

BAB 2 LANDASAN KEPUSTAKAAN

Bab 2 ini berisi penjelasan dan uraian kajian pustaka, teori, konsep ataupun metode dari kepastakaan penelitian lainnya yang berkaitan dengan permasalahan pada penelitian ini. Pada landasan kepastakaan terdapat dasar teori dari berbagai sumber pustaka penelitian lainnya yang menunjang penelitian ini.

2.1 Tinjauan Pustaka

Penelitian mengenai pengenalan emosi berdasarkan ekspresi mikro dilakukan untuk melihat psikologi seseorang masih terus diteliti oleh para pakar terutama pada kasus investigasi seperti penelitian yang dilakukan oleh Andelin & Rusu (2015) menjelaskan tentang emosi yang ditunjukkan saat investigasi dari seorang yang mengidap *psychopathy* dari ekspresi mikro. Investigasi dilakukan pada seorang narapidana pemuda dewasa yang dihukum Karena melakukan kejahatan pembunuhan. Terpidana tersebut diinvestigasi prosedur wawancara yang berlangsung selama 22 menit. Berdasarkan hasil wawancara, terpidana menunjukkan pola perilaku psikopat dengan didominasi oleh ekspresi emosi wajah negatif.

Selain dengan pakar, pengenalan ekspresi mikro kini dapat dilakukan oleh sistem cerdas. Sistem pengenalan emosi berdasarkan ekspresi mikro akan memiliki aplikasi yang luas dalam keamanan nasional, interogasi polisi, dan diagnosis klinis. Mengembangkan sistem seperti itu membutuhkan basis data berkualitas tinggi dengan sampel pelatihan yang memadai. Yan, et al (2014) membuat basis data *micro-expression* yang kemudian basis data tersebut diberi nama CASME. CASME merupakan singkatan dari institusi yang melakukan penelitian ini yaitu *Chinese Academy of Sciences Micro-Expression*. Berisi 195 ekspresi mikro spontan yang difilmkan dibawah 60fps dengan resolusi 1280 x 720 piksel. *Paper* ini juga melakukan *review* terhadap basis data ekspresi mikro yang sudah ada. Pada penelitian selanjurnya dibentuk basis data dengan nama CASME II dengan ukuran lebih kecil yaitu 640 x 480 piksel dan memiliki resolusi sebesar 200 fps.

Dengan tersedianya basis data ekspresi mikro maka akan memudahkan penelitian dibidang ekspresi mikro. Seperti penelitian yang dilakukan oleh Huang, et al (2015) menggunakan metode *spatiotemporal Local Binary Pattern* yang digabungkan dengan *integral projection* agar memperoleh hasil yang efektif dan efisien untuk pemahaman ekspresi mikro. Memperoleh akurasi 59,51% dan 54,88% untuk masing-masing basis data yang digunakan yaitu CASME2 dan SMIC2.

Metode LBP selain untuk pengenalan ekspresi mikro seperti penelitian diatas juga dapat digunakan untuk pengenalan wajah seperti penelitian yang dilakukan oleh Turiyanto, Purwanto, & Dikairono (2014) menerapkan LBP untuk pengenalan wajah dengan membandingkan histogram dari hasil gambar wajah yang telah dilakukan proses *Local Binary Pattern*. Perbandingan dilakukan dengan melihat korelasi antara histogram pada beberapa gambar wajah.

Tabel 2.1 Tinjauan Pustaka

No	Nama Penulis, Tahun dan Judul	Persamaan	Perbedaan	
			Penelitian terdahulu	Rencana penelitian
1	Xiaohua Huang, et al. (2015). Facial Micro-Expression Recognition using Spatiotemporal <i>Local Binary Pattern</i> with Integral Projection	Pengenalan emosi berdasarkan ekspresi mikro wajah menggunakan (Pietikäinen, Hadid, Zhao, & Ahonen, 2011) <i>Local Binary Pattern</i>	Menggunakan metode spatiotemporal <i>Local Binary Pattern</i> yang ditambah dengan Integral Projection	Menggunakan metode <i>Local Binary Patten</i>
2	Wen-Jing Yan, et al. (2014). <i>For micro-expression recognition: Database and suggestions</i>	Pengenalan emosi berdasarkan ekspresi mikro wajah	Hanya membangun Dataset, menghasilkan basis data yang diberi nama CASME	Membangun sistem yang dapat mengenali emosi dari ekspresi mikro
3	Wen-Jing Yan, et al. (2014). <i>CASME II: An Improved Spontaneous Micro-Expression Basis data and the Baseline Evaluation</i>	Pengenalan emosi berdasarkan ekspresi mikro wajah	Membangun basis data yang merupakan penelitian lanjutan dari penelitian yang selanjutnya, menghasilkan basis data yang diberi nama CASME II	Membangun sistem yang dapat mengenali emosi dari ekspresi mikro
4	Turiyanto, et al. (2014). Penerapan Teknik Pengenalan Wajah Berbasis Fitur <i>Local Binary Pattern</i> pada Robot Pengantar Makanan	Menggunakan metode Haar-Cascade untuk deteksi wajah dan menggunakan metode <i>Local Binary Pattern</i> .	Pengenalan wajah	Pengenalan ekspresi mikro pada wajah

Tabel 2.1 (lanjutan)

No	Nama Penulis, Tahun dan Judul	Persamaan	Perbedaan	
			Penelitian terdahulu	Rencana penelitian
5	Emanuel I. Andelin, Alina S. Rusu. (2015). Investigation of facial micro-expressions of emotions in psychopathy – a case study of an individual in detention	Pengenalan emosi berdasarkan ekspresi mikro wajah	Pengenalan emosi dilakukan secara tradisional, yaitu menggunakan bantuan pakar	Pengenalan emosi dilakukan secara otomatis menggunakan sistem yang akan dirancang

2.2 Dasar Teori

2.2.1 Ekspresi Mikro

Ekspresi mikro adalah gerakan wajah singkat yang mengungkapkan emosi bahwa seseorang mencoba untuk menyembunyikan. Pada tahun 1969, Ekman menganalisis video wawancara dari pasien terserang depresi yang mencoba bunuh diri dan menemukan ekspresi mikro. Ekspresi mikro ditampilkan dalam durasi yang pendek. Durasi yang berlaku adalah 0,5 detik. Selain itu, ekspresi mikro biasanya terjadi dengan intensitas rendah.

Ekspresi wajah dianggap lebih dari reaksi emosional belaka. Ekspresi wajah dikenal untuk mengkoordinasikan interaksi sosial melalui fungsi informatif dan memotivasi (Keltner & Kring, 1998).

Menurut teori Darwin tentang seleksi alam (1872), emosi bersifat universal dalam hal memiliki sejenis pola ekspresi dan fungsi pada tingkat komunikasi antar manusia, terlepas dari bias budaya. Emosi yang universal didefinisikan oleh Keltner, et al (2003) adalah sebagai berikut: terkejut, bahagia, marah, sedih, takut, jijik, merendakan.



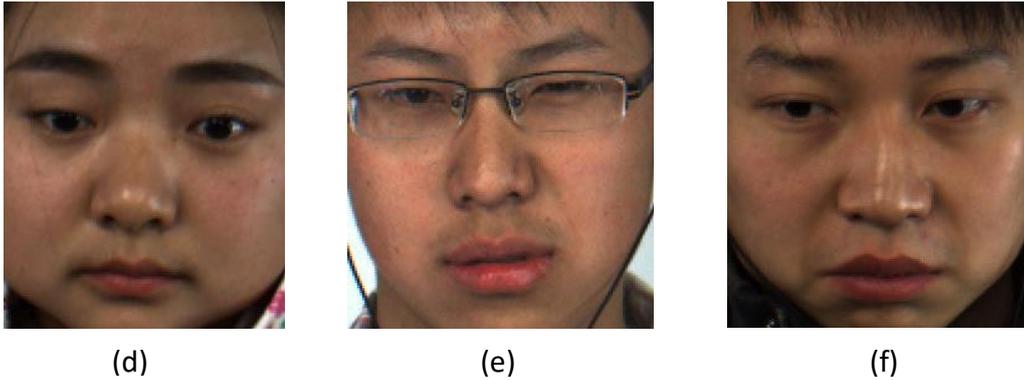
(a)



(b)



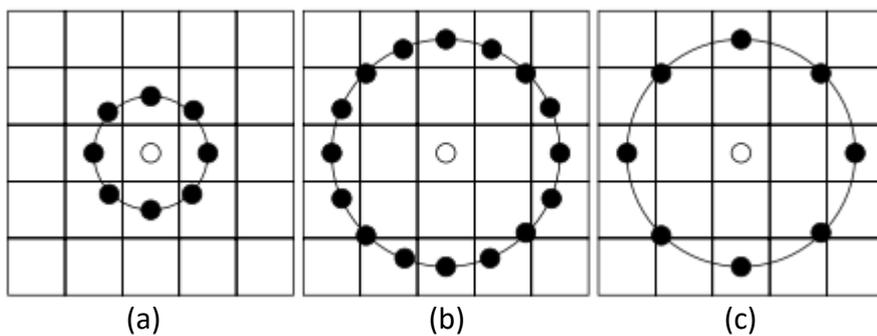
(c)



Gambar 2.1 Ekspresi mikro berbagai emosi: (a)bahagia, (b)sedih,(c)takut, (d)terkejut, (e)jijik, (f)represi

2.2.2 Local Binary Pattern

Local Binary Pattern pertama kali diperkenalkan oleh Ojala et al, didefinisikan sebagai ukuran tekstur *grayscale* yang invarian, disebut invarian Karena hamper tidak dipengaruhi oleh pencahayaan yang berbeda. LBP ampuh untuk mendeskripsikan suatu tekstur, mempunyai daya pembeda yang akurat, dan juga memiliki toleransi terhadap perubahan *grayscale* yang monotonik (Turiyanto, Purwanto, & Dikairono, 2014).



Gambar 2.2 Jarak dan banyak piksel tetangga yang terpilih (a) $R=1; P=8$, (b) $R=2; P=16$, (c) $R=2; P=8$

Sumber: Pietikäinen, et al. (2011)

Operator LBP adalah operator yang merubah gambar menjadi gambar berlabel bilangan bulat yang menggambarkan tampilan *small-scale*. Label yang sering digunakan untuk menggambarkan statistic LBP adalah histogram (Pietikäinen, Hadid, Zhao, & Ahonen, 2011). Istilah *Local Binary Pattern* dimaksudkan sebagai representasi intensitas piksel ketetanggaannya dari piksel yang sedang diolah (Turiyanto, Purwanto, & Dikairono, 2014). LBP memiliki nilai jarak untuk

menentukan ketetangaan yang akan dipilih, jarak biasa disimbolkan dengan R dan banyaknya tetangga yang dipilih disimbolkan dengan P . Besar jarak dan ketetangaan yang dipilih digambarkan dalam Gambar 2.2 (Pietikäinen, Hadid, Zhao, & Ahonen, 2011).

Berikut logika untuk *Local Binary Pattern*, untuk setiap piksel p , buat *windowing* 8-bit $P_1 P_2 P_3 P_4 P_5 P_6 P_7 P_8$, dimana $P_i = 0$ jika nilai dari i lebih kecil dari titik pusat (piksel yang sedang diolah) dan bernilai 1 sebaliknya. Fungsi LBP didefinisikan sebagai berikut (Turiyanto, Purwanto, & Dikairono, 2014):

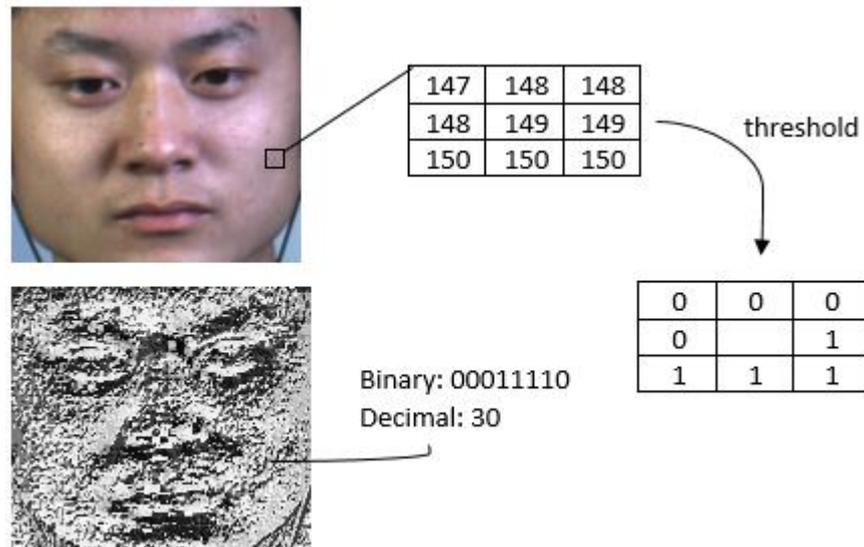
$$LBP_{P,R}(x_c, y_c) = \sum_{p=0}^{P-1} s(g_p - g_c) 2^p \quad (2.1)$$

Fungsi $s(x)$ didefinisikan sebagai berikut:

$$s(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases} \quad (2.2)$$

Keterangan:

- P : jumlah piksel tetangga
- R : nilai radius
- g_c : nilai piksel x, y
- g_p : nilai piksel tetangga

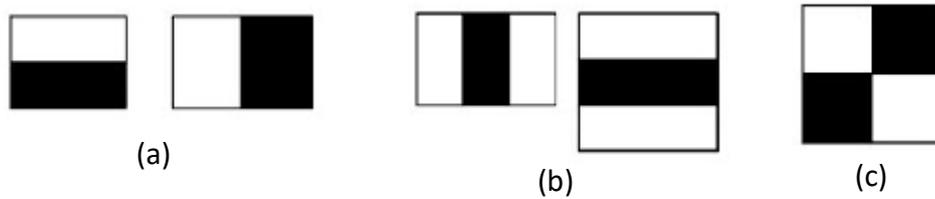


Gambar 2.3 Proses Kalkulasi piksel LBP

2.2.3 Haar Cascade

Haar Cascade adalah metode yang biasanya dipakai untuk pendeteksian objek. Bekerja dengan mendeteksi fitur-fitur yang telah ditetapkan dan dilakukan pencarian fitur-fitur tersebut terhadap gambar yang ingin dideteksi. Setiap fitur adalah suatu nilai yang diperoleh dari penjumlahan daerah persegi putih dikurangi

dengan penjumlahan daerah persegi hitam (Turiyanto, Purwanto, & Dikairono, 2014). Fitur-Fitur yang digunakan diperlihatkan pada Gambar 2.4.



Gambar 2.4 Fitur pada *Haar Cascade OpenCV*
(a)Edge Features, (b)Line Fitur, (c)Four-Rectangle feature

Sumber: https://docs.opencv.org/3.3.0/d7/d8b/tutorial_py_face_detection.html

Deteksi wajah juga bisa dilakukan oleh metode ini dengan *OpenCV* menggunakan *Haar-cascade classifier*. Ketika terdapat sebuah gambar *face detector* ini akan men-*scanning* setiap lokasi gambar tersebut dan mengklasifikasikannya sebagai wajah atau bukan wajah. Klasifikasi ini menggunakan sebuah permasalahan yang tetap. Untuk menjalankan deteksi wajah pada *OpenCV* digunakan *classifier* yaitu berupa file XML yang biasanya digunakan adalah "*haarcascade_frontalface_default.xml*" (Turiyanto, Purwanto, & Dikairono, 2014).

2.2.4 Histogram

Histogram adalah frekuensi kemunculan suatu nilai intensitas piksel melalui grafik (bar). Histogram banyak digunakan untuk beberapa teknik pemrosesan citra pada domain spasial, seperti perbaikan citra, segmentasi, klasifikasi. Terdapat 2 jenis histogram, yaitu histogram global dan histogram lokal. Jenis histogram ditentukan oleh cakupan daerah citra yang dikenai perhitungan akumulasi. Histogram global merupakan histogram yang memiliki cakupan akumulasi perhitungan pada seluruh daerah citra. Sedangkan histogram lokal merupakan histogram yang memiliki cakupan akumulasi perhitungan pada luasan tertentu citra. Histogram global tidak cocok untuk citra yang kompleks, pendekatan yang lebih baik adalah menggunakan histogram lokal karena memberikan informasi obyek yang lebih spesifik (Widodo dan Harjoko, 2015). Histogram dapat digunakan untuk membandingkan gambar karena tingkat kedetailan dan korelasi sangat efektif (Turiyanto, Purwanto, & Dikairono, 2014).

2.2.5 Klasifikasi *K-Nearest Neighbor (K-NN)*

K-NN merupakan salah satu metode klasifikasi terkenal dalam *machine learning*. Metode ini menggunakan fungsi jarak untuk melakukan klasifikasi, contohnya jarak *Euclidean* untuk membandingkan sampel data lebih dekat dengan sampel data lain pada kelas tertentu. Metode ini cukup sederhana tetapi memiliki

kecepatan dalam konvergensi yang relatif tinggi karena harus mengakses seluruh data latih setiap kali data uji baru akan diklasifikasikan. Berikut tahapan Algoritme K-NN (Priambodo, Dewi, & Triwiratno, 2015):

1. Mendefinisikan nilai k
Nilai k adalah nilai yang dipakai untuk menentukan berapa banyak tetangga terdekat pada data latih terhadap data *input*.
2. Penentuan jarak antara data *input* dengan semua data latih
Jarak digunakan untuk memilah individu yang sama ke dalam satu kelompok.
3. Urutkan data berdasarkan hasil penentuan jarak.
Data *input* yang telah dihitung jaraknya dengan data latih kemudian diurutkan dari yang paling kecil.
4. Bentuk kelompok berdasarkan nilai dengan ketetanggaan terdekat.
Dari hasil pengurutan data berdasarkan jarak kemudian dipilih sebanyak k untuk dijadikan kelompok.
5. Pilih nilai dengan kemunculan paling sering.
Dari kelompok yang telah dibentuk dipilih nilai atau kelas yang sering muncul sebagai prediksi klasifikasi data *input*.

2.2.6 Jarak *Euclidean*

Jarak *Euclidean* adalah metode untuk pengukuran jarak dari dua atau lebih vektor yang paling sering digunakan (Hartono & Lusiana, 2017). Pengukuran jarak *Euclidean* menggunakan perhitungan akar dari kuadrat dua atau lebih perbedaan vektor. Metode ini dapat digunakan untuk mendeteksi tingkat kemiripan citra dengan cara mengisi dua atau lebih vektor yang berbeda dengan nilai fitur pada citra yang ingin dideteksi tingkat kemiripannya. Metode ini menggunakan jarak paling minimum untuk menentukan kemiripan suatu citra. Jarak *Euclidean* dari dua vektor dihitung dengan Persamaan 2.3 (Nugraheny, 2015):

$$d(x, y) = \sqrt{\sum_{r=1}^n (x_i - y_i)^2} \tag{2.3}$$

Keterangan:

$d(x, y)$: jarak *Euclidean* antara x_i dengan y_i
 x_i : data pada x ke- i yang akan dilakukan perhitungan

y_i : data pada y ke- i yang akan dilakukan perhitungan

2.2.7 Jarak *Manhattan*

Jarak *Manhattan* adalah metode untuk pengukuran jarak dari dua atau lebih vektor. Pengukuran jarak *Manhattan* menggunakan perhitungan nilai absolut dari pengurangan dua atau lebih vektor. Metode ini dapat digunakan untuk mendeteksi tingkat kemiripan citra dengan cara mengisi dua atau lebih vektor yang berbeda dengan nilai fitur pada citra yang ingin dideteksi tingkat kemiripannya. Metode ini menggunakan jarak paling minimum untuk menentukan kemiripan suatu citra. Pencarian citra dengan menggunakan jarak *Manhattan* baik untuk citra yang kaya akan tekstur (Hartono & Lusiana, 2017). Jarak *Manhattan* dari dua vektor dihitung dengan Persamaan 2.4 (Nugraheny, 2015):

$$d(x, y) = \sum_{r=1}^n ||x_i - y_i|| \quad (2.4)$$

Keterangan:

$d(x, y)$: jarak *Euclidean* antara x_i dengan y_i
 x_i : data pada x ke- i yang akan dilakukan perhitungan
 y_i : data pada y ke- i yang akan dilakukan perhitungan

2.2.8 Jarak *Chebyshev*

Jarak *Chebyshev* adalah metode untuk pengukuran jarak dari dua atau lebih vektor dari nilai yang paling maksimal. Pengukuran jarak *Chebyshev* menggunakan perhitungan nilai perbandingan dua atau lebih vektor dan diambil nilai paling maksimal dari vektor-vektor yang dibandingkan. Jarak *Chebyshev* dari dua vektor dihitung dengan Persamaan 2.5 (Ponnmoli & Selvamuthukumar, 2014):

$$d(x, y) = \max_{i=1,2,\dots,n} (|x_i - y_i|) \quad (2.5)$$

Keterangan:

$d(x, y)$: jarak *Euclidean* antara x_i dengan y_i
 x_i : data pada x ke- i yang akan dilakukan perhitungan
 y_i : data pada y ke- i yang akan dilakukan perhitungan

2.2.9 Akurasi

Akurasi adalah nilai ukur untuk validasi pada suatu metode. Akurasi adalah perbedaan antara harapan hasil tes dan nilai referensi yang diterima. Akurasi biasanya dinyatakan sebagai presentase. Akurasi dapat ditentukan melalui

metode penambahan baku (*standard addition method*). Dalam metode penambahan baku selisih dari jumlah keseluruhan data dan jumlah salah dibandingkan dengan jumlah keseluruhan data (Riyanto, 2014). Perhitungan metode penambahan baku dapat dihitung dengan Persamaan 2.6:

$$\% \text{ penambahan baku} = \frac{C1-C2}{C1} \quad (2.6)$$

Keterangan:

C1: jumlah data

C2: jumlah data salah/tidak sesuai harapan