

## BAB II DASAR TEORI

### 2.1. KAJIAN PUSTAKA

Pada tahun 2021 Nur Rachmat, Yohannes dan Adhytio Mahendra. Pada penelitian ini membahas tentang ikan laut yang berjumlah 7.000 citra disegmentasi warna oleh HSV dengan mengambil diubah menjadi *grayscale* yang diproses dengan HOG, untuk klasifikasi jenis ikan laut menurut SVM. Teknik pengujian dan pendistribusian dataset menggunakan metode *fold cross-validation Leave One Out* (LOO). Hasil pengujian klasifikasi SVM menggunakan perkalian *linier* dan perkalian *polinomial* dengan *3-Fold*, *4-Fold* dan *5-Fold*. Akurasi terbesar yaitu pada untuk ikan *Black Sea Spart* bernilai 94,06%. Pada ikan *Gilt Head Bream* akurasi terbesar bernilai 94,74%. Lalu untuk ikan *Hourse Mackarel* diperoleh nilai akurasi terbesar bernilai 94,74. Pada ikan jenis *Red Mullet* memperoleh nilai akurasi terbesar bernilai 94,76%. Kemudian untuk ikan *Red Sea Bream* nilai akurasi terbesar bernilai 94,86%. Untuk jenis *Sea Bass* memiliki akurasi terbesar bernilai 77,86% lalu yang terakhir pada jenis *Striped Red Mullet* dapat hasil nilai akurasi tertinggi bernilai 94,41%. Perbedaan pada penelitian ini menggunakan klasifikasi ikan laut yang terdapat 7.000 gambar menggunakan metode SVM kernel *linear* dan kernel *polynomial*. Penelitian yang diusulkan memiliki studi kasus yang berbeda dengan penulis karena pada penelitian ini akan digunakan kernel RBF [9].

Pada tahun 2020 idagli, ali dan aliozturk dengan judul “*Goruntu Siniflandirmada derin ogrenme yontemlerinin karsilastirilmesi* (Perbandingan metode *deep learning* pada klasifikasi citra)”. Pada penelitian ini dilakukan eksperimen melalui *deep learning* berbasis *Convolution Neural Network* (CNN), *Residential Energy Services Network* (ResNet) dan *Network In Network* (NIN). Pada model pembelajaran mendalam untuk set data CIFAR-10 dijalankan secara terpisah yang menggunakan mesin hanya memiliki *Central Processing Unit* (CPU) dan mesin yang memiliki CPU dan *Graphical Processing Unit* (GPU). Pada mesin yang hanya memiliki CPU waktu pelatihan R-Model, A-Model dan E-Model

masing-masing memiliki waktu sekitar 415 Jam, 129 jam dan 3,5 jam. Tingkat akurasi yang didapatkan pada set data validasi pada masing-masing senilai 82,76%, 87,64% dan 83,47%. Pada mesin yang hanya memiliki CPU dan GPU waktu pelatihan R-Model, A-Model dan E-Model masing-masing memiliki waktu sekitar 4,5 Jam, 2,20 jam dan 1,82 jam. Tingkat akurasi yang didapatkan pada set data validasi pada masing-masing senilai 82,61%, 87,96% dan 82,43%. Untuk dataset lainnya untuk menjalankan model pada mesin yang berisi CPU dan GPU. Perbedaan pada penelitian ini dengan melakukan eksperimen melalui metode *deep learning* yang berbasis CNN, ResNet, dan NIN [10].

Pada tahun 2021 Oguzhan Ulucan, Diclehan Karakaya, Mehmet Turkan dengan judul “*A Large-Fish Dataset for Fish Segmentation and Classification*”. Pada penelitian ini menggunakan metode SVM yang dirancang khusus untuk menganalisis menggunakan fitur yang diekstraksi secara acak dari gambar 70% untuk pelatihan dan 20% digunakan untuk menguji algoritme. Pada penelitian ini menggunakan penggabungan fitur yang berbeda dengan melakukan eksperimen menggunakan algoritma BoF memiliki tingkat akurasi rata-rata pelatihan pengujian terendah senilai 84,36% dan 81,55%. Lalu pada pelatihan dan pengujian terbaik dengan fitur kontras GLCM diadopsi dalam pengklasifikasian SVM dengan akurasi rata-rata 98,74%. Pada saat menggunakan fitur *energy* menghasilkan akurasi uji senilai 97,15% untuk ikan *trout* dan namun dengan nilai keberhasilan senilai 97,28% dicapai bila dikombinasikan dengan fitur kontras. Dengan menggunakan fitur kontras dan energi menghasilkan akurasi yang lebih tinggi dibandingkan menggunakan penggabungannya. Adopsi menggunakan CNNsF dalam *algoritme* SVM menghasilkan tingkat akurasi yang jauh lebih tinggi daripada BoF dan untuk tingkat keberhasilan fitur kontras dan energi GLCM dalam proses pelatihan. Perbedaan pada penelitian yang diusulkan menggunakan metode SVM yang khusus untuk menganalisis fitur yang di ekstraksi dengan penggabungan fitur yang berbeda menggunakan algoritma BoF dan fitur kontras dan energi GLCM [11].

Pada tahun 2021 Michael Coe, and Stefanie Gutschmidt dengan judul “*Computer Vision Estimation of Physical Parameters and Its Application to Power Requirements of Natural and Artificial Swimmers*”. Pada penelitian ini menggunakan metode Ika-Fit dan Segmentasi dan kontur gambar pake *python open*

CV. Menggunakan metode Ika-Fit menunjukkan pada penelitian ini akurasi yang baik saat mengukur luas permukaan, *volume* dan massa spesies ikan dan *platform* robot berbentuk ikan. Akurasi kisaran 20% untuk luas permukaan untuk sistem rekayasa yang tidak termasuk dalam bentuk ikan juga dapat diperoleh [12].

Pada tahun 2021 Ambuj Kumar Agarwal, Raj Gaurang Tiwari, Vikas Khullar dan Rajesh Kumar Kaushal dengan judul “*Transfer Learning Inspires Fish Species Classification*”. Pada penelitian ini teknik untuk mengenali ikan laut menggunakan *Convolutional Neural Network* (CNN) dengan klasifikasi *ResNet-50*, *VGG16* dan *InceptionResNetV2* yang telah dilatih sebelumnya menggunakan *transfer learning*. Pada saat mengklasifikasi akurasi yang terbaik mencapai 90% dengan nilai *precision*, *recall* dan *F1-Score* [6]. Penjelasan secara singkat perbandingan penelitian kajian pustaka akan dijelaskan pada tabel 2.1.

**Tabel 2. 1 Tabel Perbandingan Penelitian Kajian Pustaka**

No	Judul	Penulis(tahun)	Metode	Akurasi	Perbedaan penelitian
1	Klasifikasi jenis ikan laut menggunakan metode SVM dengan Fitur HOG dan HSV	Nur Rachmat, Yohanes, Adhytio Mahendra (2021)	Metode <i>Support Vector Machine</i> dengan segmentasi warna HSV dan ekstraksi fitur HOG untuk klasifikasi ikan laut. Menggunakan kernel <i>linear</i> dan <i>polynomial</i>	Akurasi penelitian ini mengklasifikasi ikan laut metode SVM dengan kataristik fitur HOG dan HSV mencapai tingkat akurasi terbesar dengan bernilai 94,86%	Kernal RBF
2	Görüntü Siniflandirmada Derin Öğrenme Yöntemlerinin Karşılaştırılması	Idagli, ali, aliozturk (2020)	Metode <i>deep learning</i> berbasis <i>Convolutional Neural Network</i>	Akurasi pada kumpulan data validasi masing masing yaitu	SVM

No	Judul	Penulis(tahun)	Metode	Akurasi	Perbedaan penelitian
	(Perbandingan Metode <i>deep learning</i> pada klasifikasi citra)		(CNN), <i>Residential Energy Services Network</i> (ResNet), <i>Network in Network</i> (NIN)	82,61%,87,95% dan 82,43%	
3	<i>A Large-Scale Dataset for Fish Segmentation and Classification</i>	Oguzhan Ulucan, Diclehan Karakaya, Mehmet Turkan (2021)	SVM	Akurasi fitur terkuat yaitu Algoritma Bof 84,36% dan 81.55%. Fitur kontras GLCM diadopsi dalam pengklasifikasi SVM rata-rata 98,74. Fitur <i>energy</i> tingkat akurasi uji 97,15% pada ikan <i>trout</i> namun tingkat keberhasilan 97,28% .	
4	<i>Computer Vision Estimation of Physical Parameters and Its Application to Power Requirements of Natural and Artificial Swimmers</i>	Michael Coe, and Stefanie Gutschmidt (2021)	Metode Ika-Fit dan Segmentasi dan kontur gambar pake <i>python open CV</i>	Akurasi 20% untuk luas permukaan untuk sistem rekayasa yang tidak berbentuk ikan juga dapat diperoleh	Metode SVM

No	Judul	Penulis(tahun)	Metode	Akurasi	Perbedaan penelitian
5	<i>Transfer Learning Inspires Fish Species Classification</i>	Ambuj Kumar Agarwal, Raj Gaurang Tiwari, Vikas Khullar dan Rajesh Kumar Kaushal (2021)	<i>Convolutional Neural network</i> (CNN) dengan klasifikasi ResNet-50, VGG16 dan <i>InceptionResNetV2</i> yang telah dilatih sebelumnya menggunakan transfer learning	akursi yang terbaik mencapai 90% dengan nilai <i>precision</i> , <i>recall</i> dan <i>F1-Score</i>	Metode SVM

Pada tabel 2.1 merupakan perbandingan penelitian kajian Pustaka yang terdapat lima penelitian yang menggunakan dataset berupa jenis ikan laut. Pada perbandingan penelitian ini terdapat banyak metode dengan akurasi tertinggi mencapai 97,28% dengan menggunakan metode fitur kontras GLCM.

## 2.2. DASAR TEORI

Pada dasar teori ini membahas tentang materi yang berkaitan dengan karakteristik fisik ikan laut, citra digital (RGB), *machine learning*, SVM, kernel RBF, HOG, HSV dan *confusion matrix*.

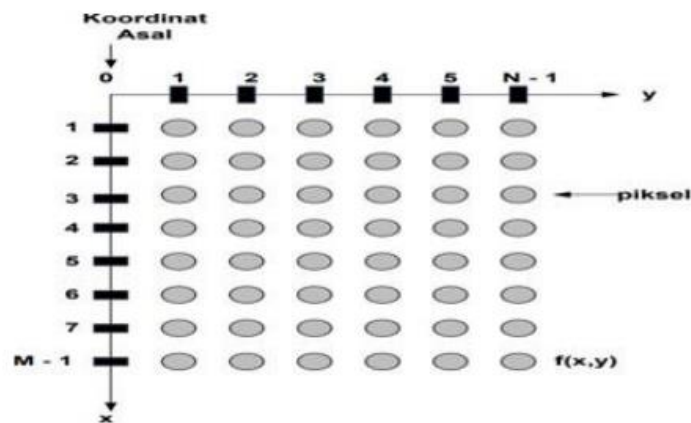
### 2.2.1 Ikan Laut

Ikan merupakan hewan *vertebrata*, mempunyai insang yang fungsinya untuk bernafas dan sirip yang memiliki fungsi untuk berenang. Ikan dapat ditemukan disebagian besar perairan laut dunia baik air laut maupun air laut dengan bentuk dan tipe yang berbeda. Ciri yang dimiliki dari kelas ikan adalah memiliki tulang rawan sejati, memiliki sirip tunggal ataupun ganda, tubuh sirip berlendir dan bagian tubuh yang jelas antara kepala, badan dan ekor [13].

### 2.2.2 Citra Digital

Citra digital adalah representasi visual dari data dalam bentuk diskret atau terpisah yang mewakili gambar atau objek dalam dunia nyata.. Setiap piksel mewakili suatu nilai atau kombinasi nilai yang mencerminkan informasi tentang warna, intensitas cahaya, atau atribut *visual* lainnya dari bagian gambar yang sesuai. Citra digital bisa berupa gambar berwarna atau hitam-putih (*grayscale*), dan dapat dihasilkan melalui berbagai perangkat seperti kamera digital, pemindai (*scanner*), dan alat lainnya. Citra digital sering digunakan dalam berbagai bidang, termasuk pengolahan citra, pengenalan pola, grafika komputer, visi komputer, kedokteran, astronomi, dan yang lainnya [14].

Sebuah Citra dapat didefinisikan sebagai fungsi dua dimensi  $f(x,y)$  yang dimana  $x$  dan  $y$  adalah sebuah koordinat spasial dan *amplitudo*  $f$  pada setiap pasang  $(x,y)$  yang disebut dengan intensitas dari citra titik tersebut. Jika  $x$  dan  $y$  berhingga dan diskrit maka disebut dengan citra digital. Citra digital terdapat dari sejumlah elemen terbatas yang masing-masing dengan posisi dan nilai, Elemen  $x$  dan  $y$  disebut dengan elemen gambar atau pels atau piksel yang ditampilkan pada gambar 2.1 [15].

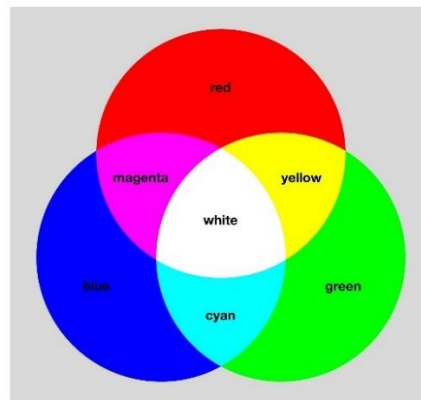


**Gambar 2. 1 Koordinat Dalam Citra Digital [16]**

Berdasarkan pada gambar 2.1 yang dimana secara sistematis tertulis sebagai intensitas( $x$  (baris).  $y$  (kolom)). Nilai fungsi pada titik tersebut mengartikan besar intensitas yang diterima citra atau warna dari piksel titik tersebut. Besar intensitas tersebut tergantung pada panyulan objek yang artinya  $f(x,y)$  sama dengan energi yang diberikan oleh sumber cahaya yang menghasilkan besar intensitas  $f(x,y)$ . Pada

citra digital dapat diwakilkan dengan sebuah matrik yang terdiri dari M berupa kolom dan N berupa baris yang dimana memotong antara kolom dan baris tersebut disebut piksel (*picture element*) yang merupakan elemen terkecil pada citra [16].

Ruang warna RGB merupakan warna primer dari suatu gambar dengan tiga warna primer yaitu merah, hijau dan biru. Kisaran dengan nilai yang dimiliki gambar RGB pada setiap piksel citra yaitu bernilai 0 hingga bernilai 225 pada gambar 2.2.

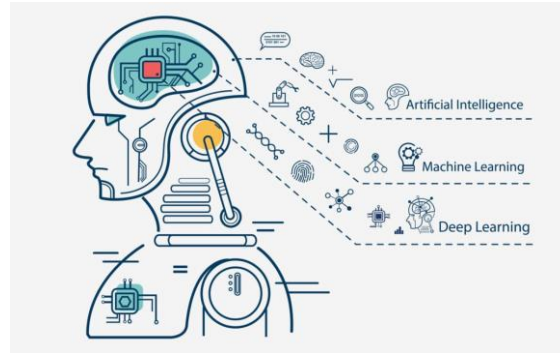


**Gambar 2. 2 Komponen RGB [17]**

Pada gambar 2.2 menunjukkan komponen RGB, Ruang warna dapat ditampilkan dalam bentuk histogram, yang dimana puncak pada histogram menunjukkan intensitas yang menonjol. Untuk rentang nilai yang dalam RGB, jika kemunculan nilai intensitas mendekati 0, maka citra bisa dikatakan dengan gelap. Akan tetapi jika nilai intensitas mendekati 255, maka citra bisa dikatakan dengan terang. Jadi, untuk mendapatkan kualitas citra yang baik, maka distribusi nilai pada histogram harus sama untuk setiap nilai intensitas piksel [18].

### **2.2.3 Machine Learning**

*Machine Learning* (ML) merupakan *Artificial Intellegant* (AI) yang focus untuk mempelajari data tentang pengembangan sistem yang dapat dipelajari secara mandiri tanpa harus diprogram ulang berulang kali oleh manusia. *Machine Learning* memerlukan data yang *valid* sebagai proses pelatihan sebelum digunakan dalam pengujian untuk hasil yang optimal yang dapat ditampilkan pada gambar 2.3.



**Gambar 2. 3 *Machine Learning*** [19]

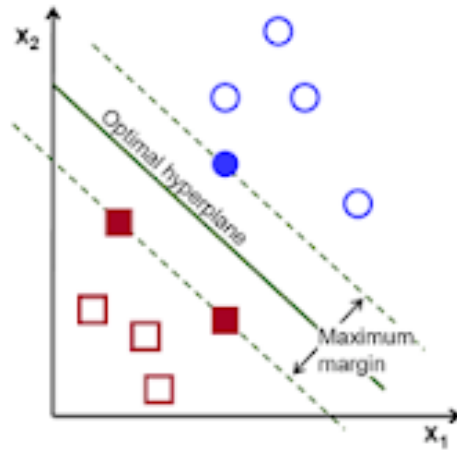
Pada gambar 2.3 merupakan *machine learning* hasil pengembangan berbasis *Artificial Intellegent* ini dapat memberikan kemudahan dan dapat langsung diterapkan kepada masyarakat luas. Simulasi pada model *Machine Learning* terkait dengan *statistic* komputasi yang bertujuan untuk pembuatan prediksi melalui *computer* [20].

#### **2.2.4 *Support Vector Machine (SVM)***

*Support Vector Machine (SVM)* adalah metode *supervised machine learning* yang menggunakan algoritma klasifikasi untuk *problem* klasifikasi yang dapat memisahkan 2 kelas atau lebih. *Support Vector Machine (SVM)* dapat dibagi menjadi dua jenis berdasarkan *hyperplane* yang digunakan adalah *linear* dan *non-linear*. *Support Vector Machine (SVM) linear* menggunakan *hyperplane linear* untuk memisahkan dua kelas. *Hyperplane linear* adalah garis atau permukaan datar yang memisahkan dua kelas dalam ruang fitur. Dapat digunakan dengan data yang terlihat dan dapat dipisahkan dengan baik oleh garis atau permukaan datar. Tetapi, jika data tidak dapat dipisahkan dengan baik oleh *hyperplane linear*, maka *SVM linear* mungkin tidak menghasilkan hasil yang memuaskan. Sedangkan *Support Vector Machine (SVM) non-linear* menggunakan klasifikasi lebih dari dua kelas dengan menggunakan kernel. Kernel SVM memungkinkan pemetaan data ini ke dimensi yang lebih tinggi melalui fungsi kernel seperti kernel *linear*, *polynomial* atau kernel *Radial Basis Functin (RBF)*, yang di mana *hyperplane* pemisah *non-linear* dapat ditemukan. Prinsip utama *Support Vector Machine (SVM)* untuk mencari *hyperplane* yang terbaik sebagai pemisah dua kelas pada ruang *input*.



*Hyperline* dapat berupa line pada *two dimension* dan dapat berupa *flat plane* pada *multiple plane* yang ditampilkan pada gambar 2.4 [21] .



**Gambar 2. 4 Support Vector Machine [22]**

Pada gambar 2.4 *Support Vector Machine* (SVM) adalah metode klasifikasi integritas karena pada proses pelatihan memerlukan tujuan pembelajaran. SVM adalah algoritma operasi *non linear* untuk mengonversi data pelatihan asli ke dimensi yang lebih tinggi. Dalam hal ini dimensi baru akan mencari *hyperplane* untuk dekomposisi linear dan dengan pemetaan *nonlinear* yang sesuai ke dimensi yang cukup tinggi, data dari kedua kelas tersebut masih dapat dipisahkan dengan *hyperline*.

Menggunakan pelatihan SVM umumnya lama, namun pada metode ini sangat akurat karena mampu menangani model *non-linear* yang kompleks. *Support Vector Machine* (SVM) kurang rentan terhadap *overfitting* dibandingkan dengan metode yang lain. *Support Vector Machine* (SVM) dapat digunakan untuk prediksi dan klasifikasi, misalnya pada deteksi tulisan tangan, pengenalan objek, pengenalan suara dan lainnya [23]. Untuk mencari nilai SVM dilakukan pada persamaan 2.1:

$$f(x) = \sum_i^N \alpha_i y_i K(x_i, x) + b \quad (2.1)$$

Keterangan :

$\alpha_i$  merupakan koefisien dari hasil proses optimasi, lalu  $y_i$  merupakan label kelas dari data pelatihan,  $K$  merupakan fungsi kernel dan  $b$  merupakan bias.

### 2.2.5 Kernel Radial Basis Function (RBF)

*Kernel Radial Basis Function* (RBF) merupakan salah satu kernel yang banyak digunakan karena mirip dengan distribusi *Gaussian*. Fungsi RBF pada dua titik pada  $X_1$  dan  $X_2$  untuk menghitung kesamaan atau seberapa dekat antara satu sama lain. Kernel RBF direpresentasikan secara matematis pada persamaan 2.2:

$$K(x, x') = e^{-\gamma \|x-x'\|^2} \quad (2.2)$$

Keterangan:

- $x, x'$  = Jarak antara dua *vector* data
- $\gamma$  = Parameter untuk mengontrol seberapa tajam pengaruh dari setiap data terhadap data lainnya. Semakin besar nilai maka semakin tajam pengaruhnya
- $\|x - x'\|$  = Jarak antara dua *vector* dengan menggunakan *Euclidean*
- $e^{-\gamma \|x-x'\|^2}$  = Mengubah jarak menjadi skala nilai antara 0 dan 1 yang menghasilkan representasi data dalam ruang dimensi yang lebih tinggi yang dapat membantu untuk menemukan *hyperplane* pemisah yang lebih kompleks.

Kernel *Radial Basis Function* (RBF) populer karena memiliki kemiripannya dengan Algoritma *K-Nearest Neighborhood*. Kernel RBF memiliki keunggulan K-NN dan mengatasi masalah kompleksitas ruang karena kernel RBF *Support Vector Machine* hanya perlu menyimpan *vector* pendukung selama pelatihan dan bukan seluruh dataset. Kernel RBF *Support Vector Machine* diimplementasikan pada *scikit-learn library* dan memiliki dua *hyperparameter* yang terkait [24] .

### 2.2.6 Histogram of Oriented Gradient (HOG)

*Histogram of oriented gradients* (HOG) yaitu salah satu pengolahan citra yang ditujukan untuk pengenalan objek. Teknik ini menghitung nilai *gradient* untuk wilayah gambar tertentu. Setiap gambar memiliki atribut yang diwakili oleh distribusi gradiennya. Fitur ini dicapai dengan membagi gambar menjadi area kecil

yang disebut sel. Setiap sel terdapat *histogram gradien*. Kombinasi *histogram* ini digunakan sebagai deskriptor untuk mewakili objek [25].

Langkah-langkah dalam proses *Histogram of Oriented Gradients* (HOG) adalah mengubah citra dari citra RGB menjadi citra *grayscale*. Langkah selanjutnya dilakukan dengan menghitung nilai gradien yang dihasilkan pada setiap piksel sebagai citra *grayscale*. Setelah mendapatkan nilai grup *orientasi*, proses selanjutnya adalah melakukan orientasi *binning* dan membentuk blok per grup sebagai representasi fitur, proses selanjutnya adalah melakukan normalisasi blok dimana fungsinya untuk mendapatkan nilai setiap blok yang akan direduksi. tumpang tindih [26]. Untuk menghitung nilai *magnitude* dan arah *gradient* dapat dilihat pada persamaan 2.3 dan 2.4

$$M = \sqrt{G_x^2 + G_y^2} \quad (2.3)$$

dan

$$\theta = \left| \tan^{-1} \left( \frac{G_y}{G_x} \right) \right| \quad (2.4)$$

Keterangan :

- $M$  = Menghitung nilai magnitudo *gradient*
- $\theta$  = Menghitung arah gradien setiap piksel
- $G_x$  dan  $G_y$  = Merupakan komponen *horizontal* dan *vertical* dari citra dengan mengambil turunan parsial terhadap koordinat x dan y

### 2.2.7 Hue, Saturation, Value (HSV)

Model yang dipakai dalam analisis warna untuk citra digital adalah HSV, atau *Hue*, *Saturation*, dan *Value*. *Hue* menggambarkan warna sebenarnya dari warna merah, ungu, kuning. Digunakan untuk mengidentifikasi untuk warna merah, hijau, dan lainnya. Saturasi menunjukkan intensitas warna. *Value* menunjukkan kecerahan warna dan bervariasi dari 0 hingga 100%. Pada *value* bernilai 0 warnanya jadi hitam, dan jika *value* bernilai lebih tinggi menunjukkan variasi warna yang lebih cerah dan lebih banyak variasi baru [27]. Untuk mengkonversi RGB ke HSV dengan persamaan 2.5, 2.6.dan 2.7:

$$H = \tan\left(\frac{3(G-B)}{(R-B)+(R-B)}\right) \quad (2.5)$$

$$S = 1 - \frac{\min(R,G,B)}{v} \quad (2.6)$$

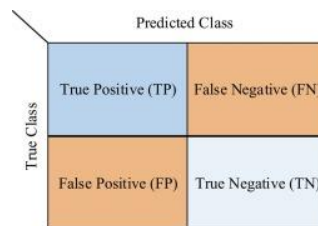
$$V = \frac{R+G+B}{3} \quad (2.7)$$

Keterangan :

- $H$  = Merupakan Hue
- $R, G, B$  = Merupakan warna red (R), green (G) dan blue (B)
- $V$  = Merupakan Value
- $S$  = Merupakan Saturasi
- $\min$  = Merupakan minimal atau nilai terkecil

### 2.2.8 Confusion Matrix

*Confusion matrix* atau dikenal juga dengan *error matrix* untuk memberikan informasi yang membandingkan hasil klasifikasi yang dilakukan oleh sistem dengan hasil klasifikasi sebenarnya ditampilkan pada gambar 2.5.



**Gambar 2. 5 Confusion Matrix [28].**

Pada gambar 2.5 *confusion matrix* berupa bentuk matriks yang menggambarkan asal prediksi model dengan membandingkan prediksi yang benar dan prediksi yang salah terhadap data uji yang sebenarnya. Memiliki istilah yang merupakan hasil dari proses klasifikasi pada *confusion matrix* yaitu :

1. *True Positive (TP)*  
Merupakan data yang bernilai *positif* dan juga hasil yang diprediksi benar sebagai *positif*

2. *True Negative* (TN)

Merupakan data yang bernilai negatif dan juga hasil yang diprediksi benar sebagai *negatif*

3. *False Positive* (FP) – *Type I Error*

Merupakan data yang bernilai *negatif* tapi hasil yang diprediksi sebagai data *positif*

4. *False Negative* (FN) – *Type II Error*

Merupakan data yang bernilai *positif* akan tetapi nilai yang diprediksi sebagai data *negatif* [29].

### 2.2.9 Accuracy

Akurasi atau *Accuracy* merupakan suatu metode pengujian yang didasarkan dengan sejauh mana nilai prediksi yang mendekati nilai sebenarnya (*actual*). Jadi jika nilai sejumlah data yang diperoleh lalu kemudian diklasifikasi dengan benar, maka akan menghasilkan nilai akurasi yang tinggi dari prediksi tersebut. Berikut persamaan 2.8 cara mencari nilai akurasi :

$$\text{Akurasi} = \frac{\text{TP} + \text{TN}}{(\text{TP} + \text{TN} + \text{FP} + \text{FN})} \times 100 \% \quad (2.8)$$

Keterangan : TP merupakan *True Positif*, TN merupakan *True Negatif*, FP merupakan *False Positif* dan FN merupakan *False Negatif*

### 2.2.10 Precision

Presisi atau *precision* adalah metrik evaluasi dalam klasifikasi yang mengukur seberapa akurat prediksi *positif* yang dibuat oleh model. Presisi mencerminkan persentase dari prediksi *positif* model yang sebenarnya benar. Ketika presisi tinggi, model cenderung menghasilkan sedikit *false positive* dan mampu mengidentifikasi kelas *positif* dengan baik. Dengan kata lain, presisi yang tinggi menunjukkan kemampuan model dalam melakukan prediksi positif secara tepat dan minim kesalahan dalam mengklasifikasikan data ke dalam kelas positif. Berikut persamaan 2.9 cara mencari nilai presisi :

$$\text{Presisi} = \frac{\text{TP}}{(\text{TP}+\text{FP})} \times 100 \% \quad (2.9)$$

Keterangan : TP merupakan *True Positif* dan FP merupakan *False Positif*

### 2.2.11 Recall

*Recall* atau juga dikenal sebagai *sensitivitas*, merupakan metrik evaluasi dalam klasifikasi yang mengukur seberapa baik model dapat mengidentifikasi semua jumlah positif yang sebenarnya (*true positives*) dari total jumlah positif yang ada dalam data. Jika *recall* tinggi, ini menandakan bahwa model mampu mengenali sebagian besar jumlah positif dalam data, dan kemungkinan untuk mengabaikan jumlah positif yang sebenarnya (*false negatives*) relatif rendah. *Recall* dapat membantu mengukur seberapa baik model dalam mengidentifikasi semua jumlah positif yang ada dalam data. Berikut persamaan 2.10 cara mencari nilai *recall* :

$$\text{Recall} = \frac{\text{TP}}{(\text{TP}+\text{FN})} \times 100 \% \quad (2.10)$$

Keterangan : TP merupakan *True Positif* dan FN merupakan *False Negatif*

### 2.2.12 F1-Score

*F1-Score* adalah sebuah ukuran evaluasi yang mengkombinasikan antara presisi dan *recall* untuk memberikan gambaran keseluruhan mengenai kinerja suatu model klasifikasi. Penggunaan metrik ini sering terjadi ketika terdapat kecenderungan untuk saling bertentangan antara presisi dan recall, dan jika ingin menemukan titik keseimbangan di antara keduanya. Berikut persamaan 2.11 cara mencari nilai *f1-score* :

$$\text{F1 - Score} = 2 \times \frac{(\text{Recall} \times \text{Precision})}{(\text{Recall} + \text{Precision})} \times 100 \% \quad (2.11)$$

Keterangan : Metrik evaluasi yang menggabungkan antara nilai presisi dan nilai recall untuk memberikan hasil\performa model klasifikasi [30].