juga dapat menyatakan perasaan emosional sedih, gembira, atau marah. Kita dapat mencari pendapat tentang produk-produk, merek atau orang-orang dan menentukan apakah mereka dilihat positif atau negatif di web (Saraswati, 2011). Hal ini memungkinkan kita untuk mencari informasi tentang:

a. Deteksi Flame (rants buruk)

b. Persepsi produk baru.

c. Persepsi Merek.

d. Manajemen reputasi.

2.2.7 Text Preprocessing

Tahap *text preprocessing* adalah tahap awal dari *text mining*. Tahap ini mencakup semua rutinitas, dan proses untuk mempersiapkan data yang akan digunakan pada operasi knowledge discovery sistem text mining (Feldman & Sanger, 2007). Tindakan yang dilakukan pada tahap ini adalah toLowerCase, yaitu mengubah semua karakter huruf menjadi huruf kecil dan Tokenizing yaitu proses penguraian deskripsi yang semula berupa kalimat menjadikata-kata dan menghilangkan delimiter seperti titik (.), koma (,), spasi dan karakter angka yang

ada pada kata tersebut (Weiis et al, 2005). Langkah umum pada *text preprocessing*:

1. *Parsing*

   Tulisan dalam sebuah dokumen bisa jadi terdiri dari berbagai macam bahasa, character sets, dan format. Parsing Dokumen berurusan dengan pengenalan dan "pemecahan" struktur dokumen menjadi komponen-komponen terpisah.

2. *Lexical Analysis*

   Lexing atau Tokenisasi proses penghilangan angka, tanda baca dan karakter selain huruf alfabet, karena dianggap sebagai pemisah kata (*delimiter*) dan tidak memiliki pengaruh terhadap pemrosesan teks. Pada tahapan ini juga dilakukan proses *case folding* dan *cleaning*. *Case folding* dimana semua huruf diubah menjadi huruf kecil. *Cleaning* adalah proses membersihkan dokumen dari komponen-komponen yang tidak memiliki hubungan dengan informasi yang ada pada dokumen, seperti tag html, link, script, dsb. *Stopword Removal,* tahap pemilihan kata-kata penting dari hasil token, yaitu kata-kata apa saja yang akan digunakan untuk mewakili dokumen.

3. *Phrase Detection*

   Pada langkah ini tidak hanya dilakukan tokenisasi per kata, namun juga mendeteksi adanya 2 kata atau lebih yang menjadi frase

4. *Stemming*

   Proses pengubahan bentuk kata menjadi kata dasar atau tahap mencari root kata dari tiap kata hasil filtering.

2.2.8 Naive Bayes

Naïve Bayes Classifier merupakan sebuah metoda klasifikasi yang berakar pada teorema *Bayes*. Metode pengklasifikasian dengan menggunakan metode probabilitas dan statistik yg dikemukakan oleh ilmuwan Inggris *Thomas Bayes*, yaitu memprediksi peluang di masa depan berdasarkan pengalaman di masa sebelumnya sehingga dikenal sebagai Teorema *Bayes.* Ciri utama dari *Naïve Bayes Classifier* ini adalah asumsi yang sangat kuat (naïf) akan independensi dari masing-masing kondisi / kejadian. Naïve Bayes untuk setiap kelas keputusan, menghitung probabilitas dengan syarat bahwa kelas keputusan adalah benar, mengingat vektor informa si obyek. Algoritma ini mengasumsikan bahwa atribut obyek adalah independen. Probabilitas yang terlibat dalam memproduksi perkiraan akhir dihitung sebagai jumlah frekuensi dari ” master ” tabel keputusan (Delen, 2008).

Naive Bayes Classifier bekerja sangat baik dibanding dengan model classifier lainnya. Hal ini dibuktikan oleh Xhemali , Hinde Stone dalam jurnalnya “*Naïve Bayes vs. Decision Trees vs. Neural Networks in the Classification of Training Web Pages”* mengatakan bahwa “*Naïve Bayes Classifier* memiliki tingkat akurasi yg lebih baik dibanding model *classifier* lainnya”(Widianto, 2019). Beberapa tahapan dari proses algoritma *naïve bayes* yaitu :

1. Menghitung jumlah kelas/label
2. Menghitung jumlah kasus per kelas
3. Kalikan semua variable kelas
4. Bandingkan hasil per kelas

Bentuk umum *teorama bayes* adalah sebagai berikut :

$$P(H|X) = \frac{P(X|H)P(H)}{P(X)}$$

Keterangan :

X = data dengan kelas yang belum diketahui

H = hipotesa data X merupakan suatu kelas spesifik

P(H|X) = probibalitas hipotesis H berdasarkan kondisi X (posterior probibality)

P(H) = probibalitas hipotesis H (prior probability)

Untuk menjelaskan metode Naive Bayes, perlu diketahui bahwa proses klasifikasi memerlukan sejumlah petunjuk untuk menentukan kelas apa yang cocok bagi sampel yang dianalisis tersebut. Karena itu, metode Naive Bayes di atas disesuaikan sebagai berikut :

$$P(C|F_1 \dots F_n) = \frac{P(C)\, P(F_1 \dots F_n|C)}{P(F_1 \dots F_n)}$$

Di mana Variabel C merepresentasikan kelas, sementara variabel $F_1 \dots F_n$ merepresentasikan karakteristik petunjuk yang dibutuhkan untuk melakukan klasifikasi. Maka rumus tersebut menjelaskan bahwa peluang masuknya sampel karakteristik tertentu dalam kelas C (Posterior) adalah peluang munculnya kelas C(sebelum masuknya sampel tersebut, seringkali disebut prior), dikali dengan peluang kemunculan karakteristik karakteristik sampel pada kelas C (disebut juga likelihood), dibagi dengan peluang kemunculan karakteristik karakteristik sampel secara global (disebut juga evidence). Rumus diatas dituliskan secara sederhana :

$$Posterior = \frac{Prior\ x\ likelihood}{evidence}$$

2.2.9 *Term Frequency – Inverse Document Frequency (TF-IDF)*

TF-IDF adalah ukuran statistik yang menggambarkan pentingnya suatu istilah terhadap sebuah dokumen dalam sebuah kumpulan dokumen atau korpus (Wikipedia). TF-IDF digunakan untuk memberikan pembobotan hubungan suatu kata atau istilah terhadap keseluruhan ulasan. Frekuensi kemunculan kata di dalam ulasan menunjukkan seberapa penting kata itu di dalam ulasan sehingga ulasan dapat diklasifikasikan ke dalam kelas yang sesuai (Muktafin, et al., 2020). Nguyen, et al. pada 2018 menjelaskan TF-IDF sebagai berikut,

a. Term Frequency (TF)

   TF adalah frekuensi kemunculan dari suatu kata dalam dokumen. Dalam model ini, setiap dokumen direpresentasikan sebagai vektor 0s dan 1s. Jika sebuah kata ada dalam dokumen, posisinya yang sesuai dalam vektor dikodekan sebagai "1" dan jika tidak, itu dikodekan sebagai 0.

$$TF = \frac{frekuensi\ kata\ dalam\ dokumen}{jumlah\ kata\ dalam\ dokumen}$$

b. Inverse Document Frequency (IDF)

   IDF adalah nilai atau ukuran dari sebuah kata berdasarkan kepentingan relative dari kata tersebut dalam keseluruhan corpus. Semakin sedikit dokumen yang mengandung kata tersebut, maka nilai IDF semakin besar.

$$IDF = \log(\frac{jumlah\ total\ dokumen}{jumlah\ dokumen\ yang\ mengandung\ kata})$$

c. Term frequency - inverse document frequency (TF-IDF)

TF-IDF adalah hasil dari skor TF dan IDF untuk kata tertentu. TF-IDF merepresetasikan dokumen sebagai vector yang berisi skor TF-IDF untuk setiap kata dalam dokumen. TF-IDF mengurangi dampak dari fitur yang sering muncul namun kurang informatif.

$$TFIDF = TF \times IDF$$

2.2.10 *Particle Swarm Optimization*

Particle Swarm Optimization (PSO) didasarkan pada perilaku sekawanan burung atau ikan. Algoritma PSO meniru perilaku social terdiri dari tindakan individu dan pengaruh dari individu-individu lain dalam suatu kelompok. Dalam *Particle Swarm Optimization* (PSO), kawanan diasumsikan mempunyai ukuran tertentu dengan setiap partikel posisi awalnya terletak di suatu lokasi yang acak dalam ruang multidimensi. Setiap partikel diasumsikan memiliki dua karakteristik: posisi dan kecepatan (Santosa, 2011). *Particle Swarm Optimization* merupakan algoritma berbasis populasi yang mengeksploitasi individu dalam pencarian. Dalam PSO populasi disebut *swarm* dan individu disebut *particle*. Tiap partikel berpindah dengan kecepatan yang diadaptasi dari daerah pencarian dan menyimpannya sebagai posisi terbaik yang pernah dicapai. Algoritma PSO terdiri dari tiga tahap, yaitu pembangkitan posisi serta kecepatan partikel, *update velocity* (update kecepatan), *update position*(update posisi) (Kusmarna et al., 2015).

Pada algoritma PSO ini, pencarian solusi dilakukan oleh suatu populasi yang terdiri dari beberapa partikel. Populasi dibangkitkan secara random dengan

batasan nilai terkecil dan terbesar. Setiap partikel melakukan pencarian solusi yang

optimal dengan melintasi ruang pencarian (*search space*). Hal ini dilakukan dengan cara setiap partikel melakukan penyesuaian terhadap posisi terbaik dari partikel tersebut (*local best*) dan penyesuaian terhadap posisi partikel terbaik dari seluruh kawanan (*global best*) selama melintasi ruang pencarian. Jadi, penyebaran pengalaman atau informasi terjadi di dalam partikel itu sendiri dan antara suatu partikel dengan partikel terbaik dari seluruh kawanan selama proses pencarian solusi. Setelah itu, dilakukan proses pencarian untuk mencari posisi terbaik setiap partikel dalam sejumlah iterasi tertentu sampai didapatkan posisi yang relatif *steady* atau mencapai batas iterasi yang telah ditetapkan. Pada setiap iterasi, setiap solusi yang direpresentasikan oleh posisi partikel, dievaluasi performansinya dengan cara memasukkan solusi tersebut kedalam *fitness function.*

Setiap partikel diperlakukan seperti sebuah titik pada suatu dimensi ruang tertentu. Kemudian terdapat dua faktor yang memberikan karakter terhadap status partikel pada ruang pencarian yaitu posisi partikel dan kecepatan partikel [Kennedy and Eberhart, 1995]. Berikut ini merupakan formulasi matematika yang menggambarkan posisi dan kecepatan partikel pada suatu dimensi ruang tertentu :

$$X_i(t) = x_{i1}(t), x_{i2}(t), \dots, x_{iN}(t) \qquad (1)$$

$$V_i(t) = v_{i1}(t), v_{i2}(t), \dots, v_{iN}(t) \qquad (2)$$

Keterangan :

X = posisi partikel

V = kecepatan partikel

i = indeks partikel

t = iterasi ke-t

N = ukuran dimensi ruang

Berikut ini merupakan model matematika yang menggambarkan mekanisme updating status partikel Kennedy and Eberhart [1995]:

$$V_i(t) = V_i(t-1) + c_1 r_1 \left(X_i^L - X_i(t-1)\right) + c_2 r_2 \left(X^G - X_i(t-1)\right) \quad (3)$$

$$X_i(t) = V_i(t) + X_i(t-1) \quad (4)$$

Keterangan :

$X_i^L = x_{i1}^L, x_{i2}^L, \dots, x_{iN}^L \quad mempresentasikan\ local\ best\ dari\ partikel\ ke\ i$

$X^G = x_{i1}^G, x_{i2}^G, \dots, x_{iN}^G \quad mempresentasikan\ global\ best\ seluruh\ kawanan$

Sedangkan c_1 dan c_2 adalah suatu konstanta yang bernilai positif yang biasanya disebut sebagai learning factor. Kemudian r_1 dan r_2 adalah suatu bilangan random yang bernilai antara 0 sampai 1. Persamaan (3) digunakan untuk menghitung kecepatan partikel yang baru berdasarkan kecepatan sebelumnya, jarak antara posisi saat ini dengan posisi terbaik partikel (*local best*), dan jarak antara posisi saat ini dengan posisi terbaik kawanan (*global best).* Kemudian partikel terbang menuju posisi yang baru berdasarkan persamaan (4). Setelah algoritma PSO ini dijalankan dengan sejumlah iterasi tertentu hingga mencapai kriteria pemberhentian,maka akan didapatkan solusi yang terletak pada global best. Algoritma PSO dapat dijabarkan dengan langkah-langkah sebagai berikut :

Misal mempunyai fungsi berikut minimasi *f(x)* dimana $x^{(B)} \leq x \geq x^{(A)}$ (5)

$x^{(B)}$ adalah batas bawah dan $x^{(A)}$adalah batas atas dari *x*.

1. Asumsikan bahwa ukuran kelompok atau kawanan (jumlah partikel) adalah N. Untuk mengurangi jumlah evaluasi fungsi yang diperlukan untuk menemukan solusi, sebaiknya ukuran N tidak terlalu besar, tetapi juga tidak terlalu kecil, agar ada banyak kemungkinan posisi menuju solusi terbaik atau optimal. Jika terlalu kecil sedikit kemungkinan menemukan posisi partikel yang baik. Terlalu besar juga akan membuat perhitungan jadi panjang. Biasanya digunakan ukuran kawanan adalah 20 sampai 30 partikel.
2. Bangkitkan populasi awal x dengan rentang $x^{(B)}$ dan $x^{(A)}$ secara random sehingga didapat $x_1, x_2, \dots, x_N$. Partikel j dan kecepatannya pada iterasi i dinotasikan sebagai $x_j^{(i)}$dan $v_j^{(i)}$sehingga partikel-partikel awal ini dinotasikan

   $$x_1(0), x_2(0), \dots, x_N(0)$$

   Vector $$v_1(0), v_2(0), \dots, v_N(0)$$

   disebut partikel atau vektor koordinat dari partikel (seperti kromosom dalam algoritma genetika). Selanjutnya lakukan evaluasi nilai fungsi tujuan untuk setiap partikel dan nyatakan dengan, $f\big(x_1(0)\big), f\big(x_2(0)\big), \dots, f\big(x_N(0)\big)$
3. Hitung kecepatan dari semua partikel. Semua partikel bergerak menuju titik optimal dengan suatu kecepatan tertentu. Awalnya semua kecepatan dari partikel diasumsikan sama dengan nol. Set iterasi i = 1.
4. Pada iterasi ke-i, temukan 2 parameter penting untuk setiap partikel j yaitu s Nilai terbaik sejauh ini dari $x_j^{(i)}$(koordinat partikel j pada iterasi i) dan nyatakan sebagai $P_{best,j}$, dengan nilai fungsi tujuan paling rendah (kasus

minimasi), $f\left[x_j\,(i)\right]$ yang ditemui sebuah partikel j pada semua iterasi sebelumnya.

Nilai terbaik untuk semua partikel $x_j^{(i)}$yang ditemukan sampai iterasi ke-i, G_{best} dengan nilai fungsi tujuan paling kecil/minimum diantara semua partikel untuk semua iterasi sebelumnya $f\left[x_j\,(i)\right]$

5. Hitung kecepatan partikel *j* pada iterasi ke *i* dengan rumus sebagai berikut:

$$V_j(i) = V_j(i-1) + c_1 r_1\left[P_{best,j} - x_j(i-1)\right] + c_2 r_2\left[G_{best} - x_j(i-1)\right],$$
$$j = 1,2,\dots N \qquad (6)$$

dimana c_1 dan c_2 masing-masing adalah learning rates untuk kemampuan individu (*cognitive*) dan pengaruh sosial (kawanan), dan r_1 dan r_2 bilangan random yang berdistribusi uniforml dalam interval 0 dan 1. Jadi parameter c_1 dan c_2 menunjukkan bobot dari memory (position) sebuah partikel terhadap memory (posisi) dari kelompok (swarm). Nilai dari c_1 dan c_2 biasanya adalah 2 sehingga perkalian $c_1 r_1$ dan $c_2 r_2$ memastikan bahwa partikel-partikel akan mendekati target sekitar setengah selisihnya

6. Hitung posisi atau koordinat partikel j pada iterasi ke-i dengan cara

$$x_j(i) = x_j(i-1) + v_j(i), \qquad j = 1,2,\dots,N \qquad (7)$$

Evaluasi nilai fungsi tujuan untuk setiap partikel dan nyatakan sebagai

$$f[x_1(i)], f[x_1(i)], \dots, f[x_N(i)]$$

7. Cek apakah solusi yang sekarang sudah konvergen. Jika posisi semua partikel menuju ke satu nilai yang sama, maka ini disebut konvergen. Jika belum konvergen maka langkah 4 diulang dengan memperbarui iterasi *i* = *i* + 1, dengan cara menghitung nilai baru dari $P_{best,j}$ dan G_{best}. Proses iterasi ini

dilanjutkan sampai semua partikel menuju ke satu titik solusi yang sama. Biasanya akan ditentukan dengan kriteria penghentian (*stopping criteria*), misalnya jumlah selisih solusi sekarang dengan solusi sebelumnya sudah sangat kecil.

2.2.11 *Evaluasi Model Klasifikasi*

Evaluasi model klasifikasi adalah pengukuran performa model klasifikasi dengan membandingkan nilai aktual dengan nilai prediksi. C*onfusion matrix* metode yang digunakan mengukur kinerja suatu model klasifikasi. *Confusion Matrix* adalah pengukuran performa untuk masalah klasifikasi *machine learning* dimana keluaran dapat berupa dua kelas atau lebih (Anggreany, 2020). Pada pengukuran kinerja menggunakan *confusion* matrix, terdapat 4 (empat) istilah sebagai representasi hasil proses klasifikasi (Solichin, 2017).

Keempat istilah tersebut adalah sebagai berikut:

a. True Positive (TP): jika data aktual positif diprediksi positif
b. False Negative (FN): jika data aktual positif diprediksi negatif
c. False Positive (FP): jika data aktual negatif diprediksi positif
d. True Negative (TN): jika data aktual negatif diprediksi negatif

Tabel 2.2 *Confusion Matrix (Ramadhan & Muslim, 2018)*

Kelas	Prediksi Positif	Prediksi Negatif
Aktual Positif	TP	FN
Aktual Negatif	FP	TN

Berdasarkan nilai *True Negative* (TN), *False Positive* (FP), *False Negative* (FN), dan *True Positive* (TP) diperoleh nilai akurasi, presisi, *recall* dan *f-measure* (Ramadhan & Muslim, 2018). Berikut penjelasan dari masing masing nilai tersebut,

a. Akurasi

Nilai akurasi merupakan persentase yang menunjukan seberapa akurat sistem dapat mengklasifikasikan data secara benar. Nilai akurasi merupakan perbandingan antara data yang terklasifikasi benar dengan keseluruhan data (Solichin, 2017),

$$akurasi = \frac{(TP + TN)}{(TP + TN + FP + FN)}$$

b. Presisi

Presisi merupakan tingkat ketepatan antara informasi yang diminta oleh pengguna dengan jawaban yang diberikan oleh sistem.

$$presisi\ positif = \frac{TP}{(TP + FP)}$$

c. *Recall*

Recall atau sensitivitas merupakan kualitas seberapa lengkap hasil relevan yang ditampilkan oleh sistem prediksi kelas.

$$recall\ positif = \frac{TP}{(TP + FN)}$$

d. *F-Measure*

F-measure mengukur akurasi data test, digunakan ketika ingin menyeimbangkan presisi dengan *recall*.

$$F - Measure = \frac{2.(recall.precision)}{(recall + precision)}$$

2.2.12 *Synthetic Minority Oversampling Technique (SMOTE)*

Synthetic Minority Oversampling Technique (SMOTE) merupakan metode oversampling yang digunakan untuk menangani masalah ketidakseimbangan kelas. Teknik ini mensintesis sampel baru dari kelas minoritas untuk menyeimbangkan dataset dengan membuat instance baru dari kelas minoritas dengan membentuk kombinasi cembung dari instance tetangga. SMOTE dibagi menjadi langkah-langkah berikut: de-noising, oversampling, dan penyaringan, bagian ini terutama memperkenalkan proses operasi denoising sampel. Kumpulan data awal dihilangkan noise dan diklasifikasikan sebelumnya oleh algoritma mesin vektor pendukung, untuk menemukan sampel minoritas dengan klasifikasi yang salah dan mengidentifikasi sampel noise. Selanjutnya menentukan kategori sampel tetangga untuk setiap sampel yang salah klasifikasi dan menghilangkan sampel kebisingan dari kelas minoritas.