BAB II
METODE PENELITIAN

2.1 Objek penelitian

Objek pada penelitian ini adalah data sosial media *twitter* yang di dapatkan dari proses (*crawling*), dan peneliti melakukan beberapa tahapan dalam melakukan penelitian meliputi, peristiwa halving sebagai topik sentimen yang dianalisis, analisis sentimen sebagai metode klasifikasi sentimen yang akan diaplikasikan, serta pembobotan fitur yang diterapkan menggunakan TF-IDF, dan *Naïve Bayes Classifier* sebagai algoritma klasifikasi yang akan diimplementasikan untuk melakukan klasifikasi sentimen terhadap data *tweet* terkait peristiwa bitcoin halving.

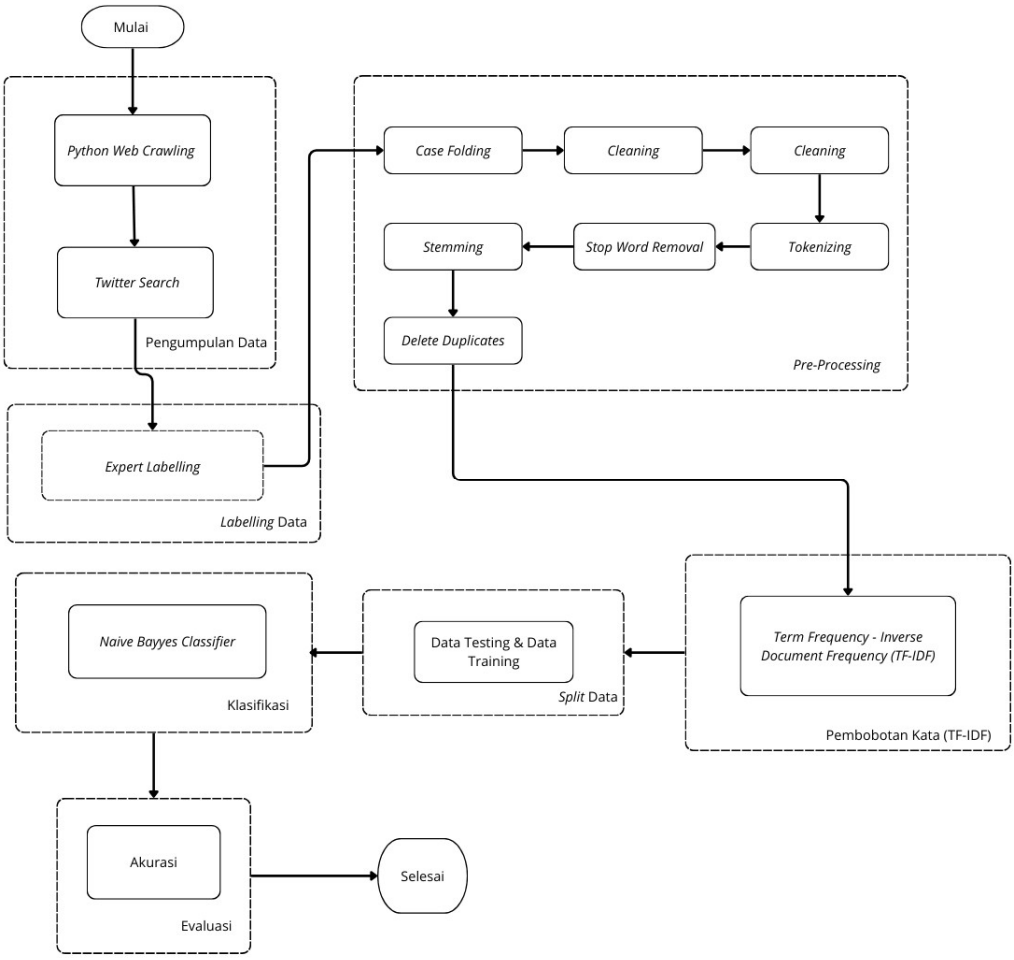
2.2 Alat dan bahan

Alat yang digunakan dalam penelitian ini meliputi perangkat keras dan perangkat lunak. Perangkat keras yang digunakan peneliti adalah laptop dengan spesifikasi AMD Ryzen 5 Series 5000H, RAM 16GB, dan penyimpanan SSD 512GB. Perangkat lunak yang digunakan mencakup *Google Collaboratory* versi 1.0.0 yang dapat diakses di https://colab.research.google.com/, Python versi 3.8.10, Node.js versi 14.16.0 digunakan untuk menjalankan *Tweet Harvest* versi 2.6.0 yang fungsinya *collect* data berdasarkan kata kunci yang di pakai. yang digunakan dalam penelitian ini berupa *library Python* seperti API *Twitter*, (i) *Pandas* versi 2.0.3 untuk manipulasi dan analisis data, (ii) *Numpy* versi 1.25.2 untuk operasi data numerik, (iii) *Re* versi 2.2.1 untuk operasi *Regex,* (iv) NLTK versi 3.8.1 untuk pemrosesan bahasa alami, (v) *Scikit-Learn* versi 1.2.2 untuk machine learning dan analisis data, (vi) *Matplotlib* versi 3.7.1 untuk visualisasi data, (vii) *Sastrawi* versi 1.0.1 untuk stemming bahasa indonesia.

Bahan yang digunakan dalam penelitian ini adalah dataset *tweet* terkait peristiwa bitcoin halving di sosial media *twitter*. Data tersebut menjadi sumber utama dalam untuk analisis percakapan dan opini publik mengenai peristiwa bitcoin halving di *twitter*.

2.3 Prosedur Penelitian

Penelitian dimulai dengan pengumpulan data dari *twitter* menggunakan *python*. Selanjutnya, proses labeling dilakukan oleh ahli bahasa (*expert*) untuk memberikan label sentimen pada data, apakah positif atau negatif. Setelah itu pra-pemrosesan data meliputi *case folding*, *cleansing*, *tokenizing*, *stopwords removal* dan *stemming.* Ekstraksi fitur dilakukan dengan metode TF-IDF. Data dibagi menjadi data latih dan data uji. Data latih digunakan untuk melatih model *Naive Bayes* dalam mengklasifikasikan sentimen menjadi positif dan negatif. Terakhir, kinerja model dievaluasi menggunakan *Confusion Matrix* untuk menilai kualitas klasifikasi sentimen pada data teks *twitter* seperti yang ditampilkan dalam Gambar 2.1.



Gambar 2. 1 *Kerangka Penelitian*

2.3.1 Pengumpulan data

Pengumpulan data dilakukan melalui *crawling* data di platform media sosial *Twitter* menggunakan *library Tweet-Harvest* yang dikembangkan dengan *Node.js* dan dapat diakses dengan bahasa pemrograman *Python*. *Library Tweet-Harvest* digunakan sebagai alat untuk mengumpulkan dan menganalisis informasi dari *Twitter* dengan kata kunci "*Bitcoin Halving lang:id*". *Tweet-Harvest* adalah *tools* untuk mengumpulkan data dari *Twitter* dengan memanfaatkan *auth_token Twitter*(Yuniarossy *et al*., 2024). Dengan menggunakan *library Tweet-Harvest*, peneliti dapat mengekstrak *tweet* yang relevan dengan topik tertentu terkait

konteks penelitian. Berdasarkan lampiran (2.1) Tahapan-tahapan pengumpulan data yang dilakukan dalam penelitian adalah sebagai berikut:

a. Tahap pertama adalah token autentikasi dari twitter, token autentikasi tersebut disimpan di variabel *twitter_auth_token*. Token tersebut digunakan untuk melakukan akses API Twitter untuk mengambil data tweet yang diperlukan dalam penelitian.
b. Tahap kedua adalah instalasi *library pandas* dan *Node*.js. Library *pandas* digunakan untuk manipulasi dan analisis data yang telah dikumpulkan. dan *Node.js* digunakan untuk menjalankan *tweet-harvest* dan mengakses API *Twitter.*
c. Pada tahap ketiga, peneliti menentukan nama *file* dan format penyimpanan data yang akan digunakan untuk menyimpan *tweet* yang dikumpulkan. Data *tweet* yang telah di *crawling* disimpan dalam format CSV untuk memudahkan proses analisis selanjutnya.
d. Pada tahap keempat, yaitu tahap *read* dan visualisasi data yang telah di *crawling.* Data yang telah dikumpulkan dalam format CSV dibaca ke dalam dataframe menggunakan *library pandas*. Kemudian *dataframe* tersebut ditampilkan untuk memeriksa data yang berhasil diambil.
e. Tahap terakhir, peneliti memeriksa jumlah dataset yang ada di dalam *dataframe*. Proses ini dilakukan dengan menghitung panjang dataframe, langkah tersebut dilakukan untuk memastikan jumlah data yang telah dikumpulkan sesuai dengan kebutuhan penelitian.

2.3.2 Labeling Data

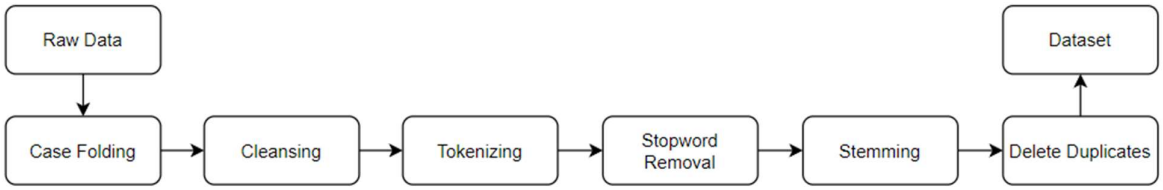
Pada proses klasifikasi teks pada dataset yang telah didapatkan sebagai bagian dari tugas akhir (skripsi), peneliti memerlukan bantuan dari seorang ahli bahasa yang memiliki pengalaman dalam pelabelan data. Oleh sebab itu peneliti mengirim permintaan melalui website *project.co.id* untuk mencari individu yang sesuai dengan kriteria tersebut. Dalam permintaan tersebut, peneliti menjelaskan bahwa peneliti mencari lulusan yang saat ini bekerja di bidang terkait seperti guru atau dosen, atau memiliki keahlian khusus dalam bahasa. Calon

ahli diminta untuk mengajukan penawaran yang mencantumkan pekerjaan saat ini, yaitu pengalaman relevan dalam pelabelan data, serta gelar akademik (Lampiran 1). Peneliti juga memberikan kesempatan bagi calon ahli untuk mengajukan pertanyaan lebih lanjut jika diperlukan. Pedoman untuk melakukan pelabelan oleh ahli bahasa dalam data ini adalah sebagai berikut:

a. Positif : Bitcoin merupakan aset masa depan, antusiasme dan optimisme pada peristiwa halving, serta harapan akan peningkatan nilai Bitcoin menjadi respons yang mendukung, motivasi untuk pertumbuhan, serta analisis positif tentang dampak halving menambah keyakinan dalam komunitas.
b. Negatif : penggunaan kata kata kasar dan kotor, skeptisme terhadap peristiwa halving, kekhawatiran terhadap volatilitas pasar yang tidak terduga, serta ketidakpastian mengenai stabilitas jangka panjang pada nilai bitcoin.

2.3.3 *Pre-Processing*

dataset yang sudah dilabeli, Selanjutnya adalah tahapan *Preprocessing*. Tahapan ini merupakan salah satu komponen utama yang penting di dalam *text mining* (S. Ramadhani *et al*., 2022). *Preprocessing* merupakan metode dalam penambangan data yang mengubah data asli menjadi format yang terstruktur dan lebih mudah dimengerti (Barus, 2022). Tahapan *preprocessing* mencakup enam langkah yaitu *case folding*, *cleansing*, *tokenizing*, *stopword removal*, *stemming*, dan penghapusan duplikat. Data yang dihasilkan dari proses crawling masih mengandung duplikat yang disebabkan oleh *tweet* yang diposting berulang kali, sehingga perlu dilakukan penghapusan data duplikat.(Fitriyah & Kartikasari, 2023) berdasarkan lampiran (2.4), (2.5) dan (2.6), tahapan preprocessing pada gambar 2.2 sebagai berikut :



Gambar 2. 2 *Tahapan Preprocessing*

a. *Case Folding*

Case folding merupakan proses mengonversi semua teks menjadi huruf kecil (*lowercase*) untuk menghilangkan variasi bentuk huruf besar dan kecil dalam analisis teks. Fungsinya adalah menyeragamkan representasi teks agar lebih konsisten dan mengurangi dimensi fitur yang perlu diproses saat analisis teks.

b. *Cleansing*

tahap pembersihan teks yang melibatkan eliminasi elemen-elemen yang tidak relevan atau tidak penting dalam teks, seperti tanda baca, karakter khusus, atau karakter yang tidak diinginkan lainnya. Hal ini dilakukan untuk membersihkan teks dari *noise* atau gangguan yang tidak diperlukan.

c. *Tokenizing*

Setelah cleansing tahap selanjutnya adalah *tokenizing*, *tokenizing* merupakan proses pembagian teks menjadi bagian-bagian yang lebih kecil, yang disebut *token*. *Token* dapat berupa kata-kata atau karakter, Langkah ini memungkinkan teks untuk dipecah menjadi unit-unit yang lebih kecil, yang nantinya akan diolah dalam analisis teks atau pemodelan.

d. *Stopword Removal*

Stopword removal adalah tahap dalam pemrosesan teks yang melibatkan penghapusan kata-kata penghenti dari teks. Kata-kata penghenti merupakan kata-kata yang sangat umum dalam suatu bahasa dan cenderung tidak memberikan banyak informasi penting dalam analisis teks. Dengan menghapus kata-kata penghenti, peneliti dapat fokus pada kata-kata kunci yang lebih bermakna dalam teks untuk analisis lebih lanjut.

e. *Stemming*

Selanjutnya adalah *stemming, Stemming* merupakan langkah dalam pemrosesan teks yang melibatkan pemangkasan akhiran atau awalan kata untuk menghasilkan akar kata atau bentuk dasarnya. Fungsinya adalah untuk mengurangi variasi kata yang mungkin muncul dalam teks yang sama, sehingga kata-kata yang memiliki akar yang sama akan dianggap sama dalam analisis teks. Hal ini membantu meningkatkan konsistensi dan efektivitas dalam pemrosesan teks serta mengurangi kompleksitas dalam pengolahan data.

f. *Delete Duplicate*

Tahapan terakhir adalah menghapus data duplikat (delete duplicate) adalah langkah penting dalam proses pembersihan data untuk memastikan kualitas dan keakuratan dataset. Proses ini penting dalam analisis karena duplikasi data dapat mengganggu hasil analisis dan menyebabkan bias dalam interpretasi.

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

Selanjutnya, dilakukan pembobotan tiap kata menggunakan TF-IDF (*term frequency, inverse document frequency)*. TF-IDF Merupakan proses perhitungan atau pengekstrakan kata menjadi sebuah angka berbentuk vektor yang digunakan untuk menentukan bobot dari sebuah kata dalam sebuah dokumen atau korpus. Bobot ini berguna untuk menentukan seberapa penting kata tersebut dalam sebuah dokumen(Tri Putra *et al*., 2023). Pada proses pembobotan kata, terdapat beberapa langkah yang dilakukan yaitu mencari *term frequency, inverse document frequency* serta *term frequency – inverse document frequency*(S. Ramadhani *et al.,* 2022). Proses pembobotan kata melibatkan penggunaan *TfidfVectorizer* dari *scikit-learn* untuk mengubah teks menjadi representasi numerik, di mana representasi ini bergantung pada bobot kata-kata yang dihitung menggunakan TF-IDF. (Br Sinulingga & Sitorus, 2024). Tahapan pembobotan kata TF-IDF yang dilakukan pada penelitian ini pada (lampiran 2.10) adalah sebagai berikut :

a. Tahap pertama peneliti melakukan import library Pandas untuk membaca dan mengelola data, NumPy untuk operasi numerik, dan *TfidfVectorizer* dari *scikit-learn* untuk mengubah teks menjadi representasi TF-IDF.
b. Tahap kedua adalah pembacaan data, di mana peneliti menggunakan *Pandas* untuk membaca dataset *'dataset_uji.csv'* ke dalam *dataframe*. Selanjutnya, dari *dataframe* tersebut, peneliti mengambil kolom *'cleaned_text*' sebagai daftar dokumen yang telah melalui proses *'stemming'*, dan kolom *'Sentimen'* sebagai daftar sentimen.
c. Pada tahap ketiga, persiapan dan penerapan TF-IDF. peneliti membuat sebuah *instance TfidfVectorizer* dan mengaplikasikannya untuk mengkonversi dokumen menjadi matriks TF-IDF. Proses ini mentransformasikan teks ke dalam representasi numerik yang mencerminkan relevansi setiap kata dalam dokumen.
d. Tahap terakhir adalah print output dari metode TF-IDF. Dan langkah berikutnya adalah menampilkan array position dari dokumen dataset, dan output numerik TF-IDF untuk setiap term.

   Berikut cara penghitungan TF-IDF dapat menggunakan rumus persamaan (2.1) di bawah ini:

$$tf_{t.d} = \frac{Jumlah\ kemunculan\ kata\ t\ dalam\ dokumen\ d}{Jumlah\ total\ kata\ dalam\ dokumen\ d} \quad (2.1)$$

Term Frequency (TF) mengukur seberapa sering sebuah kata muncul dalam sebuah dokumen. Dan Terkait rumus IDF pada persamaan (2.2) yaitu:

$$idf_d = log\frac{N}{n_t} \quad (2.2)$$

Inverse Document Frequency (IDF) mengukur seberapa penting sebuah kata dengan memperhitungkan seberapa sering kata tersebut muncul dalam semua dokumen. N

melambangkan jumlah total dokumen dalam kumpulan teks. Terkait rumus TF-IDF score pada persamaan (2.3) yaitu:

$$tfidf_{t.d} = tf_{t.d} \text{ X } idf_d \quad (2.3)$$

Keterangan:

t = Kata kunci, term

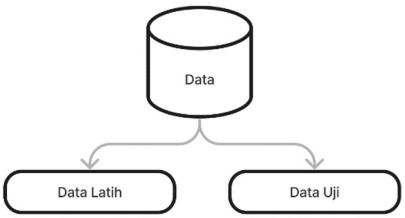
d = Dokumen

t.d = nilai TF-IDF untuk kata t dalam dokumen d

Tf = Banyaknya t (kata) yang dicari dalam dokumen

Idf = Banyak t kebalikan dari kata yang dicari

2.3.5 Split Data



Gambar 2. 3 *Split Data*

Split data adalah proses membagi dataset yang digunakan dalam penelitian menjadi dua atau lebih bagian, gambar 2.3 merupakan ilustrasi proses *split data* yang biasanya digunakan untuk menguji model atau algoritma. Dataset tersebut umumnya dibagi menjadi data latih (*training*) dan data uji (*testing*). Data latih digunakan untuk melatih algoritma, sedangkan data uji digunakan untuk mengevaluasi kinerja algoritma tersebut (Putri *et al*., 2023). Oleh karena itu pada penelitian ini data dibagi menjadi dua tahap, split data pada penelitian ini menggunakan rasio 90:10,80:20,70:30, Tahapan split data dan dilampirkan pada lampiran (2.11) sebagai berikut :

a. Langkah pertama adalah mengimpor fungsi train_test_split dari *sklearn.model* selection yang digunakan untuk membagi data.

b. Langkah kedua adalah membagi data dari term_frequency_all (fitur TF-IDF) dan kolom label df_preprocessed['Sentimen'] dari DataFrame dengan proporsi data uji sebesar 10%,20%,30% dan sisanya sebagai data latih.
c. Langkah ketiga adalah menampilkan jumlah data yang dihasilkan dari proses pembagian data tersebut.

2.3.6 *Naïve Bayes Classifier*

Naïve Bayes Classifier adalah metode klasifikasi yang didasarkan pada teorema Bayes. Metode ini menggunakan pendekatan probabilitas dan statistik yang dikembangkan oleh ilmuwan Inggris, Thomas Bayes untuk memprediksi peluang di masa depan berdasarkan data historis (Asmara *et al*., 2020). Metode ini mengasumsikan bahwa setiap fitur (kata) bersifat independen atau tidak saling bergantung satu sama lain. Naive Bayes merupakan model klasifikasi probabilistik dan statistik yang sederhana yang mengklasifikasikan data berdasarkan probabilitas tertinggi dari suatu kelas dengan mempertimbangkan nilai-nilai fitur yang diberikan(Fikri *et al*., 2020). Metode ini memiliki keunggulan karena memerlukan jumlah data latih yang relatif sedikit untuk menentukan parameter yang diperlukan dalam proses klasifikasi (Berliani & Lestari, 2024). Berdasarkan lampiran (2.11) tahapan klasifikasi sentimen menggunakan Naïve Bayes sebagai berikut :

a. Tahap pertama import library '*TfIdfVectorizer*' untuk mengubah teks menjadi representasi numerik berbasis TF-IDF. Dan '*Multinomial'* untuk inisiasi model *Naïve Bayyes.*

   Serta '*accuracy_score','classification_report'* dari '*sklearn.metrics'* untuk evaluasi performa model.
b. Tahap kedua peneliti menginisialisasi '*TfIdfVectorizer'* dengan parameter khusus, yaitu '*max_features=1000'* untuk menentukan jumlah fitur maksimum yang akan dipertahankan.

Dan '*min_df=5'* untuk mengabaikan kata-kata yang muncul kurang dari lima kali dalam dokumen, dan '*max_df=0.7'* untuk mengabaikan kata-kata yang muncul lebih dari 70% dokumen. Kemudian peneliti mengubah teks dalam data latih ('*x_train'*) dan data uji *('x_test'*) menjadi matrix TF-IDF menggunakan '*fit_transform'* dan '*transform*'.

c. Tahap ketiga peneliti menginisiasi model Naïve Bayyes dengan parameter '*alpha=0.1'* yang menentukan tingkat regularisasi *smoothing Laplace*.

d. Tahap keempat, peneliti melatih model Naïve Bayes menggunakan data latih '*x_train_tfidf'* dan label '*y_train'*.

e. Tahap kelima, melakukan prediksi pada data uji, peneliti menggunakan model yang telah di latih untuk melakukan prediksi pada data uji '*X_test_tfidf'*.

f. Tahap terakhir, peneliti melakukan evaluasi performa model dengan, menghitung akurasi menggunakan '*accuracy_score'* dan menampilkan laporan klasifikasi menggunakan '*classification_report'* yang memberikan matrix evaluasi seperti, presisi, recall dan f1-score.

Secara garis besar model naïve bayes adalah (2.4) sebagai berikut:

$$P(X/Y) = \frac{P(X \cap Y)}{P(Y)} \tag{2.4}$$

Dimana :

P(X/Y) = persentase X dalam Y

P(X ∩ Y) = Kata tertentu dalam kelas (tweet)

P(Y) = Jumlah kemunculan kata berlabel tertentu (Positif, Negatif)

Evaluasi pada penelitian ini menggunakan *Confusion matrix* sebagai alat untuk mengukur tingkat akurasi. *Confusion matrix* merupakan representasi visual yang *powerful* dan sangat berguna untuk memperkirakan performa model dengan menghitung jumlah prediksi yang benar dan salah, meliputi *True Positive* (TP) yaitu jumlah data positif yang di prediksi

dengan akurat, *False Positive* (FP) yaitu jumlah data negatif yang keliru diprediksi sebagai positif , *False Negative* (FN) yaitu jumlah data positif yang keliru diprediksi sebagai negatif, dan *True Negative* (TN) yaitu jumlah data negatif yang di prediksi dengan tepat sebagai negatif (Noor Hasan, 2024). Dengan menganalisis *Confusion matrix* secara cermat, peneliti memperoleh informasi berharga tentang kekuatan dan kelemahan model dalam mengklasifikasikan sentimen dengan tepat. Visualisasi Confusion Matrix ditampilkan berdasarkan lampiran (2.12).

Tabel 2. 1 *Confusion Matrix*

		Predicted Class	
		Positive	Negative
Actual Class	Positive	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

2.3.7 Evaluasi

Akurasi adalah salah satu matrix evaluasi yang paling umum digunakan dalam klasifikasi, Akurasi mengukur seberapa sering model klasifikasi memberikan prediksi yang benar dibandingkan dengan keseluruhan prediksi yang dibuat. Secara umum nilai akurasi mengindikasikan rasio data *tweet* yang terdeteksi dengan benar dalam dataset pengujian. Secara sederhana, akurasi mengukur seberapa dekat prediksi sistem dengan prediksi yang dibuat oleh manusia (Azhari *et al.,* 2021). Pada penelitian ini akurasi merupakan parameter penting dalam mengevaluasi performa sistem klasifikasi, yang memberikan gambaran tentang seberapa baik sistem dapat mengklasifikasikan data. akurasi dapat dihitung dengan menggunakan persamaan (2.5) berikut:

$$Akurasi = \frac{TP + TN}{TP + FP + FN + TN} \tag{2.5}$$