

BAB 2 LANDASAN TEORI

Untuk dapat mengimplemetasi *fisherfaces* sebagai pengenalan wajah (*face recognition*) dan *haar cascade* sebagai *face detection*, maka dimanfaatkan beberapa teori yang mendukung penelitian yang sedang dilakukan. Teori yang digunakan adalah sebagai berikut:

2.1 Pengenalan Wajah (Face Recognition)

Pengenalan wajah *Face Recognition* adalah teknologi yang memungkinkan mengenali identitas *biometric* yaitu wajah seseorang melalui citra digital. Tidak seperti *biometric* lainnya, *face recognition* dapat dimanfaatkan untuk kasus yang melibatkan banyak orang sekaligus. Pada umumnya *face recognition* digunakan dengan tujuan keamanan, karena pada wajah memiliki karakteristik yang berbeda.

Hingga saat ini, *face recognition* terus berkembang hingga dapat mengidentifikasi wajah secara 3-D maupun *skin texture*. Terdapat juga kekurangan dari *face recognition*, seperti kurangnya pencahayaan pada citra digital dari kamera maka pemerosesan *face recognition* tidak berjalan dengan optimal. Sistem *Face Recognition* terdapat struktur sebagai berikut:

1. *Face Detection*, merupakan proses mendeteksi posisi wajah dari input visual yang diberikan.
2. *Feature Extraction*, setelah wajah terdeteksi dilakukan ekstraksi agar mendapatkan fitur-fitur penting atau ciri dari wajah.
3. *Face Recognition*, merupakan proses pengenalan wajah berdasarkan dari hasil ekstraksi wajah.

2.2 Smart Home

Smart home dapat mencakup untuk berbagai bidang, seperti keamanan rumah, kenyamanan rumah, maupun kemudahan dalam mengoperasikan suatu barang yang terhubung langsung pada *smart home platform*. *Smart home* pada dasarnya merupakan *multi-user platform* yang dapat mengakses merupakan orang yang tinggal di rumah tersebut. Ketersediaan internet yang sudah tersedia

hampir di seluruh rumah dan personal yang memiliki *smartphone* masing-masing, dapat memungkinkan mempunyai fasilitas *smart home*. Beriringan dengan naiknya kriminalitas pada perumahan sehingga sangat menyulitkan untuk mengatur keamanan pada lingkungan rumah. *Smart home* atau rumah pintar sering juga disebut sebagai *eHome*, merupakan sistem otomatisasi yang canggih seperti mengontrol pencahayaan rumah dan suhu rumah, peralatan multi-media dalam memantau dan mengaktifkan aparat keamanan (alarm dan peringatan), dan banyak fungsi lainnya. [5]

2.3 Kriminalitas

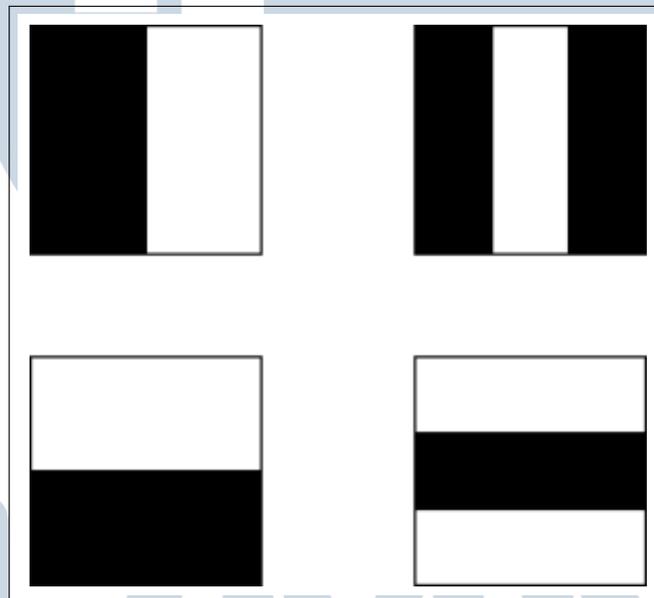
Kejahatan/kriminalitas atau pelanggaran adalah perbuatan yang dapat diancam hukuman sesuai dengan aturan-aturan serta undang-undang yang berlaku. Kriminalitas bisa berbagai bentuk jenis tindakan dan juga merupakan perbuatan yang dapat merugikan secara ekonomis dan psikologis. Dapat diartikan bahwa, tindakan kriminalitas adalah segala sesuatu perbuatan yang, melanggar hukum dan melanggar norma-norma sosial, sehingga masyarakat menentangnya. Tindakan kejahatan/kriminalitas terjadi terdapat beberapa faktor. Faktor penyebab kriminalitas dikelompokkan menjadi faktor dari dalam diri dan faktor luar dari pelaku. Faktor dari dalam diri merupakan faktor biologis dan faktor kualitas dari diri sendiri. Faktor dari luar diri, yaitu lingkungan, ekonomi, bacaan harian, atau film. [12]

2.4 Raspberry Pi

Raspberry Pi merupakan suatu perangkat mini computer yang dikembangkan oleh *Raspberry Pi Foundation*. *Raspberry Pi* ini memiliki processor Quad Core Cortex-A72 ARM (v8) 64-bit SoC speed 1.5GHz, VideoCore VI support OpenGL, dengan RAM LPDDR4 4GB. Terdapat slot untuk Micro-SD Card untuk mempermudah dan penyimpanan jangka panjang, sistem operasi yang akan ditanam ke Micro-SD Card akan memudahkan untuk melakukan pergantian atau ditukar. *Raspberry Pi* dapat menjalankan berbagai sistem operasi seperti Raspbian(sistem operasi linux yang dikembangkan untuk *Raspberry Pi*, Ubuntu, Windows 10 IoT Core, dan distribusi linux lainnya.

2.5 Haar Cascade

Metode *Haar Cascade* menggunakan fitur *Haar-like* yang dicetuskan oleh Paul Viola dan Michael Jones pada tahun 2001 atau lebih sering dikenal sebagai metode Viola-Jones [13]. Metode ini banyak digunakan untuk deteksi objek karena struktur yang tidak kompleks, dari metode ini juga bisa mendapatkan rating deteksi yang cukup tinggi [14]. Fitur *Haar-like* mengklasifikasikan berbagai fitur yang terdapat pada objek dengan posisi, bentuk, dan ukuran yang berbeda. *Haar Cascade* menggunakan fitur *haar-like* yang menghitung nilai area dengan perbedaan nilai kecerahan.



Gambar 2.1. Contoh *Haar-like feature*

Perhitungan nilai fitur menggunakan *Haar-like feature* dengan menghitung perbedaan dari jumlah kecerahan daerah gelap dan terang dalam area tertentu. Integral dari gambar dinyatakan secara matematis, sebagai berikut [14].

$$II(x_1, y_1) = \sum_{x < x_1} \sum_{y < y_1} I(x, y) \quad (2.1)$$

Dimana $II(x_1, y_1)$ merupakan integral gambar dan $I(x_1, y_1)$ adalah gambar orisinal yang dimasukkan. Jumlah kecerahan pada area tertentu menggunakan integral pada gambar didapatkan melalui persamaan berikut.

$$S_{pixel} = P_{RB} - P_{RT} - P_{LB} + P_{LT} \quad (2.2)$$

Dimana:

1. S_{pixel} adalah jumlah dari $pixel$.
2. P_{RB} adalah nilai dari kanan bawah (*right bottom*) dari area integral gambar.
3. P_{RT} adalah nilai dari kanan atas (*right top*) dari area integral gambar.
4. P_{LB} adalah nilai dari kiri bawah (*left bottom*) dari area integral gambar.
5. P_{LT} adalah nilai dari kiri atas (*left top*) dari area integral gambar.

Saat menggunakan fitur *two-rectangle* pada *Haar-like feature*, nilai dari area tertentu dapat dihitung dengan menggunakan enam koordinat dari integral gambar.

2.6 Fisherface

Dasar dari algoritma *fisherface* ini adalah *Fisher's Linear Discriminant* (FLD). FLD ditemukan oleh Robert Fisher pada tahun 1936 untuk klasifikasi taksonomi [15]. *Fisher's Linear Discriminant* atau FLD biasa juga dikenal sebagai *Linear Discriminant Analysis* (LDA), FLD merupakan kembangan yang diperuntukan untuk metode *fisherface* pada pengenalan wajah atau *face recognition*. Tidak hanya menggunakan FLD tetapi juga membutuhkan metode yang dapat mereduksi dimension lainnya yaitu, *Principle Component Analysis* (PCA). PCA mencari kombinasi linear dari fitur yang dapat memaksimalkan total varian yang terdapat dalam data.

Fitur ekstraksi merupakan proses yang memperoleh perbedaan karakteristik dari data citra wajah dengan citra wajah lainnya. Fitur ekstraksi juga merupakan teknik penyelesaian dari *pattern recognition problem* atau permasalahan pengenalan pola sebagai PCA yang digunakan untuk *face recognition* yang sudah dikenalkan oleh Turk dan Pentland pada tahun 1991 [16]. Berdasarkan prosedur *eigenfaces* sebagai berikut.

1. Memperoleh N dari *training images* I_1, I_2, \dots, I_N
2. Mempresentasikan setiap gambar I_i sebagai vektor, setiap gambar memiliki ukuran $p.q \times 1$ atau $p.q = n$
3. Mencari vektor rata-rata Ψ untuk gambar N

$$\Psi = \frac{1}{N} \sum_{i=1}^N I_i \quad (2.3)$$

Dimana:

- Ψ adalah rata-rata vektor gambar
- N adalah hasil nilai dari *images training*
- I adalah gambar yang akan di-*training*

4. Mengurangi rata-rata dari setiap vektor I_i agar mendapatkan satu set vektor Φ_i , dengan mengurangi dari setiap vektor gambar agar dapat membedakan dari setiap gambar.

$$\Phi_i = I_i - \Psi \quad (2.4)$$

Dimana:

- Φ_i adalah hasil dari pengurangan gambar I_i dengan rata-rata.
- I_i adalah vektor dari gambar ke- i
- Ψ adalah rata-rata vektor gambar

5. Mencari Kovarians matriks C :

$$C = AA^T = A^T A, \text{ Dimana } A = [\Phi_1, \Phi_2, \dots, \Phi_N] \quad (2.5)$$

Dimana:

- C membuat matriks lebih sederhana dengan meletakkan satu vektor gambar yang telah dimodifikasi pada setiap kolom.
- A adalah matriks $n \times N$
- A^T adalah matriks $N \times n$
- Jika menghitung menggunakan AA^T maka akan kehabisan *memory* karena akan menghasilkan matriks $n \times N$, maka untuk mendapatkan nilai Kovarians atau C menggunakan $A^T A$.

6. Menghitung *Eigenvector* u dan *eigen value* d dari C

Fisher's Linear Discriminant (FLD) atau *Linear Discriminant Analysis* (LDA) digunakan dalam mencari arah yang efisien untuk membedakan. Mengingat kumpulan N gambar sampel, x_1, x_2, \dots, x_N . Fungsi LDA sebagai berikut.

$$W_{LDA} = \arg \max_w \left| \frac{W^T S_b W}{W^T S_w W} \right| \quad (2.6)$$

$$S_b = \sum_{i=1}^l N_i (\mu^i - \mu) (\mu^i - \mu)^T \quad (2.7)$$

$$S_w = \sum_{i=1}^l \sum_{j=1}^{N_i} (x_j^i - \mu^i) (x_j^i - \mu^i)^T \quad (2.8)$$

Dimana:

- μ adalah rata-rata dari semua sampel.
- N_i adalah jumlah sampel kelas i .
- μ^i adalah rata-rata dari kelas i .
- x_j^i adalah nilai sampel ke- j pada kelas i

2.7 Confusion Matrix

Confusion matrix bisa dikenal sebagai *contingency table* atau sebuah *error matrix*, merupakan tabel yang memvisualisasikan kinerja dari algoritma. Tabel dari *confusion matrix* yang diakronimkan menjadi *true positive*, *false positive*, *true negative*, dan *false negative*[17], dari *confusion matrix* ini juga dapat mengukur untuk *accuracy* ketepatan dari algoritma yang diimplementasi. Pada Tabel 2.1 merupakan gambaran hasil dari *Confusion Matrix*.

Tabel 2.1. Confusion Matrix

		Kelas Sebenarnya	
		Positive	Negative
Kelas Prediksi	Positive	True Positive	False Negative
	Negative	False Positive	True Negative

1. *True Positive* (TP) adalah jumlah data dengan kelas positif yang diklasifikasi positif.
2. *False Positive* (FP) adalah jumlah data dengan kelas positif yang diklasifikasi negatif.
3. *True Negative* (TN) adalah jumlah data dengan kelas negatif yang diklasifikasi negatif.

4. *False Negative* (FN) adalah jumlah data dengan kelas negatif yang diklasifikasi positif.

Setelah menghitung metrik pada Tabel 2.1 untuk mengukur performa proses pembelajaran yang telah dilakukan oleh mesin, metrik evaluasi digunakan untuk mengukur *Accuracy*, *Precision*, *Recall*, dan *F1-Score*.

1. *Accuracy*

Accuracy adalah nilai prediksi (*positive* maupun *negative* dari keseluruhan jumlah data)

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

2. *Precision*

Precision merupakan nilai prediksi *positive* yang dijumlahkan berdasarkan total data yang teridentifikasi kelas *positive*.

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

3. *Recall*

Recall akan menampilkan nilai dari total data yang teridentifikasi sebagai kelas positif dan dibagi oleh keseluruhan sampel yang memiliki label positif

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

4. *F1-Score*

F1-Score adalah perbandingan antara *precision* dengan *recall*

$$F1 - Score = 2 \times \frac{Recall \times Precision}{Recall + Precision} \quad (2.12)$$