BAB 2
LANDASAN TEORI

Berikut adalah studi literatur yang dilakukan terkait dengan penelitian yang dilakukan. Studi literatur terkait yang dilakukan adalah mengenai Twitter, analisis sentimen, *cyberbullying*, pembelajaran mesin, *text preprocessing* dan proses-prosesnya, *multilayer perceptron*, *convolutional neural network*, dan Evaluasi Model.

2.1 Twitter

Twitter adalah media sosial yang menghubungkan penggunanya dengan cerita-cerita, pemikiran, ide dan berita terbaru tentang apa saja yang sedang terjadi di dunia dan dianggap menarik oleh banyak orang yang sudah ada sejak 21 Maret 2006. Twitter memungkinkan penggunanya untuk mengirim dan menerima pesan sepanjang maksimal 280 karakter. Sampai tahun 2022, tercatat 217 juta pengguna aktif Twitter secara global [10]. Secara global, per 2020, *tweet* yang dikirim oleh orang-orang adalah sebanyak 500 juta per hari. Selain sebagai media sosial, Twitter juga sekarang berfungsi sebagai portal berita.

2.2 Analisis Sentimen

Analisis sentimen adalah salah satu sub-bidang pembelajaran mesin yang mempelajari dan mengenali emosi, pendapat, penilaian, dan pandangan dari sekumpulan teks untuk mengidentifikasi karakter suatu kumpulan kata. Proses dari analisis sentimen adalah klasifikasi dokumen berbentuk teks dalam kelas-kelas berjumlah tertentu (biasanya positif dan negatif) [11].

Analisis sentimen juga sering disebut *opinion mining*, dan dapat didefinisikan juga sebagai studi komputasi untuk pengenalan dan mengekspresikan opini, sentimen, dan emosi atau pandangan dalam suatu teks [12].

2.3 Cyberbullying

Para ahli mendefinisikan *cyberbullying* sebagai berikut:

1. *Cyberbullying* adalah perlakuan kasar yang dilakukan oleh seseorang atau sekelompok orang, menggunakan bantuan alat elektronik yang dilakukan

berulang-ulang dan terus menerus pada seorang target yang kesulitan membela diri[13].

2. *Cyberbullying* adalah penggunaan teknologi untuk mengintimidasi, menjadikan korban, atau mengganggu individu atau sekelompok orang[14].

Dari kedua pengertian tersebut, dapat disimpulkan bahwa *cyberbullying* adalah suatu tindakan intimidasi dan mengganggu secara verbal yang dilakukan pada korban dalam dunia maya.

Dalam dataset yang secara khusus digunakan dalam penelitian ini, ada lima tipe *cyberbullying* yang akan menjadi target penelitian, yaitu:

1. Usia (*Age*)

   Bentuk *cyberbullying* ini menyerang hal-hal yang berhubungan dengan usia korban.

2. Etnisitas (*Ethnicity*)

   Bentuk *cyberbullying* ini menyerang hal-hal yang berhubungan dengan etnis atau suku bangsa korban.

3. Jenis Kelamin (*Gender*)

   Bentuk *cyberbullying* ini menyerang hal-hal yang berhubungan dengan jenis kelamin atau suku bangsa korban.

4. Agama (*Religion*)

   Bentuk *cyberbullying* ini menyerang hal-hal yang berhubungan dengan agama atau kepercayaan korban.

5. Bukan *Cyberbullying*

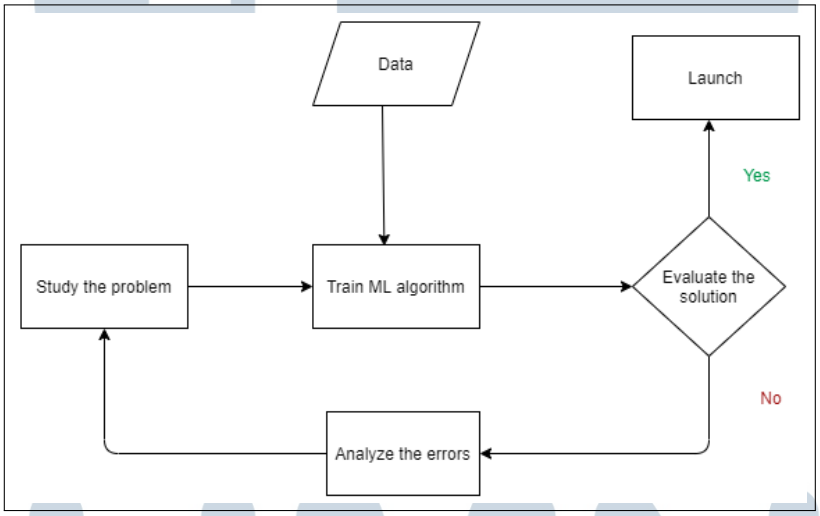
   Cuitan (*tweet* ini tidak ada indikasi *cyberbullying*.

6. *Cyberbullying* bentuk lain

   Bentuk *cyberbullying* ini *tweets* yang tidak termasuk dalam golongan usia/etnisitas/jenis kelamin/agama.

2.4 Pembelajaran Mesin

Singkatnya, pembelajaran mesin, atau *machine learning* adalah studi dari pemrograman agar komputer dapat belajar dari pengalaman sesuai dengan tugas yang diberikan dengan ukuran performa. *Machine learning* baik digunakan jika data yang digunakan berjumlah besar, lingkungan sistem yang senantiasa berubah (fluktuatif), karena *machine learning* mampu beradaptasi dengan data baru, dan saat masalah yang dihadapi benar-benar rumit untuk dipecahkan dengan pendekatan pemrograman konvensional. Secara umum, langkah-langkah dalam machine learning dapat dilihat pada Gambar 2.1.



Gambar 2.1. Langkah-langkah dalam pembelajaran mesin
Sumber: [15]

Pertama, data akan dibagi menjadi data latih (*training*) dan data uji (*testing*), dimana persentase data latih harus lebih besar dari data uji. Data latih akan digunakan untuk membuat model *machine learning*, kemudian data uji tentu saja akan digunakan untuk melakukan pengujiannya. Setelah diuji, akan dilakukan analisis terhadap solusinya dan analisis terhadap error-nya dan dilakukan evaluasi menggunakan metrik evaluasi (contoh: akurasi, *recall*, *precision*, *F1-score*). Jika sudah mencapai metrik evaluasi yang diinginkan, maka model bisa di-*deploy*.

Ada tiga jenis pendekatan dalam pembelajaran mesin, yaitu:

1. Pembelajaran yang dilatih dan tidak dilatih dengan supervision/pengawasan manusia (*supervised learning*, *unsupervised learning*, dan *reinforcement learning*)

(a) *Supervised learning* – data latih yang diberikan pada algoritma *machine learning* sudah termasuk hasil yang diinginkan, yang disebut label. Pekerjaan *supervised learning* yang biasa dilakukan adalah klasifikasi dan prediksi. Algoritma-algoritma yang yang biasa dipakai dalam *supervised learning* antara lain adalah KNN, Regresi Linear, *Support Vector Machine*, Pohon Keputusan, serta Jaringan Saraf Tiruan.

(b) *Unsupervised Learning* - kebalikan dari *supervised learning*, dimana di sini, data latihnya tidak ada label. Algoritma-algoritma yang biasa dipakai dalam *unsupervised learning* adalah: *Clustering (k-means/modes/medoids*, *hierarchical clustering*, dan *expectation maximization*), *Association rule learning* (Algoritma Apriori dan Eclat), dan Visualisasi serta reduksi dimensionalitas (PCA, LLE).

(c) *Reinforcement Learning* - berbeda dengan sistem-sistem pembelajaran di atas, disini, sistem pembelajaran disebut *agent*, dan dalam konteks ini, *agent* akan mempelajari lingkungannya, memilih dan melakukan sesuatu, dan mendapatkan *rewards* atau *punishments*.

2. Pembelajaran yang bisa atau tidak bisa belajar secara bertahap langsung (*online learning*, *batch learning*).

(a) *Batch learning* – dalam *batch learning*, atau nama lainnya, *offline learning*¸ sistem pembelajaran tidak bisa belajar secara bertahap dan harus dilatih dengan semua data yang ada.

(b) *Online learning* – dalam *online learning*, sistem dilatih secara bertahap dengan memberikan data instance secara sekuensial atau berurutan, baik secara individual atau dalam grup kecil yang disebut *mini-batches*.

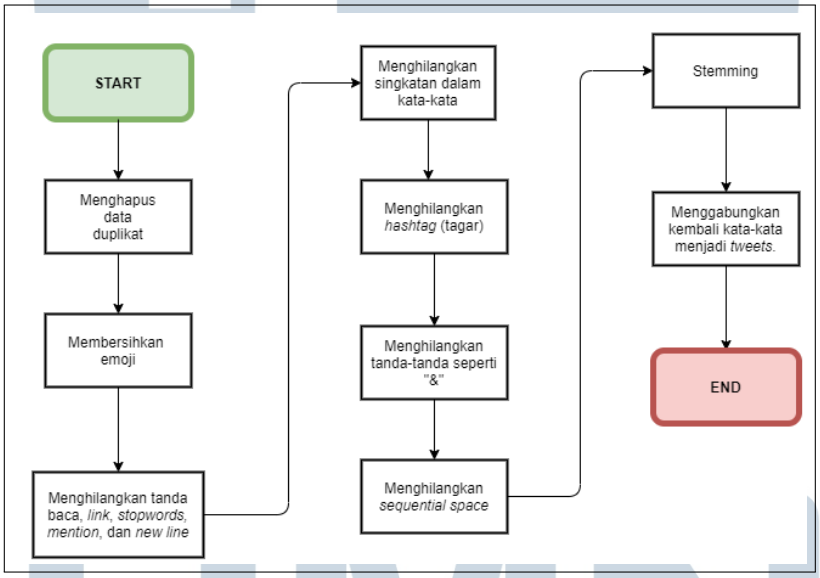
3. Pembelajaran yang membandingkan data baru dengan yang diketahui atau mendeteksi pola-pola dalam data latih untuk membangun model bersifat prediktif (*instance-based learning* dan *model-based learning*).

(a) *Instance-based learning* – sistem melakukan pembelajaran secara langsung, kemudian melakukan generalisasi dari kasus-kasus baru yang ada berdasarkan ukuran kesamaan.

(b) *Model-based learning* – sistem melakukan pembelajaran dengan membangun model dari sekelompok contoh-contoh yang ada untuk melakukan suatu generalisasi.

2.5 Text Preprocessing

Text preprocessing adalah salah satu langkah yang sangat penting dalam melakukan implementasi *machine learning* dalam permasalahan yang berhubungan dengan bahasa, dalam kasus penelitian ini, teks. *Text preprocessing* berfungsi membersihkan data teks dari *noise* sehingga data teks yang akan diberikan kepada model *machine learning* bersih dan lengkap, karena data dalam *machine learning* harus bersih serta lengkap. *Text preprocessing* akan memproses data teks yang ada menjadi bentuk data lain yang lebih mudah diproses oleh komputer. Alur *preprocessing* teks dapat dilihat pada Gambar 2.2.



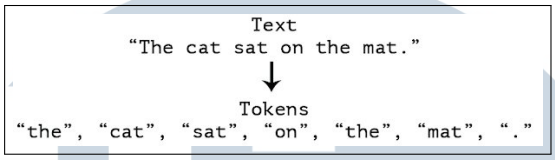
Gambar 2.2. Ilustrasi langkah-langkah *text preprocessing*

Beberapa teknik yang digunakan dalam *text preprocessing* adalah:

2.5.1 Tokenization

Tokenization adalah teknik pengolahan bahasa alami yang mengolah teks mentah, kemudian dipotong-potong menjadi *token* kecil. *Token* kecil ini adalah daftar kata-kata, karakter-karakter, dan bilangan-bilangan, dan pada saat yang bersamaan juga berhubungan dengan menghapus karakter-karakter seperti tanda baca dan spasi. *Token* juga bisa dapat didefinisikan sebagai sekumpulan karakter dalam suatu dokumen tertentu yang dikelompokkan bersama sebagai satu unit semantik untuk diolah[16]. Sebelum melakukan *tokenization*, semua huruf pada teks akan diubah menjadi huruf kecil atau kapital [17]. Pada proses ini juga

dilakukan proses penyaringan terhadap kata yang berawalan bukan huruf[18]. Ilustrasi proses *tokenization* dapat dilihat pada Gambar 2.3.



Gambar 2.3. Ilustrasi proses *tokenization*
Sumber: [19]

2.5.2 Stopword Removal

Stopword removal adalah proses menampilkan daftar kata umum yang maknanya tidak penting dan tidak digunakan untuk kemudian dihilangkan. Proses ini dapat mengurangi jumlah kata yang akan disimpan oleh sistem[7].

2.5.3 Case Folding

Pada tahap ini, semua huruf dalam data akan dikonversi menjadi huruf-huruf kecil (*lowercase*). Huruf-huruf yang diambil adalah dari ”a” sampai ”z”, dan karakter-karakter selain huruf-huruf tersebut akan dianggap sebagai *delimiter*[20].

2.5.4 Noise Removal

Noise removal adalah proses penghilangan digit-digit karakter dan potongan-potongan dari teks yang dapat mengganggu dalam proses analisis teks [21]. Proses ini juga dapat disebut *Data Cleansing*, yang menghilangkan koma, titik, tanda baca, *emoji*, *link*, dan *mention* yang mengganggu untuk mengurangi jumlah noise[7].

2.5.5 Menghilangkan Angka

Proses ini menghilangkan angka-angka yang ada dalam teks[22]. Angka-angka perlu dihilangkan juga dari *tweets* karena angka-angka termasuk *noise* yang berpotensi mempengaruhi algoritma *machine learning*.

2.5.6 Stemming

Stemming adalah proses mengembalikan suatu kata ke bentuk dasarnya. Dengan proses *Stemming*, setiap kata dengan imbuhan akan menjadi bentuk dasarnya sehingga dapat mengoptimalkan proses analisis teks[23].

2.6 Word Embedding

Word Embedding adalah representasi vektor dari kata-kata dengan melakukan penanaman makna semantik dan sintaksis dari teks yang tidak berlabel. *Word Embedding* adalah alat yang ampuh dan banyak digunakan dalam *Natural Language Processing*, termasuk analisis sentimen[24]. Metode word embedding menggunakan teknik pembelajaran model neural network untuk melakukan representasi vektor dari kosakata konstan yang berasal dari kumpulan teks. Secara umum, ada 3 model yang digunakan untuk word embedding, yaitu Word2Vec, GloVe (Global Vector), dan FastText [25].

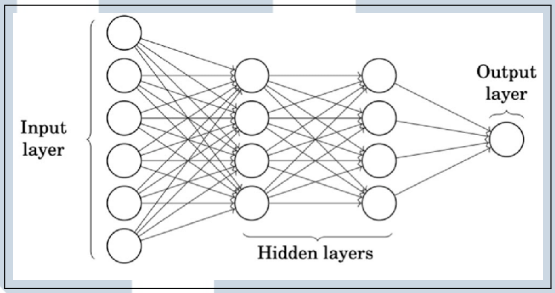
2.7 FastText

FastText merupakan jaringan syaraf tiruan dengan tiga komponen sebagai berikut, yaitu lapisan untuk *word embedding*, lapisan tersembunyi, dan lapisan keluaran[26].FastText digunakan dalam bentuk *library* bersifat *open-source* yang diciptakan oleh Facebook AI Research. *Library* ini dapat digunakan untuk *text representation* dan *text classification*[27]. Seperti Word2Vec, FastText juga digunakan untuk *word embedding*. Melalui FastText, kata yang menjadi input akan direpresentasikan dalam bentuk vektor dengan urutan penempatan sedemikian rupa supaya kata-kata yang bermakna sama akan muncul bersama, dan kata-kata yang bermakna berbeda tidak akan muncul bersama. FastText juga dapat digunakan untuk memperoleh vektor untuk kata-kata *out-of-vocabulary* (OOV)[28].

Terdapat dua jenis arsitektur dalam FastText, yaitu *Continuous Bag-of-Words* (CBOW) dan *Skip-gram*. Perbedaan mendasar dari kedua arsitektur ini adalah CBOW memprediksi probabilitas kata sebagai target yang diberikan konteks sebagai input, sedangkan Skip-gram adalah kebalikan dari CBOW, sehingga konteks dapat berupa satu kata atau kelompok kata[27].

2.8 Multilayer Perceptron

Multilayer Perceptron (MLP) adalah sebuah jaringan saraf tiruan turunan dari Perceptron yang terdiri dari JST umpan balik atau *feed-forward* dengan satu atau lebih *hidden layer* yang memetakan set data input ke output yang sesuai[29]. Biasanya, jaringan ini terdiri dari satu *layer* masukan, setidaknya satu *layer* neuron komputasi di tengah, dan sebuah *layer* komputasi keluaran. Sinyal masukan ditambatkan dengan arah maju per *layer*. Gambar 2.4 menunjukkan ilustrasi contoh arsitektur *Multilayer Perceptron*.



Gambar 2.4. Arsitektur MLP
Sumber:[30]

Pada Gambar 2.4, terdapat dua *layer* tersembunyi dengan empat neuron, serta satu *layer* keluaran dengan satu *neuron*. Setiap *layer* dalam *multilayer perceptron* mempunyai fungsi khusus. *Layer* masukan atau *input layer* berfungsi menerima sinyal masukan, *layer* keluaran atau *output layer* berfungsi menerima sinyal keluaran dari *layer* tersembunyi dan memunculkan sinyal/nilai/kelas keluaran dari seluruh jaringan.

ANN yang sering digunakan biasanya terdiri dari tiga atau empat *layer*, termasuk satu atau dua *layer* tersembunyi. Setiap *layer* bisa berisi 10 sampai 1.000 neuron, tetapi biasanya, dalam kebanyakan penerapannya, digunakan satu atau dua layer karena setiap penambahan satu *layer* akan meningkatkan beban komputasi secara eksponensial[31]. Berikut adalah langkah-langkah untuk *multilayer Perceptron*[32].

1. Inisialisasi semua bobot dengan bilangan acak kecil.
2. Jika kondisi untuk menghentikan *looping* belum dipenuhi, ulangi langkah 2-8.
3. Untuk setiap set data latih, lakukan langkah 3-8.

4. Tiap unit masukan menerima sinyal dan meneruskan ke unit tersembunyi di atasnya.

5. Hitung semua output unit tersembunyi z_j (unit tersembunyi ke-j) (j = 1,2,...,p)

$$z\_net\,j = v_{j_0} + \sum_{k=1}^{n} \tag{2.1}$$

$$z_i = f(z_{net_j}) = \frac{1}{1 + e^{(}- z\_net_j)} \tag{2.2}$$

Keterangan untuk rumus 2.1 dan 2.2:

z adalah keluaran di unit tersembunyi.

v adalah bobot dari *layer* masukan ke *layer* tersembunyi.

x adalah variabel masukan.

6. Hitung semua output jaringan di unit output y_k (k = 1, 2, ..., m)

$$y\_netk = w_{k_0} \sum_{j=1}^{p} z_j w_{kj} \tag{2.3}$$

$$y_k = f(y_n et_k) = \tag{2.4}$$

7. Hitung faktor δ unit keluaran berdasarkan keluaran di setiap unit keluaran y_k (k = 1,2,. . . ,m).

$$\delta_k = (t_k - y_k) f'(y_{net_k}) = (t_k - y_k) y_k (1 - y_k) \tag{2.5}$$

$$t_k = target$$

δ_k merupakan unit kesalahan yang akan dipakai dalam perubahan bobot *layer* di bawahnya. Hitung perubahan bobot w_{k_j} dengan laju pemahaman α

$$\Delta w_{kj} = \alpha \delta_k Z_j \tag{2.6}$$

$$(k = 1,2,...,m)(j = 0.1,...,p)$$

8. Hitung faktor δ unit tersembunyi berdasarkan kesalahan di setiap unit tersembunyi z_j (j = 1).

$$\delta net_j = \sum_{k=1}^{m} \delta_k w_{kj} \tag{2.7}$$

Faktor δ unit tersembunyi

$$\delta_j = \delta_{net_j} f'(Z_{net_j}) = \delta_{net_j} z_j (1 - z_j) \tag{2.8}$$

Hitung suku perubahan bobot v_{i_j}

$$\Delta v_{ji} = \alpha \delta_j Z_i \tag{2.9}$$

$$(j = 1, 2, ..., p; i = 1, 2, ..., n)$$

9. Hitung semua perubahan bobot. Perubahan bobot garis yang menuju ke unit keluaran yaitu:

$$w_k j(baru) = w_k j(lama) + \Delta w_k j \tag{2.10}$$

$$(k = 1, 2, ..., m; j = 0.1, ..., p)$$

Perubahan bobot garis yang menuju ke unit tersembunyi:

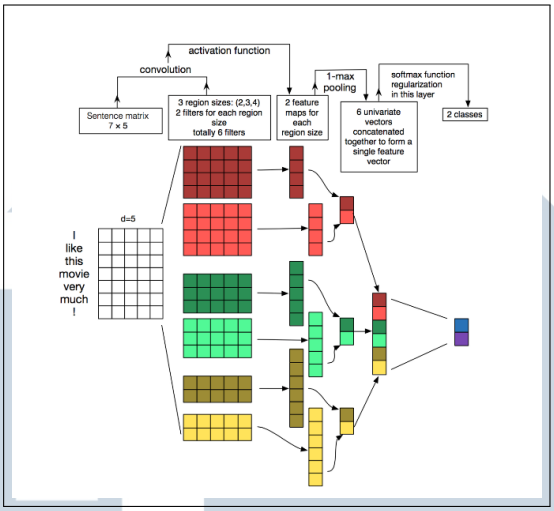
$$v_j i(baru) = v_j i(lama) + \Delta w_j i \tag{2.11}$$

$$(j = 1, 2, ..., p; i = 0.2, ..., n)$$

2.9 Convolutional Neural Network

Convolutional Neural Network adalah sekelompok jaringan saraf tiruan *feed-forward* dan menggunakan variasi dari *multilayer perceptron* yang dirancang supaya hanya memerlukan preprocessing minimum, CNN terinspirasi dari korteks visual hewan. CNN biasanya digunakan dalam computer vision, tapi baru-baru ini, juga digunakan untuk berbagai proses dalam *Natural Language processing* dan hasilnya menjanjikan[33, 34]. Perbedaan paling mendasar dari *convolutional neural network* dengan *artificial neural network* adalah, *convolutional neural network* dapat digunakan untuk mengenali gambar dan tulisan/teks. CNN diketahui sudah berhasil dalam masalah-masalah terkait teks, antara lain: pengenalan tulisan tangan, dan pengenalan objek visual[35].

Pada umumnya, *convolutional neural network* terdiri dari 3 jenis lapisan atau *layer* yaitu *convolutional layers*, *pooling layers*, dan *fully-connected layers*. Ketika ketiga lapisan tersebut ditumpuk, maka arsitektur CNN sudah terbentuk. CNN sudah memiliki berbagai macam arsitektur yang dikembangkan, diantaranya



Gambar 2.5. Arsitektur CNN
Sumber:[36]

adalah Inception, Residual Network (ResNet), LeNet, AlexNet, dan lain-lain. Arsitektur CNN untuk klasifikasi teks dapat dilihat di Gambar 2.5.

Berikut adalah fungsi-fungsi dasar dalam *Convolutional Neural Network*:

1. Seperti bentuk ANN lainnya, lapisan masukan atau input layer pasti ada. Dalam kasus ini, yang menjadi masukan adalah teks (*"I like this movie very much!"*).
2. *Input Layer* yang menerima *input* berupa teks.
3. *Convolutional layer* yang menentukan jumlah neuron yang terhubung dengan daerah local dari input. Di Gambar 2.5, ada 3 ukuran region dan 2 filter untuk setiap ukuran region.
4. *Pooling layer* yang terdiri dari 2 *feature maps* dan 6 vektor *univariate* yang ditempelkan untuk membentuk satu jenis vektor.
5. *Fully connected layer* yang dimaksud di sni adalah *multilayer perceptron*. *Fully-connected layer* memiliki beberapa *hidden layer*, *activation function*, *output layer*, dan *loss function* [37].

2.10 Evaluasi Model

Dalam melakukan evaluasi performa dari algoritma pembelajaran mesin, terutama dengan menggunakan teknik *supervised learning*, dapat digunakan

metode *confusion matrix* untuk memprediksi sebuah kategori [38]. Secara umum, *confusion matrix* dapat dilihat pada Tabel 2.1. Persamaan untuk melakukan perhitungan evaluasi performa dapat dilihat pada Persamaan 2.12 sampai 2.16.

Tabel 2.1. Tabel *confusion matrix*

	Positif (Aktual)	Negatif (Aktual)
Positif (Sistem)	True Positive	False Positive
Negatif (Sistem)	False Negative	True Negative

True Positive adalah perhitungan dari kelas positif sistem dan aktual yang hasilnya sama-sama positif. *True Negative* adalah perhitungan dari kelas negatif sistem dan aktual yang negatif. *False Negative* adalah perhitungan dari kelas negatif sistem tetapi kelas aktual positif. *False Positive* adalah perhitungan dari kelas positif sistem tetapi kelas aktual negatif.

Selain mengetahui nilai-nilai tersebut pada *confusion matrix*, dapat dilakukan pula perhitungan matematis untuk lebih memahami *confusion matrix*. Perhitungan yang dibuat dapat digunakan untuk mengevaluasi algoritma klasifikasi yang sudah dibuat bisa dilihat pada rumus 2.12 sampai rumus 2.16[39].

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \tag{2.12}$$

$$Precision = \frac{TP}{TP+FP} \tag{2.13}$$

$$Sensitivitas/Recall = \frac{TP}{FP+FN} \tag{2.14}$$

$$Spesifitas = \frac{TN}{TN+FP} \tag{2.15}$$

$$F1\ Score = \frac{2*(Recall*Precision)}{Recall+Precision} \tag{2.16}$$

Accuracy merupakan ketepatan klasifikasi (bernilai *true*). *Precision* merupakan kemampuan pengujian untuk mengidentifikasi kelas positif yang ada di prediksi dengan tepat yang benar. *Recall* atau sensitivitas merupakan kemampuan pengujian untuk mengidentifikasi nilai positif dari sejumlah data yang memang positif. Spesifitas adalah kemampuan pengujian untuk mengidentifikasi hasil

negatif dari sejumlah data yang benar-benar negatif hasilnya[31]. *F1 Score*, atau disebut juga *F-score*, adalah ukuran akurasi dari suatu model dalam sebuah dataset yang merupakan kombinasi dari *precision* dan *recall* dari model pembelajaran[40].