

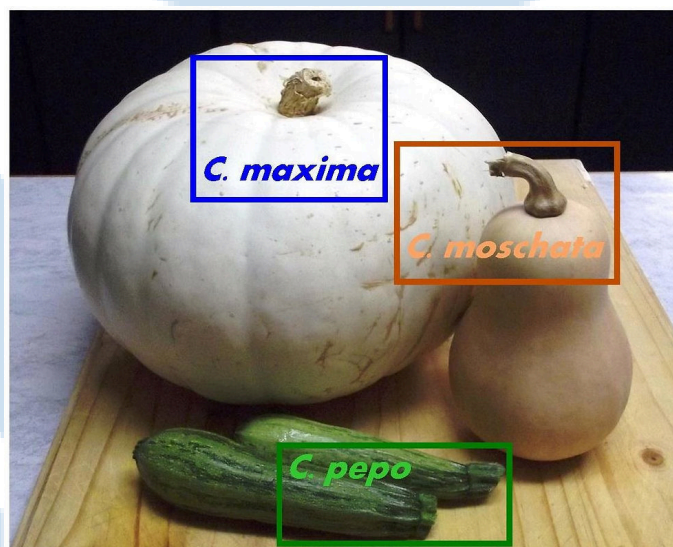
BAB 2

LANDASAN TEORI

2.1. Tanaman Labu.

Tanaman labu merupakan salah satu jenis tanaman dari keluarga *Cucurbitaceae* yang sering dibudidayakan karena mengandung banyak manfaat vitamin dan mineral bagi tubuh. Melalui laman Balai Pelatihan dan Pertanian Lampung[20], 100 gram labu mengandung 93,69 gram air, 20 kalori energi, 0,72 gram protein, 0,07 gram lemak, 4,9 gram karbohidrat, 1,1 gram serat, 15 mg kalsium, 0,57 mg zat besi, 9 mg magnesium, 30 mg fosfor, 230 mg kalium, 1 mg natrium, 0,23 mg seng, 4,7 mg vitamin C, 0,031 mg thiamin, 0,078 mg riboflavin, 0,413 mg niacin, 9 mg folat, 1,7 mg vitamin A.

Labu yang umum dikenal secara umum di Indonesia[20], diantaranya: Labu kuning (*Cucurbita pepo*), labu kabocha atau labu Jepang(*Cucurbita maxima*), dan labu madu atau *butternut squash* (*Cucurbita moschata*).



Gambar 2.1 Beberapa jenis buah dalam famili *Cucurbitaceae*

Sumber: [21]

Tanaman labu secara umum memiliki beberapa karakteristik yang

membedakannya dengan tanaman lainnya[22] sebagai berikut: Berbatang berongga yang lunak dan berair serta tidak berkayu keras (*non-woody* dan *hollow-stemmed*), berakar tunggang tunggal (*a singular taproot*) dengan banyak akar cabang yang tumbuh lurus ke bawah kemudian tumbuh mendatar ke samping, memiliki daun yang berlekuk dalam dengan bentuk hati atau menjari (*deeply lobed leaves*) dengan permukaannya yang kasar dan berbulu, memiliki bunga berumah satu dimana bunga jantan dan betina tumbuh pada tanaman yang sama.

Tanaman labu rentan terhadap kerusakan yang disebabkan beberapa faktor penyakit[2][3], diantaranya: penyakit karena jamur (*Fungal leaf spots*, *downy mildew*, *powdery mildew*, antraknosa, busuk daun, dan busuk akar), penyakit karena bakteri (*Bacterial leaf spots*) dan penyakit yang disebabkan oleh virus (*Mosaic disease*).

2.2. Penyakit Downy Mildew

Downy mildew atau embun bulu atau bulai merupakan penyakit pada tanaman yang disebabkan oleh patogen yang dari organisme *oomycota*. Mengutip dari laman Kementerian Pertanian Republik Indonesia, penyakit *downy mildew* banyak dijumpai pada area lahan yang lembab, terutama pada musim panas menjelang musim hujan, serta disebabkan oleh beberapa spesies jamur, diantaranya: *Peronosclerospora maydis*, *Peronosclerospora philippinensis*, dan *Peronosclerospora sorghi* yang dapat menurunkan produksi buah labu[23]. Jamur *P. cubensis* ketika menginfeksi daun akan memunculkan ciri-ciri sebagai berikut: memiliki miselium yang tidak bersekat, berwarna kecoklatan, memiliki bentuk bulat seperti telur, serta memiliki alat pengisap kecil dalam jumlah banyak yang berkelompok[23].



Gambar 2.2 Jamur *Pseudoperonospora cubensis*

Sumber: [24]

Patogen *Pseudoperonospora cubensis* membutuhkan inang hidup agar bisa berkembang biak, serta tidak dapat bertahan hidup di musim dingin, sehingga spora patogen *P. cubensis* akan dibawa melalui arus udara dari daerah yang lebih hangat dan lembab tergantung pola cuaca[25].

Downy mildew pada daun tanaman labu dapat diidentifikasi berdasarkan beberapa faktor[6], mulai dari munculnya bercak-bercak berwarna pucat, warna hijau dan kuning yang terbentuk pada permukaan atas daun yang kemudian akan berubah menjadi coklat, bercak daun yang membentuk sudut dan dibatasi tulang daun, buluh berwarna abu-abu keunguan gelap pada bagian bawah daun dalam kondisi kelembaban yang tinggi, daun yang tampak seperti mati akibat embun beku.

2.3. Machine Learning

Machine learning merupakan pengembangan sistem komputer dan algoritma matematika agar dapat mempelajari serangkaian data atau informasi sehingga dapat membuat prediksi selanjutnya melalui pola yang diajarkan[26]. *Machine learning* sendiri merupakan sub-kategori dari kecerdasan buatan yang memungkinkan komputer untuk mempelajari data tanpa diprogram secara eksplisit

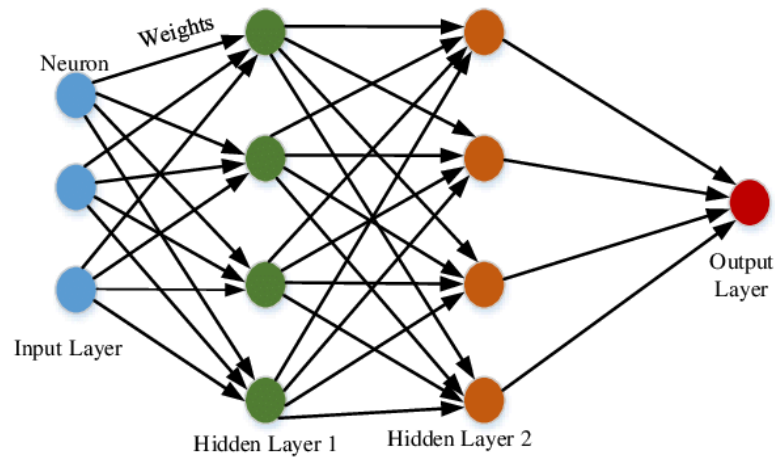
dengan mengidentifikasi pola dalam data besar, serta membuat prediksi atau keputusan berdasarkan pola tersebut[27].

Proses pembelajaran dari *machine learning* melibatkan pelatihan model pada dataset (sekumpulan data) sehingga model tersebut dapat menggeneralisasi pengetahuan ke data baru[28][29]. Machine learning berdasarkan jenis pembelajarannya dibagi menjadi 3 kategori, yaitu: *Supervised learning*, *unsupervised learning*, dan *reinforcement learning*[30].

Supervised learning adalah algoritma yang dilatih dengan data yang sudah diberi label kelas atau target seperti klasifikasi dan regresi, contoh algoritmanya adalah: Regresi Linear, *Logistic Regression*, *Decision Tree*, *Random Forest*, *Support Vector Machine (SVM)*, *Neural Network*[31]. *Unsupervised learning* merupakan algoritma yang dilatih untuk dapat menemukan pola tanpa adanya label seperti clustering dan reduksi dimensi, contoh algoritmanya adalah: *K-Means Clustering*, *Hierarchical Clustering*, *Principal Component Analysis*[32]. *Reinforcement learning* yaitu algoritma yang belajar melalui interaksi dengan lingkungan sekitarnya, dan akan mendapatkan reward maupun punishment berdasarkan tindakan yang dijalankan, contoh algoritmanya adalah: *Q-learning* dan *Deep Q-Network*.

2.4. Support Vector Machine

Penelitian deteksi penyakit pada daun tanaman labu merupakan bagian dari ranah keilmuan machine learning dengan memanfaatkan metode algoritma Support Vector Machine. Support Vector Machine (SVM) merupakan metode algoritma pembelajaran mesin jenis *supervised learning* (untuk melakukan klasifikasi dan regresi) dimana satu kelas memiliki metode jaringan syaraf tiruan yang dapat memetakan data ke ruang fitur berdimensi tinggi dan mencari *hyperplane* yang memisahkan data menjadi kelas yang berbeda dengan margin yang maksimal[33].



Gambar 2.3 Diagram Arsitektur SVM

Sumber: [34]

Hyperplane ini adalah batas pemisah linear atau nonlinear yang bertugas memisahkan titik-titik data dalam ruang fitur (*feature space*)[35]. SVM telah banyak digunakan dalam pengaplikasian saintifik, misalnya: klasifikasi citra satelit[36], prediksi kualitas udara[37], dan klasifikasi spesies[38]. SVM dapat digunakan untuk mengklasifikasi citra gambar daun yang sehat dan yang terinfeksi penyakit[39]. Dengan mengekstrak fitur-fitur yang relevan dari citra daun, seperti warna, tekstur, dan bentuk, SVM dapat melatih model untuk membedakan antara daun yang sehat dan daun yang terkena penyakit[39]. Proses klasifikasi dari data dalam SVM dilakukan dengan memperhatikan persamaan berikut:

$$f(\chi) = \text{sign} \left(\sum_{i=1}^n a_i y_i K(\chi_i^T \cdot \chi_j) + b \right) \quad (2.1)$$

Dimana:

χ_i = Data input x baris ke-i

χ_j = Data input x kolom ke-j

y_i = Kelas output baris ke-i

b = Nilai Bias

a_i = Nilai Alpha atau sebagai support vector

sign = Nilai yang besar dari 0 dilabelkan +1, Semua nilai yang kecil dari 0 dilabelkan -1.

Salah satu keunggulan utama dari SVM adalah mampu mengatasi data berdimensi tinggi dan kompleks, salah satunya dalam mengolah citra. Ini dimungkinkan melalui penggunaan fungsi kernel, yang memungkinkan SVM untuk memetakan data ke ruang berdimensi lebih tinggi agar dapat dipisahkan secara linear. Pemetaan fungsi kernel pada SVM dibedakan menjadi 3 jenis kernel yang umum digunakan[40], yaitu: Kernel Linear, Kernel *Polynomial* dan Kernel *Radial Basis Function* (RBF) atau Kernel *Gaussian*.

2.4.1. Kernel Linear

Merupakan kernel yang paling sederhana dalam melakukan pemetaan gambar. Kernel Linear dirumuskan sebagai berikut:

$$K(x_i, x_j) = x_i^T \cdot x_j \quad (2.2)$$

2.4.2. Kernel Polynomial

Kernel Polynomial merupakan kernel non-stasioner, bersifat lebih umum karena mewakili kesamaan vektor dalam sampel gambar, cocok digunakan untuk masalah dengan data latih yang telah dinormalisasi. Kernel Polynomial dirumuskan sebagai berikut:

$$K(\bar{x}_i, \bar{x}_j) = (\bar{x}_i \cdot \bar{x}_j + 1)^p \quad (2.3)$$

2.4.3. Kernel Radial Basis Function (RBF) atau Kernel Gaussian

Kernel RBF merupakan kernel utama karena merupakan kernel nonlinear yang dapat memetakan sampel ke dalam dimensi. Kernel RBF lebih sedikit memiliki *hyperparameter* bila dibandingkan kernel polynomial. Kernel RBF dirumuskan sebagai berikut:

$$K(\bar{x}_i, \bar{x}_j) = \exp\left(-\frac{\|\bar{x}_i - \bar{x}_j\|^2}{2\sigma^2}\right) \quad (2.4)$$

Pada penelitian ini, langkah langkah metode Support Vector Machine yang digunakan untuk deteksi penyakit daun labu adalah *training* SVM dan *Hyperparameter Tuning* dengan GridSearchCV. Data fitur dari CNN yang sudah diekstraksi kemudian dilakukan pelatihan dengan mencari hyperplane terbaik yang memisahkan fitur dalam kelas.

Pada SVM, model klasifikasi digunakan untuk memisahkan dua kelas dengan mencari *hyperplane* dengan margin terbesar, yang dirumuskan sebagai berikut:

$$w^T x + b = 0 \quad (2.5)$$

Dengan w adalah vektor bobot (*weight*), x adalah vektor fitur input, b adalah bias, dan w^T adalah *transpose* vektor bobot w . Kemudian dilakukan prediksi kelas untuk mencari titik data baru dengan rumus:

$$\text{Prediksi} = \text{sign}(w^T x_i + b) \quad (2.6)$$

Untuk mendefinisikan kelas negatif, titik terdekat didefinisikan dengan $w^T x + b = -1$; dan pendefinisian kelas positif dengan $w^T x + b = 1$. Jarak dari *hyperplane* tersebut adalah $2/\|w\|$, di mana $\|w\|$ adalah norma dari vektor bobot w . Dalam mengoptimasi margin dengan meminimalkan $\|w\|$ dilakukan rumus berikut:

$$\min\{w, b\} \cdot 1/2 \cdot \|w\|^2 \quad (2.7)$$

Dengan memperhatikan kendala berikut:

$$y_i (w^T x + b) \geq 1$$

Selain kernel, terdapat hyperparameter lainnya yang juga penting karena memiliki fungsi dan penentu yang berbeda, yaitu C (*Regularization parameter*) dan Γ (*Kernel Coefficient*). C mengontrol keseimbangan antara memaksimalkan margin dan meminimalkan kesalahan klasifikasi. Nilai C yang besar cenderung menghasilkan margin yang kecil dan memaksa semua titik data diklasifikasikan dengan benar (risiko *overfitting*)[41]. Γ berfungsi mengatur

pengaruh satu data terhadap garis pemisah. Gamma yang besar menyebabkan model lebih fokus pada data sekitar (berisiko *overfit*), sedangkan gamma kecil menyebabkan model lebih umum (berisiko *underfit*)[42]. Gamma relevan digunakan pada kernel non-linear sebagaimana pada fungsi kernel RBF dan *polynomial*. *Grid Search*[43] (GridSearchCV) adalah metode sistematis untuk menemukan kombinasi hyperparameter terbaik dalam suatu model pembelajaran mesin dengan melakukan percobaan dengan membandingkan semua kemungkinan kombinasi nilai-nilai parameter yang telah ditentukan sebelumnya. Untuk setiap *grid search* akan dilakukan semua kombinasi yang memungkinkan, sebagai contoh pada kode 2.1.

```
param_grid = {  
    'C': [0.1, 1, 10],  
    'gamma': ['scale', 'auto', 0.01, 0.001],  
    'kernel': ['rbf', 'linear', 'poly']  
}
```

Kode 2.1 Parameter Grid

Pada gambar berikut, *grid search* akan melakukan semua kombinasi yang memungkinkan dengan menghasilkan total 36 kombinasi, misalnya: (C=0.1, gamma=0.001, kernel='linear'), (C=0.1, gamma=0.001, kernel='rbf'), dan seterusnya hingga kombinasi ke 36.

2.5. Convolutional Neural Network

Convolutional Neural Network (CNN) merupakan salah satu arsitektur dari jaringan saraf tiruan (*Artificial Neural Network*) yang secara khusus dirancang untuk memproses data berbentuk grid, seperti citra digital dua dimensi. CNN sangat efektif dalam menangani data visual karena mampu mengekstraksi fitur spasial dari gambar dan mempertahankan hubungan spasial antara piksel, tanpa memerlukan teknik ekstraksi fitur manual[44].

CNN memiliki arsitektur CNN dengan beberapa jenis lapisan utama (gambar) , yaitu: Convolutional Layer, Activation Function (ReLU), Pooling Layer, Fully Connected Layer (FC). Convolutional Layer merupakan lapisan yang

bertugas menerapkan filter (kernel) ke input gambar, menghasilkan feature map[45]. Filter ini belajar mendeteksi pola seperti tepi, garis, dan tekstur. Operasi konvolusi memungkinkan jaringan untuk mengekstraksi fitur lokal dari suatu area citra. Operasi konvolusi dilakukan antara input (gambar) dan filter (kernel) untuk menghasilkan *feature map*, sebagai berikut:

$$S(i,j)=(I*K)(i,j)=\sum_m\sum_n I(i+m,j+n) \cdot K(m,n) \quad (2.8)$$

Dimana:

$I(i,j)$ = nilai piksel input pada posisi (i, j)

$K(m,n)$ = nilai kernel (filter) pada posisi (m, n)

$S(i,j)$ hasil output (feature map) dari posisi (i, j)

Activation Function (ReLU) merupakan fungsi aktivasi untuk meningkatkan non-linearitas, sehingga model mampu mempelajari representasi kompleks dari data input non linearitas, sebagaimana dirumuskan seperti berikut[46]:

$$f(x)=\max(0,x) \quad (2.9)$$

Fungsi ini akan menghasilkan 0 jika nilai negatif dan tetap mempertahankan nilai positif.

Pooling layer adalah lapisan untuk mereduksi dimensi *feature map* dan menjaga fitur penting. Jenis *pooling* yang umum adalah *max pooling*. Lapisan ini juga membantu mencegah *overfitting* dengan mengurangi kompleksitas model[47]. *Fully Connected Layer* (FC) adalah Lapisan yang berfungsi menggabungkan semua fitur yang telah diekstraksi dan melakukan klasifikasi akhir. *Output* dari FC *layer* biasanya diteruskan ke fungsi aktivasi *softmax* untuk klasifikasi *multiclass*[48].

Untuk menyelesaikan masalah klasifikasi *multi class* pada CNN, akan

dilakukan fungsi *loss* dengan mengukur seberapa jauh prediksi model dari label sebenarnya, lalu digunakan untuk mengarahkan proses *training* (*backpropagation*), dirumuskan sebagai berikut:

$$L = - \sum_{i=1}^C y_i \log(\hat{y}_i) \quad (2.10)$$

Fungsi *Softmax* mengubah *output* dari *layer* terakhir menjadi probabilitas antar kelas sebagaimana pada rumus berikut:

$$\hat{y}_i = \frac{e^{z_i}}{\sum_{j=1}^C e^{z_j}} \quad (2.11)$$

Dimana:

z_i = output (logit) dari neuron ke-i

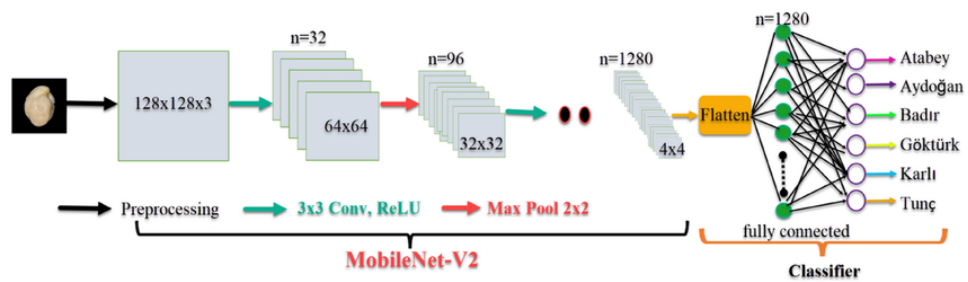
$\hat{y}_i$ = probabilitas prediksi kelas ke-i

C = jumlah kelas

Pada penelitian deteksi penyakit pada tanaman daun labu CNN, memungkinkan sistem untuk mengenali dan membedakan daun sehat maupun daun yang terinfeksi pola-pola penyakit berdasarkan warna, bentuk, tekstur, dan struktur daun tanpa perlu algoritma pendeteksian fitur secara manual[49].

2.5.1. MobileNetV2

MobileNetV2 adalah arsitektur jaringan konvolusi dalam (Convolutional Neural Network/CNN) yang dirancang khusus untuk aplikasi pada perangkat dengan sumber daya terbatas, seperti perangkat mobile dan embedded systems. Dikembangkan oleh Sandler et al[50], MobileNetV2 merupakan peningkatan dari MobileNetV1 dengan memperkenalkan *blok inverted residuals* dan *bottleneck* yang efisien, memungkinkan pengurangan kompleksitas komputasi dan parameter tanpa mengorbankan akurasi. Arsitektur ini sangat cocok untuk tugas klasifikasi gambar, termasuk deteksi penyakit daun labu dalam penelitian ini, karena menggabungkan efisiensi dengan kemampuan ekstraksi fitur yang kuat.



Gambar 2.4 Blok Diagram dari Arsitektur MobileNetV2

Sumber: [51]

MobileNetV2 terdiri dari serangkaian lapisan yang dimulai dengan konvolusi standar, diikuti oleh 17 blok *bottleneck residual*, dan diakhiri dengan lapisan *pooling* dan *fully connected*. Setiap blok *bottleneck* memiliki operasi yaitu *expansion layer*, *depthwise convolution*, dan *project layer*. *Expansion Layer* adalah lapisan konvolusi 1x1 yang meningkatkan jumlah kanal (dikenal sebagai *expansion factor*) untuk memperkaya representasi fitur. *Depthwise Convolution* adalah konvolusi *depthwise* 3x3 yang diterapkan pada setiap kanal input secara terpisah untuk mengurangi kompleksitas. *Projection Layer* adalah lapisan konvolusi 1x1 yang mengompresi jumlah kanal kembali ke dimensi yang lebih rendah[50].

Performa MobileNetV2 dapat diukur melalui kompleksitas komputasi dan jumlah parameter. Rumus dari kompleksitas komputasi atau FLOPs dirumuskan sebagaimana pada rumus 2.9.

$$FLOPs = H \cdot W \cdot K \cdot C_{in} \cdot C_{out} \quad (2.12)$$

Dimana:

H dan W = Tinggi dan lebar peta fitur.

K = Ukuran kernel (misalnya, 3 untuk konvolusi 3x3).

C_{in} = Jumlah kanal input.

C_{out} = Jumlah kanal output.

Untuk *depthwise separable convolution* yang digunakan pada MobileNetV2, kompleksitas dipecah menjadi dua bagian, yaitu *Depthwise Convolution* dan *Pointwise Convolution* sebagaimana pada rumus 2.10 dan rumus 2.11

$$\text{Depthwise Convolution} = H \cdot W \cdot K \cdot K \cdot C_{in} \quad (2.113)$$

$$\text{Pointwise Convolution} = H \cdot W \cdot C_{in} \cdot C_{out} \quad (2.14)$$

Pada *Pointwise Convolution*, terdapat total FLOPs untuk *depthwise separable convolution*, dimana jauh lebih efisien bila dibandingkan dengan konvolusi standar dengan rasio pengurangan sekitar $1/K^2$ untuk kernel 3x3. FLOPs untuk *depthwise* dirumuskan sebagaimana pada rumus 2.12

$$FLOPs_{depthwise} = H \cdot W \cdot (K \cdot K \cdot C_{in} + C_{in} \cdot C_{out}) \quad (2.15)$$

Selain FLOPs, setiap blok *bottleneck* memiliki struktur yang ditentukan oleh tiga parameter utama, yaitu:

- t yang merupakan faktor ekspansi (expansion factor), bertujuan menentukan jumlah kanal pada lapisan ekspansi (misalnya, $t = 6$).
- c adalah jumlah kanal *output* setelah proyeksi.
- n yang merupakan Jumlah pengulangan blok.
- s adalah *Stride* (1 atau 2).

Bila ada koneksi yang bersisa, maka *output* blok digabungkan dengan *input* asli, yang dirumuskan pada kode 2.13.

$$\text{Output - Projection} + \text{Input (jika } s = 1) \quad (2.16)$$

Jumlah parameter pada MobileNetV2 didominasi oleh lapisan *pointwise convolution*. Total parameter untuk satu blok dirumuskan pada kode 2.14.

$$Parameters = (C_{in} \cdot t \cdot C_{in}) + (t \cdot C_{in} \cdot c) + (c \cdot c) \quad (2.17)$$

Di mana $c \cdot c$ muncul dari bias. Dengan 17 blok dan lapisan awal/akhir, total parameter MobileNetV2 sekitar 3.5 juta, yang jauh lebih rendah dibandingkan arsitektur seperti ResNet[50].

2.6. Evaluasi Model

Evaluasi model bertujuan untuk mengetahui seberapa baik performa dari model penelitian yang telah dilakukan. Evaluasi model penelitian dilakukan dengan *classification report* yaitu mencari akurasi, presisi, *recall*, dan *F1-Score*. Akurasi merupakan proporsi prediksi benar dari total keseluruhan data, seberapa baik tingkat keakuratan model yang telah dilatih dan diuji dari semua total sampel dataset; sehingga semakin tinggi akurasi maka semakin baik model memprediksi kelas yang benar. Akurasi dirumuskan sebagaimana pada rumus ..

$$Akurasi = (TP + TN) / (TP + TN + FP + FN) \quad (2.18)$$

Dimana:

TP (True Positive): Prediksi positif benar

TN (True Negative): Prediksi negatif benar

FP (False Positive): Prediksi positif salah

FN (False Negative): Prediksi negatif salah

Presisi merupakan proporsi prediksi positif dari keseluruhan prediksi positif. Semakin tinggi presisi, maka prediksi model cenderung benar. Presisi dirumuskan sebagai berikut:

$$Presisi = TP / (TP + FP) \quad (2.19)$$

Recall mengukur proporsi sampel positif yang benar-benar diidentifikasi model dari segala sampel positif yang ada. Semakin tinggi *recall*, maka model dapat menemukan sebagian besar sampel positif. *Recall* dirumuskan sebagai berikut:

$$\text{Recall} = TP / (TP + FN) \quad (2.20)$$

F1-Score merupakan perbandingan rerata hasil dari presisi dan *recall*. Semakin tinggi *F1-score*, maka semakin seimbang presisi dan *recall* (model memiliki presisi dan *recall* yang baik). *F1-Score* dirumuskan sebagai berikut:

$$\text{F1-Score} = 2 * (\text{presisi} * \text{recall}) / (\text{presisi} + \text{recall}) \quad (2.21)$$

Selain itu, dilakukan visualisasi dalam bentuk *Confusion matrix* yang digunakan untuk menggambarkan hasil dari performa model dalam meneliti dataset yang diuji.

