

BAB 2

LANDASAN TEORI

2.1 Analisis Sentimen

Analisis sentimen adalah sebuah teknik atau cara yang digunakan untuk mengidentifikasi bagaimana sebuah sentimen diekspresikan menggunakan teks dan bagaimana sentimen tersebut bisa dikategorikan sebagai sentimen positif maupun sentimen negatif. Pendapat yang hampir sama juga digunakan untuk memahami komentar yang diciptakan oleh pengguna (internet) dan menjelaskan bagaimana produk maupun brand diterima oleh mereka. Teknik tersebut dapat menganalisis pendapat, sentimen, evaluasi, penilaian, sikap, dan emosi masyarakat terhadap entitas seperti produk, layanan, organisasi, individu, isu, peristiwa, topik, dan atribut. Selain analisis sentimen, teknik tersebut dikenal juga dengan istilah berbeda, misalnya, *opinion mining*, *opinion extraction*, *sentimen mining*, *subjectivity analysis*, *affect analysis*, *emotion analysis*, *review mining*, dan lain-lain. Dari beragam istilah tersebut, kalangan akademisi dan industri lebih sering menggunakan istilah *sentiment analysis*.

Analisis Sentimen ditujukan untuk mencari pendapat orang lain. Ini tidak hanya berlaku untuk individu tetapi juga berlaku untuk organisasi. Contohnya saat ini, jika seseorang ingin membeli produk konsumen, tidak lagi terbatas untuk meminta pendapat teman dan keluarga seseorang karena ada banyak ulasan pengguna dan diskusi tentang produk di forum publik di website. Bagisebuah organisasi, mungkin tidak perlu lagi melakukan survei, jajak pendapat, dan memfokuskan diri untuk mengumpulkan opini publik. Beberapa tahun terakhir, postingan pendapat di media sosial juga telah membantu

membentuk bisnis, mempengaruhi sentimen publik dan emosi publik [8].

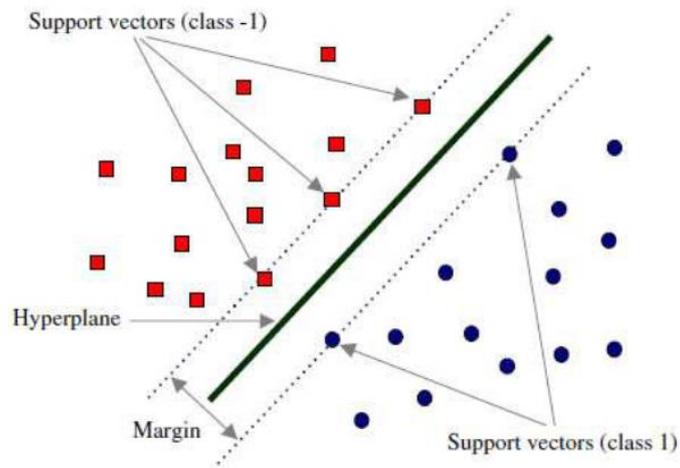
Berdasarkan penjelasan diatas, Analisis Sentimen merupakan metode untuk melakukan klasifikasi terhadap komentar atau opini yang diungkapkan dalam teks pada media digital. Analisis Sentimen banyak digunakan untuk mengambil data opini atau feed back dari konsumen suatu perusahaan untuk kepentingan perusahaan. Namun, Analisis Sentimen juga dapat dilakukan untuk menarik opini publik. Pada laporan ini Analisis Sentimen digunakan dalam klasifikasi review opini pada suatu aplikasi Kesehatan.

2.1.1 Analisis Sentimen Berdasarkan Aspek

Analisis sentimen berbasis aspek atau *Aspect-based Sentiment Analysis* merupakan perkembangan dari analisis sentimen yang hanya mengacu pada sebuah kalimat. Analisis sentimen berbasis aspek dari sebuah kalimat merupakan sebuah opini yang mengacu kepada entitas yang spesifik dan aspek yang dibahasnya. Analisis sentimen berbasis aspek bertujuan untuk mendeteksi polaritas teks tertulis berdasarkan dengan aspek tertentu. Penelitian *aspect-based sentiment analysis* terdiri dari beberapa task.

2.2 Support Vector Machine (SVM)

Support Vector Machine (SVM) merupakan salah satu algoritma *supervised learning* untuk analisis klasifikasi dan regresi. Keluaran dari algoritma ini adalah *hyper-plane* yang optimal dan memaksimalkan jentang antara dua kelas. Pada gambar di bawah menggambarkan tentang cara kerja dari SVM :



Gambar 2.1 Hyperlane pemisah kelas positif (+1) dan negatif (-1)

Pada gambar 2.1, diperlihatkan tentang dua kelas yang diwakilkan oleh kotak merah untuk kelas -1 dan bulat biru untuk kelas 1. Pada SVM, klasifikasi akan dilakukan dengan cara mencari garis pemisah (*hyper-plane*) antara kelas -1 dan kelas 1 yang optimal. *Hyper-plane* yang optimal dapat ditemukan dengan cara mengukur *margin*/jenjang antara kedua kelas. Dalam membantu hal tersebut, SVM memiliki beberapa kernel yang dapat meningkatkan akurasi yang didapat.

Dalam linear SVM, pemisah merupakan fungsi linear. Data latih dinyatakan oleh (x_i, y_i) dan $x_i = \{x_1, x_2, \dots, x_{iq}\}$ merupakan atribut (fitur) set untuk data latih kelas ke- i . Untuk $y_i \in \{-1, 1\}$ menyatakan label kelas. Pendefinisian persamaan suatu *hyperplane* pemisah yang dituliskan dengan :

$$X_i * w + b \geq 1 \quad (2.1)$$

$$X_i * w + b = 0 \quad (2.2)$$

$$X_i * w + b \geq -1 \quad (2.3)$$

Keterangan : w = nilai dari bidang normal

b = posisi bidang terhadap pusat koordinat

X_i = data input

Dengan mengetahui garis normal dan posisi terhadap bidang yang relative terhadap titik koordinat maka kita akan menentukan nilai jarak terdekat terhadap margin. Hal tersebut menjadikan sebagai pembatas atau support untuk menentukan perbedaan terhadap nilai sentiment -1 atau negatif, 0 atau netral dan 1 atau positif. Nilai jarak margin dapat dioptimalkan dengan menggunakan hyperplane dan titik terdekatnya dengan $1 / |w|$. Untuk mendapatkan titik tersebut perlu adanya persamaan Quadratic Programming (QP) Problem, dengan mencari persamaan terhadap nilai titik minimum pada persamaan 2.4 dan mendapatkan nilai constraint dengan persamaan 2.5.

$$\tau(w) = \frac{1}{2} \|w\|^2 \quad (2.4)$$

$$y_i (w x_i + b) \geq 1, i = 1, \dots, n \quad (2.5)$$

Untuk mengoptimisasi pada linear dan tidak membatasi masalah permodelan persamaan dapat diberikan lagrange Multiplier. Dengan adanya fungsi tujuan, fungsi langragian fungsi kendala dapat mempresentasikan kondisi Yang membatasi untuk meningkatkan optimasi. Dengan menambahkan metode lagrange Multiplier persamaan dapat menjadi 2.6

$$\min_{w,b} L_p(w,b,a) = \frac{1}{2}|w|^2 - \sum_{i=1}^n a_i y_i (x_i \cdot w + b) + \sum_{i=1}^n a_i \quad (2.6)$$

w dan b memberi nilai faktor terhadap L_p maka, perlu diperhatikan untuk meminimalkan nilai L_p dengan tambahan konstrain $a_i \geq 0$ maka $\frac{\delta}{\delta b} L_p(w,b,a) = 0$ ketika diperoleh persamaan 2.7, dan terhadap nilai w ketika $\frac{\delta}{\delta w} L_p(w,b,a) = 0$ maka diperoleh persamaan 2.8 berikut :

$$\sum_{i=1}^n a_i y_i = 0 \quad (2.7)$$

$$w = \sum_{i=1}^n a_i y_i x_i \quad (2.8)$$

Namun terdapat kekurangan yang umumnya w dapat bernilai ∞ atau tak terhingga sehingga tidak mendapatkan hasil yang tepat berdasarkan support vector yang diberikan perlu adanya persamaan problem LD sebagai substitusi terhadap problem L_p . Substitusi persamaan akan memperoleh dua problem LD seperti pada persamaan 2.9

$$L_D(a) = \sum_{i=1}^n a_i - \frac{1}{2} \sum_{i=1,j=1}^n a_i a_j y_i y_j x_i x_j \quad (2.9)$$

Dengan persamaan 2.9 kita dapat menentukan kostrain $\min_{w,b} L_p = \max_a L_d$ sehingga untuk menentukan margin terhadap hasil nilai $\max_a L_d$ dapat ditentukan sebagai persamaan 2.10

$$\max_a L_d \equiv \sum_{i=1}^n a_i - \frac{1}{2} \sum_{i=1,j=1}^n a_i a_j y_i y_j x_i x_j \quad (2.10)$$

Maka persamaan tersebut dapat menentukan support vector yang jelas untuk membagi berdasarkan nilai masing terhadap w dan b sehingga pemisah nilai akan menjadi faktor untuk menentukan hasil sentiment berdasarkan perhitungan persamaan diatas. Nilai support vector nantinya akan menjadi pemisah berdasarkan $a \geq 0$ yang menjadi hyperplane dan $\sum_{i=1}^n a_i y_i = 0$ menjadi menghasilkan nilai a_i dengan nilai 0.

Untuk mendapatkan nilai a_i , langkah pertama adalah mengubah setiap *feature* data *training* menjadi nilai vektor (*support vector*) = $x y$. Kemudian nilai vektor dari setiap data *training* dimasukkan ke persamaan (2.11) *kernel trick* phi (φ) berikut ini :

$$\begin{aligned} \varphi(x y) = \{ & \sqrt{x_n^2 + y_n^2} & (2.11) \\ & > 2, \text{ maka } \left[\sqrt{x_n^2 + y_n^2} - y \right. \\ & + |x - y| \sqrt{x_n^2 + y_n^2} - x \\ & + |x - y| \left. \right] \sqrt{x_n^2 + y_n^2} \\ & < 2, \text{ maka } (x y) \end{aligned}$$

Nilai x didapatkan dari persamaan *kernel linear* untuk x berikut:

$$\sum_{i=1}^n x_i^T x_j \quad (2.12)$$

Untuk mendapatkan jarak tegak lurus yang optimal dengan mempertimbangkan vektor positif, maka hasil perhitungan dari substitusi nilai x dan nilai y diberi nilai bias =1. Kemudian cari parameter a_i , dengan terlebih dahulu mencari nilai fungsi setiap data *training*, lalu mencari nilai a_i pada persamaan linear menggunakan persamaan (2.13) dengan memperhatikan $i, j = 1, \dots, n$ berikut :

$$\sum_{i=1}^n a_i T_i^T T_j \quad (2.13)$$

$$\sum_{i=1}^n a_i T_i^T T_j = y_i \quad (2.14)$$

Setelah parameter a_i didapatkan, kemudian masukkan ke persamaan (2.15) berikut:

$$w = \sum_{i=1}^n a_i T_i \quad (2.15)$$

Terakhir akan dicari nilai w dan b untuk menemukan *hyperplane* sebagai patokan proses klasifikasi dengan persamaan sebagai berikut :

$$y = wx + b, w = \sum_i^n a_i s_i \quad (2.16)$$

Nilai s_i merupakan nilai *support vector* yang telah dihitung sebelumnya. Dengan demikian proses klasifikasi selesai dengan memperhatikan *hyperplane* nya.

2.2.1 Kernel Trick

SVM dapat diperluas untuk menggambar sebuah batas keputusan non-linear dengan mentransformasi input dari *original space* atau ruang aslinya menjadi *high-dimensional space* atau ruang dimensi tinggi. Karena hubungan antara ruang *input* dan ruang transformasi bersifat non-linier, maka tujuannya adalah untuk mendapatkan batas keputusan non-linier[8]. Untuk meningkatkan akurasi dari permasalahan tersebut, SVM memiliki kernel *trick* yang dapat membantu mengatasi masalah perubahan data menjadi ruang non-linier.

Secara umum, beberapa fungsi kernel pada SVM yang sering digunakan sebagai berikut[8]:

1) *Linear Kernel*

$$K(X_i, X_j) = X_i^T \cdot X_j \quad (2.17)$$

2) *Polynomial Kernel*

$$K(X_i, X_j) = (X_i \cdot X_j + 1)^h \quad (2.18)$$

3) *Radial Basis Function (RBF)*

$$K(X_i, X_j) = e^{-\|X_i - X_j\|^2 / 2\sigma^2} \quad (2.19)$$

4) *Sigmoidal Kernel*

$$K(X_i, X_j) = \tanh(kX_i \cdot X_j - \delta) \quad (2.20)$$

2.3 Google Play Store

Google Play Store, sebelumnya dikenal dengan nama Android Market, adalah layanan distribusi digital yang dioperasikan dan dikembangkan oleh Google. Aplikasi atau situs ini berfungsi sebagai toko aplikasi resmi untuk perangkat bersertifikat yang berjalan pada sistem operasi Android, memungkinkan pengguna untuk menjelajahi dan mengunduh aplikasi yang dikembangkan dengan kit pengembangan perangkat lunak (SDK) Android dan diterbitkan melalui Google. Google Play juga berfungsi sebagai toko media digital, menawarkan musik, buku, film, dan program televisi.

Aplikasi tersebut tersedia melalui Google Play secara gratis atau dengan biaya tertentu. Aplikasi ini dapat diunduh langsung di perangkat Android melalui aplikasi seluler Play Store atau dengan menerapkan aplikasi ke perangkat dari situs web Google Play. Aplikasi yang memanfaatkan kemampuan perangkat keras perangkat dapat ditargetkan ke pengguna perangkat dengan komponen perangkat keras tertentu, seperti sensor gerak (untuk permainan yang bergantung pada gerakan) atau kamera depan (untuk panggilan video online). Google

Play Store memiliki lebih dari 82 miliar unduhan aplikasi pada tahun 2016 dan mencapai lebih dari 3,5 juta aplikasi yang diterbitkan pada tahun 2017 [9], dengan jutaan pengguna akan lebih baik apa bila aplikasi-aplikasi tersebut dapat di analisa untuk mengurangi review buruk dan memperbaiki aplikasi terhadap ulasan pengguna.

2.4 Web Srapping

Web Scrapping atau yang juga dikenal sebagai ekstraksi web atau *harvesting*, merupakan teknik mengekstrak data dari World Wide Web (WWW) dan simpan ke file sistem atau database untuk pengambilan atau analisis nanti. Biasanya, data web dihapus menggunakan Hypertext Transfer Protocol (HTTP) atau melalui web browser. Ini dilakukan baik secara manual oleh pengguna atau secara otomatis oleh bot atau *web crawler*. Fakta bahwa sejumlah data heterogen terus-menerus dihasilkan di WWW, web scraping secara luas dikenal sebagai teknik yang efisien dan kuat untuk mengumpulkan data besar

2.5 Preprocessing

Preprocessing adalah tahapan untuk mempersiapkan teks menjadi data yang akan diolah di tahapan berikutnya. Teks yang akan dilakukan pada proses pada umumnya memiliki beberapa karakteristik, berdimensi tinggi, terdapat *noise* dan terdapat struktur yang tidak baik.

Tahapan *preprocessing* yang akan dilakukan di penelitian ini yaitu, *casefolding*, *filtering*, *tokenizing*, *convert emoticon*, *convert slang*, *convert negation*, *stopword removal*, dan *stemming*.

2.5.1 Casefolding

Case Folding merupakan langkah dalam mengkonversi keseluruhan teks dalam dokumen menjadi suatu bentuk standar (biasanya huruf kecil atau *lowercase*). Contohnya adalah mengubah semua huruf dalam dokumen menjadi huruf kecil yang dimana dalam dokumen ditemui beberapa kata yang memiliki huruf besar. Hanya huruf “a” sampai dengan “z” yang diterima. Karakter selain huruf dihilangkan.

2.5.2 Filtering

Filtering merupakan proses penghilangan kata-kata (yang dianggap) sebagai kata yang jarang dicari atau jarang digunakan sebagai keywords pada proses pencarian. Filtering dalam *machine learning* adalah tahap mengambil kata-kata penting dari hasil token. Bisa menggunakan algoritma *stoplist* (membuang kata kurang penting) atau *wordlist* (menyimpan kata penting). *Stoplist/stopword* adalah kata-kata yang tidak deskriptif yang dapat dibuang dalam pendekatan *bag-of-words*. Contoh stopwords adalah “yang”, “dan”, “di”, “dari” dan seterusnya.

2.5.3 Tokenizing

Tokenizing merupakan langkah atau tahapan pemotongan string input berdasarkan tiap kata yang menyusunnya. Tokenisasi secara garis besar memecah sekumpulan karakter dalam suatu teks ke dalam satuan kata, bagaimana membedakan karakter-karakter tertentu yang dapat diperlakukan sebagai pemisah kata atau bukan. Ataupun dapat diartikan sebagai proses pemenggalan kata pada dokumen berdasarkan spasi dan tanda –(penghubung).

2.5.4 Convert Emoticon

Convert emoticon adalah proses mengkonversikan emoticon kedalam string yang sesuai dengan ekspresi emoticon itu sendiri. Convert emoticon dilakukan karena pada data komentar yang diambil dari halaman review terkadang memiliki opini yang menggunakan emoticon untuk menyampaikan ekspresi sentimen terhadap topik diskusi tersebut. Hal ini dirasa mempunyai pengaruh terhadap pengklasifikasian sentimen, oleh karena itu convert emoticon. Pada setelah menggunakan scraper untuk mengambil data yang ada di youtube, emoticon yang diambil berubah menjadi simbol-simbol. Rangkaian dari simbol itu bisa di konversikan menjadi sebuah kata yang memiliki makna untuk membantu menganalisis sentimen.

2.5.5 Convert Slang

Slang ragam bahasa tidak resmi dan tidak baku yang sifatnya musiman, dipakai oleh kaum remaja atau kelompok sosial tertentu untuk komunikasi intern dengan maksud agar yang bukan anggota kelompok tidak mengerti. Pada sosial media seperti youtube, data komentar terdapat banyak kata-kata yang tidak baku. Kata yang tidak baku akan mempengaruhi proses selanjutnya yang nantinya sulit dideteksi saat pelabelan di setiap katanya. Kata yang tidak baku tersebut akan diubah menjadi kata yang baku. Pada penelitian kali ini menggunakan fitur lokal, dimana kata-kata yang tidak baku yang terdapat dalam data akan diubah menjadi kata baku. Perubahan kata tidak baku menjadi kata baku didapat dari panduan buku gaul. Proses selanjutnya yaitu kata akan ditentukan kategorinya dan dilihat setiap stemming katanya yang menandakan bahwa kata itu sudah baku.

2.5.6 Convert Negation

Convert Negation adalah proses yang bertujuan untuk menggabungkan kata negasi dengan kata setelahnya, misal “ga mau” menjadi “gamau”. Kata yang termasuk kata negasi memiliki jumlah sebanyak 9 kata, dapat dilihat pada tabel. Proses ini dilakukan karena dapat menentukan hasil deteksi teks.

2.5.7 Stopword Removal

Stopword Removal adalah proses untuk menghilangkan kata yang tidak relevan pada hasil *parsing* sebuah dokumen teks dengan cara membandingkan dengan *stoplist* yang ada. *Stoplist* berisi sekumpulan kata yang tidak relevan namun sering muncul dalam sebuah dokumen. *Stoplist* berisi sekumpulan *stopwords*.

Setiap kata akan diperiksa apakah ada dalam *stoplist* atau tidak, jika sebuah kata termasuk kedalam *stoplist* maka kata tersebut tidak akan diproses lebih lanjut dan akan dihilangkan. Sebaliknya jika sebuah kata tidak termasuk kedalam *stoplist* maka kata tersebut akan masuk ke proses berikutnya.

2.5.8 Stemming

Stemming merupakan tahapan untuk memperkecil jumlah indeks yang berbeda dari suatu dokumen, juga untuk melakukan pengelompokan kata-kata lain yang memiliki kata dasar dan arti yang serupa namun memiliki bentuk atau form yang berbeda karena mendapatkan imbuhan yang berbeda. Sebagai contoh kata bersama, kebersamaan, menyamai, akan distem ke root word-nya yaitu “sama”. Namun, seperti halnya stopping, kinerja stemming juga bervariasi dan sering tergantung pada domain bahasa yang digunakan.

2.6 TF - IDF

TF-IDF merupakan statistik numerik yang mencerminkan betapa pentingnya sebuah kata dalam sebuah dokumen dalam koleksi atau *corpus*.

Metode ini memiliki dua perhitungan yaitu perhitungan TF dan IDF lalu menggabungkan dua perhitungan tersebut dengan mengkalikannya, rumus, Pada persamaan (2.21) dapat dilihat rumus untuk menghitung TF:

$$tf_{ij} = \frac{f_d(i)}{\max f_d(j)} \quad (2.21)$$

Dimana:

tf_{ij} = frekuensi kemunculan term i pada dokumen j dibagi total term pada dokumen j

Selanjutnya adalah untuk menghitung IDF atau *Inverse Document Frequency*, pada dasarnya IDF mengukur jumlah informasi yang diberikan oleh sebuah kata, yaitu apakah term itu biasa atau jarang terjadi di semua dokumen. Berikut ini persamaan (2.22) untuk menghitung IDF:

$$IDF_t = \log\left(\frac{D}{DF_t}\right) \quad (2.22)$$

Dimana:

IDF = Hasil *Inversed Document Frequency*

D = Jumlah dokumen

DF = Jumlah dokumen yang berisi term

Frekuensi kemunculan kata di dalam dokumen yang diberikan menunjukkan seberapa penting kata tersebut di dalam dokumen. Sehingga bobot hubungan antara sebuah kata dan sebuah dokumen akan tinggi apabila frekuensi keseluruhan dokumen yang mengandung kata tersebut akan rendah pada kumpulan dokumen. Setelah menghasilkan nilai perhitungan TF dan IDF tahap selanjutnya adalah untuk menggabungkan hasil perhitungan TF dan IDF supaya menghasilkan nilai TF-IDF. Berikut ini persamaan (2.23) untuk menghitung TF-IDF:

$$TF - IDF = TF \times IDF \quad (2.23)$$

Dengan konvensi, nilai TF-IDF meningkat secara proporsional dengan sebuah kata berapa kali muncul dalam sebuah dokumen, namun diimbangi oleh frekuensi kata di dalam corpus, yang membantu mengendalikan fakta bahwa beberapa kata lebih umum daripada yang lain.

2.7 Confusion Matrix

Confusion Matrix adalah sebuah metode perhitungan yang digunakan untuk mencari keakuratan pada hasil klasifikasi. Confusion Matrix mengandung informasi yang membandingkan hasil klasifikasi yang dilakukan oleh sistem dengan klasifikasi seharusnya.

Pada evaluasi klasifikasi terdapat empat kemungkinan yang bisa terjadi dari hasil klasifikasi suatu data. Jika data positif dan diprediksi positif maka akan dihitung sebagai true positif dan jika data positif diprediksi negatif maka akan dihitung sebagai false negatif. Pada data negatif jika diprediksi negatif akan dihitung sebagai true negatif dan jika diprediksi positif maka akan dihitung sebagai false positif. Inilah yang dinamakan Matriks Konfusi dan untuk lebih jelasnya bisa dilihat pada

Tabel 2.7.1.

Tabel 2.7.1 Confusion Matrix

	Aktual		
	Class	Positive	Negative
Prediksi	Positive	True Positive (TP)	False Positive (FP)
	Negative	False Negative (FN)	True Negative (TN)

2.7.1 Precision

Precision adalah tingkat ketepatan antara informasi yang diminta oleh pengguna dengan jawaban sistem, precision dapat dihitung dengan persamaan (2.24).

$$Precision = \frac{\text{relevant item retrieved}}{\text{retrieved item}} = \frac{TP}{TP + FP} \quad (2.24)$$

2.7.2 Recall

Recall adalah salah satu perhitungan keakuratan prediksi yang digunakan sebagai ukuran tingkat keberhasilan sistem dalam menemukan kembali sebuah informasi, recall dapat dihitung melalui persamaan (2.25).

$$Recall = \frac{\text{relevant item retrieved}}{\text{retrieved item}} = \frac{TP}{TP + FN} \quad (2.25)$$

2.7.3 F-Measure

F-Measure atau F-Score adalah *relative* hasil kombinasi nilai precision dengan nilai recall. F-measure dapat digunakan untuk

mengukur kinerja dari sistem klasifikasi yang merupakan rata-rata harmonis dari *precision* dan *recall*. F-measure dapat memberikan penilaian yang lebih seimbang. F-measure dapat dihitung dengan persamaan (2.26).

$$F - Measure = \frac{2 \cdot (recall \cdot precision)}{(recall + precision)} \quad (2.26)$$

2.7.4 Accuracy

Accuracy adalah tingkat kedekatan antara nilai prediksi dengan nilai aktual. Jika nilai akurasi tinggi maka sebuah sistem akan semakin bagus dalam melakukan prediksi, *accuracy* dapat dihitung dengan persamaan (2.27).

$$Accuracy = \frac{\text{Prediksi data benar}}{\text{Total data}} = \frac{TP + TN}{TP + TN + FP + FN} \quad (2.27)$$

2.8 Python

Python adalah bahasa pemrograman komputer, sama layaknya seperti bahasa pemrograman lain, misalnya C, C++, Pascal, Java, PHP, Perl dan lain-lain. Sebagai bahasa pemrograman, Python tentu memiliki varian, kosakata atau kata kunci, dan aturan tersendiri yang jelas berbeda dengan bahasa pemrograman lainnya.

Python menawarkan code yang ringkas dan mudah dibaca. Meskipun algoritma kompleks dan alur kerja serbaguna mendukung pembelajaran mesin dan AI, kesederhanaan python memungkinkan pengembang untuk menulis sistem yang andal. Pengembang dapat mengerahkan semua upaya mereka untuk memecahkan masalah ML daripada berfokus pada nuansa teknis bahasa tersebut. Selain itu, Python menarik bagi banyak pengembang karena mudah dipelajari.

Kode Python dapat dimengerti oleh manusia, yang membuatnya lebih mudah untuk membangun model untuk pembelajaran mesin.

Banyak programmer mengatakan bahwa Python lebih intuitif daripada bahasa pemrograman lainnya. Yang lain menunjukkan banyak kerangka kerja, pustaka, dan ekstensi yang menyederhanakan penerapan fungsi yang berbeda. Secara umum diterima bahwa Python cocok untuk implementasi kolaboratif ketika banyak pengembang terlibat. Karena Python adalah bahasa tujuan umum, Python dapat melakukan serangkaian tugas pembelajaran mesin yang kompleks dan memungkinkan Anda membuat prototipe dengan cepat yang memungkinkan Anda menguji produk untuk tujuan pembelajaran mesin.

2.9 Unified Modelling Language

UML atau Unified Modeling Language merupakan pengganti dari metode analisis berorientasi objek dan design berorientasi objek (OOAD&D/object oriented analysis and design) yang dimunculkan sekitar akhir tahun 80-an dan awal tahun 90-an. UML memiliki beberapa jenis diagram untuk mempresentasikan proses pembuatan perangkat lunak berorientasi objek, diantaranya adalah *class diagram*, *sequence diagram*, *use case diagram*, dan *activity diagram*.

2.9.1 Use Case Diagram

Use Case Diagram adalah pemodelan untuk menggambarkan behavior / kelakuan sistem yang akan dibuat. Use case diagram menggambarkan sebuah interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat. Secara sederhana, diagram use case digunakan untuk memahami fungsi apa saja yang ada di dalam sebuah sistem dan siapa saja yang dapat menggunakan fungsi-fungsi tersebut.

Simbol-simbol dalam use case diagram memiliki arti masing-masing, definisi dari simbol-simbol use case dapat dilihat pada paparan Tabel 1.1.1

2.9.2 Use Case Scenario

Use case scenario merupakan penjelasan dari setiap *use case*. *Use case scenario* terbagi menjadi tiga bagian, diantaranya :

1. Identifikasi dan inisiasi
2. Langkah yang dilakukan
3. Kondisi, asumsi dan pertanyaan.

2.9.3 Activity Diagram

Activity Diagram merupakan rancangan aliran aktivitas atau aliran kerja dalam sebuah sistem yang akan dijalankan. Activity Diagram juga digunakan untuk mendefinisikan atau mengelompokkan aluran tampilan dari sistem tersebut. Activity Diagram memiliki komponen dengan bentuk tertentu yang dihubungkan dengan tanda panah. Panah tersebut mengarah ke-urutan aktivitas yang terjadi dari awal hingga akhir.

Simbol-simbol dalam activity diagram memiliki arti masing-masing, definisi dari simbol-simbol activity diagram dapat dilihat pada paparan **Tabel 1.1.2.**

2.9.4 Class Diagram

Class diagram adalah visual dari struktur sistem program pada jenis-jenis yang di bentuk. Class diagram merupakan alur jalannya database pada sebuah sistem. Class diagram merupakan penjelasan

proses database dalam suatu program. Dalam sebuah laporan sistem maka class diagram ini wajib ada.

Simbol-simbol dalam class diagram memiliki arti masing-masing, definisi dari simbol-simbol class diagram dapat dilihat pada paparan **Tabel 1.1.3.**

2.9.5 Sequence Diagram

Sequence Diagram adalah suatu diagram yang menjelaskan interaksi objek dan menunjukkan (memberi tanda atau petunjuk) komunikasi diantara objek-objek tersebut. Sequence diagram digunakan untuk menggambarkan perilaku pada sebuah skenario dan mendeskripsikan bagaimana entitas dan sistem berinteraksi, termasuk pesan yang digunakan saat interaksi. Semua pesan dideskripsikan dalam urutan pada eksekusi. Sequence diagram berhubungan erat dengan Use Case Diagram, dimana 1 Use Case akan menjadi 1 Sequence Diagram.

Simbol-simbol dalam sequence diagram memiliki arti masing-masing, definisi dari simbol-simbol sequence diagram dapat dilihat dari paparan **Tabel 1.1.4.**

