

BAB II DASAR TEORI

2.1 TINJAUAN PUSTAKAN

Penelitian dengan judul “*Performance analysis of Local Binary Pattern and K-Nearest Neighbor on image classification of fingers leaves*” yang ditulis oleh A. D. Ningtyasa, E. B. Nababana dan S. Efendia. Penelitian tersebut bertujuan untuk klasifikasi jenis daun berbentuk jari, seperti daun papaya dan daun Chaya. Metode yang digunakan adalah ekstraksi fitur daun *Local Binary Pattern* (LBP) dibandingkan dengan metode ekstraksi fitur *Gray Level Co-Occurrence Matrix* (GMCM) dengan klasifikasi *K-Nearest Neighbor* (KNN). Hasil penelitian menunjukkan bahwa dengan menggunakan rasio NI/Np = 67%:33%, performansi sistem klasifikasikan citra daun dengan menggunakan metode ekstraksi LBP menghasilkan nilai akurasi data latih mendekati 95% dan data uji diperoleh nilai mendekati 76%. Sedangkan hasil ekstraksi menggunakan GLCM menunjukkan bahwa data pelatihan diperoleh nilai mendekati 83% dan data pengujian diperoleh nilai mendekati 58% [7].

Penelitian dengan judul “Perbandingan Metode KNN Dan LBPH Pada Klasifikasi Daun Herbal” oleh Isman, Andani Ahmad dan Abdul Latief dengan tujuan untuk membandingkan kinerja metode *Local Binnary Pattern Histogram* dengan metode *K-Nearest Neighbor* dalam klasifikasi tanaman herbal. Penelitian ini menggunakan pengujian 5 jenis daun herbal dari Sulawesi Tenggara. Total *dataset* yang digunakan sebanyak 100 *dataset*. Hasil evaluasi *Confusion Matrix*, didapatkan nilai akurasi menggunakan metode *K-Nearest Neighbour* (KNN) sebesar 97,5%, sedangkan untuk metode *Local Binary Pattern Histogram* (LBPH) mendapat total nilai akurasi sebesar 94% [8].

Penelitian dengan judul “*Local Binary Pattern* untuk Pengenalan Jenis Daun Tanaman Obat menggunakan *K-Nearest Neighbor*” dengan penulis Zulfrianto Y. Lamasigi, Maryam Hasan dan Yulianti Lasena menggabungkan metode *Local Binary Pattern* (LBP) untuk ekstraksi fitur dengan *K-Nearest Neighbor* untuk klasifikasi pengenalan jenis daun dari tanaman obat. Penelitian ini menggunakan 120 *dataset* yang dibagi menjadi 90 data *Training* dengan 6 jenis

daun tanaman obat dan percobaan 30 data *Testing*. Tanaman tersebut terdiri dari 15 daun binahong, 15 daun bayam duri, 15 daun jarak, 15 daun sirih, dan 15 daun afrika. Penggabungan kedua metode tersebut menghasilkan akurasi yang cukup baik yaitu sebesar 96.67%. Nilai akurasi tersebut berdasarkan perhitungan manual *convusion matrix* dengan nilai $k=9$ [9].

Berdasarkan beberapa Tinjauan Pustaka yang dilakukan penulis menunjukkan bahwa, penggunaan metode *Local Binary Pattern* dengan *K-Nearest neighbor* dapat digunakan untuk pembuatan model klasifikasi *Supervised Learning* (Dataset berlabel). Hasil dari penggabungan kedua metode tersebut memungkinkan menghasilkan akurasi yang baik untuk dataset *input* berupa citra tanaman atau daun.

2.2 DASAR TEORI

2.2.1 Tanaman Hidroponik

Pada awal tahun 1930 di Berkley California, William Frederick Gericke memelopori sistem hidroponik, yaitu sistem budidaya menggunakan air yang mengandung nutrisi dan mineral tanpa tanah [10]. Saat ini teknologi pertanian menggunakan hidroponik sudah sangat berkembang dan diterapkan secara luas. Hal tersebut dikarenakan pertanian hidroponik memiliki beberapa keunggulan dibandingkan dengan sistem budidaya konvensional, salah satunya mengurangi risiko yang berhubungan dengan tanah seperti gangguan serangga, jamur dan bakteri yang hidup di tanah [11]. Sistem ini juga lebih mudah dalam pemeliharaan seperti tidak melibatkan proses penyiangan dan pengolahan tanah dalam budidaya tanamannya[1][12].

Salah satu tanaman yang dapat dibudidaya secara hidroponik adalah pakcoy. Sayuran pakcoy umumnya dibudidaya secara generatif menggunakan biji yang ditanam langsung maupun melalui persemaian. Secara langsung yaitu biji akan langsung disebar pada lahan atau areal pertanaman sebagai benih. Sedangkan persemaian atau pembibitan yaitu menanam benih pada tempat khusus terlebih dahulu sampai pada umur tertentu tergantung dari jenis tanamannya, untuk memperoleh bibit yang baik dan seraga [1][13].



Gambar 2.1 Tanaman hidroponik pakcoy siap panen

Bibit tanaman pakcoy di persemaian bisaanya sekitar 2 sampai 3 minggu. Masa tanam relative tanaman hidroponik pakcoy sampai bisa dipanen adalah 2 bulan, dengan jumlah daun 3 sampai 5 helai [2]. Gambar 2.1 adalah contoh tanaman hidroponik pakcoy yang siap panen.

2.2.2 *Machine Learning*

Machine Learning merupakan serangkaian teknik yang dapat membantu dalam menangani dan memprediksi data yang sangat besar dengan cara mempresentasikan data-data tersebut dengan algoritma pembelajaran [14][15]. Proses pembelajaran dalam *Machine Learning* merupakan suatu usaha dalam memperoleh kecerdasan, yang bisaanya melalui dua yaitu proses latihan (*Training*) dan pengujian (*Testing*)[16].

Machine learning terbagi menjadi tiga kategori klasifikasi yaitu *Supervised Learning*, *Unsupervised Learning*, *Reinforcement Learning*[17].

1. *Supervised Learning*

Pada klasifikasi *supervised learning*, pembelajaran menggunakan masukan data yang telah diberi kelas atau label. Setelah itu model membuat prediksi sesuai data yang telah diberi label.

2. *Unsupervised Learning*

Penggunaan skenario *unsupervised learning*, pembelajaran menggunakan masukan data pembelajaran yang tidak diberi kelas atau label. Kemudian model mencoba untuk mengelompokan data berdasarkan karakteristik-karakteristik yang ditemui dari data masukan.

3. *Reinforcement learning*

Klasifikasi *reinforcement learning* digambarkan dengan campuran antara fase pembelajaran dan tes. Model mengumpulkan informasi pembelajar secara

aktif dengan berinteraksi ke lingkungan sehingga untuk mendapatkan balasan atau penghargaan untuk setiap aksi dari pembelajar [16][18].

Selain ketiga klasifikasi diatas ada juga yang disebut dengan *semi-supervised learning*, dimana model pembelajaran secara otomatis menggunakan data masukan tanpa label untuk pembelajaran langsung sejumlah kecil data berlabel [14][19].

Deep Learning merupakan salah satu bidang dari *machine learning* yang memanfaatkan jaringan syaraf tiruan yang bisaa diimplementasi pada permasalahan jumlah *dataset* yang besar [20]. Teknik *deep learning* memberikan arsitektur kuat pada klasifikasi *supervised learning* atau model pembelajaran yang mewakili data citra berlabel.

Terdapat dua istilah penting dalam pembangunan model *machine learning* yaitu *Training* dan *Testing*. *Training* adalah proses konstruksi model dan *Testing* adalah proses menguji kinerja dari model *learning*[21]. *Dataset* adalah kumpulan data (sampel dalam statistik). Sampel ini adalah data yang kita gunakan untuk membuat model maupun mengevaluasi model *machine learning*. *Dataset* dibagi menjadi tiga jenis yang tidak beririsan (suatu sampel pada himpunan tertentu tidak muncul pada himpunan lainnya).

1. *Training set* merupakan data yang digunakan untuk melatih atau membangun model.
2. *Development set* atau *validation set* adalah data yang digunakan untuk mengoptimisasi saat melatih model, umumnya kinerja saat latihan diuji menggunakan validation set. Hal ini berguna agar model mampu mengenali pola secara generik.
3. *Testing* yaitu data yang digunakan untuk menguji model setelah proses latihan selesai[15][22].

2.2.3 Citra Digital

Citra adalah representasi dari suatu objek bisa juga diartikan sebagai gambar dua dimensi. Citra memiliki peranan penting sebagai bentuk informasi visual, karna citra sendiri mempunyai karakteristik dan kaya akan informasi [23]. Citra dapat didefinisikan sebagai fungsi dua dimensi $f(x,y)$, dimana x dan y adalah koordinat bidang datar, dan harga f pada setiap koordinat (x,y) disebut intensitas

atau level keabuan (*gray level*) dari citra di titik itu. Jika x , y , dan f semuanya berhingga dan nilainya diskrit maka citra itu disebut citra digital [5][24].

Citra digital dapat dibedakan menjadi tiga yaitu citra biner (*binary image*), citra keabuan (*grayscale image*) dan citra warna (*color image*).

1. *Color Image* atau RGB (*Red, Green, Blue*)

Citra berwarna (*color image*) atau bisa disebut citra RGB adalah jenis citra yang menyajikan warna dalam bentuk komponen merah, hijau, dan biru. Setiap komponen warna menggunakan 8 bit (nilainya berkisar antara 0 sampai dengan 255). Dengan demikian, kemungkinan warna yang bisa disajikan mencapai 255³ atau 16.581.375 (16K) warna[14].

2. *Grayscale* (Keabu-abuan)

Piksel dari citra digital *grayscale* (keabu-abuan) memiliki gradasi warna mulai dari putih sampai hitam[25]. Rentang warna pada *grayscale image* banyak digunakan dalam dunia kedokteran (X-ray). *Grayscale* merupakan hasil rata-rata dari *color image* dengan persamaan sebagai berikut:

$$I_{BW}(x, y) = \frac{I_R(x, y) + I_G(x, y) + I_B(x, y)}{3} \quad (2.1)$$

Keterangan:

$I_R(x, y)$: nilai piksel *Red* titik (x, y)

$I_G(x, y)$: nilai piksel *Green* titik (x, y)

$I_B(x, y)$: nilai piksel *Blue* titik (x, y)

$I_{BW}(x, y)$: nilai piksel *Black and White* titik (x, y)

3. *Binary Image*

Komponen warna dari *binary image* hanya terdiri dari dua warna yaitu hitam dan putih, maka dibutuhkan 1 bit per piksel (0 dan 1) atau apabila dalam 8 bit (0 dan 255). *Binary image* cocok digunakan pada penerapan teks, sidik jari, maupun gambar arsitektur[26][27].

2.2.4 Ekstraksi Fitur

Fitur merupakan karakteristik unik atau pembeda dari suatu objek gambar. Fitur dibedakan menjadi dua yaitu fitur “alami” merupakan bagian dari gambar,

misalnya kecerahan dan tepi objek. Sedangkan fitur “buatan” merupakan fitur yang diperoleh dengan operasi tertentu pada gambar, misalnya histogram tingkat keabuan [28].

Feature Extraction atau ekstraksi fitur merupakan suatu proses pengambilan ciri (*feature*) dari suatu bentuk atau gambar sebagai pembeda yang kemudian akan dianalisis untuk proses selanjutnya. Ekstraksi fitur bertujuan untuk mencari daerah fitur yang signifikan pada gambar tergantung pada karakteristik intrinsik dan aplikasinya. Fitur tersebut dibedakan menjadi tiga yaitu berdasarkan bentuk, warna, dan tekstur[5].

Salah satu contoh metode ekstraksi fitur berdasarkan tekstur adalah metode *Local Binary Patterns* (LBP). Implementasi ekstraksi fitur tekstur untuk membedakan objek satu dengan objek lainnya dapat menggunakan ciri statistik orde pertama atau ciri statistik orde dua.

1. Ciri orde pertama didasarkan pada karakteristik histogram citra. Ciri orde pertama umumnya digunakan untuk membedakan tekstur makrostruktur (perulangan pola lokal secara periodik). Ciri orde pertama antara lain: *mean, variance, skewness, kurtosis, dan entropy*.
2. Ciri orde dua didasarkan pada probabilitas hubungan ketetanggaan antara dua piksel pada jarak dan orientasi sudut tertentu. Ciri orde dua umumnya digunakan untuk membedakan tekstur mikrostruktur (pola lokal dan perulangan tidak begitu jelas). Ciri orde dua antara lain: *Angular Second Moment, Contrast, Correlation, Variance, Inverse Different Moment, dan Entropy* [29].

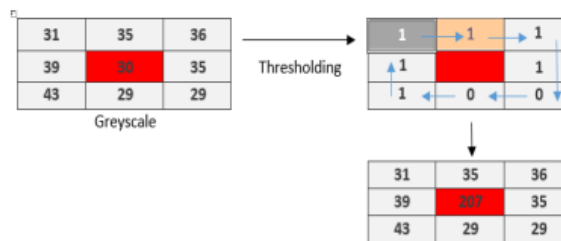
Metode LBP dapat menggambarkan tekstur dan bentuk dari citra digital yaitu dengan melakukan pembagian gambar menjadi bagian-bagian kecil dari fitur yang diekstraksi [6].

2.2.5 Local Binary Patterns (LBP)

Local Binary Pattern diperkenalkan pertama kali pada tahun 1992 oleh Timo Ojala dan David Harwood di Universitas Maryland [5][30]. *Local Binary Pattern* (LBP) merupakan salah satu metode ekstraksi fitur berdasarkan tekstur dari sebuah citra digital. Ekstraksi fitur tekstur menggunakan LBP dilakukan dengan membagi citra menjadi bagian-bagian kecil dari fitur yang diekstraksi.

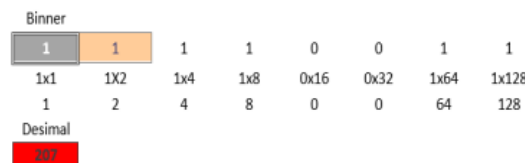
Fitur tersebut merupakan pola-pola biner yang menggambarkan piksel pada daerah yang terbagi. Fitur yang diperoleh dari gabungan daerah tersebut membentuk suatu histogram tunggal yang merupakan representasi dari sebuah citra. Citra tersebut dapat dibandingkan dengan mengukur kesamaan (jarak) dari masing-masing histogram [31].

Pada sebuah citra digital berukuran 3×3 , *Local Binary Pattern* diartikan sebagai perbandingan perbedaan nilai biner piksel pusat/tengah citra dengan 8 nilai piksel disekelilingnya. Dapat dikatakan bahwa Operator LBP bekerja dengan delapan konektivitas piksel dengan menggunakan nilai piksel pusat sebagai nilai ambang batas (*thresholding*)[32]. Jika nilai piksel pusat lebih kecil, maka nilai sekelilingnya bernilai 0 sebaliknya jika nilai piksel pusat lebih besar maka nilai sekelilingnya adalah 1. Dengan jumlah 8 piksel disekelilingnya, hasil perbandingan tersebut menghasilkan 8 nilai biner seperti terlihat pada Gambar 2.2 Hal tersebut berarti memungkinkan ada $2^8 = 256$ kombinasi kode LBP.



Gambar 2.2 Operator LBP [29]

Selanjutnya menghitung nilai LBP untuk menggantikan nilai piksel pusat atau tengah. Caranya, 8 nilai biner disusun dengan cara *clockwise* (searah jarum jam) atau *counter-clockwise* (berbalik arah jarum jam) secara konsisten. Kemudian hasilnya disimpan dalam array 8 bit biner untuk diubah menjadi desimal.



Gambar 2.3 Pengambilan 8 bit biner menjadi desimal [29]

Nilai desimal dari 8 bit dapat dinyatakan dalam persamaan (2.2):

$$LBP_{P,R}(x_c, y_c) = \sum_{p=0}^{P-1} s(g_p - g_c) 2^p \quad (2.2)$$

Keterangan:

P : banyaknya piksel tetangga

R : nilai jarak/radius

g_c : nilai piksel x dan y

g_p : nilai *grayscale* tetangga

x_c, y_c : titik piksel pusat

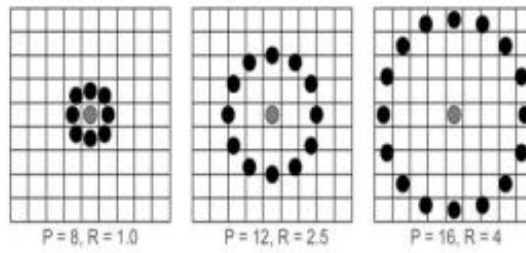
Dan fungsi $s(x)$ didefinisikan dengan:

$$s(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases} \quad (2.3)$$

fungsi *threshold* (2.3) bertujuan untuk menghilangkan variabilitas yang disebabkan oleh iluminasi kontras sehingga citra dengan berbagai pencahayaan akan mendapatkan *output* yang decimal mirip[5][33].

Nilai decimal tersebut akan disimpan dalam *array output Local Binary Pattern 2D*, sehingga dapat divisualisasikan. Proses tersebut disebut *thresholding* atau mengumpulkan biner dan menyimpan nilai decimal tersebut pada keluaran *array Local Binary Pattern*. Hal tersebut diulang untuk setiap piksel pada citra yang dimasukkan, sehingga menghasilkan *output* dengan lebar dan tinggi yang sama dengan gambar asli[5].

Sebuah lingkaran terbentuk dengan menggunakan radius R dari piksel pusat. Sedangkan titik sample P disekitar lingkaran didapat dan dibandingkan dengan nilai piksel pusat. Interpolasi untuk memperoleh nilai dari semua titik sampel pada berbagai radius dan jumlah piksel dinotasikan dengan (P, R) [34]. Gambar 2.4 menunjukkan 3 contoh gambaran ketetanggaan piksel dengan nilai P dan R yang berbeda.



Gambar 2.4 Contoh ketetanggaan piksel dengan nilai R dan P berbeda [31]

Jika koordinat dari piksel pusat adalah (x_c, y_c) maka koordinat dari tetangga $P(x_p, y_p)$ pada tepi lingkaran dengan jari-jari R dapat ditentukan dengan persamaan (2.4) dan persamaan (2.5) [35].

$$x_p = x_c + R \cos\left(\frac{2\pi p}{P}\right) \quad (2.4)$$

$$y_p = y_c + R \sin\left(\frac{2\pi p}{P}\right) \quad (2.5)$$

Keterangan:

p = jumlah tetangga

x_c = koordinat x dari pusat piksel

y_c = koordinat y dari pusat piksel

x_p = koordinat x tetangga (p)

y_p = koordinat y tetangga (p)

Citra hasil LBP operator dibagi menjadi beberapa region. Pada setiap region dilakukan proses ekstraksi histogram. Histogram merupakan diagram yang menggambarkan frekuensi setiap nilai yang muncul di seluruh piksel citra namun tidak mencerminkan urutan posisi warna piksel dalam citra[36]. Selanjutnya, histogram yang dihasilkan pada tiap-tiap region tersebut digabungkan menjadi satu [37].

Histogram citra adalah sebuah grafik yang menyatakan distribusi nilai-nilai warna atau intensitas penyebaran piksel-piksel di dalam citra. Frekuensi kemunculan nilai intensitas piksel pada suatu citra dapat diketahui melalui histogram citra tersebut. Histogram juga dapat menunjukkan kecerahan dan

kontras dari suatu citra dengan puncak histogram menyatakan intensitas piksel yang paling sering muncul pada suatu citra. Sumbu horizontal suatu histogram mewakili intensitas piksel dan sumbu vertical mewakili frekuensi kemunculan intensitas piksel. Secara matematis, histogram citra dinyatakan dengan persamaan (2.6).

$$h_i = \frac{n_i}{n}, i = 0, 1, \dots, L-1 \quad (2.6)$$

Keterangan:

h_i = Frekuensi kemunculan piksel berintensitas ke- i

n_i = Banyaknya piksel berintensitas ke- i

n = jumlah piksel keseluruhan citra

$L : 2^k$ atau kedalaman bit piksel

Histogram dibentuk dengan menggambarkan besarnya h_i pada setiap i [38].

2.2.6 *K-Nearest Neighbor*

K-Nearest Neighbor (KNN) telah mulai digunakan sejak awal tahun 1970-an untuk pengenalan pola dan estimasi statistik sebagai teknik *non-parametik*, sekarang ini mulai banyak digunakan untuk klasifikasi objek [29]. Algoritma *K-Nearest Neighbor* merupakan algoritma *supervised learning* dimana klasifikasinya berdasarkan kedekatan jarak antara data uji dengan data latih (data *Training*) yang menjadi tetangga dari data uji tersebut. Dengan kata lain tujuan dari algoritma KNN adalah untuk mengklasikasikan obyek baru berdasarkan atribut dan sample-sample dari data *Training* yang ada pada sistem, berdasarkan jarak ketetanggaan dari data uji ke data *Training* (data latih) [5][39]. Algoritma ini menggunakan *Neighborhood Classification* sebagai nilai prediksi dari nilai *instance* yang baru [40]. Persamaan algoritma KNN:

$$similarity = \frac{\sum_{i=1}^n f(T_i, S_i)}{W_i} \quad (2.7)$$

Keterangan:

T : Kasus baru

S : kasus yang ada dalam penyimpanan

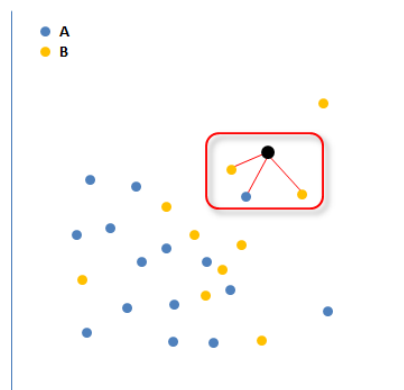
N : jumlah atribut

i : atribut individu antara 1 – n

f : fungsi *similarity* ATRIBUT I antara kasus T dan kasus S

w : bobot yang diberikan pada atribut ke- i [41].

Analisis yang dilakukan oleh algoritma *K-Nearest Neighbor* adalah pengaruh penggunaan pengukuran kemiripan dan nilai k yang digunakan terhadap akurasi sistem dalam mengklasifikasikan sebuah citra digital. Nilai k yang diuji disarankan bernilai ganjil untuk mengurangi resiko kesalahan algoritma jika peluang kemiripannya sama [41]. Penggunaan nilai k sebaiknya disesuaikan dengan banyaknya data yang ada dan ukuran dimensi yang dibentuk oleh data. Semakin banyak data, angka k yang dipilih sebaiknya semakin kecil. Namun, semakin besar ukuran dimensi data, angka k yang dipilih sebaiknya semakin besar [23][42].



Gambar 2.5 Ilustrasi Kerja Algoritma KNN [43]

Cara kerja Algoritma KNN diilustrasikan dengan dua kelas biru (A) dan kuning (B) dengan data baru berwarna hitam seperti pada Gambar 2.5 Algoritma KNN akan menghitung menghitung jarak antara data hitam ke data *point* dari *dataset*. Jika nilai K yang digunakan adalah 3, maka akan diambil 3 titik dengan jarak terdekat dengan data hitam. Pada ilustrasi diatas didapatkan titik terdekat yaitu 2 kuning dan 1 biru. Maka kelas dari data baru (hitam) tersebut adalah kelas kuning (B).

Pada penelitian ini, pengukuran kedekatan jarak berdasarkan perhitungan *Chi-Square*. *Chi-Square* digunakan pada *feature selection*[39]. Konsep dasar *Chi-*

Square yaitu menguji kemiripan dari 2 buah distribusi histogram citra tes dan distribusi histogram citra *Training* dengan cara menghitung nilai minimum dari rata-rata 2 histogram citra untuk menentukan kemiripan antara kedua citra tersebut [5][44]. Perumusan *feature selection* menggunakan *Chi-Square* digambarkan dengan persamaan (2.8) [39].

$$X^2(t, c) = \frac{N(A \times D - B \times C)^2}{(A + B) \times (C + D) \times (A + C) \times (B + D)} \quad (2.8)$$

Keterangan rumus:

c = kelas

t = term yang dicari pada kelas c

N = jumlah data training

A = jumlah data pada kelas c

B = jumlah data yang tidak ditemukan pada kelas c tapi mengandung term t

C = jumlah data pada kelas c namun tidak mengandung term t

D = jumlah data bukan kelas c dan tidak mengandung term t

2.2.7 Confusion Matrix

Confusion Matrix juga sering disebut *error matrix*, merupakan metode evaluasi yang bisa digunakan pada masalah klasifikasi. Pada dasarnya *Confusion Matrix* berbentuk table matriks yang memberikan informasi perbandingan hasil klasifikasi yang dilakukan oleh sistem (model) dengan hasil klasifikasi sebenarnya [45][46]. Terdapat 4 istilah sebagai representasi hasil proses klasifikasi pada *Confusion Matrix*. Keempat istilah tersebut adalah *True Positive (TP)*, *True Negative (TN)*, *False Positive (FP)* dan *False Negative (FN)*[47].

- 1) *True Positive (TP)* yaitu ketika data uji diklasifikasikan positif (+) dan diprediksi benar sebagai data positif (+).
- 2) *False Positive (FP) - Type I error* yaitu ketika data uji diklasifikasikan positif (+) dan diprediksi salah sebagai data negatif (-).
- 3) *True Negative (TN)* yaitu ketika data uji diklasifikasikan negatif (-) dan diprediksi benar sebagai data negatif (-).
- 4) *False Negative (FN) - Type II error* yaitu ketika data uji diklasifikasikan negatif (-) dan diprediksi salah sebagai data positif (+)[45].

Confusion Matrix akan memberi tahu seberapa baik model yang kita buat dan secara khusus memberikan informasi tentang TP, FP, TN, dan FN. Hal ini sangat berguna karena hasil dari klasifikasi umumnya tidak dapat diekspresikan dengan baik dalam satu angka saja. *Confusion Matrix* dapat digunakan untuk menghitung berbagai performance metrics untuk mengukur kinerja model yang telah dibuat seperti *accuracy*, *precision*, *recall* dan *f_score*.

1. Accuracy

Accuracy menggambarkan seberapa akurat model dapat mengklasifikasikan citra dengan benar. *Accuracy* merupakan rasio prediksi benar (positif dan negatif) dengan keseluruhan data, dapat dihitung menggunakan persamaan (2.9).

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (2.9)$$

Keterangan:

FN = False Negative

TP = True Positive

TN = True Negative

FP = False Positive

Persentase nilai akurasi hasil klasifikasi dapat dirumuskan dalam persamaan (2.10).

$$Akurasi = \frac{\text{Banyak prediksi yang benar}}{\text{Total banyak prediksi}} \times 100\% \quad (2.10)$$

2. Precision (Positive Predictive Value)

Precision menggambarkan tingkat keakuratan antara data yang diminta dengan hasil prediksi yang diberikan oleh model. Maka, *precision* merupakan rasio prediksi benar positif dibandingkan dengan keseluruhan hasil yang diprediksi positif.

$$\text{Precision} = \frac{TP}{TP + FP} \quad (2.11)$$

Keterangan:

$TP = \text{True Positive}$

$FP = \text{False Positive}$

3. *Recall* atau *Sensitivity* (*True Positive Rate*)

Recall menggambarkan keberhasilan model dalam menemukan kembali sebuah informasi. *Recall* merupakan rasio prediksi benar positif dibandingkan dengan keseluruhan data yang benar positif. Nilai *recall* dapat diperoleh dengan persamaan (2.12).

$$\text{Recall} = \frac{TP}{TP + FN} \quad (2.12)$$

Keterangan:

$FN = \text{False Negative}$

$TP = \text{True Positive}$

4. *F1_score*

F1_score menggambarkan perbandingan dari rata-rata nilai precision dan recall yang dibobotkan. Acuan *f1_score* dapat digunakan jika jumlah data *False Negatif* dan *False Positif* tidak sama atau *symmetric*. Persamaan (2.13) menunjukkan perhitungan dari *f1_score*[48].

$$f1_score = \frac{2 \times (\text{Recall} \times \text{Precision})}{\text{Recall} + \text{Precision}} \quad (2.13)$$

Keterangan:

$\text{Recall} = \text{Nilai Recall}$

$\text{Precision} = \text{Nilai Precision}$