BAB II

KERANGKA TEORI

2.1 Tinjauan Pustaka

Analisis sentimen diperlukan untuk mengetahui seperti apa preferensi wisatawan terhadap produk fasilitas dan layanan di wilayah tersebut destinasi wisata. Penelitian dengan topik analisis sentimen sudah banyak dilakukan sebelumnya seperti penelitian untuk menganalisis sentimen ulasan wisatawan di Thailand selama Pandemic Covid-19 dengan menggunakan algoritma Support Vector Machine, decision tree yang dipakai yaitu Classification and Regression Trees (CART) dan Random Forest dan hasil yang diperoleh Support Vector Machine dengan maksimal akurasi 77,4%, CART 94,3% dan Random Forest dengan akurasi tertinggi yaitu sebanyak 95,4% serta mendapatkan hasil rekomendasi hasil penelitian menyarankan untuk membantu memulihkan pariwisata di Thailand, tujuan wisata, atraksi alam, restoran, dan kehidupan malam harus dipromosikan. Selain itu, dua perhatian utama wisatawan ke Thailand harus diperhatikan yaitu covid-19 dan ketegangan politik saat ini (Leelawat dkk., 2022). Analisis Sentimen Kepuasan Pengguna Operator Seluler Indonesia pada twitter Menggunakan Metode Support Vector Machine dan Fungsi Berbasis Lexicon sebagai Fungsi yang Diperbarui. Hasil penelitian ini menunjukkan bahwa analisis sentimen menggunakan metode Support Vector Machine dan Lexicon Based Features memiliki akurasi sebesar 79%, sedangkan sistem analisis sentimen tanpa metode Lexicon Based Features memiliki akurasi sebesar 84% (Rofiqoh dkk., 2017).

Penelitian tentang ulasan pengunjung wisata borobudur dengan metode Naive Bayes, Decision Tree, Support Vector Machine mendapatkan hasil bahwa algoritma Support Vector Machine lebih dominan dibandingkan dengan algoritma lainnya, dimana akurasi algoritma Support Vector Machine sebesar 99,4% jika dibandingkan dengan algoritma Decision Tree sebesar 94% dan algoritma Naive Bayes sebesar 98% rekomendasi bagi pengelola objek wisata di Candi Borobudur mempertahankan dan meningkatkan kinerja layanan produk dan layanan terkait

budaya dan penerapan nilai-nilai sapta pesona untuk meningkatkan kepuasan dan citra pengunjung objek wisata Candi Borobudur merupakan objek wisata yang sangat penting di Indonesia (Singgalen, 2022). Pada penelitian Fanissa dkk., (2018) menggunakan metode Naive Bayes dengan pemilihan fitur pemeringkatan Query expansion dalam penelitiannya untuk mengurangi kompleksitas komputasi pemeringkatan. Informasi yang digunakan berasal dari ulasan di situs web TripAdvisor. Hasil penelitian ini menunjukkan bahwa 75 fitur hasil seleksi memiliki akurasi terbaik sebesar 86,6%.

Baid dkk. (2017) dalam penelitiannya adalah untuk memeriksa sentimen review film di situs web Internet Movie Database (IMDb) dengan membandingkan beberapa algoritma, seperti Naive Bayes, K-Nearest Neighbor, dan Random Forest. Data diproses menggunakan perangkat lunak Weka. Hasilnya menunjukkan bahwa algoritma Naive Bayes memiliki akurasi tertinggi sebesar 81,45%, sementara algoritma K-Nearest Neighbor memiliki akurasi sebesar 55,30%, dan algoritma Random Forest memiliki akurasi sebesar.

Sulton (2021) melakukan penelitian untuk mengkategorikan review baru berdasarkan data review pembaca dari situs media sosial Twitter. Penelitian menggunakan Multinomial Naive Bayes, Maximum Entropy, dan Support Vector Machine, dengan dataset 1.600.000 tweet. Jumlah data, jumlah fitur, metode pra-pemrosesan, dan algoritme klasifikasi yang dipilih memengaruhi kinerja penelitian. Hasil penelitian menunjukkan bahwa jumlah data dan algoritme klasifikasi yang dipilih memengaruhi akurasi klasifikasi. Akurasi tertinggi—sebesar 85,33%—diberikan oleh metode Multinomial Naive Bayes dibandingkan dengan metode lain. Ini menunjukkan bahwa Multinomial Naive Bayes dapat menjadi metode yang bagus untuk klasifikasi review data dari situs media sosial Twitter.

Artikel ini membahas tentang bagaimana membandingkan algoritma Support Vector Machine, Random Forest, dan Random Forest Support Vector Machine yang sangat cocok dalam teknik klasifikasi. Hasil eksperimen, disimpulkan bahwa algoritma Random Forest Support Vector Machine lebih baik daripada algoritma lainnya untuk dataset ulasan produk yang ditawarkan oleh

Amazon. Alasan hasil yang lebih baik dalam metodologi klasifikasi hibrida yang digunakan dalam artikel ini adalah karena menggunakan kelebihan dari masing-masing metode klasifikasi tradisional Random Forest dan Support Vector Machine (Al Amrani dkk., 2018). Artikel tersebut membahas analisis sentimen tweet berbahasa Malayalam menggunakan teknik machine learning dengan membandingkan akurasi tiga algoritma klasifikasi yaitu Naive Bayes, Support Vector Machine, dan Random Forest. Beberapa fitur seperti Bag-of-Words, TF-IDF, Unigram dengan Sentiwordnet, dan Unigram dengan Sentiwordnet termasuk kata negasi juga digunakan. Hasilnya menunjukkan bahwa klasifikasi dengan menggunakan fitur Unigram dengan Sentiwordnet termasuk kata negasi memiliki akurasi tertinggi menggunakan algoritma Random Forest (Soumya dkk., 2020).

Al-Smadi dkk., (2018) melakukan analisis sentimen terhadap data review hotel dalam bahasa Arab. Ada perbandingan antara dua metode klasifikasi, yaitu Recurrent Neural Network dan Support Vector Machine pada dataset sebanyak 24.028. Hasil penelitian menunjukkan bahwa metode Suport Vector Machine memiliki performansi yang lebih baik dibandingkan metode Recurrent Neural Network berdasarkan nilai F-1 dan akurasi sebesar 89,9% dan 95,4%. Aliza dkk. (2020) menerapkan metode naive Bayes dan Support Vector Machine untuk mengklasifikasikan ulasan publik di situs media sosial twitter tentang penerapan “lockdown” . Data yang digunakan adalah melalui live tweet di tanggal 19 April 2020 pada pukul 12.42-15.02. Hasil penelitian ini menunjukkan bahwa akurasi metode Support Vector Machine sebesar 87% sedangkan akurasi metode Naive Bayes sebesar 81%.

2.2 Dasar Teori

2.2.1 Web Scraping

Web Scraping adalah aktivitas mendapatkan sebagian informasi (spesifik) dari seluruh obyek sebuah situs menggunakan alat tertentu. Dalam pengertian lainnya juga *web scraping* adalah sebuah proses mengekstraksi informasi bermanfaat dari sebuah laman website *HyperText Markup Language (HTML)* dan menyimpannya ke dalam format spesifik, seperti *excel, comma-separated- values*

(CSV), atau dalam bentuk lain menggunakan PurseHub (Ade dkk., 2022). Teknik *web scraping* dapat digunakan untuk mengekstrak sejumlah data semi- terstruktur yang tertanam *(embedded)* di dalam laman HTML (Kumaresan dan Kalpana, 2022). Tujuan *Web Scraping* yaitu mengambil sebuah dokumen semi- terstruktur dari internet, umumnya berupa halaman-halaman website dalam bahasa markup seperti HTML atau Extensible HyperText Markup Language (XHTML), dan menganalisa dokumen untuk diambil data tertentu dari halaman tersebut untuk digunakan bagi kepentingan lain (Rahmattullah, 2022).

Terdapat beberapa langkah dalam proses *webscraping*, yaitu (Josi dkk., 2014):

1. *Create Scraping Template*

   Pembuat program mempelajari dokumen HTML dari website yang akan diambil informasinya dari *tag* HTML yang mengapit informasi yang akan diambil.

2. *Explore Site Navigation*

   Pembuat program mempelajari teknis navigasi pada *website* yang akan diambilinformasinya untuk ditirukan pada aplikasi *web scraper*.

3. *Automate Navigation and Extraction*

   Berdasarkan informasi yang didapatkan dari langkah 1 dan 2 diatas, aplikasi *web scraper* dibuat untuk mengotomatisasi pengambilan informasi dari *website* yang ditentukan.

4. *Extracted Data and Package History*

   Informasi yang didapat dari langkah 3 disimpan dalam tabel atau tabel-tabel *database*.

2.2.2 Text Mining

Text mining adalah salah satu aspek utama penambangan data yang bertujuan untuk mengungkap pengetahuan yang sebelumnya tidak diketahui namun memiliki potensi nilai dari data teks yang tidak terstruktur atau semi-terstruktur. Penambangan teks juga menghadapi sejumlah masalah, seperti volume data yang

besar, dimensi yang tinggi, perubahan struktur dan data yang berkelanjutan, dan adanya "gangguan" dalam data. Berbeda dengan penambangan data, yang utamanya berfokus pada pemrosesan data terstruktur, *text mining* lebih berurusan dengan data yang bersifat tidak terstruktur atau setidaknya semi terstruktur, seperti data teks (Fitri dkk., 2019). Informasi teks yang diterima dianalisis dan digabungkan sesuai dengan istilah pencarian. Ada tahapan-tahapan dalam analisis *text mining* seperti: Pengumpulan data yang dilanjutkan dengan ekstraksi sesuai dengan kebutuhan masing-masing. *Text mining* dapat digunakan dalam analisis sentimen dengan mengambil data teks dari internet. *Text mining* menghasilkan data dalam bentuk data yang tidak terstruktur atau *unstructured.* Ini membedakan text mining dari data mining.. Sebaliknya, data terstruktur digunakan dalam konteks penambangan data. Oleh karena itu, terdapat langkah-langkah tertentu yang perlu dilakukan untuk membuat data yang dihasilkan dari text mining dapat diolah, yaitu melalui langkah preprocessing.

Sebelum dimasukkan ke dalam proses utama, teks dokumen yang akan digunakan harus disiapkan terlebih dahulu. Text preprocessing mengubah data teks yang tidak terstruktur atau tidak teratur menjadi data yang terstruktur. Ini adalah proses persiapan teks dokumen atau dataset mentah ini. Secara umum, langkah-langkah yang dilakukan dalam tahapan preprocessing adalah sebagai berikut:

1. *Cleaning*

   langkah pembersihan *(cleaning)* membersihkan tweet dari kata-kata yang tidak penting. Karakter HTML, kata kunci, ikon emosi, hashtag (#), username (@username), URL (http://situs.com), dan alamat email adalah semua kata-kata yang dihilangkan.

2. *Case Folding*

   Langkah case-folding menjadi penting dalam mengonversi keseluruhan teks dokumen menjadi bentuk standar, biasanya dalam bentuk huruf kecil, karena tidak semua dokumen teks menggunakan huruf kapital secara konsisten. Proses case-folding adalah proses penyamaan case dalam dokumen, yang dilakukan untuk mempermudah pencarian dengan mengubah semua kata menjadi huruf kecil.

3. *Stopword*

   Proses identifikasi dan penghapusan kata-kata yang sering muncul dalam dokumen teks tetapi tidak signifikan untuk klasifikasi sentimen disebut sebagai langkah *stopword*. Tujuan penggunaan stopword adalah untuk menghilangkan kata-kata yang tidak penting bagi klasifikasi sentimen. Contoh stopwords dalam bahasa Indonesia adalah sebagai berikut: yang, juga, dari, dia, kami, kamu, aku, saya, ini, itu, atau, dan, tersebut, pada, dengan, adalah, yaitu, dan ke.

4. *Tokenizing*

   Tokenisasi adalah langkah di mana deretan kata dalam kalimat, paragraf, atau halaman dipisahkan menjadi token atau potongan kata tunggal, yang sering disebut sebagai "kata". Tokenisasi juga melibatkan menghilangkan beberapa karakter tertentu yang dianggap sebagai tanda baca dalam teks.

5. *Normalizing*

   Mengubah kata-kata yang tidak baku menjadi yang sesuai dengan aturan bahasa yang berlaku adalah proses yang dikenal sebagai langkah normalisasi. Tahapan ini bertujuan untuk mengembalikan bentuk penulisan dari setiap kata sesuai dengan Kamus Besar Bahasa Indonesia (KBBI). Proses ini melibatkan pencocokan kata-kata dalam dokumen data latih dan data uji dengan kata-kata dalam kamus Bahasa Indonesia baku.

6. *Stemming*

   Teks disiapkan untuk langkah berikutnya melalui proses *stemming*, langkah pengolahan teks yang bertujuan untuk mengubah kata menjadi bentuk dasarnya, atau kata dasar. Algoritma Sastrawi digunakan untuk melakukan proses stemming dalam penelitian ini..

2.2.3 Analisis Sentimen

Analisis sentimen merupakan bagian dari penelitian dalam ranah text mining yang berfokus pada pengkajian opini dari suatu dokumen teks. Dalam dunia bisnis, analisis sentimen banyak digunakan untuk menganalisis pendapat pelanggan secara otomatis tentang produk dan layanan (Liu, 2012). Informasi merupakan data

yang telah diproses sedemikian rupa sehingga memiliki nilai dan kegunaan bagi pengguna yang dapat membantu dalam pengambilan keputusan. Informasi sendiri dapat dibedakan menjadi dua, yaitu fakta dan opini (Liu, 2012). Fakta bersifat objektif pernyataan tentang sesuatu yang telah terjadi dan biasanya disertai bukti, sedangkan opini lebih banyak subyektif dalam cara seseorang mengekspresikan dirinya terhadap segala sesuatu yang terjadi menurut dirinya masing-masing persepsi dan asumsi. Analisis sentimen dapat didefinisikan sebagai tugas untuk menemukan pendapat atau opini penulis terhadap entitas tertentu. Analisis sentimen dapat didasarkan dan dinilai pada tataran dokumen, kalimat, atau kata (Fitri dkk., 2019).

Analisis sentimen, juga dikenal sebagai opini mining, adalah proses mendapatkan informasi dengan memahami, mengekstrak, dan mengolah teks secara otomatis. Salah satu cabang text mining, analisis sentimen, mulai terkenal pada tahun 2013. Pada dasarnya, analisis sentimen digunakan untuk mengetahui tanggapan dan sikap suatu kelompok atau individu terhadap topik bahasan kontekstual yang terkandung dalam dokumen secara keseluruhan. Tanggapan dan sikap tersebut dapat berasal dari pendapat, penilaian, evaluasi, keadaan afektif (yang merupakan kondisi emosional penulis saat menulis), atau komunikasi emosional (yang merupakan efek emosional yang diterima pembaca oleh penulis). (Al sari dkk., 2022).

Analisis sentimen, juga dikenal sebagai penambangan opini, adalah studi yang bertujuan untuk menganalisis pendapat orang. Fokus utama analisis sentimen adalah pendapat yang menyatakan atau menyiratkan sentimen positif atau negatif. Dalam praktiknya, analisis sentimen dilakukan dari data teks sumber, yang biasanya dapat diakses melalui media digital atau media sosial (Pratama dkk., 2023)

2.2.4 Pembobotan Kata

Setelah tahap preprocessing, pembobotan kata memberikan nilai pada setiap kata. Metode *Frequency-Inverse Term Document Frequency (TF-IDF)* digunakan untuk pembobotan kata ini.

a. *Term Frequency*

Salah satu cara untuk menjelaskan berapa banyak kata atau fitur yang muncul dalam suatu dokumen adalah dengan menggunakan teknik *Term Frequency (TF).* Rumus *Term Frequency* dapat dinyatakan dalam persamaan (2.1) (Nandini dkk., 2019):

$$tf_{t,d} = \begin{cases} 1 + log_{10} tf_{t,d}, if tf_{t,d} > 0 \\ 0 \qquad\qquad otherwise \end{cases} \tag{2.1}$$

dengan :

$tf_{t,d}$ = jumlah yang muncul di *term t* pada dokumen *d*

b. *Document Document Frequency (DF)*

Jumlah dokumen yang mengandung term tertentu, yang biasanya muncul di banyak dokumen, disebut *Document Frequency (DF).*

c. *Inverse Document Frequency (IDF)*

Invert Document Frequency (IDF), kata-kata yang paling jarang muncul dalam dokumen akan memiliki nilai yang lebih tinggi.. Rumus *Inverse Document Frequency (IDF)* dapat dinyatakan dalam persamaan (2.2) (Nandini dkk., 2019):

$$idf_t = log_{10}(\frac{N}{df_t}) \tag{2.2}$$

dengan:

N = jumlah yang terdapat di dokumen teks

df_t = jumlah dokumen yang terdapat term t

d. *Term Frequency-Inverse Document Frequency (TF-IDF)*

Algoritma pembobotan *Term Frequency-Inverse Document Frequency (TF-IDF*) terdiri dari dua sub-algoritma yang masing-masing memiliki bobot yang berbeda: *Term Frequency (TF)* dan *Inverse Document Frequency (IDF).* Algoritma ini menghasilkan keluaran dengan memberikan nilai tinggi pada fitur yang sering muncul dalam dokumen dan nilai rendah pada fitur yang jarang

muncul (Rahmattullah, 2022). Rumus *Term Frequency-Inverse Document Frequency (TF-IDF)* dalam persamaan (2.3):

$$W_{t,d} = W_{tft,d}\, x\, idf_t \tag{2.3}$$

dengan:

$W_{tft,d}$ = Nilai $term\ frequency$

df_t = jumlah dokumen yang terdapat term

2.2.5 Lexicon Based

Metode Lexicon Based menggunakan kamus sentimen, yaitu kumpulan kata-kata yang bersentimen, dan menghitung berapa kali kata-kata tersebut muncul dalam dokumen teks. Metode ini dianggap mudah, layak, dan efektif untuk melakukan analisis sentimen terhadap data media sosial. Jenis data yang cocok dengan metode Lexicon Based meliputi data dari kuesioner, Twitter, Facebook, atau jenis data media sosial lainnya yang berisi pendapat pelanggan tentang barang atau jasa tertentu (Matulatuwa, 2017). Kata sifat, keterangan, kata kerja, dan kata benda biasanya memiliki kata-kata dengan nilai sentimen.

Menentukan apakah suatu teks memiliki sentuhan positif atau negatif, metode ini menggunakan kamus yang berisi kata-kata opini. Kamus ini disebut kamus opini atau opini lexicon.. Dalam penelitian ini, data yang telah melalui tahap preprocessing akan diartikan terlebih dahulu ke dalam Bahasa Inggris. Dengan menggunakan salah satu *translator* yaitu *Deep Translator*, yang memiliki akurasi terjemahan Bahasa inggris yang bagus. Meskipun beberapa kata harus diubah secara manual. Setelah diterjemahkan, metode berbasis lexicon dengan SentiWordNet akan digunakan. SentiWordNet adalah sumber leksikal yang dimaksudkan untuk membantu menambang pendapat dan menganalisis perasaan. Semua informasi ini tersedia secara publik dan dapat diakses secara gratis. (Cernian dan Sgarciu, 2015). *SentiWordNet, WordNet* dapat digunakan untuk menambang pendapat domain dan menganalisis perasaan. *SentiWordNet* tetap menggunakan format *WordNet* untuk menyusun *synset* dan *glossesnya.* Informasi dalam *SentiWordNet* disajikan dalam format yang mengikuti struktur berikut (Cernian

dan Sgarciu, 2015):

- #POS : berisi kelas (*part of speech*) dari *synset* (kata).
- #ID : berisi kode unik untuk setiap *synset* dengan kelas kata tertentu.
- #PosScore : berisi nilai sentimen positif dari *synset*.
- #NegScore : berisi nilai sentimen negatif dari *synset*
- #SynsetTerms : berisi daftar sinonim dari *synset*. Setiap sinonim dipisahkan dengan spasi dan mengandung informasi indeks dari *synset* yang menandai seberapa sering *synset* dalam konteks tersebut digunakan.
- #Gloss : berisi makna dan konteks dari *synset*.

Setelah dilakukan pembobotan kata dengan *SentiWordNet* tahap selanjutnya adalah memberikan penilaian pada setiap kalimat berdasarkan pembobotan dari kata-kata dari kalimat tersebut.

Nilai sentimen, yang diperoleh dengan menjumlahkan nilai *PosScore* atau *NegScore* masing-masing kata dan membaginya dengan nilai total *PosScore* dan *NegScore*, digunakan untuk menentukan berat yang diberikan. Dapat dilihat pada Persamaan (2.4) dan Persamaan (2.5).

$$sum\_(kata) = \sum_{i=1}^{n} PosScore(kata) \quad (2.4)$$

$$sum\_(kata) = \sum_{i=1}^{n} NegScore(kata) \quad (2.5)$$

Persamaan (2.6) menunjukkan perhitungan nilai total dari PosScore dan NegScore kata.

$$total\_score = sum\_PosScore + sum\_NegScore \quad (2.6)$$

Persamaan (2.7) dan Persamaan (2.8) menunjukkan perhitungan nilai sentimen kata pada kelas positif dan negatif.

$$senti\_score\_(kata) = \frac{sum\_PosScore(kata)}{total\_score(kata)} \quad (2.7)$$

$$senti\_score\_(kata) = \frac{sum\_NegScore(kata)}{total\_score(kata)} \quad (2.8)$$

Nilai sentimen suatu kata adalah bilangan real yang berkisar antara 0 dan 1; nilai yang lebih tinggi pada kata tersebut menunjukkan sentimen yang lebih positif, dan nilai yang lebih rendah pada kata tersebut menunjukkan sentimen yang

lebih rendah. Setiap kelas positif dan negatif akan memiliki nilai sentimen ini dimasukkan ke dalam perhitungan posterior. Persamaan (2.9) menunjukkan hasil perhitungan (Goel, dkk., 2017).

$$P(c|w) = \frac{(P(w|c)+senti\_score)*P(c)}{P(w)+Senti\ Score} \quad (2.9)$$

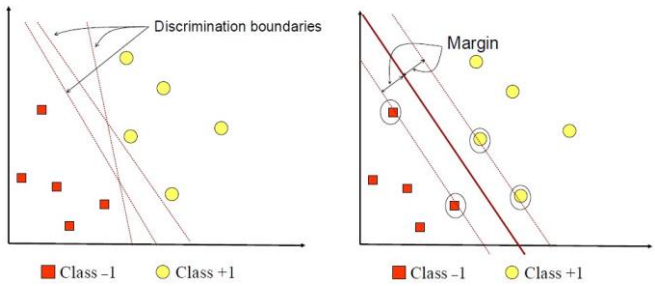
Selanjutnya adalah tahap justifikasi dari kalimat yang sudah dilakukan pembobotan nilai. Sebuah kalimat dianggap berorientasi positif jika jumlah nilai positifnya lebih besar dari jumlah nilai negatifnya; jika jumlah nilai positifnya sama dengan jumlah nilai negatifnya, maka kalimat dianggap netral atau objektif; dan jika jumlah nilai positifnya lebih rendah dari jumlah nilai negatifnya, maka kalimat dianggap negatif (Pamungkas dan Putri, 2016). Persamaan (2.10) menunjukkan hasil.

$$Sentencesentiment \begin{cases} positiveifSpositive > Snegative \\ neutalifSpositive = Snegative \\ negativeifSnegatif > Spositive \end{cases} \quad (2.10)$$

2.2.6 Support Vector Machine

Klasifikasi dan regresi, teknik prediksi *Support Vector Machine (SVM)* masih sangat baru (Handayani, dkk., 2020). Jenis klasifikator SVM adalah biner, linier, dan non-probabilistik. Mencari garis (hyperplane) yang optimal biasanya adalah ide awal yang mendasari pemahaman klasifikasi dengan SVM (Mutawalli, dkk., 2019). Gambar 2.1 menunjukkan warna kuning untuk data positif (+1) dan merah untuk data negatif (-1). Tujuannya adalah untuk membedakan dua kelas data, yaitu kelas positif (+1) dan negatif (-1).

Gambar 2.1 menunjukkan proses SVM secara umum. Grafik di sebelah kiri gambar menunjukkan beberapa garis pemisah (discrimination boundaries) yang mungkin ada pada SVM untuk set data tertentu, sedangkan grafik di sebelah kanan menunjukkan discrimination boundaries dengan margin maksimum. Margin, juga dikenal sebagai "batas pemisah", adalah jarak antara dua kelas data yang paling dekat pada bidang hyperplane.



Gambar 2. 1 Proses SVM dalam menemukan hyperplane

Sumber: (Syaifudin dkk., 2018)

Rumus Support Vector Machine untuk fungsi pada persamaan 2.11 berikut:

$$f(x) = w^T x + b \tag{2.11}$$

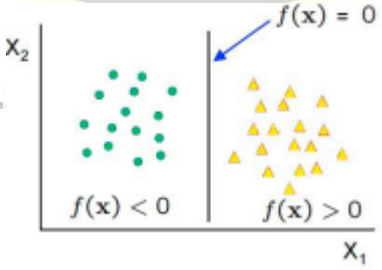
dengan

$w^T = parameter\ bobot$

$x = \ vektor\ input$

$b = bias$

Ruang data input diklasifikasikan menjadi dua kategori karena SVM diwakili oleh bidang hyperplane *f(x)* yang membagi ruang data menjadi dua area yang berbeda secara geometris. Ini ditunjukkan pada gambar 2.2, di mana *f(x)* kurang dari 0 atau negatif akan diklasifikasikan pada ruang berwarna hijau dan *f(x)* lebih dari 0 atau positif akan diklasifikasikan pada ruang berwarna kuning :



Gambar 2. 2 Klasifikasi ruang data

Sumber : (Amrani dkk., 2018)

Fungsi *f(x)* menunjukkan hyperplane yang memisahkan dua wilayah dan memudahkan dalam klasifikasi dataset. Dua wilayah yang dibuat secara geometris oleh hyperplane sesuai dengan dua kategori data di bawah dua label kelas. Karena hiperruang direpresentasikan oleh sebuah garis, maka hal ini juga

dapat diwakili secara matematis dengan persamaan 2.12 berikut:

$$w^T x_i + b \geq +1 \qquad (2.12)$$
$$w^T x_i + b \leq -1$$

Hyperplane juga dapat ditulis dalam persamaan 2.13 berikut

$$f(x) = sgn(w^T x + b) \qquad (2.13)$$

Sgn() atau signum dikenal juga sebagai fungsi tanda yang bertujuan untuk mengubah nilai contohnya 1 dengan positif, 0 dengan negatif dan -1 dengan negatif dengan pesamaan 2.14 berikut :

$$sgn(x) = \begin{cases} 1\ if\ x > 0 \\ 0\ if\ x = 0 \\ -1\ if\ x < 0 \end{cases} \qquad (2.14)$$

Untuk menghitung jarak D dari titik x hyperplane seperti gambar 2.3 dapat dirumuskan dalam persamaan 2.15 berikut:

$$D = \frac{|W^T x + b|}{||w||} \qquad (2.15)$$

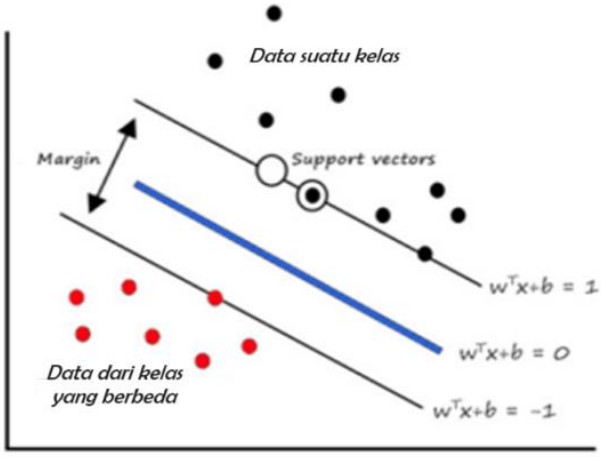
dengan

D = Jarak support vector dengan hyperplane

$W^T = parameter\ bobot$

x = Vektor Input

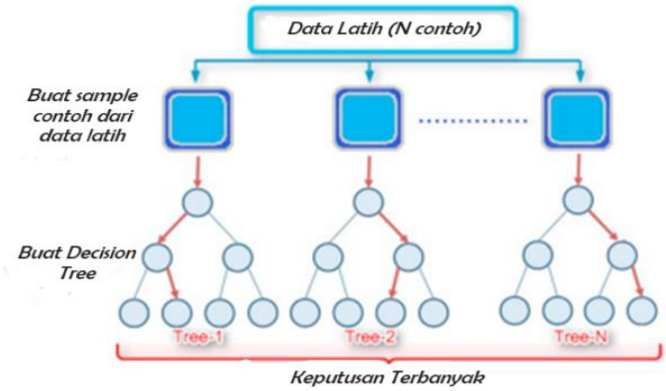
||w|| = Norma Euclidean dari vektor bobot w



Gambar 2. 3 Jarak D dari titik X (Amrani dkk., 2018)

2.2.7 Random Forest

Secara resmi Random Forest pertama kali diusulkan oleh Leo Breiman dan Adèle Cutler pada tahun 2001, Random Forest adalah bagian dari teknik pembelajaran mesin. Algoritma ini menggabungkan konsep subruang acak dan "mengantongi". Keseluruhan algoritme pohon keputusan dilatih dengan beberapa pohon keputusan yang diterapkan pada subkumpulan data yang sedikit berbeda. Random Forest adalah model ansambel berbasis pohon yang bisa digunakan untuk regresi dan klasifikasi. Ini mencapai akurasi tinggi prediksi dengan menggabungkan beberapa peserta didik miskin (pohon keputusan) dari data pelatihan dan pemilihan fitur acak. Random Forest merupang kumpulan Decision Tree. Gambar 2.4 menjelaskan bagaimana tahapan-tahapan pada algoritma decision tree.



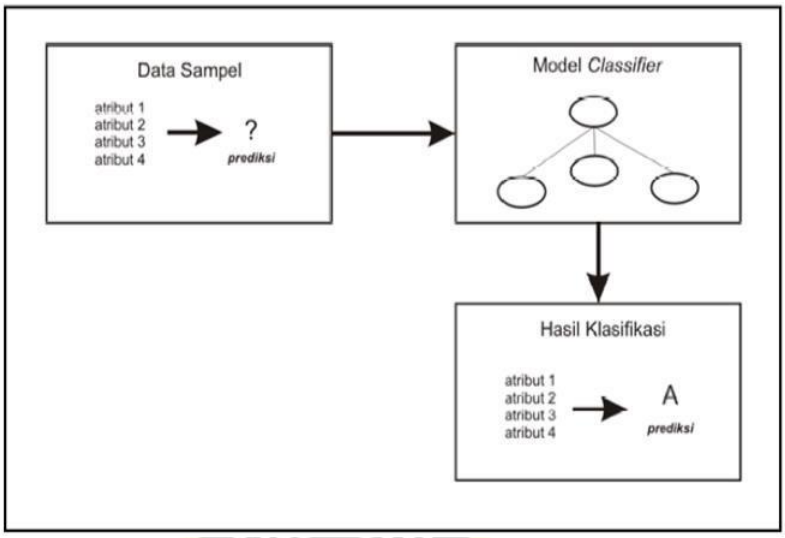
Gambar 2. 4 Representasi Tahapan Random Forest (Amrani dkk.,2018)

Sesuai namanya Random Forest membangun hutan dari kumpulan pohon-pohon. Masing-masing pohon dibangun menggunakan data sample dari data training dengan teknik bootsrap sampling. Data training kemudian diklasifikasikan berdasarkan pohon yang dibangun. Setiap pohon mengklasifikasikan data uji ke dalam kelas data aktif diklasifikasikan dalam kategori dengan suara terbanyak, yang disebut majority voting.

2.2.8 Klasifikasi

Pengelompokan objek data teramati ke dalam suatu kelas tertentu berdasarkan kelas-kelas yang sudah ada dikenal sebagai klasifikasi.. Sedangkan menurut (Liu, 2012) Klasifikasi adalah metode analisis data yang digunakan untuk membuat model prediksi untuk memberi label atau kelas pada data.

Menurut Han dan Kamber (2006) Klasifikasi dilakukan dalam dua tahap. Tahap pertama mencakup pembuatan model; tahap kedua melibatkan analisis rekaman database yang mencakup serangkaian kelas saat ini. Setiap rekaman diasumsikan memiliki kelas yang telah ditentukan sebelumnya berdasarkan atribut label kelas. Oleh karena itu, klasifikasi ini masuk dalam kategori pembelajaran yang diawasi. Ini adalah tahap yang juga disebut sebagai pembelajaran atau pelatihan. Gambar 2.5 menunjukkan bagaimana algoritma klasifikasi akan menganalisis data latihan untuk membangun model klasifikasi. Salah satu cara untuk menggambarkan tahap pembelajaran adalah sebagai tahap pembentukan fungsi atau pemetaan $y = f(x)$, di mana y adalah kelas hasil prediksi dan x adalah rekaman yang ingin dipediksi kelasnya.kelasnya.



Gambar 2. 5 Proses Klasifikasi (Han dan Kamber, 2006).

Ada berbagai ukuran yang dapat digunakan untuk menilai dan mengevaluasi kualitas hasil klasifikasi. Tabel confusion matrix, misalnya, digunakan untuk mengevaluasi kinerja model klasifikasi atau algoritma prediksi (Paulina dkk., 2020).

Tabel 2. 1 Confusion Matrix.

	Actual Class		
	Label	*Positive*	*Negative*
Prediction class	*Positive*	*True Positif (TP)*	*False Positif (FP)*
	Negative	*False Negatif (FN)*	*True Negatif (TN)*

1. *Accuracy*

   Accuracy adalah proporsi dari prediksi yang benar. Nilai aktual dan prediksi diukur dengan akurasi. (Nandini dkk.,2019). Rumus *Accuracy* dapat dilihat pada persamaan (2.16):

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \; X \; 100\% \qquad (2.16)$$

   dengan

$TP = True\ Positif$

$TN = True\ Negatif$

2. *Precision*

Precision adalah proporsi jumlah dokumen teks yang relevan yang diidentifikasi oleh sistem di antara semua dokumen yang dipilihnya. Ini digunakan sebagai tingkat ketepatan informasi yang diminta dengan jawaban yang diberikan oleh system (Nandini dkk., 2019). Rumus *Prescision* dapat dilihat pada persamaan (2.17):

$$Precision = \frac{TP}{\sum TP+FP}\ X\ 100\% \qquad (2.17)$$

3. *Recall*

Recall adalah persentase jumlah dokumen teks yang relevan yang diidentifikasi oleh sistem dari semua dokumen teks relevan yang ada dalam koleksi (Nandini dkk., 2019). Rumus *Recall* dapat dilihat pada persamaan (2.18):

$$Recall = \frac{TP}{TP+FN}\ X\ 100\% \qquad (2.18)$$

4. *F-Measure*

F-Measure digunakan untuk mengukur kinerja sistem secara keseluruhan dalam pengklasifikasian (Nandini dkk., 2019). Rumus *F-Measure* dapat dilihat dengan persamaan (2.19);

$$F - Measure = \frac{2\ x\ Precision\ X\ Recall}{Precision+Recall}\ X\ 100\% \qquad (2.19)$$