

BAB III METODE PENELITIAN

Pada bagian ini akan melakukan program *python* pada *google colab* menggunakan dataset sekunder yang terdapat pada kaggle yaitu *RiceLeafs datasets* untuk dijadikan data latih.

3.1 Alat dan Bahan

Pada penelitian ini membutuhkan beberapa alat dan bahan yang dapat dilihat dibawah ini :

1. *Laptop*, Perangkat keras yang digunakan pada penelitian ini menggunakan laptop HP dengan spesifikasi yang dimiliki berupa sistem operasi *Windows 11 Home Single Language*, prosesor yang digunakan *11th Gen Intel(R) Core(TM) i3-1115G @ 3.00GHz 2.90 GHz*. Penggunaan laptop pada penelitian ini berfungsi sebagai alat untuk membantu penulis dalam melakukan pemrograman.
2. Dataset, diperoleh dari <https://www.kaggle.com/datasets/vbookshelf/rice-leaf-diseases?resource=download> dengan nama dataset *RiceLeafs datasets* total 1260 sampel citra daun tanaman padi berpenyakit dengan ukuran dimensi *1600x1600 pixel*. Dengan rincian seperti tabel 3.1 Dataset ini memiliki tiga jenis kelas penyakit dengan jumlah sampel 420 sampel citra perkelas. Setiap kelas memiliki perbedaan pola dan jenis daun pada tanaman padi [23].

Tabel 3. 1 Pembagian Jenis Penyakit Daun tanaman Padi

No	Nama Kelas	Jumlah Data
1	<i>Bacterial leaf blight</i>	420
2	<i>Brown spot</i>	420
3	<i>Leaf smut</i>	420



Gambar 3. 1 a) leaf Scald, b) Narrow Brown Spot c) Brown spot [23]

Gambar 3.1 merupakan sampel dari penyakit yang terdapat didataset terdapat penyakit *brown spot*, *leaf scald* dan *narrow brown spot*

3. Library *Tensorflow*, dapat menggabungkan banyak model dan algoritma *machine learning* termasuk *deep learning (neural network)*.
4. *Google Colaboratory*, perangkat lunak ini digunakan untuk melakukan program *python* dan simulasi hasil data latih.

3.2 ALUR PENELITIAN

Penelitian ini dilakukan dalam beberapa tahapan sebagai berikut :

3.2.1 Studi Literatur

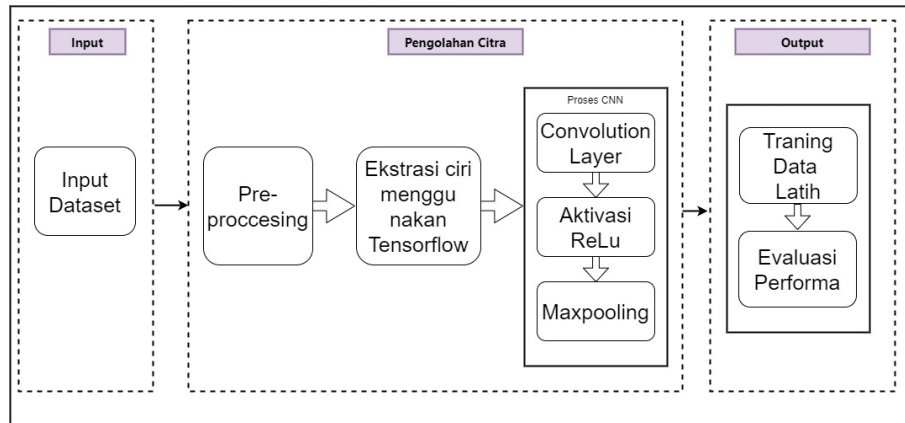
Pada penelitian “Identifikasi Penyakit Tanaman Padi Menggunakan *Convolutional Neural Network (CNN)*” hal pertama yang dilakukan adalah mencari referensi – referensi sebanyak mungkin yang berkaitan dengan topik penelitian yang dilakukan dan dapat membantu proses pembuatan perancangan baik dari internet maupun dari buku.

3.2.2 Identifikasi Masalah

Dalam penelitian “Identifikasi Penyakit Tanaman Padi Menggunakan *Convolutional Neural Network (CNN)*” Pada tumbuhan padi terdapat beberapa penyakit. Jenis penyakit yang ada biasanya juga memiliki banyak kemiripan sehingga keterbatasan visual petani dalam mengidentifikasi penyakit tanaman padi jadi salah memprediksi penyakit yang ada.

3.2.3 Perancangan Sistem

Untuk tahapan proses pembuatan sistem klasifikasi CNN yang terdapat pada penelitian tugas akhir ini sebagai berikut :



Gambar 3. 2 Blog Diagram system

1. Input dataset seperti terlihat pada gambar 3.2, merupakan tahapan utama untuk memasukkan gambar yang telah didapat pada kaggle.
2. *Pre-Processing*, merupakan proses untuk menyeragamkan unsur citra dan *resize* menjadi lebih kecil dengan ketentuan ukuran dimensi 224 x 224 *pixel*.
3. Ekstrasi fitur, *Tensorflow* dapat melatih dan menjalankan *neural network* untuk keperluan mengklasifikasikan.
4. *Conventional Layer*, Pada *conventional layer* terdapat operasi 7 konvolusi antara citra input dan filter.
5. Aktivasi *ReLU*, merupakan tahapan untuk melakukan *thresholding* dengan nilai nol terhadap nilai piksel pada input citra.
6. *Maxpooling*, Nilai yang diambil adalah 2x2.
7. *Traning Data Latih*, pada tahapan data latih dilakukan proses pembelajaran pada citra yang kemudian hasil keluarannya berupa model yang akan disimpan untuk digunakan dalam proses pengujian.
8. *Traning Data Uji*, Citra yang telah diambil akan diproses oleh algoritma CNN yang digunakan dengan epochs yang diuji yaitu nilai 40 iterasi.

3.2.4 Analisis Data

Parameter yang diinisiasi pada penelitian ini antara lain hasil *training* dan *testing* yang diperoleh dari jumlah *epoch*, optimasi, dan *loss function* yang didapatkan. Kemudian penggunaan beberapa jumlah *epoch* dan nilai *Overfitting*

pada setiap pembelajaran model untuk mendapatkan hasil yang ideal. Jumlah *epoch* yang digunakan yaitu 40 iterasi. *Epoch* merupakan proses ketika seluruh dataset sudah melalui proses training atau pelatihan dalam *neural network* sampai proses itu kembali ke awal. Sedangkan nilai *Dropout* yang digunakan yaitu 0,05. *Dropout* ini digunakan untuk mencegah *Overfitting*. *Overfitting* terjadi ketika data yang digunakan untuk pelatihan mencapai hasil terbaik, tetapi jika diuji pada data yang berbeda, akurasi tidak optimal. *batch_size* merupakan jumlah contoh pelatihan dalam satu *forward/backward pass*. Semakin tinggi nilai *batch_size* maka akan semakin banyak memori yang dibutuhkan. Model CNN terbaik didapatkan apabila telah melakukan pelatihan atau training dengan percobaan menggunakan beberapa parameter yang telah ditentukan.

3.2 PRE-PROCESSING

Dengan dataset yang berjumlah 1260 sampel citra yang terbagi dari tiga kelas data penyakit (*Brown spot, leaf Scald* dan *Narrow Brown Spot*). Pada proses pemberian label peneliti membagi data yang telah *download* dari *kaggle*. Pembagian ini dilakukan untuk memisahkan data yang akan ditraining dan data untuk *testing*. Data yang digunakan untuk mentraining data harus memiliki jumlah yang lebih banyak dari jumlah validasi agar hasil pelatihan mendapat akurasi yang baik. Folder training berisikan 3 *subfolder* yang masing – masing berisikan penyakit daun padi *Brown spot, leaf Scald, Narrow Brown Spot* dengan jumlah 350 citra untuk 1 kelas penyakit daun padi. Folder kedua yaitu validasi berisikan 3 *subfolder* yang masing – masing berisikan penyakit daun padi *Brown spot, leaf Scald, Narrow Brown Spot* dengan jumlah 70 citra untuk 1 kelas penyakit daun padi.

Untuk menyeragamkan unsur citra yang akan dilakukan analisis karena unsur dari masing-masing citra yang sangat bervariasi dan untuk memperjelas fitur dari citra. Selanjutnya akan di-*resize* menjadi lebih kecil dengan ketentuan ukuran dimensi 224 x 224 *pixel* seperti terlihat pada gambar 3.3. setelah gambar di-*resize* akan disimpan pada variabel X yang nantinya akan dilakukan pemanggilan pada saat training dan testing.



Gambar 3. 3 flowchart Proses Pre-Processing

3.3 EKSTRAKSI CIRI

Untuk mendapatkan ciri dari sebuah citra daun padi yang telah diambil, proses tersebut bertujuan melakukan pengambilan data dan informasi yang terdapat pada citra. Dengan menggunakan Tensorflow dapat menggabungkan banyak model dan algoritma *machine learning* termasuk *deep learning (neural network)*. Metode untuk bekerja dengan *bounding box* dan untuk menyesuaikan kontras, kecerahan. *Tensorflow* dapat melatih dan menjalankan *neural network* untuk keperluan

mengklasifikasikan pengenalan object. *Tensorflow* menyediakan bagaimana data yang digunakan bergerak melalui tampilan grafis atau serangkaian pemrosesan node. Setiap node di dalam grafik mewakili operasi matematika, dan setiap koneksi atau ujung antara node tersebut merupakan baris data multidimensi (*multidimensional data array*) atau yang disebut juga tensor.

Ekstraksi ciri dilakukan dengan memasukkan data gambar yang telah dilakukan pre-processing. Membangun model *Neural Network* untuk tugas pengenalan objek dengan membangun 7 layer konvolusi, *maxpooling*, *fatten* dan *fully connected layers*. Dalam lapisan Conv2D dalam jaringan saraf konvolusi (CNN) melakukan operasi konvolusi 2D pada data masukan. Operasi ini melibatkan pergeseran sekumpulan filter yang dapat dipelajari di atas data masukan, menghitung hasil perkalian titik antara filter dan jepitan masukan pada setiap posisi. Proses ini membantu jaringan untuk mempelajari pola lokal dan fitur dari data masukan. Rumus untuk menghitung bentuk keluaran dari lapisan Conv2D :

$$Output\ Height = \frac{(Input\ Height - Filter\ Height + 2 \times Padding)}{Stride} + 1 [22]$$

$$Output\ Width = \frac{(Input\ width - Filter\ width + 2 \times Padding)}{Stride} + 1 [22]$$

Dalam penelitian ini menggunakan Conv2D (224, 3, *padding='same'*, *activation='relu'*), maka perhitungan :

$$Output\ Height = \frac{(224 - 3 + 2 \times 1)}{1} + 1$$

$$= 224$$

$$Output\ Width = \frac{(224 - 3 + 2 \times 1)}{1} + 1$$

$$= 224$$

Jadi, bentuk keluaran dari lapisan Conv2D akan menjadi (224,224, Jumlah filter). Di lapisan *MaxPooling* beroperasi dengan memilih nilai maksimum dari setiap area dalam filter yang digeser ke sepanjang gambar. Ini membantu untuk mengurangi jumlah parameter dalam model dan berfungsi untuk mengurangi dimensi data dan mempertahankan fitur yang paling menonjol dari area yang diperiksa dalam gambar. Rumus untuk menghitung bentuk keluaran dari lapisan *MaxPooling2D* :

$$Output\ Height = \frac{(Input\ Height - Pool\ Size)}{Stride} + 1 [22]$$

$$Output\ Width = \frac{(Input\ width - Pool\ Size)}{Stride} + 1 [22]$$

Dalam penelitian ini menggunakan *MaxPooling2D(2,2)*, maka perhitungan :

$$Output\ Height = \frac{(224 - 2)}{2} + 1$$

$$= 112$$

$$Output\ Width = \frac{(224-2)}{2} + 1$$

$$= 112$$

Jadi, bentuk keluaran dari lapisan *MaxPooling* dalam penelitian ini akan menjadi (112, 112, Jumlah Saluran). Lapisan *Flatten* dalam jaringan saraf konvolusi (CNN) digunakan untuk mengubah tensor multidimensi menjadi *vektor* satu dimensi. Ini diperlukan sebelum menghubungkan tensor ke lapisan *Dense* untuk pemrosesan lebih lanjut. Dalam penelitian ini inputan yang diperoleh dari *MaxPooling* merupakan tensor 2D yang akan diubah menjadi *vektor* 1D. Lapisan *Dense* atau *Fully Connected Layer* dalam jaringan saraf merupakan lapisan yang menerapkan transformasi *linier* ke data yang masuk. Dalam lapisan yang terhubung sepenuhnya, *neuron* menerapkan transformasi *linier* ke *vektor input* melalui matriks bobot. Transformasi *non-linear* kemudian diterapkan pada produk melalui fungsi aktivasi *non-linear f*.

$$y_{jk}(x) = f \left(\sum_{i=1}^{n_H} W_{jk} X_i + W_{j0} \right) [22]$$

Perkalian titik antara matriks bobot W dan *vektor* masukan x. Istilah bias (W0) dapat ditambahkan di dalam fungsi *non-linear*. Bobot mengatur kontribusi masing-masing fitur input terhadap *neuron* dalam lapisan yang lebih dalam, sementara bias memungkinkan model untuk menangkap pola yang mungkin tidak diwakili oleh fitur input saja. Fungsi aktivasi digunakan untuk memperkenalkan *non-linearitas*. Bobot dan bias digunakan untuk menghitung *output* dari setiap *neuron* dalam jaringan saraf tiruan. Selama pelatihan, bobot dan bias diperbarui secara iteratif berdasarkan perbandingan antara hasil prediksi model dan nilai yang diharapkan (y) menggunakan algoritma pembelajaran yang sesuai, bobot dan bias

disesuaikan menggunakan algoritma optimisasi. Untuk nilai bias ini adalah bagian dari proses pelatihan yang diatur oleh algoritma optimisasi. Sehingga hanya perlu memastikan bahwa model yang telah bangun memiliki bias di setiap lapisan dan bahwa pelatihan dilakukan dengan benar. Setiap *neuron* dalam lapisan *hidden layer* terhubung dengan setiap *neuron* dalam lapisan sebelumnya. Selanjutnya model yang telah dibangun akan dikompilasi dengan menentukan fungsi kerugian, *optimizer* dan melakukan pelatihan model. Setelah pelatihan selesai akan dilakukan evaluasi kinerja menggunakan data uji. Model yang telah dilatih digunakan untuk melakukan prediksi pada gambar baru.

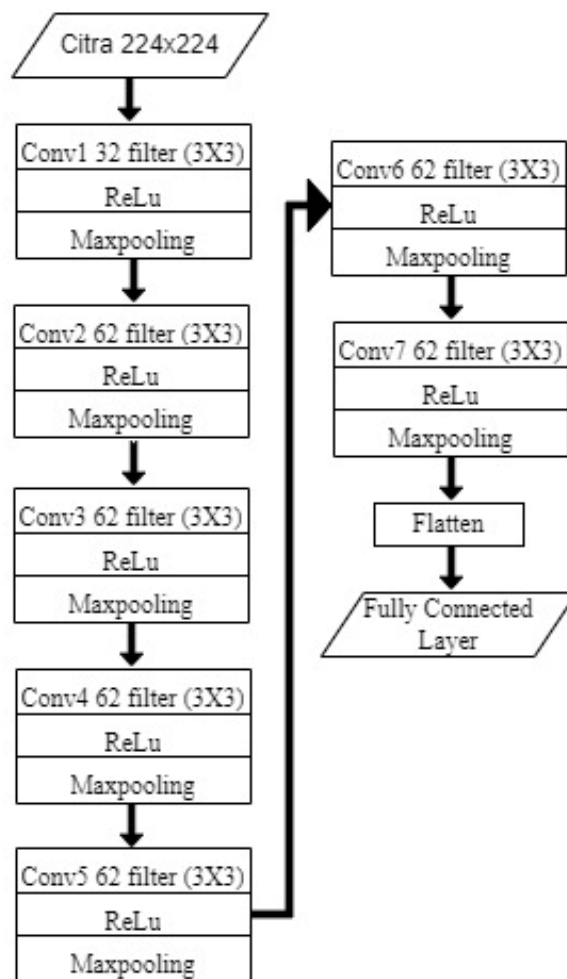
3.4 DATA LATIH

Pada tahapan data pelatihan, proses pembelajaran dilakukan pada gambar dan kemudian buatan model yang disimpan untuk digunakan dalam proses pengujian. Susunan model adalah proses melatih data citra latih untuk mengidentifikasi objek dan mengategorikannya sesuai dengan kelasnya. Pada penelitian ini metode yang digunakan merupakan salah satu cabang algoritma *deep learning* yaitu CNN dengan arsitektur 7 kovolusi layer. Data latih yang digunakan adalah 80% dari total keseluruhan data sehingga didapat data latih sebanyak 1050 gambar dengan masing-masing kelas sebanyak 350 sampel. Sampel citra yang terbagi dari tiga kelas data penyakit (*Brown spot*, *leaf Scald* dan *Narrow Brown Spot*). Komputasi dilakukan menggunakan mode single GPU.

Data *train* akan digunakan untuk melakukan proses pelatihan menggunakan algoritma CNN. Pada saat yang sama, proses pengujian dilakukan dengan menggunakan pengujian data, sehingga menghasilkan hasil prediksi. Proses *Modelling* CNN dimulai dengan langkah 1 yakni pendefinisian kelas, dilanjutkan dengan langkah kedua yakni pembuatan arsitektur model. Pembuatan arsitektur model menggunakan acuan pada metode CNN. Tipe lapisan pada CNN terbagi menjadi dua, lapisan pertama merupakan lapisan ekstraksi fitur terletak pada awal arsitektur yang tersusun atas beberapa lapisan dan disetiap lapisan tersusun atas *neuron* yang terkoneksi pada daerah lokal dari lapisan sebelumnya.

Seperti yang ditunjukkan pada gambar 3.4 terdapat arsitektur model 7 layer CNN yang dibangun. Layer pertama terdapat *Convolutional Layer* (Conv2D), Lapisan ini menerapkan operasi konvolusi pada data masukan menggunakan filter

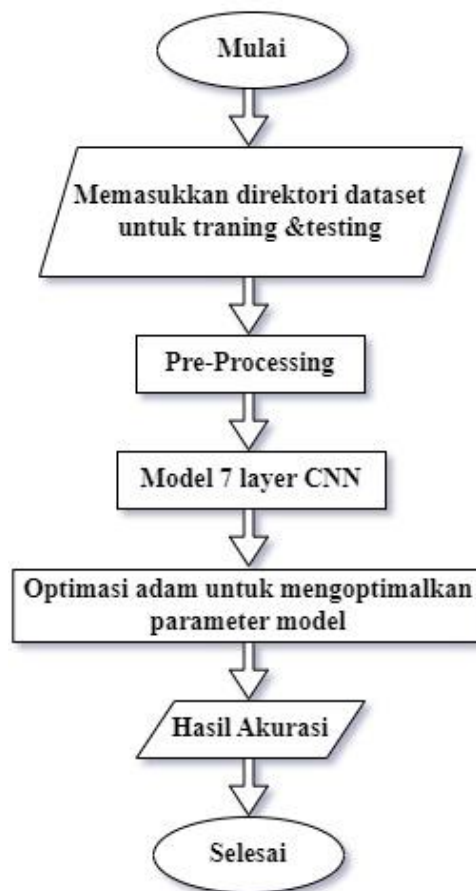
yang dapat dipelajari. Ini membantu dalam mengekstrak fitur-fitur penting dari gambar. Setelah lapisan konvolusi, fungsi aktivasi seperti ReLU (*Rectified Linear Unit*) biasanya diterapkan untuk memperkenalkan *non-linearitas* ke dalam model. *MaxPooling* atau *AveragePooling*, lapisan ini digunakan untuk mengurangi dimensi spasial dari data dengan mengambil nilai maksimum atau rata-rata dari area dalam filter yang bergerak di sepanjang gambar. Ini membantu dalam mengurangi jumlah parameter dan komputasi yang dibutuhkan oleh model. *Flatten Layer*, Lapisan ini mengubah tensor multidimensi menjadi vektor 1D, mempersiapkannya untuk lapisan *Dense* selanjutnya. *Dense Layer* (*Fully Connected Layer*), lapisan ini menghubungkan setiap *neuron* dengan setiap *neuron* dalam lapisan sebelumnya. Ini bertanggung jawab untuk transformasi *linier* dan memungkinkan model untuk mempelajari hubungan yang lebih kompleks antara fitur-fitur masukan dan output yang diinginkan.



Gambar 3. 4 Diagram tahapan data latih

3.5 DATA TESTING

Tahap testing melibatkan evaluasi kinerja model yang telah dilatih pada data uji yang belum pernah dilihat model sebelumnya selama pelatihan. Pada gambar 3.5 tujuan utama dari tes ini adalah untuk memahami seberapa baik model CNN menggeneralisasi dan secara akurat memprediksi data baru yang disimpan pada *directory*. Penelitian ini menggunakan skenario data test 210 data dengan masing-masing kelas sebanyak 70 sampel. Citra yang telah diambil akan diproses oleh algoritma CNN yang digunakan dengan jumlah *epochs* 40 iterasi. Dengan menggunakan 40 epoch akan meminimalkan komputasi yang berlangsung pada saat training data, semakin banyak epoch yang digunakan maka akan semakin banyak waktu yang dibutuhkan pada saat pelatihan. Kemudian data baru akan melakukan prediksi pada klasifikasi jenis penyakit daun padi. Selama pengujian, penting untuk memastikan bahwa evaluasi dilakukan secara objektif dan menggunakan kumpulan data pengujian yang representatif sehingga hasil pengujian dapat diandalkan.



Gambar 3. 5 Diagram tahapan data uji

3.6 EVALUASI

Hasil yang telah diperoleh pada penelitian ini di evaluasi untuk mengetahui performa model dalam mengidentifikasi penyakit tanaman padi melalui citra daun. Pada tahap evaluasi, performa model diukur dari nilai *metric score* yang diperoleh berdasarkan *confussion matrix*. Berdasarkan *precision*, *recall* dan *f1-score* yang diperoleh pada tahap evaluasi yang menggambarkan tingkat keberhasilan model dalam mengidentifikasi penyakit tanaman padi berdasarkan citra daun.

Tabel 3. 2 Confussion Matrix[24]

		AKTUAL		
		Brownspot	Leaf Scale	Narrow
P R E D I K S I	Brownspot	TP	FN	FN
	Leaf Scale	FP	TP	FN
	Narrow	FP	FP	TP

Seperti yang terlihat dalam tabel 3.2 *True Postive* (TP) merupakan kondisi dimana model memprediksi data ada di kelas Positif dan yang sebenarnya data memang ada di kelas Positif. Dalam memprediksi penyakit pada daun padi yang sebenarnya *brown spot* diprediksi *brown spot*. *False Positive* (FP) merupakan kondisi dimana model memprediksi data berada di kelas positif, tetapi sebenarnya data ada di kelas negatif. Dalam memprediksi penyakit pada daun padi yang sebenarnya *leaf Scald* diprediksi *brown spot*. Sementara *False Negative* (FN) merupakan kondisi dimana model memprediksi data ada di kelas negatif, namun yang sebenarnya data ada di kelas positif. Dalam memprediksi penyakit pada daun padi yang sebenarnya *Narrow Brown Spot* diprediksi *brown spot*. Selanjutnya mengukur seberapa persen akurasi, *precisson* dan *recall* dari sistem.

1. Accuracy

Accuracy merupakan *metric* yang digunakan untuk menghitung rasio prediksi benar (positif dan negatif) dengan keseluruhan data saat

melakukan pelatihan. *Metric accuracy* dirumuskan pada persamaan berikut :

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} [24]$$

2. *Precision*

Precision digunakan untuk menghitung presisi atau data prediksi positif dari keseluruhan hasil yang diprediksi positif. *Precision* dirumuskan pada persamaan berikut :

$$Precision = \frac{TP}{TP+FP} [24]$$

3. *Recall*

Recall digunakan untuk menghitung jumlah prediksi pada kelas yang sama. Secara definisi *recall* adalah perbandingan antara *True Positive* (TP) dengan banyaknya data yang sebenarnya positif. Rumus *Recall* dituliskan dalam persamaan berikut :

$$Recall = \frac{TP}{TP+FN} [24]$$

4. *F1-Score*

F1-Score merupakan *harmonic-mean* atau perbandingan rata-rata *precision* dan *recall* yang dibobotkan. Rentang nilai *F-1 Score* adalah 0 sampai 1 dengan nilai terburuk adalah 0 dan nilai terbaik adalah 1. *F1-Score* mengindikasikan bahwa model klasifikasi memiliki *precision* dan *recall* yang baik. dan nilai *F1-Score* dirumuskan pada persamaan berikut :

$$F1\ Score = 2 \times \frac{(Recall \times Precision)}{(Recall + Precision)} [24]$$