**Pengenalan Ekspresi Wajah dengan metode *Local Directional Pattern* dan *Support Vector Machine***

***Face Expression Recognition using Local Directional Pattern method and Support Vector Machine***

**Tugas Akhir**

Diajukan untuk memenuhi sebagian dari syarat

untuk memperoleh gelar Sarjana Teknik

Telkom School of Computing

Teknik Informatika

Telkom University

**Satryo Bakuh Sem Matabei**

**1103104184**



**Telkom School of Computing**

**Teknik Informatika**

**Telkom University**

**Bandung**

**2014**

**Lembar Pernyataan**

Dengan ini saya menyatakan bahwa Tugas Akhir dengan judul “**Pengenalan Ekspresi Wajah dengan metode Local Directional Pattern dan Support Vector Machine**” beserta seluruh isinya adalah benar-benar karya saya sendiri dan saya tidak melakukan penjiplakan atau pengutipan dengan cara-cara yang tidak sesuai dengan etika keilmuan yang berlaku dalam masyarakat keilmuan. Atas pernyataan ini, saya siap menanggung resiko/sanksi yang dijatuhkan kepada saya apabila kemudian ditemukan adanya pelanggaran terhadap etika keilmuan dalam karya saya ini, atau ada klaim dari pihak lain terhadap keaslian karya saya ini.

Bandung, 16 Juli 2014

Yang membuat pernyataan,

Satryo Bakuh Sem Matabei

**Lembar Pengesahan**

**Pengenalan Ekspresi Wajah dengan metode *Local Directional Pattern* dan *Support Vector Machine***

***Face Expression Recognition using Local Directional Pattern method and Support Vector Machine***

##### Satryo Bakuh Sem Matabei

##### 1103104184

Tugas akhir ini telah diterima dan disahkan untuk memenuhi sebagian dari syarat untuk memperoleh gelar Sarjana Teknik  
Telkom School of Computing

Teknik Informatika

Telkom University

Bandung, 16 Juli 2014

Menyetujui

Pembimbing I Pembimbing II

Tjokorda Agung Budi Wirayuda, ST. MT Febryanti Sthevanie, ST. MT

NIP: 06830333-1 NIP: 14881300-2

# Abstrak

Ekspresi wajah merupakan bentuk respon yang kelihatan untuk menunjukan tanggapan seseorang dalam berkomunikasi. Dengan ekpresi wajah seseorang akan mengetahui respon lawan bicara secara non-verbal teradap apa yang dia ucapkan. Oleh karena itu kemampuan mengenali ekpresi wajah merupakan salah satu bentuk kompetensi sosial.

Tugas akhir ini merupakan penelitian terhadap kemampuan *Local Directional Pattern* dan *Support Vector Machine* dalam melakukan proses pengenalan ekpresi wajah secara *real-time*. *Local Directional Pattern* menggunakan pendekatan berbasis rupa sehingga lebih cocok untuk sistem waktu nyata. Hal ini dikarenakan pendekatan berbasis rupa memiliki komputasi yang lebih sederhana dibandingkan pendekatan berbasis fitur geometri.

Hasil penelitian menunjukan bahwa sistem memiliki kemampuan akurasi 88.92% dalam mendeteksi ekpresi wajah manusia dengan parameter nilai respon kernel arah terbaik ( untuk *Local Directional Pattern* adalah 3, jumlah region yang terbaik untuk membagi citra wajah adalah . Hasil penelitian juga menunjukan bahwa kernel pemetaan pada *Support Vector Machine* yang cocok untuk sistem waktu nyata adalah linear kernel.

**Kata Kunci**: pengenal ekpresi wajah otomatis, local directional pattern, support vector machine, waktu nyata, kernel arah

# Abstract

Facial expressions is a form of response that appear to show a person’s response in communication. With face expressions, someone will know the other person’s response in non-verbal about what he or she said. Therefore, the ability to recognize the face expressions is one of the social competence form.

This final project is a study about the ability of Local Directional Pattern and Support Vector Machine in the process of face expressions recognition in real time. Local Directional Pattern using the approaches based on appearance so that is more suitable for real time system. This is because the approaches based on appearance has a simpler computation than approaches based on geometric features.

The result of study show the system reach 88.92% of accuracy ability in detecting the human face expression with ​​best direction kernel response value parameter (k) for Local Directional Pattern is 3 and the best number of region to divide the face image is . The result of study also show the mapping of kernel in Support Vector Machine that suitable for real time system is the linear kernel.

**Keywords**: automatic face expressions recognition, local directional pattern, support vector machine, real time, kernel direction

# Lembar Persembahan

Pada bagian ini penulis ingin mengucapkan banyak terima kasih kepada semua pihak yang telah memberikan bantuan secara langsung maupun tidak langsung. Karena tanpa bantuan penulis tidak dapat melakukan penelitian ini dengan baik. Dengan segala kerendahan hati penulis ingin mengucapkan terima kasih kepada:

1. Tuhan Yang Maha Esa yang selalu menyertai dan memberikan penulis kesehatan serta pengetahuan untuk menyelesaikan tugas akhir ini.
2. Kedua orang tua penulis yang walaupun jauh selalu memberikan dukungan moril maupun materi. Terima kasih karena tidak pernah memberikan tekanan terhadap penulis dengan memberikan pertanyaan tentang pendamping hidup. Terima kasih buat Ibu yang selalu dengan sabar menanyakan kemajuan tugas akhir.
3. Bapak Tjokorda Agung Budi Wirayuda selaku pembimbing I, yang telah banyak meluangkan waktunya untuk melakukan diskusi dan arahan yang sangat berguna untuk menyelesaikan tugas akhir ini.
4. Ibu Febryanti Sthevanie, ST sebagai pembimbing II, yang juga telah meluangkan waktunya untuk melakukan diskusi dan arahan yang sangat berguna untuk menyelesaikan tugas akhir ini, khususnya saat penulis berusaha menyelesaikan tugas akhir I.
5. Bapak Gandeva Bayu Satria yang merupakan dosen wali penulis yang sebelumnya, yang telah menyempatkan waktunya untuk mengurus proses perwalian penulis. Terima kasih juga atas dukungan dan bantuan yang diberikan selama penulis menyelesaikan tugas akhir ini.
6. Ibu Shinta Yulia Puspitasari yang merupakan dosen wali penulis, yang telah bersedia untuk membantu penulis dalam mengurus keperluan-keperluan tugas akhir ini.
7. Semua dosen Universitas Telkom atas pelajarannya baik secara materi maupun moral.
8. Semua anak Surfivers, IF-34-05 (Arif, Gugun, Doni, Aul, Dika, Miko, Teon, Damar, Yayat, Rio, Alif, Okta, Cendol, Adzan, Tusti, Selvi, Elis, Lia, Ocha, Vinca, Heru, Merlin, Kartika, Cantika, Vero, Jeje, Dery, Agung, Pandu, Niila, Ilham, Yoel, Nyoman) , yang memberikan suka duka dari mulai masuk perkuliahan hingga akhir perkuliahan, khusunya untuk Taufik, Adit dan Riyad yang selalu mengisi kesendirian dan kekosongan penulis sehingga menjauhkan penulis dari kegiatan yang berbahaya.
9. Teman-teman penulis di ProClub yang menemani penulis setiap harinya seperti Kak Agung, Kendy, Byan, Ryan, Wahyu, Reza, Rama, Sendi, Arif, Atoy dan juga khususnya Widya yang selalu memaksa penulis untuk mengerjakan tugas akhir ini. Juga pembimbing ProClub Kak Dody Qori Utama.

# Kata Pengantar

Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa yang telah memberikan rahmat dan karunianya, sehingga penulis dapat menyelesaikan tugas akhir penulis yang berjudul “**Pengenalan Ekpresi Wajah dengan metode *Local Directional Pattern* dan *Support Vector Machine***” sebagai salah satu syarat untuk menyelesaikan program Sarjana Teknik Informatika Universitas Telkom.

Dengan selesainya tugas akhir ini tidak terlepas dari bantuan banyak pihak yang telah memberikan masukan-masukan kepada penulis. Untuk itu penulis mengucapkan terima kasih.

Penulis menyadari bahwa tugas akhir ini masih jauh dari sempurna. Oleh karena itu, penulis sangat mengharapkan kritik dan saran yang membangun dari semua pihak. Penulis berharap tugas akhir ini dapat dikembangkan dan dilengkapi sehingga dapat berguna di masyarakat.

# Daftar Isi

[Abstrak iv](#_Toc407982152)

[Abstract v](#_Toc407982153)

[Lembar Persembahan vi](#_Toc407982154)

[Kata Pengantar vii](#_Toc407982155)

[Daftar Isi viii](#_Toc407982156)

[Daftar Gambar x](#_Toc407982157)

[Daftar Tabel xi](#_Toc407982158)

[Daftar Istilah xii](#_Toc407982159)

[1. Pendahuluan 1](#_Toc407982160)

[1.1 Latar Belakang 1](#_Toc407982161)

[1.2 Perumusan Masalah 2](#_Toc407982162)

[1.3 Batasan Masalah 2](#_Toc407982163)

[1.4 Tujuan 2](#_Toc407982164)

[1.5 Metodologi penyelesaian masalah 2](#_Toc407982165)

[1.6 Hipotesa 3](#_Toc407982166)

[1. Gambar wajah, dapat diklasifikasikan kedalam 7 kategori, yaitu: marah, senang, terkejut, jijik, takut, sedih, dan netral. 3](#_Toc407982167)

[2. Hasil klasifikasi yang didapatkan oleh sistem, memiliki akurasi diatas percobaan sebelumnya, yaitu LDP-SVM, dengan akurasi diatas 85%. 3](#_Toc407982168)

[2. Dasar Teori 4](#_Toc407982169)

[2.1 Ekspresi Wajah 4](#_Toc407982171)

[2.2 Konsep dasar Citra Digital 4](#_Toc407982172)

[2.2.1 Ruang Warna 4](#_Toc407982173)

[2.2.2 Pra-pengolahan Citra 6](#_Toc407982174)

[2.2.3 Video Digital 6](#_Toc407982175)

[2.3 Ekstrasi Fitur 7](#_Toc407982176)

[2.3.1 Haar cascade Face Detector 7](#_Toc407982177)

[2.3.2 LDP 11](#_Toc407982178)

[2.4 Klasifikasi Fitur 12](#_Toc407982179)

[2.4.1 Support Vector Machines 13](#_Toc407982180)

[2.5 Akurasi 17](#_Toc407982181)

[2.6 Kecepatan 17](#_Toc407982182)

[3. Analisa dan Perancangan Sistem 18](#_Toc407982183)

[3.1 Gambaran Umum Sistem 18](#_Toc407982185)

[3.2 Analisa Kebutuhan Sistem 19](#_Toc407982186)

[3.2.1 Spesifikasi Perangkat Keras 19](#_Toc407982187)

[3.2.2 Spesifikasi Perangkat Lunak 19](#_Toc407982188)

[3.2.3 Analisa kebutuhan data 19](#_Toc407982189)

[3.2.4 Analisa Masukan dan Keluaran Sistem 20](#_Toc407982190)

[3.3 Perancangan Sistem 20](#_Toc407982191)

[3.3.1 Deteksi Wajah 21](#_Toc407982192)

[3.3.2 Pra-pengolahan Citra 22](#_Toc407982193)

[3.3.3 Pengenalan Ekpresi 23](#_Toc407982194)

[4. Analisa dan Pengujian Aplikasi 28](#_Toc407982195)

[4.1 Gambaran Umum Pengujian 28](#_Toc407982200)

[4.1.1 Skenario Pengujian 29](#_Toc407982201)

[4.1.2 Hasil Pengujian dan analisis 31](#_Toc407982202)

[4.2 Rangkuman Analisa 34](#_Toc407982203)

[5. Kesimpulan dan Saran 35](#_Toc407982204)

[5.1 Kesimpulan 35](#_Toc407982206)

[5.2 Saran 35](#_Toc407982207)

[Daftar Pustaka 36](#_Toc407982208)

[Lampiran A : Detail Hasil Pengujian 37](#_Toc407982209)

[1.1 Hasil uji skenario 1 37](#_Toc407982210)

[1.2 Hasil uji skenario 2 37](#_Toc407982211)

[1.3 Hasil uji skenario 3 37](#_Toc407982212)

[1.4 Hasil uji skenario 4 38](#_Toc407982213)

[1.5 Hasil uji skenario 5 38](#_Toc407982214)

# Daftar Gambar

Gambar 2‑1 Citra RGB 5

Gambar 2‑2 Citra Grayscale 5

Gambar 2‑3 Haar Features 8

Gambar 2‑4 Perubahan Citra Asli Menjadi Integral Image 9

Gambar 2‑5 Feature Pool 9

Gambar 2‑6 Algoritma Gentle AdaBoost 10

Gambar 2‑7 Cascade Classifier 10

Gambar 2‑8 Posisi respon tepi dan posisi bit biner LDP 11

Gambar 2‑9 Perhitungan kode LDP dengan 11

Gambar 2‑10 Matriks kernel 8 arah 12

Gambar 2‑11 LDP Descriptor 12

Gambar 2‑12 Hyperplane (bidang pemisah) 13

Gambar 2‑13 Transformasi dari vektor input ke feature space 14

Gambar 2‑14 Transformasi untuk data yang tidak dapat dipisahkan secara linier 15

Gambar 2‑15 Contoh klasifikasi dengan metode One-against-rest 16

Gambar 2‑16 Contoh klasifikasi dengan metode One-against-one 16

Gambar 3‑1 Gambaran umum sistem 18

Gambar 3‑2 Contoh Data Latih CK 20

Gambar 3‑3 *Flowchart* sistem 21

Gambar 3‑4 Citra RGB dikonversi ke Grayscale 22

Gambar 3‑5 *Cropping* daerah wajah 23

Gambar 3‑6 Hasil *Histogram Equalization* 23

Gambar 3‑7 Matriks Citra 24

Gambar 3‑8 Konvolusi dengan kernel 8 arah 24

Gambar 3‑9 Hasil konvolusi 25

Gambar 3‑10 Pengurutan hasil konvolusi 25

Gambar 3‑11 Matriks baru 25

Gambar 4‑1 Proses Utama AFER 28

Gambar 4‑2 Pembagian data 29

# Daftar Tabel

Tabel 2‑1 Contoh kombinasi nilai RGB 5

Tabel 2‑2 Karateristik video digital 6

Tabel 2‑3 Contoh 4 SVM biner dengan metode One-against-all 15

Tabel 2‑4 Contoh 6 SVM biner dengan metode One-against-one 16

Tabel 3‑1 Daftar jumlah data citra tiap ekpresi 20

Tabel 3‑2 Parameter DetectMultiscale 22

Tabel 3‑3 Parameter SVM 26

Tabel 4‑1 Komposisi data 29

Tabel 4‑2 Detail Skenario 30

Tabel 4‑3 Skenario 1 31

Tabel 4‑4 Skenario 2 32

Tabel 4‑5 Skenario 3 32

Tabel 4‑6 Skenario 4 33

Tabel 4‑7 Skenario 5 33

Tabel 4‑8 Skenario 6 34

# Daftar Istilah

*False alarm* Sebuah deteksi yang salah. Deteksi yang seharusnya bukan area suatu objek.

*Hit rate* Rasio terdeteksinya suatu objek dari semua objek yang ada

*Histogram* Representasi persebaran data dalam bentuk grafik

*Kirsch edge masks* Kernel arah untuk mendeteksi sudut

Feature space Dimensi ruang untuk memetakan titik-titik fitur citra

# Pendahuluan

## Latar Belakang

Studi expresi wajah adalah bidang yang telah dipelajari sejak era Aristoteles. Studi ini mempelajari sifat seseorang melalui expresi wajah. Aspek ini dinilai mampu menyampaikan 55% dari pesan. Dibandingkan dengan bahasa (7%), serta intonasi (38%)[7].

Automatic facial expression analisis, ialah suatu permasalahan yang menantang dan menarik, dengan dampak aplikasi yang berdampak di banyak area, seperti interaksi manusia-komputer, teori psikologi manusia, dan study simulasi emosi.[4] Selain itu, expresi wajah juga merupakan salah satu metode komunikasi yang sangat powerful, natural[5]. Beberapa metode extrasi-ciri berbasis histogram konvolusi pada area muka telah banyak diupayakan seperti *Local Phase Quantization* yang menggunakan *Short-Term Fourier Transform*, *Local Binary Pattern* yang membandingkan intensitas suatu pixel dengan pixel tetangga-tetangganya, dan *Local Directional Pattern* yang mengaplikasikan kirschedge matrix sebagai basis dari konvolusinya yang dinyatakan sebagai teknik state-of-the-art yang dapat memberikan hasil paling memuaskan[1]. Hasil dari preprocessing menggunakan *local directional pattern* tersebut, berikutnya akan diklasifikasikan dengan menggunakan metode supervised learning yang disebut neural network ke dalam 7 kategori yaitu senang, sedih, marah, jijik, takut, terkejut dan netral.

Neural network merupakan metode *supervised learning* untuk klasifikasi yang cukup populer sejak tahun 1960-an. Namun sempat sering dinyatakan sering *outperformed* oleh metode yang lebih simple seperti SVM. Namun beberapa tahun terakhir, sebuat metode yang mengutilisasi arsitektur *neural network* yang disebut *deep-learning* seperti *deep neural network*, *convolutional deep neural networks*, dan d*eep belief network*, sering menghasilkan produk *state-of-the-art* untuk berbagai bidang riset seperti *computer vision*, *automatic speech recognition*, *natural language processing*, dan *music/audio signal recognition*[11]*.* Oleh karena itu, pada tugas akhir ini, penulis ingin mencoba mengkombinasikan kedua teknik diatas dengan harapan didapatkan system yang dapat mengklasifikasikan expresi wajah dengan akurasi lebih dari atau sama dengan pendekatan yang menggunakan SVM yaitu diatas 85%.

## Perumusan Masalah

Berdasarkan masalah yang telah diutarakan di latar belakang, masalah yang harus dijawab oleh penulis ialah:

1. Bagaimana bentuk model learning yang cocok untuk diterapkan pada sistem.(jumlah hidden layer, jumlah input node, jumlah output node)
2. Seberapa tinggi tingkat *accuracy* yang dapat dicapai oleh sistem.
3. Pre-Processing apa saja yang perlu dilakukan oleh system sebelum system melakukan klasifikasi gambar.
4. Bagaimana performansi neural network yang digunakan untuk mengklasifikasikan expressi wajah pada gambar.

## Batasan Masalah

Batasan masalah dalam penelitian ini adalah:

1. Data input merupakan file gambar.
2. Gambar yang menjadi data input dipastikan memiliki gambar wajah tampak depan meliputi kedua mata, dan mulut.
3. Data uji yang akan digunakan ialah data uji *Cohn-Kanade yang* dapat diakses melalui <http://www.consortium.ri.cmu.edu/ckagree/> dan *Japanese Female Facial Expression* (JAFFE) dapat diakses melalui situs <http://www.kasrl.org/jaffe_info.html>

## Tujuan

Tujuan yang ingin dicapai dalam penelitian ini adalah :

1. Menganalisis proses pengenalan ekpresi wajah menggunakan LDP pada ekstrasi fitur dan ANN pada klasifikasi fitur

## Metodologi penyelesaian masalah

Dalam penelitian ini, penulis menggunakan pendekatan sistematis untuk pemecahan masalah diatas. Tahapan dari pendekatan system yang digunakan adalah sebagai berikut:

1. Studi Literatur  
   Mempelajari literature dan paper mengenai *local directional pattern, multi-layer perceptron*, *backpropagation,* serta pengimplementasian *neural network*
2. Pengumpulan Data  
   Data uji yang akan digunakan ialah data uji *Cohn-Kanade yang* dapat diakses melalui <http://www.consortium.ri.cmu.edu/ckagree/> dan *Japanese Female Facial Expression* (JAFFE) dapat diakses melalui situs <http://www.kasrl.org/jaffe_info.html>.
3. Implementasi Sistem  
   Mengimplimentasikan hasil rancangan system dari tahap pertama
4. Pengujian Sistem  
   Menguji system yang telah diimplementasikan dengan menggunakan data gambar yang didapat dari database wajah online.
5. Analisis Hasil  
   Hasil dari tahap ketiga, lalu dianalisis sesuai dengan rumusan masalah yang telah ditetapkan. Yaitu dengan melihat konfigurasi *neural network* yang dapat memberikan akurasi tertinggi.
6. Pembuatan Laporan  
   Berdasarkan analisis hasil, penulis membuat laporan yang lengkap mengenai hasil penelitian.

## Hipotesa

Berdasarkan perumusan masalah yang telah ditetapkan, hipotesa awal yang penulis tetapkan adalah sebagai berikut:

1. Gambar wajah, dapat diklasifikasikan kedalam 7 kategori, yaitu: marah, senang, terkejut, jijik, takut, sedih, dan netral.
2. Hasil klasifikasi yang didapatkan oleh sistem, memiliki akurasi diatas percobaan sebelumnya, yaitu LDP-SVM, dengan akurasi diatas 85%.

# Dasar Teori



## Ekspresi Wajah

*Expression Recognition*adalah teknik automatisasi proses pengenalan ekspresi dimana computer berperan sebagai pengamat, berusaha menangkap pesan apa yang ditunjukan oleh sebuah wajah pada suatu media visual.

## Konsep dasar Citra Digital

Citra (image) adalah gambar pada bidang dua-dimensi, maka sebuah citra merupakan dimensi spasial atau bidang yang berisi informasi warna yang tidak bergantung waktu (Munir, 2004). Pada umumnya, citra digital direpresentasikan dalam bentuk matrix 2 dimensi dengan ukuran w \* h, dimana w ialah lebar dan h adalah tinggi dari citra tersebut.

### Pra-pengolahan Citra

Proses yang dilakukan pada citra untuk mendapatkan citra yang kualitasnya lebih baik dari citra sebelumnya dan melakukan standarisasi terhadap citra untuk memudahkan proses pada tahap selanjunya. Proses yang biasanya dilakukan pada tahap ini adalah pengubahan citra RGB menjadi citra Grayscale.

## Ekstrasi Fitur

Tujuan dari ekstrasi fitur adalah untuk menemukan represantasi dari citra wajah yang efektif dan efisien, untuk selanjutnya menjadi paramater pada proses klasifikasi. Dua pendekatan untuk melakukan ekstrasi fitur adalah geometrik *feature-based* dan *appearance-based* (Jabid, et al., October 2010).

Pada pendekatan geometrik, parameter utama ialah bentuk dan relasi dari komponen wajah seperti posisi, jarak dan kemiringan.

Sementara itu pendekatan appeareance-based merupakan approximasi berbasis konvolusi, pada area wajah, seperti yang dilakukan pada experiment ini yang menggunakan Kirschedge Matrix sebagai dasar konvolusinya.

### Konvolusi

Menurut teori konvolusi, konvolusi pada fungsi kontinyu ialah operator dari dua fungsi yang menghasilkan inverse dari transformasi laplace kedua fungsi tersebut Atau secara lebih formal dinotasikan sebagai:

(1)

(Webb, et al., 1999)

Sementara pada fungsi diskrit, konvolusi dinotasikan sebagai:

(2)

(William, et al., Numerical Recipe in Pascal)

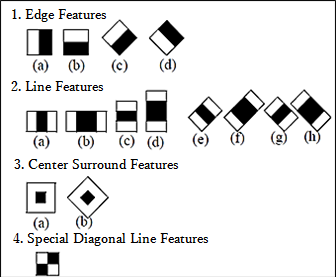
### Haar cascade

*Haar cascade* merupakan klasifier yang umum digunakan pada i*mage processing* untuk mengenali suatu bentuk. Pada kasus ini, *Haar Cascade* digunakan untuk melakukan seleksi Area Mata. *Haar Cascade* terdiri dari beberapa klasifier yang terdiri dari aturan – aturan sederhana yang memiliki bobot. Dimana klassifier tersebut terurut dari yang paling sederhana (paling tidak akurat), hingga yang paling komplex (paling akurat), sehingga apabila klasifier sederhana menyatakan daerah tersebut tidak lolos, klasifier yang lebih komplex tidak perlu diterapkan pada daerah tersebut. *Haar Cascade* terdiri dari empat komponen penting, yaitu *Haar Fature,* *Integral Image*, *AdaBoost*, dan *Cascade Classifier*.

#### Haar Feature

*Haar Feature* merupakan fitur sederhana yang digunakan untuk proses klasifikasi pada data berbentuk Image. *Haar Feature* terdiri dari 2 bagian, yaitu bagian hitam dan putih dimana bagian hitam dan putih memiliki jumlah (count) yang sama Nilai dari *Haar Feature* ialah selisih jumlah intensitas bagian putih dengan bagian hitam.

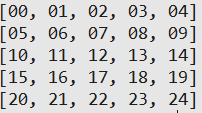
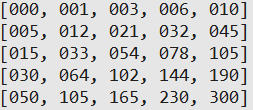
jenis-jenis *Haar feature* yang umum digunakan:



Gambar 2.1 Haar Features

#### Integral Image

Integral Image atau Summed-Area-Table merupakan algoritma untuk menghitung total intensitas pada matrix 2 dimensi. Hal tersebut, terlihat seperti pada gambar berikut.

Gambar 2.2 Matrix 2 dimensi dan Summed-Area-Tablenya

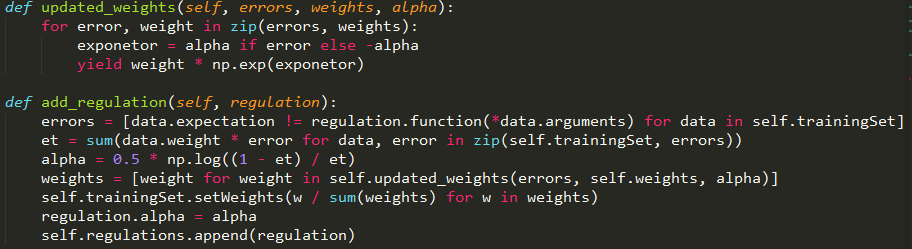
Dengan menggunakan data pada Integral Image, jumlah intensitas kumulatif pada suatu gambar dapat dihitung dengan:

(3)

Dimana:

#### AdaBoost

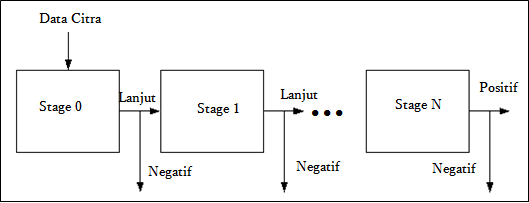
*Boosting* merupakan metode untuk menyatukan berbagai *classifiers* lemah menjadi sebuah *classifier* yang kuat (Kruppa, et al., 2003). Pada *adaboost* boosting dilakukan dengan cara memberikan bobot pada masing – masing dataset dan juga classifiers lemah tersebut. Pada umumnya, classifiers lemah tersebut ialah suatu pertidaksamaan yang membagi dimensi parameter input menjadi 2 bagian.



Gambar 2.2 Algoritma utama pada Adaboost.

#### Cascade Classifier

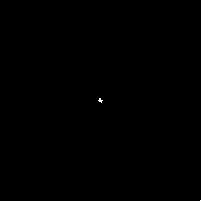
*Cascade classifier* merupakan proses seleksi dengan menggunakan Adaboost Classifier yang telah diurutkan dari yang paling sederhana hingga yang paling kompleks. Hal ini dilakukan karena proses Adaboost diatas baik yang lemah, maupun yang kuat cenderung menghasilkan klassifier yang memiliki akurasi true-positif tinggi, dan klassifier yang lebih kompleks memiliki false-positive hit-rate yang lebih sedikit. Dengan methode ini, daerah yang sudah dinyatakan negatif oleh klasifier lemah dibuang, dan tidak digunakan untuk proses selanjutnya sehingga menghemat proses komputasi.



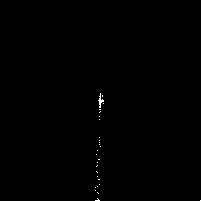
Gambar 2‑3 Cascade Classifier

### Harris Corner Detection

Percobaan ini, selain menggunakan Haar Cascade untuk menyeleksi area mata, juga menggunakan algoritma Harris corner detection, untuk memastikan bahwa pada area tersebut setidaknya, terdapat suatu sudut. Harris corner detection ialah metode untuk mencari sudut pada citra digital. Hal ini dilakukan dengan mencari *feature point*. Proses pendeteksian *feature point* tersebut ialah dengan melihat nilai eigen dari kovariant dari distribusi turunan intensitas pada citra digital tersebut terhadap sumbu X, dan sumbu Y.

D:\TA.prototype\!laporan\data2\middle.jpg 

Gambar 2.4 Distribusi turunan pada area yang tidak memiliki perubahan ekstrim (memiliki eigen value senilai [0.017, 0.012])

D:\TA.prototype\!laporan\data3\middle.jpg 

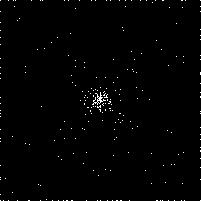
Gambar 2.5: Distribusi turunan pada gambar bergaris

(memiliki eigen value senilai [22.965, 213.447])

D:\TA.prototype\!laporan\data1\middle.jpg 

Gambar 2.6 Distribusi turunan intensitas pada gambar bersudut.

(memiliki eigen value senilai [170.463, 335.073])

D:\TA.prototype\!laporan\data4\middle.jpg 

Gambar 2.7 Distribusi turunan intensitas pada gambar berbentuk donat.

(memiliki eigen value senilai [207.614, 274.981])

### LDP (Local Directional Pattern)

*Local Directional Pattern* merupakan salah satu metode ekstrasi fitur yang menggunakan pendekatan berbasis appearance (Jabid, et al., October 2010).

Sebagaimana dijelaskan diatas, apabila konvolusi dari fungsi diskrit satu peubah f terhadap fungsi g dinyatakan sebagai:

(1)

Maka konvolusi pada bidang diskrit dengan dua peubah dinyatakan sebagai:

(2)

LDP, merupakan extraksi fitur berbasiskan 8 fungsi diskrit yang disebut sebagai *Kirsrchedge Matrix*, yang didefisikan sebagai berikut:



Gambar 2.8 8 arah Kirschedge Matrix

*LDP Descriptor* merupakan histogram yang menyimpan distribusi kode LDP pada citra untuk sebagai informasi yang mendeskripsikan citra tersebut. Histogram ini tidak menampung distribusi kode LDP secara langsung namun menyimpannya tiap region. Sebelum menghitung distribusi kode LDP, citra dibagi menjadi region kemudian dihitung histogram tiap region dan akhirnya histogram tiap region tersebut digabungkan untuk menjadi satu region utuh yang mendeskripsikan citra tersebut. Rumus yang digunakan untuk menghitung histogram tiap region adalah :

(4)

dimana adalah kode LDP. Untuk setiap nilai , sebuah histogram memiliki angka biner atau setiap merupakan vektor dengan panjang Karena citra dibagi kedalam region dan setiap region memiliki histogramnya masing-masing maka panjang akhir dari LDP deskriptor sebuah citra berukuran . Alasan pembagian citra menjadi beberapa region adalah untuk menampung informasi lokasi dari fitur wajah.



Gambar 2‑11 LDP Descriptor

## Klasifikasi Fitur

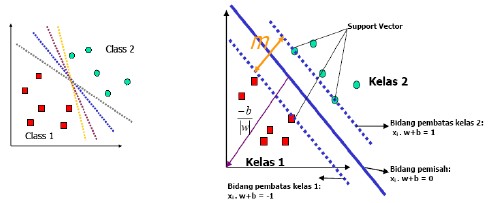
Klasifikasi merupakan proses untuk mengindetifikasi termasuk grup atau kelas manakah sebuah data baru, kelas-kelas ini telah didapatkan sebelumnya dengan melakukan pengujian terhadap data-data sebelumnya. Klasifikasi fitur bertujuan untuk menemukan model atau fungsi fitur yang membedakan kelas data untuk memperkirakan kelas dari data fitur baru yang tidak memiliki label. *Template Matching*, *Linear Discriminant Analysis*, *Linear Programming*, dan SVM adalah beberapa teknik klasifikasi yang dapat digunakan untuk melakukan klasifikasi terhadap citra ekpresi wajah (Shan, et al., 2005). Berdasarkan penelitian sebelumnya (Shan, et al., 2005), SVM memiliki performansi terbaik dibandingkan teknik-teknik lainnya yang telah disebutkan.

### Support Vector Machines

*Support Vector Machine* (SVM) dikembangkan oleh Boser, Guyon, Vapnik, dan pertama kali dipresentasikan pada tahun 1992 di *Annual Workshop on Computational Learning Theory*. Support Vector Machine (SVM) adalah sistem pembelajaran yang menggunakan ruang hipotesis berupa fungsi-fungsi linier dalam sebuah ruang fitur (feature space) berdimensi tinggi, dilatih dengan algoritma pembelajaran yang didasarkan pada teori optimasi dengan mengimplementasikan learning bias yang berasal dari teori pembelajaran statistik. Prinsip dasar SVM adalah linear classifier, dan selanjutnya dikembangkan agar dapat bekerja pada problem non-linear. dengan memasukkan konsep kernel trick pada ruang kerja berdimensi tinggi.

#### SVM pada Linearly Separable Data

*Linearly separable* data merupakan data yang dapat dipisahkan secara linier. Misalkan {X1, …, Xn} adalah dataset dan {+1,-1} adalah label kelas dari data Xi. Pada Gambar 2‑12 dapat dilihat berbagai alternatif bidang pemisah yang dapat memisahkan semua data set sesuai dengan kelasnya. Namun, bidang pemisah terbaik tidak hanya dapat memisahkan data tetapi juga memiliki margin paling besar.



Gambar 2‑12 Hyperplane (bidang pemisah)

#### SVM pada NonLinearly Separable Data

Untuk mengklasifikasikan data yang tidak dapat dipisahkan secara linier formula SVM harus dimodifikasi karena tidak akan ada solusi yang ditemukan. Oleh karena itu, kedua bidang pembatas harus diubah sehingga lebih fleksibel (untuk kondisi tertentu) dengan penambahan variabel ξ,>0 dan mengubah konstrain menjadi .

Pencarian bidang pemisah terbaik dengan dengan penambahan variable ξ sering juga disebut soft margin hyperplane. Dengan demikian formula pencarian bidang pemisah terbaik berubah menjadi:

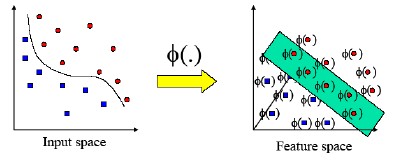
(5)

(6)

(7)

C adalah parameter yang menentukan besar penalti akibat kesalahan dalam klasifikasi data dan nilainya ditentukan oleh pengguna.

Metode lain untuk mengklasifikasikan data yang tidak dapat dipisahkan secara linier adalah dengan mentransformasikan data ke dalam dimensi ruang fitur (feature space) sehingga dapat dipisahkan secara linier pada feature space.



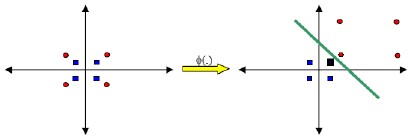
Gambar 2‑13 Transformasi dari vektor input ke feature space

Caranya, data dipetakan dengan menggunakan fungsi pemetaan (transformasi) x → ϕ(x) ke dalam feature space sehingga terdapat bidang pemisah yang dapat memisahkan data sesuai dengan kelasnya (seperti pada gambar).

Misalkan terdapat data set yang datanya memiliki dua atribut dan dua kelas yaitu kelas positif dan negatif. Data yang memiliki kelas positif adalah{(2,2),(2,−2),(−2,2),(−2,−2)}, dan data yang memiliki kelas negatif {(1,1),(1,−1),(−1,1),(−1,−1)}. Apabila data ini digambarkan dalam ruang dua dimensi (gambar dibawah) dapat dilihat data ini tidak dapat dipisahkan secara linier. Oleh karena itu, digunakan fungsi transformasi berikut:

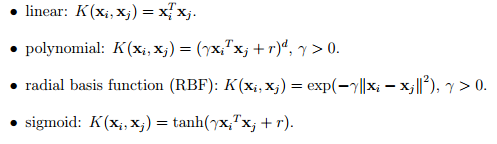
(8)

Data sesudah transformasi adalah {(6, 2), (6,6), (2,6), (2,2)} untuk kelas negatif, dan {(1,1) , (1,−1 ), (−1,1), (−1,−1)} untuk kelas positif. Selanjutnya pencarian bidang pemisah terbaik dilakukan pada data ini.



Gambar 2‑14 Transformasi untuk data yang tidak dapat dipisahkan secara linier

Feature space dalam prakteknya biasanya memiliki dimensi yang lebih tinggi dari vektor input (input space). Hal ini mengakibatkan komputasi pada feature space mungkin sangat besar, karena ada kemungkinan feature space dapat memiliki jumlah feature yang tidak terhingga. Selain itu, sulit mengetahui fungsi transformasi yang tepat. Untuk mengatasi masalah ini, pada SVM digunakan ”kernel trick”. Empat kernel dasar yang biasa digunakan adalah :



#### Metode Multiclass SVM

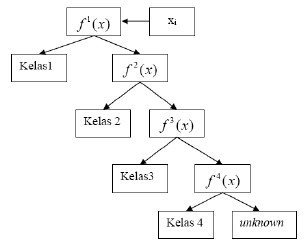
SVM sebenarnya dirancang untuk melakukan klasifikasi biner. Karena itu untuk melakukan klasifikasi terhadap kasus yang memiliki kelas lebih dari 2 maka dikembangkan beberapa pendekatan untuk menggabungkan beberapa klasifier biner. Pendekatan tersebut ialah dengan metode *One-Against-Rest* (Jabid, et al., October 2010) dan *One-Against-One*.

#### Metode One-Against-Rest

Metode ini digunakan untuk membangun k buah model SVM biner, dimana k adalah jumlah kelas atau label. Setiap model klasifikasi ke-*i* dilatih dengan menggunakan keseluruhan kelas yang lain.

Tabel 2‑3 Contoh 4 SVM biner dengan metode One-against-all

|  |  |  |
| --- | --- | --- |
|  |  | Hipotesis |
| Kelas 1 | Bukan kelas 1 |  |
| Kelas 2 | Bukan kelas 2 |  |
| Kelas 3 | Bukan kelas 3 |  |
| Kelas 4 | Bukan kelas 4 |  |



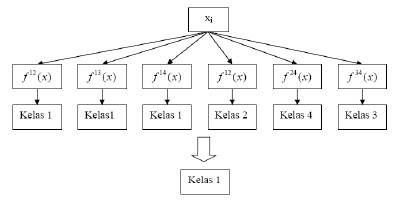
Gambar 2‑15 Contoh klasifikasi dengan metode One-against-rest

#### Metode One-Against-One

Metode ini membangun buah model SVM biner, dimana *k* adalah jumlah kelas atau label. Setiap model klasifikasi dilatih dengan menggunakan data dari dua kelas.

Tabel 2‑4 Contoh 6 SVM biner dengan metode One-against-one

|  |  |  |
| --- | --- | --- |
|  |  | Hipotesis |
| Kelas 1 | Kelas 2 |  |
| Kelas 1 | Kelas 3 |  |
| Kelas 1 | Kelas 4 |  |
| Kelas 2 | Kelas 3 |  |
| Kelas 2 | Kelas 4 |  |
| Kelas 3 | Kelas 4 |  |



Gambar 2‑16 Contoh klasifikasi dengan metode One-against-one

Jika data dimasukkan ke dalam fungsi hasil pelatihan dan hasilnya menyatakan menyatakan adalah kelas , maka suara untuk kelas ditambah satu. Kelas dari data akan ditentukan dari jumlah suara terbanyak. Jika terdapat dua buah kelas yang jumlah suaranya sama, maka kelas yang indeksnya lebih kecil dinyatakan sebagai kelas dari data. Jadi pada pendekatan ini terdapat buah permasalahan quadratic programming yang masingmasing memiliki variabel ( adalah jumlah data pelatihan). Contohnya, terdapat permasalahan klasifikasi dengan 4 buah kelas. Oleh karena itu, digunakan 6 buah SVM biner seperti pada tabel 2 dan contoh penggunaanya dalam memprediksi kelas data baru dapat dilihat pada gambar contoh klasifikasi dengan metode one-againts-one.

## Akurasi

Salah satu parameter yang digunakan untuk menetukan performansi sistem dalam tugas akhir ini adalah akurasi. Akurasi adalah tingkat kedekatan pengukuran kuantitas terhadap nilai sebenarnya. Rumus perhitungan akurasi yang digunakan adalah :

(9)

Dimana :

* : akurasi rata-rata dari t pengujian
* : banyaknya kombinasi latih-uji untuk setiap kasus pada setiap skenario
* : nilai yang diukur
* : nilai yang diharapkan

## Kecepatan

Rumus pengukuran kecepatan yang digunakan dalam tugas akhir ini adalah:

(10)

Dimana :

* : jumlah frame yang diproses tiap detik (fps)
* : banyaknya *frame* atau citra yang diproses
* : lamanya waktu yang dibutuhkan untuk memroses 1 frame dalam satuan mili second

# Analisa dan Perancangan Sistem



## Gambaran Umum Sistem

Setiap ekpresi wajah memiliki ciri yang berbeda yang dibentuk oleh posisi otot-otot pada wajah. Perbedaan ciri ini dihasilkan oleh perbedaan posisi alis, kerutan diantara alis, keadaan mata, kerutan disekitar hidung, dan keadaan mulut. Dengan melihat kesamaan ciri lokal pada wajah yang dibentuk oleh otot-otot wajah tersebut dapat dibangun sebuah sistem yang dapat mengenali ekspresi wajah.

Sistem yang dibangun diberi kode AFER (Automatic Facial Expression Recognition), dimana sistem ini ditujukan untuk melakukan pemrosesan terhadap data berbentuk citra dari *stream* video untuk mengenali ekspresi wajah seseorang didalam video tersebut.

Sistem ini memilki 2 mode yaitu : pengujian dan implementasi. Perbedaan kedua mode ini hanya terletak pada teknik dalam menerima data masukan dan teknik menghasilkan data keluaran. Pada mode pengujian, sistem menerima data inputan berupa kumpulan citra dari kumpulan file berekstensi JPEG dan menghasilkan sebuah model untuk digunakan sebagai model pengambilan keputusan pada tahap pengujian SVM. Pada mode implementasi, sistem menerima inputan berupa citra yang diterima dari *frame-frame* video *stream,* melakukan pemrosesan terhadap setiap citra dan mengeluarkan teks yang menunjukan kelas dari ekpresi yang terdapat pada citra*.*

Mulai

Capture Frame

Write to View

Capture Thread

Deteksi Wajah (Haar)

Ada wajah?

Pra-pengolahan

Ekstrasi Fitur (LDP)

Klasifikasi(SVM)

Kelas

Recognition Thread

≈ 0.2 Sec

No

Yes

Gambar 3‑1 Gambaran umum sistem

## Analisa Kebutuhan Sistem

AFER ini melakukan tugasnya dalam beberapa tahap dimana tahap-tahap ini merupakan fungsionalitas dari sistem. Kebutuhan sistem untuk memenuhi fungsionalitas tersebut adalah :

1. Sistem dapat mengambil *frame*-*frame* dari video yang sedang merekam dan mengubah setiap *frame* tersebut menjadi citra diam
2. Sistem dapat mendeteksi lokasi wajah pada citra diam
3. Sistem dapat melakukan Pra-Pengolahan terhadap data citra yang mengandung wajah
4. Sistem dapat melakukan ekstrasi fitur dari citra wajah
5. Sistem dapat melakukan klasifikasi ekpresi wajah dari fitur dan menentukan kelas dari fitur wajah

### Spesifikasi Perangkat Keras

Spesifikasi perangkat keras yang digunakan adalah :

1. Processor : Intel Core i7-3517U CPU 2.4GHz
2. Memory : 8 GB DDR3 1600 MHz SDRAM
3. Grafis : NVIDIA® GeForce® GT 635M dengan 2GB DDR3 VRAM
4. WebCam : UVC HD Webcam Resolusi 640x480
5. HDD : 500 GB Serial ATA 5400 RPM

### Spesifikasi Perangkat Lunak

Spesifikasi perangkat lunak yang digunakan adalah :

1. Sistem operasi : Windows 8 Pro 64-bit
2. Tools pemrograman : Visual Studio 2013
3. Library pemrosesan citra : emguCV
4. Bahasa pemrograman : C#
5. Paradigma pemrograman : Multithreading

### Analisa kebutuhan data

Data latih dan Data uji terdiri dari 385 citra wajah yang berasal dari Cohn-Kanade AU-Coded Facial Expression Database[[1]](#footnote-1). Data merupakan citra grayscale dengan resolusi 640x480 pixel, yang diambil dari 123 subjek dalam ruangan tertutup dengan pencahayaan terang dan subjek menghadap lurus ke kamera. Data terdiri dari 7 jenis ekpresi untuk setiap subjek seperti contoh yang dipaparkan pada Gambar 3‑2. Tidak semua subjek menghasilkan ekpresi yang jelas (beberapa ambigu) sehingga hanya 385 citra yang dilabeli dengan kelas ekpresi. Pembagian data latih dan data uji selanjutnya akan dijelaskan pada BAB 4.



Gambar 3‑2 Contoh Data Latih CK

Tabel 3‑1 Daftar jumlah data citra tiap ekpresi

|  |  |  |
| --- | --- | --- |
| **No** | **Ekpresi** | **Jumlah citra** |
| 1 | Marah | 35 |
| 2 | Jijik | 46 |
| 3 | Takut | 43 |
| 4 | Senang | 77 |
| 5 | Netral | 77 |
| 6 | Sedih | 35 |
| 7 | Terkejut | 72 |

### Analisa Masukan dan Keluaran Sistem

Data masukan untuk sistem pada mode pengujian berupa file berekstensi JPEG dengan nama file sebagai penentu kelas ekpresi wajah yang terdapat pada citra. Keluaran sistem pada mode pengujian berupa model SVM yang akan digunakan sebagai model prediksi pada mode implementasi.

Data masukan untuk sistem pada mode implementasi berupa *frame-frame* yang diterima dari video *stream*. Tiap *frame* ini dikonversi menjadi data citra digital. Keluaran sistem pada mode implementasi berupa teks yang berisi kelas ekspresi hasil prediksi terhadap citra wajah yang diproses oleh sistem.

## Perancangan Sistem

Pada tugas akhir ini dibangun sistem pengenal ekpresi wajah yang bekerja melalui 2 tahap utama yaitu :

1. Deteksi Wajah
2. Pengenalan Ekspresi

Berikut *flowchart* sistem :

Gambar 3‑3 *Flowchart* sistem

### Deteksi Wajah

Merupakan tahap untuk mencari adanya wajah pada setiap *frame* dari video *stream*, jika ditemukan maka akan dilakukan pra-pengolahan terhadap citra tersebut untuk menghasilkan citra yang konsisten dalam segi resolusi, warna, tingkat pencahayaan dan kontras, serta untuk mengurangi tingkat komputasi pada tahap berikut, namun jika tidak ditemukan maka sistem akan kembali mencari wajah pada *frame* berikut. Proses deteksi wajah menggunakan fungsi DetectMultiscale dari *Haar Cascade Face Detector* yang dapat mendeteksi beberapa wajah dalam satu citra, namun dalam tugas akhir ini fungsi tersebut dimodifikasi agar hanya mengambil salah satu wajah saja. Parameter yang digunakan pada fungsi ini dijelaskan pada Tabel 3‑2.

Tabel 3‑2 Parameter DetectMultiscale

|  |  |  |  |
| --- | --- | --- | --- |
| No | Parameter | Nilai | Keterangan |
| 1 | Jenis *Image* | *GPUImage* | *Graphics Processing Unit* (GPU) merupakan perangkat keras yang mirip dengan CPU namun dibuat khusus untuk memroses citra. GPU memperlakukan semua citra sebagai citra 3D, walaupun citra yang diproses 2D. GPU dioptimasi untuk mengolah citra 3D sehingga ketika digunakan untuk mengolah citra 2D yang lebih sederhana performansi yang didapatkan lebih cepat. Citra yang diproses dengan GPU disebut *GPUImage* |
| 2 | scaleFactor | 1.1 | Menentukan seberapa besar skala untuk memperkecil ukuran citra dalam proses deteksi |
| 3 | minNeighbors | 10 | Jumlah minimal pixel sekitar daerah deteksi yang akan dianggap sebagai wilayah wajah |
| 4 | minSize | Empty | Menentukan minimal ukuran objek yang akan dideteksi. |

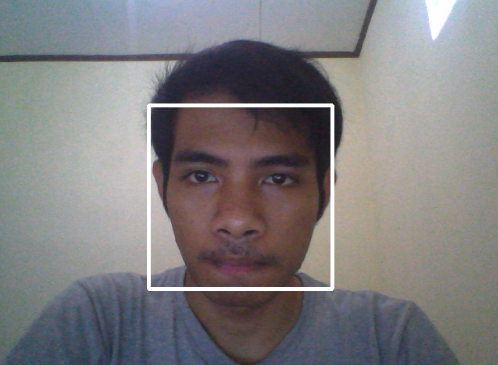
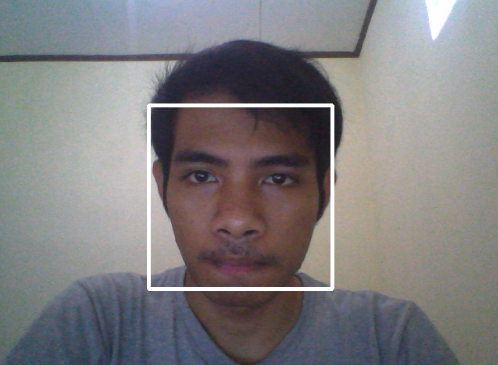
### Pra-pengolahan Citra

Pra-pengolahan citra yang dilakukan antara lain :

1. Konversi citra RGB ke *Grayscale*
2. Pembuangan daerah citra selain wajah terbesar yang terdeteksi (cropping)

Area wajah yang dihasilkan oleh *Haar Cascade Face Detector* berbentuk persegi, dimana area telinga ikut serta didalam daerah ini, sementara area telinga tidak banyak berpengaruh terhadap proses pengenalan ekpresi wajah sehingga daerah ini juga dibuang saat dilakukan pembuangan daerah gambar bukan wajah, hal ini terbukti menambah tingkat akurasi sistem dan dapat dilihat hasil pengujiannya BAB 4.

1. Pengubahan resolusi citra menjadi 135 x 184
2. *Histogram Equalization* untuk menjaga tingkat pencahayaan dan kontras yang konsisten

Gambar 3‑4 Citra RGB dikonversi ke Grayscale



Gambar 3‑5 *Cropping* daerah wajah



Gambar 3‑6 Hasil *Histogram Equalization*

### Pengenalan Ekpresi

Pada tahap ini terbagi menjadi 2 proses yaitu Ekstrasi Fitur yang menggunakan metode *Local Directional Pattern* dan Klasifikasi Fitur yang menggunakan *Support Vector Machine*.

#### Ekstrasi Fitur

Pada proses ekstrasi fitur LDP akan dilakukan penghitungan kode LDP yang akan digunakan untuk menggantikan nilai intensitas pixel, karena itu untuk citra berukuran dengan kernel arah yang memiliki ukuran , akan dilakukan penghitungan kode LDP sebanyak kali. Proses penghitungan kode LDP dilakukan dengan mengalikan sub matriks berukuran dengan kernel arah sebanyak 8 kali. Berikut adalah proses perhitungan kode LDP:

Perhitungan kode LDP akan dilakukan mulai dari pixel pada indeks ke agar dapat dikonvolusi dengan matriks kernel.



Gambar 3‑7 Matriks Citra

Pixel yang ditandai merah adalah pixel yang akan dihitung kode LDP-nya sehingga akan diambil sub matriks berukuran dengan pixel tersebut sebagai titik tengahnya. Sub matriks ini akan dikonvolusikan dengan 8 matriks kernel arah.



Gambar 3‑8 Konvolusi dengan kernel 8 arah

Nilai respon bit hasil konvolusi pada tiap arah tidak akan digunakan semuanya, karena biasanya hanya beberapa arah saja sudah cukup untuk digunakan pada LDP, jika nilai respon bit semua arah digunakan akan memakan waktu proses lebih lama namun hasilnya sama dengan pemilihan beberapa arah yang paling menonjol. Hasil konvolusi yaitu diambil nilai absolutnya dan diurutkan dari besar ke kecil. dari (dalam kasus ini ) tertinggi bernilai 1 dan sisanya bernilai 0. Kemudian diurutkan kembali dari sampai dan dikonversi menjadi nilai desimal.



Gambar 3‑9 Hasil konvolusi



Gambar 3‑10 Pengurutan hasil konvolusi

Berdasarkan Gambar 3‑10 nilai desimalnya adalah 82. Nilai ini adalah kode LDP dan digunakan untuk menggantikan nilai intensitas pixel yang sedang dihitung kode LDP-nya. Matriks baru hasil perhitungan kode LDP akan berukuran dengan . Pada Gambar 3‑11 daerah matriks yang ditandai abu-abu tidak digunakan lagi.



Gambar 3‑11 Matriks baru

Setelah menghitung kode LDP pada setiap *pixel,* matriks baru dengan ukuran direpresentasikan pada histogram LDP sebagai LDP *Descriptor*. Histogram ini merupakan gabungan dari sub histogram yang mewakili tiap region pada matriks. Kode LDP memiliki kemungkinan nilai, dengan maka ada 56 kemungkinan nilai, jumlah ini mewakili panjang satu vektor sub histogram. Dengan demikian panjang satu vektor LDP *Descriptor* adalah , dengan (matriks dibagi menjadi 7 kolom dan 7 baris) maka panjang satu vektor LDP *Descriptor* adalah 2744. Vektor LDP *Descriptor* menampung ciri yang mendeskripsikan sebuah citra wajah sehingga vektor inilah yang akan digunakan oleh SVM untuk diolah menjadi model. Untuk meningkatkan akurasi SVM dalam melakukan klasifikasi, diharapkan nilai yang diolah oleh SVM berkisar antar 0 sampai 1, oleh karena itu dilakukan normalisasi terhadap LDP *Descriptor* sebelum digunakan oleh SVM.

#### Klasifikasi Fitur

Pada klasifikasi fitur ditahap *training*, digunakan fungsi SVMTrain pada emguCV dengan parameter yang dipaparkan pada Tabel 3‑3.

Tabel 3‑3 Parameter SVM

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Parameter** | **Nilai** | **Keterangan** |
| 1 | Tipe Kernel | Linear | Untuk sistem real-time dibutuhkan kernel yg memiliki komputasi cepat, Linear kernel lebih cepat dibandingkan dengan RBF dan Poly karena tidak ada pemetaan yang dilakukan Untuk data yang memilki dimensi yang tinggi linear kernel meimiliki tingkat akurasi yang sebaik RBF namun dengan komputasi yang lebih cepat. Karena data dengan dimensi tinggi biasanya telah dapat dipisahkan secara linear sehingga tidak perlu lagi dipetakan ke dimensi yang lebih tinggi dengan RBF kernel ataupun kernel lainnya dan jika dipetakan pun tidak akan meningkatkan akurasinya, hanya akan menambah lama waktu komputasi. Definisi dimensi yang tinggi disini adalah ketika jumlah feature data mencapai ribuan dan lebih banyak daripada data yang diolah.[15] |
| 2 | Tipe SVM | C\_SVC | Mengimplementasikan pendekatan one-againts-one |
| 3 | Nilai C | 0.1 | Nilai pinalti untuk kesalahan perhitungan, semakin kecil nilai C maka margin untuk hyperplane yang digunakan semakin kecil juga dan sebaliknya. Jika nilai C terlalu besar maka SVM akan kehilangan kemampuan generalisasi karena SVM mencoba untuk mencocokan data latih dengan data uji semirip mungkin. |
| 4 | Maksimum Iterasi | 10000 kali | Jumlah perulangan pada rumus perhitungan quadratic |
| 5 | Epsilon | 0.00001 | ε mempengaruhi kehalusan model SVM dan mempengaruhi jumlah support vector, jadi kompleksitas dan kemampuan melakukan generalisasi bergantung pada nilai ini. Semakin tinggi semakin sedikit jumlah support vector dan sebaliknya. |
| 6 | kFold | 7 | Jumlah kelompok untuk grid Search dalam pencarian nilai C yang optimal |

Pada tahap ini juga dilakukan *Cross-Validation* untuk mencari parameter optimal untuk nilai C. Prinsip utama dari *Cross-Validation* adalah dengan membagi data uji kedalam kFold bagian . Setiap bagian memiliki jumlah data yang sama banyaknya. Tiap 1 bagian kemudian diuji dengan kFold-1 bagian lainnya sebagai data latih untuk mendapat nilai *error-rate* terkecil dari setiap nilai C yang telah didefinisikan. Nilai *error-rate* dari setiap parameter C kemudian dibandingkan. Nilai C yang menghasilkan *error-rate* terkecil akan dipilih dan digunakan untuk melakukan training terhadap keseluruhan data uji.

# Analisa dan Pengujian Aplikasi



## Gambaran Umum Pengujian

Peformansi dari Sistem Pengenalan Ekspresi Wajah Berbasis Pengolahan data video diukur dari sisi akurasi dan waktu proses yang dibutuhkan. Yang dimaksud dengan akurasi adalah benar tidaknya hasil keluaran sistem yang menyatakan ekspresi yang terdapat pada citra.

Pengujian sistem AFER ditujukan untuk mendapatkan pengaturan parameter optimal dari LDP dan SVM sehingga memberikan akurasi yang tinggi dalam mengenali ekspresi wajah. Parameter optimal diperoleh dengan melakukan perubahan terhadap beberapa parameter dengan mengatur parameter lain pada nilai tertentu. AFER terdiri atas 3 proses utama yang saling berkaitan dan memberikan efek terhadap akurasi sistem.

Gambar 4‑1 Proses Utama AFER

Dengan memperhatikan tahapan proses yang dilalui dalam pengolahan data maka, tahapan proses pengujian yang dilakukan pada AFER meliputi:

1. Pengujian efek pemotongan gambar hasil deteksi wajah dengan mengikutsertakan area telinga dan tidak
2. Setting parameter LDP yang meliputi:
   1. Mendapatkan nilai yang optimal pada LDP dengan mencoba 4 nilai yang berbeda
   2. Mendapatkan pembagian region yang optimal dengan dengan menguji 4 kombinasi jumlah kolom dan baris yang berbeda
3. Pengujian parameter SVM yang meliputi:
   1. Menguji jenis kernel SVM yaitu Linear dan RBF
   2. Melihat efek kFold yang berbeda pada SVM terhadap akurasi sistem
4. Pengujian menyeluruh dengan menggunakan parameter optimal yang diperoleh dari pengujian sebelumnya.

### Skenario Pengujian

Data yang digunakan terdiri dari 385 citra wajah yang berasal dari Cohn-Kanade AU-Coded Facial Expression Database dengan komposisi data seperti yang ditunjukkan pada Tabel 4‑1.

Tabel 4‑1 Komposisi data

|  |  |  |
| --- | --- | --- |
| **No** | **Ekpresi** | **Jumlah citra** |
| 1 | Marah | 35 |
| 2 | Jijik | 46 |
| 3 | Takut | 43 |
| 4 | Senang | 77 |
| 5 | Netral | 77 |
| 6 | Sedih | 35 |
| 7 | Terkejut | 72 |

Dengan mempertimbangkan keseimbangan data maka untuk setiap ekpresi digunaknan data sebanyak 30 citra, sehingga total data yang dimiliki adalah 30 x 7 = 210 data citra, data ini kemudian disebut *balanced data*. Data tersebut dibagi menjadi 5 kelompok dengan total data setiap kelompok adalah 42 data (setiap ekpresi terwakili dengan 6 data). Dari 5 kelompok data, 3 kelompok digunakan sebagai data training dan 2 kelompok lainnya digunakan sebagai data uji. Pemilihan kelompok yang digunakan sebagai kelompok testing maupun training dilakukan secara kombinasional sehingga jumlah pengujiannya adalah . Hasil dari setiap pengujian akan diambil rata-ratanya untuk mendapatkan nilai akurasi.

210 data citra

(seimbang antar ekpresi)

Set A

42 data

Set B

42 data

Set C

42 data

Set D

42 data

Set E

42 data

Proporsi Data Latih : 3 Kelompok

Proporsi Data Uji : 2 Kelompok

Kombinasi Latih-Uji : 10 Pengujian

Jumlah data setiap pengujian = 84 Data

Gambar 4‑2 Pembagian data

Data yang telah dipersiapkan digunakan untuk melakukan pengujian terhadap AFER dimana detail skenario pengujian ditunjukkan pada Tabel 4‑2.

Tabel 4‑2 Detail Skenario

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Nama Skenario | Tujuan | Parameter Tetap | Parameter Berubah | Data |
| 1 | Skenario 1 | Membuktikan pengaruh area telinga terhadap proses pengenalan ekspresi | **LDP**  *k* : 3  Region : 8 x 5  **SVM**  Kernel : Linear  kFold : 7 | Citra wajah tanpa telinga dan Citra wajah mengandung telinga | Tipe : *Balanced data*  Jumlah : 210 data |
| 2 | Skenario 2 | Mendapatkan nilai sebagai penentu jumlah nilai respon arah yang menonjol dalam pembentukan kode LDP | Menggunakan hasil terbaik skenario 1  **LDP**  Region : 8 x 5  **SVM**  Kernel : Linear  kFold : 7 | **LDP**  *k* : 1,2,3,4  Nilai *k* 5,6,7 tidak diuji karena memiliki panjang vektor yang sama dengan *k* 3,2,1 | Tipe : *Balanced data*  Jumlah : 210 data |
| 3 | Skenario 3 | Mendapatkan jumlah region untuk membagi citra wajah yang digunakan dalam pembentukan LDP *Histogram.* | Menggunakan hasil terbaik skenario 1  **LDP**  *k* : hasil terbaik skenario 2  **SVM**  Kernel : Linear  kFold : 7 | **LDP**  Region : 8x5, 7x7, 9x7, 9x9 | Tipe : *Balanced data*  Jumlah : 210 data |
| 4 | Skenario 4 | Mengetahui jenis kernel yang tepat yang digunakan pada SVM | Menggunakan hasil terbaik skenario 1  **LDP**  *k* : hasil terbaik skenario 2  Region : hasil terbaik skenario 3  **SVM**  kFold : 7 | **SVM**  Kernel : Linear,  RBF | Tipe : *Balanced data*  Jumlah : 210 data |
| 5 | Skenario 5 | Mendapatkan nilai kFold untuk pemilihan nilai koefisien C yang optimal | Menggunakan hasil terbaik skenario 1  **LDP**  *k* : hasil terbaik skenario 2  Region : hasil terbaik skenario 3  **SVM**  Kernel : hasil terbaik skenario 4 | **SVM**  kFold :  5, 7, 9 | Tipe : *Balanced data*  Jumlah : 210 data |
| 6 | Skenario 6 | Mendapatkan performansi sistem melalui kecepatan dalam mengklasifikasikan ekpresi wajah | Menggunakan hasil terbaik skenario 1  **LDP**  *k* : hasil terbaik skenario 2  Region : hasil terbaik skenario 3  **SVM**  Kernel : hasil terbaik skenario 4  kFold : hasil terbaik skenario 5 |  | Tipe : Un*balanced data*  Jumlah : 385 data |

### Hasil Pengujian dan analisis

#### Skenario 1

Karena LDP melakukan ekstrasi ciri dengan melihat ciri lokal pada sebuah citra maka fitur wajah apa saja yang terkandung didalam citra yang akan diekstrak dapat mempengaruhi hasil ekstrasi cirinya. Karena itu pada skenario 1 dilakukan pengujian terhadap pengaruh keberadaan area telinga terhadap hasil ekstrasi ciri suatu citra wajah. Hasil pengujian ini dapat dilihat pada Tabel 4‑3.

Tabel 4‑3 Skenario 1

|  |  |  |  |
| --- | --- | --- | --- |
| Nomor | Panjang vektor fitur LDP | Pengujian | Akurasi (%) |
| 1 | 392 | Dengan area telinga | 75.69 |
| 2 | 392 | Tanpa area telinga | 87.61 |

Ditemukan bahwa citra wajah tanpa telinga memiliki akurasi yang lebih tinggi (87.61%) dalam mengenali ekspresi dibanding citra wajah dengan telinga (75.69%). Hasil ini diakibatkan karena area ini memiliki tingkat kesamaan yang cukup tingga pada semua citra ekskpresi wajah, sehingga ciri lokal yang dimiliki area ini menjadi noise bagi SVM dalam membentuk model sistem, karena itu dengan diabaikannya area ini, tingkat akurasi dapat ditingkatkan.

#### Skenario 2

Besar nilai *k* menunjukan jumlah arah yang dominan dalam menentukan nilai kode LDP. Nilai *k* mempengaruhi jumlah kemungkinan nilai untuk sebuah kode LDP, semakin banyak kemungkinan nilai maka semakin panjang vektor fitur LDP. Panjang vektor LDP ini mewakili jumlah dimensi input pada SVM. Hasil pengujian nilai *k* dapat dilihat pada Tabel 4‑4.

Tabel 4‑4 Skenario 2

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Nilai | Panjang vektor fitur LDP | Akurasi (%) | Waktu Proses (ms) | | | | Kecepatan  (fps) |
| Deteksi | LDP | SVM Test | Total |
| 1 | 392 | 83.69 | 150.5 | 131.94 | 2 | 285.44 | 3.5 |
| 2 | 1372 | 86.30 | 153.17 | 147.25 | 2 | 303.42 | 3.29 |
| 3 | 2744 | 87.61 | 166.91 | 138.57 | 2 | 308.48 | 3.24 |
| 4 | 3430 | 87.65 | 200.67 | 132.9 | 2 | 336.57 | 2.97 |

Setelah melakukan pengujian 4 nilai *k*, diketahui bahwa pengujian dengan nilai *k*=4 menghasilkan akurasi tertiggi, namun karena hanya meningkatkan akurasi 0.04% dibandingkan dengan *k*=3 tetapi jumlah vektornya semakin banyak maka parameter *k* yang dipilih adalah *k*=3.

#### Skenario 3

Region merupakan komponen penting dalam LDP karena region yang membagi sebuah citra menjadi beberapa area lokal, karena itu pada skenario 3 akan dilakukan pengujian terhadap beberapa jumlah region. Hasil pengujian ini dapat dilihat pada Tabel 4‑5.

Tabel 4‑5 Skenario 3

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Region | Panjang vektor fitur LDP | Akurasi (%) | Waktu Proses (ms) | | | | Kecepatan  (fps) |
| Deteksi | LDP | SVM Test | Total |
|  | 2240 | 87.61 | 165.6 | 124.39 | 2 | 292.99 | 3.41 |
|  | 2744 | 88.92 | 179.59 | 115.65 | 2 | 298.24 | 3.35 |
|  | 3528 | 89.04 | 190.92 | 123.09 | 2 | 317.01 | 3.15 |
|  | 4536 | 88.33 | 181.77 | 132.63 | 3 | 318.4 | 3.14 |

Setelah melakukan pengujian pada skenario 3, diketahui bahwa pengujian dengan region 9x7 menghasilkan akurasi tertiggi, namun karena peningkatan akurasi yang terjadi hanya sedikit (0.12%) maka untuk alasan meminimalkan kecepatan pemrosesan makan dipilihlah region 7x7.

#### Skenario 4

SVM kernel yang sering digunakan untuk melakukan pemrosesan citra adalah Linear dan RBF, karena itu pada skenario 4 akan dilakukan pengujian terhadap kedua kernel ini. Hasil pengujian skenario 4 dapat dilihat pada Tabel 4‑6.

Tabel 4‑6 Skenario 4

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Kernel | Akurasi (%) | Waktu Proses (ms) | | | | Kecepatan  (fps) |
| Deteksi | LDP | SVM Test | Total |
| Linear | 88.92 | 179.59 | 115.65 | 2 | 297.24 | 3.36 |
| RBF | 88.69 | 179.59 | 115.65 | 5 | 300.24 | 3.33 |

Setelah melakukan pengujian pada skenario 4, diketahui bahwa pengujian kernel Linear memiliki akurasi yang paling tinggi dengan kecepaan pemrosesan yang paling baik. Hal ini dimungkinkan karena kernel RBF mencoba melakukan pemetaan fitur ke dimensi yang lebih tinggi, padahal jumlah fitur LDP sudah memiliki dimensi yang tinggi,sehingga waktu yang diperlukan semakin lama.

#### Skenario 5

Dalam mendapatkan parameter SVM yang terbaik biasa dipergunakan teknik *Grid-Search,* teknik ini membagi data uji menjadi kFold kelompok dan melakukan *training-testing* menggunakan kombinasi-kombinasi parameter SVM yang dicari nilai optimalnya. Pada skenario ini dilakukan pengujian terhadap nilai kFold untuk mendapatkan hasil akurasi dan kecepatan pemrosesan yang ideal. Hasil pengujian skenario 5 dapat dilihat padaTabel 4‑7.

Tabel 4‑7 Skenario 5

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| kFold | Akurasi | Waktu Proses (ms) | | | | Kecepatan  (fps) |
| Deteksi | LDP | SVM Test | Total |
| 5 | 88.34 | 179.59 | 115.65 | 2 | 297.24 | 3.36 |
| 7 | 88.92 | 179.59 | 115.65 | 2 | 297.24 | 3.36 |
| 9 | 88.72 | 179.59 | 115.65 | 2 | 297.24 | 3.36 |

Setelah melakukan pengujian pada skenario 5, diketahui bahwa pengujian menggunakan kFold=7 menghasilkan akurasi yang paling tinggi sehingga nilai ini yang dpilih. Nilai kFold=7 menunjukan bahwa ketika SVM melakukan Grid-Search, data latih dibagi menjadi 7 kelompok untuk mencari parameter C terbaik.

#### Skenario 6

Pada pengujian ke-6 ini tidak lagi menggunakan metode membagi data kedalam 5 kelompok untuk menentukan data *training* dan data *testing* namun menggunakan pembagian 1 banding seluruh data, dimana 1 data sebagai data testing dan data sisanya sebagai training untuk meniru lingkungan sebenarnya dimana sistem diaplikasikan.

Hasil pengujian skenario 6 dapat dilihat pada Tabel 4‑8.

Tabel 4‑8 Skenario 6

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Marah | Jijik | Takut | Senang | Netral | Sedih | Kaget |
| Marah | 84.03 | 2.5 | 0.64 | 0 | 5.5 | 7.33 | 0 |
| Jijik | 0.7 | 99.3 | 0 | 0 | 0 | 0 | 0 |
| Takut | 1.38 | 0.7 | 88.9 | 0 | 8.3 | 0.72 | 0 |
| Senang | 0 | 0 | 0 | 100 | 0 | 0 | 0 |
| Netral | 6.25 | 0 | 9.03 | 0 | 78.47 | 6.25 | 0 |
| Sedih | 8.33 | 0 | 0.69 | 0 | 4.18 | 84.72 | 2.08 |
| Kaget | 0 | 0 | 0 | 0 | 0 | 0 | 100 |

Pada pengujian ini juga didapatkan lama rata-rata pemrosesan tiap *frame* selama 297.24 ms atau 3.36 fps dan akurasi rata-rata 90.77%.

## Rangkuman Analisa

Setelah melakukan pengujian terhadap pengaruh area telinga terhadap diferensiasi suatu ekspresi, ditemukan bahwa citra wajah tanpa telinga memiliki akurasi yang lebih tinggi (87.61%) dibanding citra wajah dengan telinga (75.69%). Selanjutnya pengujian pada parameter LDP, menyimpulkan bahwa parameter yang cocok adalah *k* sebesar 3 danjumlah region sebesar . K*=*3 artinya LDP hanya melihat 3 nilai respon kernel arah teratas yang digunakan sebgai pembentuk kode LDP. Jumlah region artinya LDP membagi citra menjadi area lokal. Selain itu, nilai ini juga menunjukan bahwa pembagian citra wajah menjadi region menyebabkan area lokal alis, mata, hidung, dan mulut sebagai penentu ciri sebuah ekspresi wajah terbagi dengan baik.

Setelah melakukan pengujian terhadap kernel pemetaan, ditemukan bahwa Linear kernel memiliki kecepatan proses hingga 14 kali lipat dibandingkan RBF kernel namun dengan tingkat akurasi yang hampir sama. Hal ini dikarenakan RBF tetap melakukan pemetaan dimensi terhadap data masukan padahal jumlah dimensi data masukan sudah cukup besar sehingga tidak diperlukan lagi adanya pemetaan ke dimensi yang lebih tinggi untuk dapat dipisahkan secara linear.

Dari Tabel 4‑8 terlihat bahwa ekspresi kaget dan senang memiliki akurasi 100%, artinya ciri wajah pada ekspresi ini paling mudah dibedakan dengan ciri wajah pada ekspresi lainnya. Sementara ekspresi netral memilki akurasi paling kecil karena kemiripan cirinya dengan ekspresi marah, takut dan sedih cukup besar untuk data CK.

Sistem berhasil membedakan ekspresi wajah manusia walaupun terdapat subjek yang beragam sehingga membuktikan bahwa teknik ini bukanlah teknik deteksi objek yang melihat kesamaan ciri secara keseluruhan namun merupakan sistem yang dapat mengenali ciri sebuah ekspresi berdasarkan ciri lokalnya.

# Kesimpulan dan Saran



## Kesimpulan

1. Metode pengenalan ekspresi dapat dilakukan dengan melakukan ekstrasi fitur yang terdapat pada wajah berupa pola tepi dan sudut tanpa harus mengetahui posisi elemen wajah seperti mata dan mulut. Hasil ekstrasi ini kemudian digunakan untuk melatih algortima klasifikasi untuk menemukan ciri dari setiap ekspresi agar selanjutnya dapat digunakan untuk mengenali ekpresi-ekpresi wajah.
2. Sistem dapat mengenali ekspresi wajah dari 123 subjek berbeda yang menghadap lurus dengan kamera dengan akurasi 88.92% dan kecepatan rata-rata pemrosesan untuk tiap *frame* 297.24 ms atau 3.36 fps, dihasilkan dari pengaturan parameter *k*=3, jumlah region 7x7, kernel SVM Linear, dan kFold=7.

## Saran

1. Ektraksi ciri dan *matching* memakan waktu pemrosesan sistem paling banyak, sehingga diperlukan metode ekstraksi dan *matching* yang lebih cepat.
2. Melakukan percobaan dengan pemilihan ciri yang lain seperti *loop* dan *delta*
3. Melakukan implementasi LIP ke sistem rekognisi biometrik lainnya seperti telapak tangan, pembuluh vena, wajah, dan lain sebagainya.

# Daftar Pustaka

Admin, t.thn. *Web ITTekom. Citra Digital.* [Online]   
Available at: http://www.ittelkom.ac.id/library/index.php?view=article-&catid=15%3Apemrosesansinyal&id=344%3Acitradigital&option=com\_content&Itemid=15  
[Diakses 25 April 2013].

Author, t.thn. *OpenCV wiki.* [Online]   
Available at: http://opencv.willowgarage.com/wiki/FaceDetection  
[Diakses 24 April 2013].

Chao, C. C. et al., May 12, 2011. Efficacy of a Facial Expression Recognition Training Software for Taiwanese School-Aged Children with Asperger's Disorder. *In International Meeting for Autism Research.*

Cho, et al., 2009. *FPGA-Based Face Detection System Using Haar Classifiers.* ACM, s.n.

Gonzales, R. C., Woods, R. E. & Eddins, S. L., 2003. *Digital Image Processing using Matlab.* s.l.:Pearson Prentice Hall.

H, W., Flannery, B. P., Teukolsky, S. A. & Vetterling, W. T., Numerical Recipe in Pascal. *1989.* s.l.:Cambridge University Press.

Jabid, T., Kabir, M. H. & Chae, O., October 2010. Robust Facial Expression Recognition Based on Local Directional Pattern.. *In ETRI Journal,* 32(5).

Kruppa, H., Santana, M. C. & Schiele, B., 2003. Fast and Robust Face Finding via Local Context. *Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance,* pp. 157-164.

Lajevardi, S. M. & Hussain, Z. M., February 15-18, 2009. Facial Expression Recognition: Gabor Filters versus Higher-Order Correlators. *In international conference on communication, computer and power (icccp’09) muscat.*

Leszczyński, M., 2010. Image Preprocessing for Illumination Invariant Face Verification. *Journal of Telecommunications and Information Technologies.*

Lienhart, Rainer, Kuranov, A. & Pisarevsky, V., 2003. Empirical Analysis of Detection Cascades of Boosted Classifiers for Rapid Object. *Pattern Recognition,* pp. 297-304.

S, G. & S., N., 2009. Social adjustment, academic adjustment, and the ability to identify emotion in facial expressions of 7-year-old children. *J Genet Psychol,* Volume 3, pp. 234-43.

Shan, C., Gong, S. & McOwan, P. W., 2005. Robust facial expression recognition using local binary patterns. *IEEE.*

Viola, P., Jones, M. J. & Snow, D., 2003. *Detecting Pedestrians Using Patterns of Motion and Appearance.* IEEE, s.n.

Wilson, P. I. & Fernandez, D. J., 2006. Facial Feature Detection Using Haar Classifiers. *Journal of Computing Sciences in Colleges.*

# Lampiran A : Detail Hasil Pengujian

## Hasil uji skenario 1

|  |  |  |
| --- | --- | --- |
| No\Area | Tanpa Telinga | Dengan Telinga |
| 1 | 0.7695238 | 0.8571429 |
| 2 | 0.7805238 | 0.8928571 |
| 3 | 0.7807224 | 0.8809524 |
| 4 | 0.7911111 | 0.8871429 |
| 5 | 0.7995238 | 0.8992381 |
| 6 | 0.8012476 | 0.9027619 |
| 7 | 0.7252381 | 0.8889524 |
| 8 | 0.7086192 | 0.8383333 |
| 9 | 0.7121225 | 0.8452381 |
| 10 | 0.7011524 | 0.8693476 |
| Rata-rata | 0.75697847 | 0.87619667 |

## Hasil uji skenario 2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No\*k* | 1 | 2 | 3 | 4 |
| 1 | 0.8095238 | 0.8809524 | 0.8571429 | 0.9166667 |
| 2 | 0.8095238 | 0.8571429 | 0.8978571 | 0.8690476 |
| 3 | 0.8809524 | 0.8690476 | 0.8899524 | 0.8459381 |
| 4 | 0.8333333 | 0.8928571 | 0.8571429 | 0.8809524 |
| 5 | 0.8095238 | 0.8571429 | 0.8455381 | 0.8928571 |
| 6 | 0.8690476 | 0.8571429 | 0.9047619 | 0.8591429 |
| 7 | 0.8452381 | 0.8214286 | 0.8809524 | 0.9285714 |
| 8 | 0.797619 | 0.8690476 | 0.9033333 | 0.9047619 |
| 9 | 0.8333333 | 0.8095238 | 0.8554381 | 0.8575429 |
| 10 | 0.8809524 | 0.9166667 | 0.8691476 | 0.8099619 |
| Rata-rata | 0.83690475 | 0.86309525 | 0.87612667 | 0.87654429 |

## Hasil uji skenario 3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No\*Region* | 8x5 | 7x7 | 9x7 | 9x9 |
| 1 | 0.8571429 | 0.9166667 | 0.9285714 | 0.9047619 |
| 2 | 0.8978571 | 0.8809524 | 0.9047619 | 0.8690476 |
| 3 | 0.8899524 | 0.9285714 | 0.8809524 | 0.8928571 |
| 4 | 0.8571429 | 0.8928571 | 0.8928571 | 0.8809524 |
| 5 | 0.8455381 | 0.8452381 | 0.8809524 | 0.8571429 |
| 6 | 0.9047619 | 0.9404762 | 0.9285714 | 0.9285714 |
| 7 | 0.8809524 | 0.8571429 | 0.8809524 | 0.8571429 |
| 8 | 0.9033333 | 0.9047619 | 0.8690476 | 0.9047619 |
| 9 | 0.8554381 | 0.8095238 | 0.8333333 | 0.8214286 |
| 10 | 0.8691476 | 0.9166667 | 0.9047619 | 0.9166667 |
| Rata-rata | 0.87612667 | 0.88928572 | 0.89047618 | 0.88333334 |

## Hasil uji skenario 4

|  |  |  |
| --- | --- | --- |
| No\*Kernel* | Linear | RBF |
| 1 | 0.9166667 | 0.9047619 |
| 2 | 0.8809524 | 0.8928571 |
| 3 | 0.9285714 | 0.8928571 |
| 4 | 0.8928571 | 0.8928571 |
| 5 | 0.8452381 | 0.8809524 |
| 6 | 0.9404762 | 0.9285714 |
| 7 | 0.8571429 | 0.8571429 |
| 8 | 0.9047619 | 0.9166667 |
| 9 | 0.8095238 | 0.797619 |
| 10 | 0.9166667 | 0.9047619 |
| Rata-rata | 0.88928572 | 0.88690475 |

## Hasil uji skenario 5

|  |  |  |  |
| --- | --- | --- | --- |
| No\kFold | 1 | 2 | 3 |
| 1 | 0.9281714 | 0.9166667 | 0.9071429 |
| 2 | 0.8690476 | 0.8809524 | 0.8978571 |
| 3 | 0.8842619 | 0.9285714 | 0.8899524 |
| 4 | 0.8925571 | 0.8928571 | 0.8571429 |
| 5 | 0.8452381 | 0.8452381 | 0.8455381 |
| 6 | 0.8946667 | 0.9404762 | 0.9247619 |
| 7 | 0.8824524 | 0.8571429 | 0.8819524 |
| 8 | 0.8928571 | 0.9047619 | 0.9133333 |
| 9 | 0.8410286 | 0.8095238 | 0.8554381 |
| 10 | 0.9045619 | 0.9166667 | 0.8991476 |
| Rata-rata | 0.88348428 | 0.88928572 | 0.88722667 |

1. http://www.pitt.edu/~emotion/ck-spread.htm [↑](#footnote-ref-1)