



### **Hak cipta dan penggunaan kembali:**

Lisensi ini mengizinkan setiap orang untuk mengubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

### **Copyright and reuse:**

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Pengolahan Citra Digital**

Pengolahan citra digital merupakan istilah umum yang digunakan untuk berbagai macam teknik yang untuk memperoleh keberadaannya dalam melakukan manipulasi dan modifikasi citra dengan segala cara. (Fazarul, 2016). Citra digital adalah citra hasil dari representasi yang dilakukan berdasarkan pendekatan antara sampling dan kuantisasi pada suatu baris dan kolom, sehingga sampling pada sebuah citra akan menyatakan besar atau kecil ukuran piksel (titik). Sedangkan, pada citra dan kuantisasi menyatakan besarnya suatu nilai pada tingkat kecerahan yang akan menyatakan didalam nilai pada tingkat keabuan (grayscale) akan disesuaikan dengan jumlah bit biner yang akan digunakan oleh mesin. (Hanung, 2010).

#### **2.2 Fishers Linear Discriminant**

*Fishers Linear Discriminant* (FLD) adalah penggabungan antara metode pengelompokan pola dengan menggunakan *Principal Component Analysis* (PCA) dan *Linear Discriminant Analysis* (LDA). (Manjunath, 2007). FLD bertujuan untuk menghasilkan matriks sebaran untuk mempermudah klasifikasi dan pengenalan objek. *Distance Classifier* yang umum digunakan yaitu *Euclidean Distance* yang pada proses ini dilakukan klasifikasi setelah pengambilan fitur dari citra selanjutnya dicari jarak terdekat dengan fitur yang ada di dalam database kemudian diperoleh hasil klasifikasi citra yang di uji. (Mrs.Dipti, 2016)

*Fishers Linear Discriminant* memanfaatkan kedua metode pengelompokan pola tersebut yang bertujuan melakukan reduksi dimensi ruang fitur dengan cara

diskriminasi atau klasifikasi dibandingkan dengan menggunakan PCA yaitu dengan cara memaksimalkan rasio pada penyebaran antar kelas tersebut terhadap penyebaran yang ada didalam kelas untuk memudahkan klasifikasi. (Ruei-Sung Lin, 2004). Berikut rumus Fishers Linear Discriminant pada persamaan 2.1, persamaan 2.2 dan persamaan 2.3.

$$J(w) = \frac{|wS_b w^T|}{|wS_w w^T|} \quad (2.1)$$

$$S_B = \sum_{i=1}^n N_i (m_i - m)(m_i - m)^T \quad (2.2)$$

$$S_w = \sum_{i=1}^n \sum_{x \in X_i} (x - m_i)(x - m_i)^T \quad (2.3)$$

Keterangan :

J = Mean

T = Transpose Matriks

W = Matriks Transformasi

S<sub>B</sub> = Between-Class Scatter

S<sub>w</sub> = Within-Class Scatter

n = jumlah kelas

N<sub>i</sub> = jumlah image pada kelas ke-i

X<sub>i</sub> = image ke-k

X = rata – rata total dari keseluruhan image

M<sub>i</sub> = rata-rata image pada kelas ke-i

## 2.3 Laplacian

*Laplacian* adalah untuk mendeteksi tepi dari objek dalam citra yang disebut sebagai operator turunan kedua. Pada turunan kedua atau *Laplacian* digunakan untuk memperbaiki citra dikarenakan memiliki respon yang baik untuk meningkatkan kedetailan. (Irmayani, 2019) *Laplacian* memiliki zerocrossing

(persilangan nol), berupa titik yang terdapat pergantian tanda nilai turunan kedua sedangkan tepi yang landai tidak terdapat persilangan nol. Pada persilangan nol merupakan lokasi tepi yang akurat dan sering digunakan pada tepi yang lebih curam. (Irmayani, 2019) Berikut rumus Laplacian pada persamaan 2.4.

$$\nabla^2 f = \frac{\partial^2 f}{\partial^2 x} + \frac{\partial^2 f}{\partial^2 y} \quad (2.4)$$

Turunan kedua dalam pendekatan pada arah x dan arah y, diperoleh dalam persamaan 2.5 dan persamaan 2.6 sebagai berikut.

$$\frac{\partial^2 f}{\partial^2 x} = f(x + 1, y) + f(x - 1, y) - 2f(x, y) \quad (2.5)$$

$$\frac{\partial^2 f}{\partial^2 y} = f(x, y + 1) + f(x, y - 1) - 2f(x, y) \quad (2.6)$$

## 2.4 Laplacian of Gaussian

*Laplacian* merupakan filter turunan yang bertujuan untuk deteksi area yang memiliki perubahan cepat (rapid changes) seperti tepi (edge) pada citra. Namun, *Laplacian* memiliki kelemahan yaitu sangat rentan atau sensitif pada pengolahan citra yang mendeteksi tepi maka diperlukan cara memperhalus terlebih dahulu digunakannya fungsi *Gaussian*. Dengan demikian dikenal sebagai adanya fungsi turunan baru atau *Laplacian of Gaussian* (LoG). *Laplacian of Gaussian* merupakan salah satu deteksi tepi yang dikembangkan dari turunan kedua digunakan untuk mengekstrak informasi dari tepi gambar yang diekspos dan kemudian gambar yang dibangun dilengkapi dengan informasi yang diperkaya untuk mengekstrak hasil yang baik. (Muhammad Sarif, dkk, 2013). *Laplacian of Gaussian* bertujuan untuk deteksi tepi agar lebih akurat yang khusus pada tepi yang terlihat lebih curam agar citra tidak mengalami gangguan dengan

cara melakukan *filtering* pada gambar. Sehingga, dapat mengurangi gangguan berupa tepi palsu pada citra karena citra yang digunakan akan melakukan proses penyaring terlebih dahulu menggunakan *Gaussian*. Berikut rumus *Laplacian of Gaussian* pada persamaan 2.7.

$$\nabla^2 G(x, y) = \left[ \frac{x^2 + y^2 - 2\sigma^2}{\sigma^4} \right] e^{-\frac{x^2 + y^2}{2\sigma^2}} \quad (2.7)$$

Fungsi  $\nabla^2 G(x, y)$  merupakan turunan kedua dari fungsi *Gaussian* dan melakukan korelasi yang digunakan dalam filter *Gaussian* pada rumus *Laplacian of Gaussian* pada persamaan 2.8 sebagai berikut.

$$h(x, y) = f(x, y) * g(x, y) = \sum_{k=1}^m \sum_{l=1}^n f(k, l, g(x - k, y - l)) \quad (2.8)$$

## 2.5 Confusion Matrix

*Confusion matrix* adalah metode yang sering dipakai untuk melakukan suatu perhitungan akurasi dan mengukur kinerja dari metode klasifikasi dengan konsep data mining atau sistem pendukung keputusan. *Confusion matrix* adalah salah satu tools analitik prediktif yang dapat digunakan untuk menampilkan dan membandingkan suatu nilai *actual* atau disebut nilai sebenarnya dengan nilai hasil prediksi model yang menggunakan suatu hasil dari konsep evaluasi seperti *Accuracy*, *Precision*, *Recall*, dan *F1-Score*. (Christian, 2021)

Pada proses pengukuran kinerja dengan digunakannya *confusion matrix* ada empat nilai yang dihasilkan atau sebagai merepresentasi hasil dari suatu klasifikasi. Dari keempat tersebut merupakan *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN). Nilai *True Positive* (TP) adalah jumlah data yang bernilai positif yang terdeteksi dan diprediksi benar sebagai data positif, sedangkan Nilai *True Negative* (TN) adalah jumlah data yang

bernilai negatif yang terdeteksi dan diprediksi benar sebagai data negatif. Sementara itu, Nilai *False Positive* (FP) adalah jumlah data yang bernilai negatif yang terdeteksi namun diprediksi benar sebagai data positif, sedangkan Nilai *False Negative* (FN) adalah jumlah data yang bernilai negatif terdeteksi namun diprediksi benar sebagai data negatif. (I Gusti Ayu, 2021) Berikut dilihat pada tabel 2.1 *Confusion Matrix*.

Tabel 2.1 *Confusion Matrix*

Class		True Values	
		True	False
Prediction	Positive	TP (True Positive) Correct result	FP (False Positive) Unexpected result
	Negative	TN (True Negative) Correct absence of result	FN (False Negative) Missing result

Berdasarkan dari nilai klasifikasi positif dan negatif pada *Confusion Matrix* dapat diperoleh suatu evaluasi yang digunakan untuk melakukan perhitungan pada nilai akurasi, presisi, recall dan F1-score. Dari keempat evaluasi memiliki manfaat dalam mengukur antara performa dari klasifikasi dan melakukan suatu prediksi.

*Accuracy* merupakan hasil dari presentase pada total data yang telah teridentifikasi dan memiliki nilai untuk melakukan perbandingan data yang sudah melakukan klasifikasi dengan benar pada seluruh data. Berikut rumus *accuracy* pada persamaan 2.9

$$. Accuracy = \frac{TP+TN}{TP+FP+FN+TN} * 100\% \quad (2.9)$$

*Precision* merupakan rasio prediksi positif untuk membandingkan antara True Positive (TP) dengan seberapa banyaknya data yang diprediksi Positif. Berikut rumus *Precision* pada persamaan 2.10.

$$Precision = \frac{TP}{TP+FP} * 100\% \quad (2.10)$$

*Recall* merupakan rasio prediksi positif untuk dibandingkan dengan keseluruhan data yang benar positif pada bagian dari jumlah total data yang positif. Berikut rumus *Recall* pada persamaan 2.11.

$$Recall = \frac{TP}{TP+FN} * 100\% \quad (2.11)$$

*F1-Score* merupakan perbandingan rata-rata antara Precision dengan Recall yang dibobotkan untuk memahami performa dan representasi standar dari nilai-nilai tersebut. Berikut rumus *F1-Score* pada persamaan 2.12.

$$F1 - Score = 2 * (Recall * Precision) / (Recall + Precision) \quad (2.12)$$