

**PENDEKATAN PENGENALAN EMOSI MANUSIA MENGGUNAKAN  
*EXTREME LEARNING MACHINE***





**Prasetia Utama Putra  
5235107381**

Skripsi ini ditulis untuk memenuhi sebagian persyaratan dalam  
memperoleh gelar Sarjana Pendidikan




**PENDIDIKAN TEKNIK INFORMATIKA DAN KOMPUTER  
JURUSAN TEKNIK ELEKTRO  
FAKULTAS TEKNIK  
UNIVERSITAS NEGERI JAKARTA**

**2015**

## HALAMAN PENGESAHAN

NAMA DOSEN	TANDA TANGAN	TANGGAL
Hamidillah Aje, M.T (Dosen Pembimbing I)		27-01-2015
Widodo, M. Kom (Dosen Pembimbing II)		27-1-2015

## PENGESAHAN PANITIA UJIAN SKRIPSI

NAMA DOSEN	TANDA TANGAN	TANGGAL
Drs. Wisnu Djatmiko, MT (Ketua Penguji)		
Prasetyo Wibowo Y, M.Eng (Dosen Penguji)		19-1-2015
Lipur Sugiyanta, Ph.D (Dosen Ahli)		16-01-2015

Tanggal Lulus : 7 Januari 2015

## HALAMAN PERNYATAAN

Dengan ini saya menyatakan bahwa:

1. Karya tulis skripsi saya ini adalah asli dan belum pernah diajukan untuk mendapatkan gelar akademik sarjana, baik di Universitas Negeri Jakarta maupun di perguruan tinggi lain.
2. Karya tulis ini adalah murni gagasan, rumusan, dan penelitian saya sendiri dengan arahan dosen pembimbing.
3. Dalam karya tulis ini tidak terdapat karya atau pendapat yang telah ditulis atau dipublikasikan orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan dicantumkan dalam daftar pustaka.
4. Pernyataan ini saya buat dengan sesungguhnya dan apabila di kemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka saya bersedia menerima sanksi akademik berupa pencabutan gelar yang telah diperoleh karena karya tulis ini, serta sanksi lainnya sesuai dengan norma yang berlaku di Universitas Negeri Jakarta.

Jakarta, 7 Januari 2015

Prasetia Utama Putra

5235107381

# PENDEKATAN PENGENALAN EMOSI MANUSIA MENGGUNAKAN

## *EXTREME LEARNING MACHINE*

PRASETIA UTAMA

### ABSTRAK

Pengenalan emosi manusia telah menjadi permasalahan yang cukup mendapat perhatian dalam bidang interaksi manusia dan komputer. Dengan tujuan untuk membentuk sebuah interaksi yang lebih natural diantara manusia dan komputer; komputer haruslah dapat membedakan dan merespons emosi manusia. Dalam penelitian ini, sebuah pendekatan untuk mengenali emosi manusia diusulkan. Pendekatan menggunakan HAAR-*classifier* untuk mendeteksi mata, alis mata dan mulut pada wajah, dan, untuk mengekstrak fitur dari atribut wajah tersebut, metode yang diusulkan menggunakan Gabor *wavelet*. Sebelum mengklasifikasikan fitur, reduksi dimensi fitur menggunakan PCA dilakukan. Pendekatan yang diusulkan menggunakan SLFNs dengan *Extreme Learning Machine* (ELM) untuk mengklasifikasi fitur. Dalam eksperimen ini, pendekatan diuji dalam dua kasus, personalisasi dan generalisasi wajah, dengan sepuluh subjek mengekspresikan enam emosi dasar dan kondisi netral. Performa ELM dievaluasi dengan membandingkan hasil pengujian dengan performa k-NN dan *Support Vector Machine* (SVM). Pada kasus generalisasi wajah, ELM mencapai performa 93.36%, k-NN mencapai performa 87.20%, dan SVM mencapai 85.8%. Sedangkan pada kasus personalisasi wajah, ELM mencapai performa 30.41%, k-NN mencapai 25.41%, dan SVM mencapai 14.06%.

Kata Kunci: *Human Emotion Recognition, Extreme Learning Machine, Gabor Wavelet, Human Computer Interaction, SLFNs, Deteksi Tepi*

**AN APPROACH OF HUMAN EMOTION RECOGNITION USING  
EXTREME LEARNING MACHINE**

**PRASETIA UTAMA**

**ABSTRACT**

Human emotion recognition has been challenging issue in field of human-computer interaction. In order to form an interaction that is more natural between human and computer, the computer should be able to discern and respond to human emotion. In this paper, an approach for recognizing human emotion is proposed. The proposed approach operates HAAR-classifier to detect mouth, eyes, and eyebrow on face. To extract features from them, it uses Gabor wavelet. Before classifying the features, PCA is performed to reduce its dimension. The proposed approach employs SLFNs with ELM as its learning algorithm to classify the features. In this experimental, the proposed approach is tested in two cases, personalize and generalize face case, with ten subjects expressing six basic emotions and neural state. The robustness of ELM is evaluated with comparing it to k-NN and SVM. In generalize face case, the recognition rate of ELM reaches 93.36%, k-NN obtains recognition rate 87.20%, and the performance of SVM reaches 85.8%. While in personalize face case, the ELM obtains 30.41%, the performance of k-NN reaches 25.41, and the recognition rate of SVM reaches 14.06%.

Keywords: Human Emotion Recognition, Extreme Learning Machine, Gabor Wavelet, Human Computer Interaction, SLFNs, Edge Detection

## KATA PENGANTAR

Puji syukur terhadap rahmat dan nikmat yang selalu di berikan Allah SWT, sehingga skripsi penelitian berjudul “Pengenalan Pengenalan Emosi Manusia Menggunakan *Extreme Learning Machine*” dapat di selesaikan. Tidak lupa juga sholawat serta salam kepada Nabi Muhammah, SAW, yang telah memberikan pengaruh besar terhadap peradaban umat manusia.

Penulis menyadari bahwa dalam proses penulisan skripsi ini banyak mengalami kendala, namun berkat berkah dari Allah SWT, serta bantuan, bimbingan, kerjasama dari berbagai pihak, sehingga kendala-kendala yang dihadapi tersebut dapat diatasi. Untuk itu penulis menyampaikan ucapan terima kasih kepada:

1. Pak Hamidillah Ajie, selaku pembimbing yang tak pernah lelah mengarahkan dan memberikan nasihat dalam penyusunan skripsi penelitian ini.
2. Pak Widodo, yang selalu sabar dalam membimbing penulis dalam pemilihan judul dan penyusunan skripsi ini, serta yang telah memberikan pandangan baru kepada penulis terhadap bidang *Soft Computing*.
3. Pak Wisnu Djatmiko, atas komentar dan masukan yang diberikan pada hasil penelitian.
4. Pak Lipur Sugiyanta, atas komentar dan masukan yang diberikan pada hasil penelitian.

5. Pak Prasetyo Wibowo, atas komentar dan masukan yang diberikan pada hasil penelitian.

Dikarenakan keterbatasan waktu yang dimiliki oleh penulis dalam penyusunan skripsi ini, penulis menyadari bahwa skripsi ini masih memiliki banyak kekurangan, oleh karena itu penulis berharap kritik dan saran yang dapat di pergunakan dalam melakukan perbaikan di masa mendatang.

Jakarta, 7 Januari 2015

Prasetia Utama Putra

5235107381

## DAFTAR ISI

HALAMAN PENGESAHAN.....	i
HALAMAN PERNYATAAN .....	ii
ABSTRAK.....	iii
ABSTRACT.....	iv
KATA PENGANTAR .....	v
DAFTAR ISI....	vii
DAFTAR TABEL.....	xi
DAFTAR GAMBAR .....	xii
DAFTAR LAMPIRAN.....	xv
BAB I PENDAHULUAN .....	1
1.1 Latar Belakang .....	1
1.2 Identifikasi Masalah .....	3
1.3 Pembatasan Masalah .....	4
1.4 Perumusan Masalah.....	5
1.5 Tujuan Penelitian.....	5
1.6 Manfaat Penelitian.....	5
BAB II KERANGKA TEORETIK, KERANGKA BERPIKIR, DAN HIPOTESIS PENELITIAN .....	6
2.1 Kerangka Teoretik.....	6
2.1.1 Emosi Manusia .....	6
2.1.2 <i>Face Feature</i> .....	8
2.1.3 <i>Facial Feature Detection</i> .....	10



2.1.3.1	HAAR Like-Features .....	10
2.1.3.2	<i>Rectangle Features</i> .....	11
2.1.3.3	<i>Integral Image</i> .....	13
2.1.3.4	<i>Learning Classification Function</i> .....	14
2.1.3.5	<i>The Attentional Cascade</i> .....	16
2.1.4	<i>Facial Feature Extraction</i> .....	17
2.1.4.1	<i>Convolution Filter</i> .....	17
2.1.4.2	<i>Gabor Wavelet</i> .....	18
2.1.5	<i>Principal Component Analysis</i> .....	20
2.1.6	<i>Artificial Neural Network</i> .....	21
2.1.7	<i>Network Architecture</i> .....	23
2.1.7.1	<i>Learning Method</i> .....	24
2.1.7.2	<i>Extreme Learning Machine</i> .....	25
2.1.8	<i>Root Mean Squared Error</i> .....	27
2.1.9	<i>Support Vector Machine</i> .....	28
2.1.10	<i>k-Nearest Neighbour</i> .....	29
2.1.11	Metode Penelitian Implementasi .....	31
2.2	Kerangka Berpikir .....	31
2.3	Hipotesis Penelitian .....	33
BAB III	METODOLOGI PENELITIAN .....	34
3.1	Tempat dan Waktu Penelitian .....	34
3.2	Metode Penelitian .....	34
3.3	Rancangan Penelitian .....	34
3.3.1	Rancangan Pendekatan .....	34

3.3.2	Ekstraksi Fitur.....	35
3.3.3	Konversi Citra ke <i>Gray-Scale</i> .....	35
3.3.4	<i>Equalization</i> Histogram.....	36
3.3.5	Deteksi Fitur.....	37
3.3.6	<i>Cropping</i> Mata dan Mulut.....	37
3.3.7	Deteksi Tepi Menggunakan Gabor <i>Wavelet</i> .....	38
3.3.8	Reduksi Fitur <i>Space</i> .....	39
3.3.9	Klasifikasi Fitur.....	39
3.4	Instrumen Penelitian.....	40
3.4.1	Perangkat Keras.....	40
3.4.2	Perangkat Lunak.....	40
3.5	Perancangan Uji Coba.....	42
3.6	Prosedur Penelitian.....	412
BAB IV	HASIL PENELITIAN DAN PEMBAHASAN.....	43
4.1	Hasil Penelitian.....	43
4.1.1	Hasil Pengujian Pendekatan.....	43
4.1.1.1	Hasil Pengujian Kasus Personalisasi Wajah.....	43
4.1.1.2	Hasil Pengujian Kasus Generalisasi Wajah.....	45
4.2	Pembahasan.....	46
4.2.1	Pembahasan Hasil Pengujian Pendekatan.....	46
BAB V	KESIMPULAN DAN SARAN.....	48
5.1	Kesimpulan.....	48
5.2	Saran.....	49
DAFTAR PUSTAKA	.....	50

LAMPIRAN 1: HASIL PENGUJIAN PENDEKATAN PADA KASUS

PERSONALISASI WAJAH.....	54
TENTANG PENULIS .....	56

## DAFTAR TABEL

Table 2.1	Motion cues pada wajah untuk enam emosi dasar .....	9
Tabel 2.2	Kemungkinan rectangle – features pada area deteksi berdimensi 5 x 5.....	12
Tabel 3.1	Detail dataset.....	41
Tabel 4.1	Hasil pengujian ELM, k-NN, dan SVM pada kasus personalisasi wajah .....	44
Tabel 4.2	Performa pendekatan pada kasus personalisasi wajah, nilai $C^{25}$ dan $\tilde{L}=1500$ .....	44
Table 4.3	Hasil pengujian ELM, k-NN, dan SVM pada kasus generalisasi wajah .....	45
Tabel 4.4	Performa pendekatan pada kasus generalisasi wajah, nilai $C^{25}$ dan $\tilde{L}=1500$ .....	45

## DAFTAR GAMBAR

Gambar 2.1	Tujuh individu dari ras yang berbeda mengekspresikan berbagai emosi .....	7
Gambar 2.2	<i>Facial features categories</i> : mata, alis mata, hidung, dan mulut .....	9
Gambar 2.3	Contoh <i>rectangle features</i> , nilai dari fitur adalah jumlah <i>pixel</i> pada bagian putih dikurangi jumlah <i>pixel</i> pada bagian hitam. (a) dan (b) merupakan <i>two – rectangle features</i> , (c) dan (d) adalah <i>three – rectangle features</i> , dan (d) menggambarkan <i>four – rectangle features</i> .....	11
Gambar 2.4	Kemungkinan <i>rectangle – features</i> dengan dimensi 2 x 1 pada area deteksi berdimensi 5 x 5.....	12
Gambar 2.5	<i>Integral image</i> .....	14
Gambar 2.6	Algoritma AdaBoost .....	15
Gambar 2.7	Sistematis proses <i>attentional cascade</i> .....	16
Gambar 2.8	(a) <i>source image</i> , (b) <i>kernel</i> , (c) <i>filtered image</i> . Nilai <i>pixel</i> dari <i>filtered image</i> pada posisi (1,1) = $(9*0)+(2*1)+(7*0)+(2*1)$ .....	18
Gambar 2.9	Gabor <i>wavelet</i> dengan tujuh orientasi dan lima skala.....	19
Gambar 2.10	(a) <i>Kernel complex</i> dan (b) <i>visualisasi kernel Gabor Wavelet</i> dengan ukuran 3 x 3, $k_{max}=\pi/2$ , $\mu=3$ , $v=0$ , $\sigma=2\pi$ , dan $f=\sqrt{2}$ .....	20
Gambar 2.11	<i>Model neuron</i> .....	22
Gambar 2.12	(a) <i>Single Layer Neural Networks Model</i> , (b) <i>Multi Layer Neural Networks Model</i> , (c) <i>Recurrent Networks Model</i> .....	24

Gambar 2.13	ELM <i>algorithm</i> .....	26
Gambar 2.14	Ilustrasi SVM .....	28
Gambar 2.15	Algoritma SVM.....	29
Gambar 2.16	Ilustrasi k-NN.....	30
Gambar 2.17	Kerangka berpikir dalam penyusunan skripsi pendekatan pengenalan emosi manusia.....	32
Gambar 3.1	Sistem <i>Human Emotion Recognition</i> .....	35
Gambar 3.2	(a) citra input, (b) hasil konversi citra ke dalam <i>gray-scale</i> .....	36
Gambar 3.3	(a) <i>Gray-scale</i> dari input gambar, (b) citra setelah ekualisasi histogram, (c) histogram-plot dari citra (a), (d) histogram-plot dari citra (b) .....	36
Gambar 3.4	Deteksi wajah, daerah mata dan mulut menggunakan HAAR- <i>like</i> <i>features</i> .....	37
Gambar 3.5	(a) Area mulut, (b) area mata .....	37
Gambar 3.6	Gabor <i>filter bank</i> dengan delapan orientasi untuk mendeteksi tepi pada daerah mata.....	38
Gambar 3.7	Hasil <i>filtering</i> pada daerah mata menggunakan Gabor <i>wavelet</i> untuk enam emosi dasar dan posisi netral, (a) marah, (b) jijik, (c) takut, (d) senang, (e) netral, (f) sedih, (g) terkejut .....	38
Gambar 3.8	Gabor <i>filter bank</i> dengan delapan orientasi untuk mendeteksi tepi pada mulut.....	38
Gambar 3.9	Hasil <i>filtering</i> pada daerah mulut menggunakan Gabor <i>wavelet</i> untuk enam emosi dasar dan posisi netral, (a) marah, (b) jijik, (c) takut, (d) senang, (e) netral, (f) sedih, (g) terkejut .....	39

Gambar 3.10	Arsitektur SLFNs yang dipergunakan dalam pendekatan.....	40
Gambar 3.11	Subjek 1 – 10, gambar (a – j) .....	41
Gambar 4.1	Hasil pengujian pendekatan pada kasus I, (a) Tingkat pengenalan, (b) RMSE pendekatan .....	44

## DAFTAR LAMPIRAN

LAMPIRAN 1: HASIL PENGUJIAN PENDEKATAN PADA KASUS PERSONALISASI WAJAH. ....	54
--	----



# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Komputer merupakan salah satu elemen penting dalam perkembangan masyarakat modern. Tidak dapat dipungkiri bahwa penggunaan komputer telah merambah hampir seluruh komponen kehidupan manusia. Dengan semakin meningkatnya penggunaan komputer, intensitas interaksi antara manusia dan komputer pun meningkat. Oleh karena itu tuntutan dalam menyediakan aplikasi yang dapat memberikan interaksi natural, interaksi yang terjadi antara manusia, kepada pengguna menjadi satu tantangan tersendiri bagi para *software provider* (Sears & Jacko, 2009). Contoh aplikasi yang memberikan interaksi natural adalah CU animator (Ma, Yan, & Cole, 2002), sebuah aplikasi yang memungkinkan pengguna untuk berkomunikasi dengan animasi secara natural. Untuk dapat mewujudkan hal tersebut, sebuah komputer harus dapat merespons apa yang sedang dirasakan oleh penggunanya (Shan, Gong, & McOwan, 2009).

Emosi merupakan sebuah respons dari kejadian yang berkorelasi dengan situasi yang sedang dialami oleh manusia (Newen & Barlassina, 2013). Oleh karena itu, menggunakan emosi sebagai dasar dalam membentuk hubungan yang lebih natural sangat relevan. Dalam mengenali emosi pada manusia, ekspresi pada wajah pengguna dapat dipergunakan sebagai dasar pengklasifikasian. Hal tersebut dikarenakan ekspresi wajah merupakan salah satu bentuk respons yang dikeluarkan oleh manusia ketika mengalami emosi tertentu, dengan kata lain

ekspresi wajah berkaitan erat dengan emosi yang sedang dirasakan oleh manusia (Darwin, 1998).

Penelitian dalam pengenalan emosi telah mendapatkan perhatian khusus dari para pakar di bidang *Computer Vision* dari tahun 1990 (Ioannou, et al., 2005). Hingga saat ini berbagai penelitian dengan berbagai metode telah diusung untuk menangani isu tersebut. Sarode dan Bhatia (2010) mengusung sebuah pendekatan menggunakan proyeksi tepi dan relasi geometri dalam mengenali ekspresi manusia. Sawada dan Samad (2011) menggunakan deteksi tepi dan *Support Vector Machine* (SVM) dalam penelitiannya untuk mengenali emosi manusia. Sedangkan dalam penelitian lain yang dilakukan oleh Quraishi, et al. (2012), digunakan deteksi tepi Sobel dan *Multiple Layer Perceptron* (MLP) dengan *Back Propagation* (BP) dalam pendekatan yang diusungnya. Dalam penelitian terbaru, Uar, et al. (2014) mengusulkan pendekatan menggunakan OSELM dan *curvelet transform*; pendekatan tersebut memiliki hasil yang memuaskan diuji pada JAFFE and Cohn-Kanade *database*.

Salah satu keterbatasan yang dialami oleh metode – metode diatas adalah *unclear result* dalam kasus *limited features attribute*, dikarenakan, sebagian besar dari penelitian sebelumnya menggunakan seluruh atribut wajah untuk mengenali emosi manusia. Oleh karena itu dalam penelitian ini diusung sebuah pendekatan yang ditujukan untuk kasus tersebut. Dalam pendekatan yang diusung, Gabor *wavelet* dipilih sebagai metode untuk mengekstrak fitur pada mata dan mulut. Berdasarkan penelitian oleh Marcelja (1980) dan Daugman (1985), diketahui bahwa fungsi Gabor dapat memodelkan sel sederhana dari *visual cortex* mamalia. Selain itu, performa Gabor *wavelet* dalam proses ekstraksi fitur telah dibuktikan

oleh Samad dan Sawada (2011), dimana hasil dari proses filterisasi mata dan mulut menggunakan Gabor *wavelet* menghasilkan pola unik pada setiap emosi. Sedangkan dalam proses klasifikasi digunakan *Extreme Learning Machine* (ELM) dan *Single Layer Feedforward* (SLFNs). Alasan penggunaan ELM dikarenakan ELM memiliki *time consumption* yang jauh lebih rendah dibanding SVM dan BP, serta performa generalisasi yang tinggi. Dinyatakan dalam sebuah jurnal; dimana ELM dengan *Gaussian Kernel*, *Least Square Support Vector Machine* (LSSVM), dan SVM diuji pada kasus *multiclass* menggunakan *Letter dataset*, dari hasil yang didapatkan ELM lebih cepat 8.06 kali dibandingkan LSSVM dan 7.23 kali dibanding SVM dalam proses *training*. Sedangkan untuk generalisasi, performa ELM jauh lebih baik dibanding SVM dan LSSVM (Huang, Zhou, Ding, & Zhang, 2012).

## 1.2 Identifikasi Masalah

Masalah-masalah yang menjadi perhatian dalam skripsi ini adalah:

1. Pengenalan emosi merupakan salah satu solusi untuk menciptakan interaksi yang natural antara manusia dan komputer.
2. Ekspresi wajah manusia dapat digunakan untuk memprediksi emosi yang sedang dirasakan pengguna.
3. Fitur pada wajah manusia dapat diekstraksi menggunakan deteksi tepi dengan Gabor *wavelet*.
4. *Extreme Learning Machine* (ELM) memiliki performa yang tinggi dalam kasus klasifikasi dibandingkan dengan *Support Vector Machine* (SVM) dan *Back Propagation* (BP), sehingga ELM dapat digunakan

untuk mengklasifikasikan fitur dalam kasus pengenalan emosi manusia.

5. Belum terdapat penelitian mengenai pengenalan emosi manusia menggunakan ekspresi wajah dengan atribut fitur yang terbatas.

### 1.3 Pembatasan Masalah

Untuk dapat mendapatkan hasil yang maksimal, maka dalam penelitian ini dilakukan beberapa pembatasan masalah:

1. Dalam pengenalan emosi, variabel yang dipergunakan dalam klasifikasi adalah ekspresi pada wajah.
2. Dalam penelitian ini, uji coba akan dilakukan menggunakan enam emosi dasar manusia: marah, jijik, takut, senang, sedih, dan terkejut, serta kondisi netral.
3. Foto wajah diambil dari posisi depan.
4. Tingkat *brightness* foto harus cukup tinggi.
5. Pendekatan diuji coba dengan *open database* Feedtum (Wallhoff, 2006).
6. Atribut wajah yang digunakan dalam penelitian ini hanya mata dan mulut.
7. Tidak terdapatnya asesoris pada wajah yang menutupi area mulut dan mata, seperti: kacamata, masker, ataupun cadar.

#### **1.4 Perumusan Masalah**

Berdasarkan latar belakang permasalahan, identifikasi dan pembatasan masalah, maka perumusan masalah yang perlu dijawab dapat diuraikan sebagai berikut:

1. Bagaimana performa dari pendekatan pengenalan emosi manusia menggunakan ELM dalam mengenali enam emosi dasar manusia dan kondisi netral berdasarkan ekspresi pada wajah menggunakan fitur atribut mata dan mulut?

#### **1.5 Tujuan Penelitian**

Penelitian ini memiliki tujuan untuk dapat menghasilkan sebuah metode dengan performa yang baik dalam mengenali emosi manusia, serta dapat diimplementasikan dengan mudah.

#### **1.6 Manfaat Penelitian**

Berdasarkan pada permasalahan yang telah dijelaskan sebelumnya, penelitian ini akan memiliki kontribusi dalam pengembangan aplikasi berbasis *human being*.

**BAB II**  
**KERANGKA TEORETIK, KERANGKA BERPIKIR, DAN**  
**HIPOTESIS PENELITIAN**

**2.1 Kerangka Teoretik**

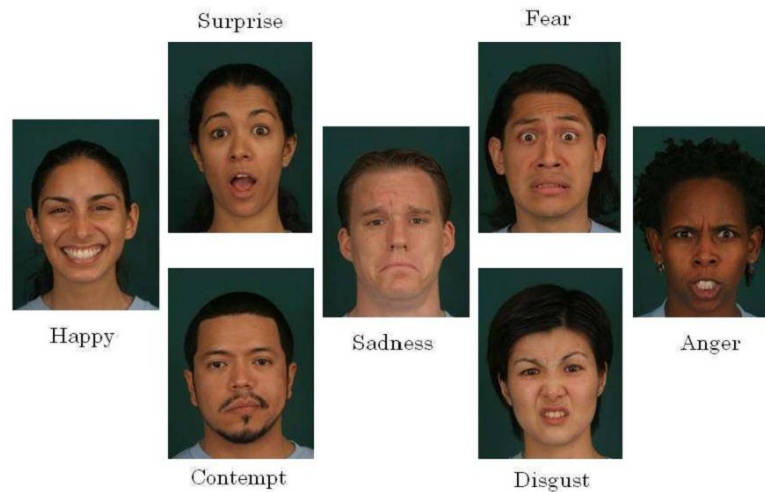
**2.1.1 Emosi Manusia**

Kehidupan setiap manusia pasti diwarnai dengan emosi yang berubah – ubah, entah itu sedih, marah, terkejut ataupun takut. Tidak dapat dipungkiri bahwa emosi merupakan salah satu elemen penting dalam kehidupan manusia, tanpa emosi, manusia tidak akan dapat bertahan hingga saat ini, karena emosi merupakan sebuah aktual respons dari manusia dalam merespons lingkungannya (Matsumoto, 2009). Dalam sebuah artikel berjudul “*The Origin of Universal Human Emotions*”, Matsumoto menyatakan bahwa emosi lebih dari sekedar perasaan, emosi merupakan sebuah reaksi biologis, karena emosi merupakan sebuah respons psikologis dari sistem saraf manusia. Emosi juga merupakan kejadian psikologis, hal ini didasari dari proses terjadinya emosi yang membutuhkan proses psikis. Lebih dari itu, emosi juga digolongkan sebagai sebuah kejadian sosial, dikarenakan emosi merupakan hasil dari interaksi sosial (Matsumoto, 2009). Sedangkan James (1884) mempunyai pendapat bahwa emosi merupakan sebuah reaksi psikologis yang didorong oleh faktor eksternal. Dari pernyataan di atas dapat disimpulkan bahwa emosi adalah respons psikologis yang diungkapkan oleh manusia untuk menyatakan apa yang sedang dirasakannya. Individu akan merespons sebuah kejadian yang sama dengan emosi yang berbeda – beda. Contohnya, ketika individu A merasa senang melihat sebuah pentas musik

Rock, individu B dapat merespons pentas tersebut dengan rasa marah dikarenakan tidak menyukai kebisingan.

Secara universal, terdapat enam emosi yang diklasifikasi dalam kelas emosi dasar: marah, sedih, senang, jijik, terkejut, dan takut (Black & Yacoob, 1995) . Dikatakan universal dikarenakan setiap manusia dapat mengenali dan mengekspresikan keenam emosi tersebut (Matsumoto, 2009). C dan Hager (2003) mengemukakan bahwa manusia mengekspresikan emosi dalam berbagai cara menurut dengan kebudayaannya, namun manusia mengekspresikan emosi – emosi tertentu (emosi dasar) dengan cara yang sama melalui ekspresi wajah.

Dari pernyataan – pernyataan di atas dapat disimpulkan bahwa manusia memiliki ekspresi wajah yang serupa satu sama lain dalam menyatakan emosi dasar (Gambar 2.1).

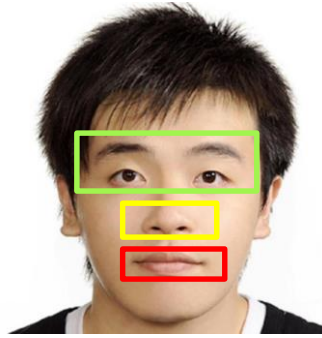


Gambar 2.1. Tujuh individu dari ras yang berbeda mengekspresikan berbagai emosi. Sumber: (Matsumoto, 2009).

### 2.1.2 *Face Feature*

Manusia dapat mengenali wajah seorang individu dengan mudah, namun bagi komputer hal tersebut merupakan hal yang sulit (Liu & Wechsler, 2003). Kemampuan manusia dalam mengidentifikasi wajah sangatlah menakjubkan, manusia dapat mengenali wajah seseorang dari jarak yang jauh. Hal ini dikarenakan kemampuan manusia untuk dapat mengidentifikasi wajah hanya dengan menggunakan karakteristik simetris wajah. Perbedaan bentuk dan struktur hidung, mulut, dagu, dan mata, pada setiap individu, menciptakan perbedaan yang dapat digunakan dalam membedakan wajah satu individu dengan individu lainnya (Gu, Su, & Du, 2003). Secara umum, terdapat tujuh kategori fitur yang terdapat pada wajah manusia: dahi dan alis, mata dan alis mata, hidung, mulut, rahang, pipi, dagu, dan telinga (C & Hager, 2003). Dalam proses identifikasi otomatis wajah manusia, tiga kategori fitur, mata, hidung dan mulut, adalah kategori yang paling banyak dipergunakan, hal ini didasari ketiga kategori tersebut memiliki karakteristik yang sangat berbeda pada setiap individu, selain itu kategori tersebut dapat diekstrak dengan mudah (Gu, Su, & Du, 2003). Tidak hanya dalam pengenalan wajah manusia, ketiga kategori fitur tersebut juga dipergunakan sebagai variabel dalam mengidentifikasi emosi manusia. Dalam jurnal berjudul “*Emotion Recognition Through Facial Expression Analysis Based on Neurofuzzy Network*”, fitur pada mata, alis mata, hidung, dan mulut (Gambar 2.2) dipergunakan sebagai *input* pada neuron (Ioannou, et al., 2005).





Gambar 2.2. *Facial features categories*: mata, alis mata, hidung, dan mulut.  
Sumber: (Lou, 2011).

**TABEL 2.1. *Motion cues* pada wajah untuk enam emosi dasar.**

<b>Emosi</b>	<b><i>Motion Cues</i></b>
<b>Senang</b>	Sudut mulut naik dan turun.
<b>Sedih</b>	Sudut mulut turun. Bagian dalam alis naik.
<b>Terkejut</b>	Alis melengkung. Mata terbuka lebar, dan bagian putih mata lebih terekspos. Posisi rahang turun sedikit.
<b>Takut</b>	Alis naik. Mata terbuka. Mulut terbuka sedikit.
<b>Jijik</b>	Bibir bagian atas naik. Batang hidung berkerut. Posisi pipi naik.
<b>Marah</b>	Alis mata turun. Bibir ditekan rapat. Mata terbuka lebar.

Ekman dan Friesen (1967) menyatakan bahwa kepala (wajah) merupakan bagian yang lebih informatif dalam mengidentifikasi emosi manusia, sedangkan tubuh manusia dianggap mempunyai kemampuan lebih dalam “menggambarkan” intensitas emosi. Hal ini didasari kemampuan wajah untuk membentuk ekspresi yang berbeda dari setiap emosi. Secara singkat *motion cues* pada wajah untuk setiap emosi dapat digambarkan dalam tabel 2.1 (Daniel & Kim, 1996).

### 2.1.3 Facial Feature Detection

Terdapat banyak teknik yang dapat digunakan untuk mendeteksi fitur wajah, mulai dari metode sederhana seperti segmentasi warna hingga metode rumit yang menggabungkan beberapa pendekatan. Akan tetapi hampir seluruh metode tersebut mempunyai masalah dalam hal *time consumption* (Bradski, 1998). Dikemukakan oleh Wilson dan Fernandez, permasalahan tersebut dikarenakan sebuah gambar merupakan kumpulan dari warna dan intensitas cahaya, pemrosesan setiap warna pada setiap *pixel* tentu akan memerlukan waktu yang cukup lama, di samping itu, proses deteksi menggunakan bentuk fitur pada wajah bukanlah proses komputasi yang sederhana, disebabkan bentuk yang bervariasi dari setiap fitur wajah pada setiap individu (Wilson & Fernandez, 2006). Viola dan Jones (2004) menciptakan sebuah algoritma, *Haar-classifiers*, untuk mendeteksi objek visual dengan kemampuan deteksi 15 *frames* per detik. Dikarenakan akurasi yang tinggi dan *time consumption* yang rendah, metode yang diusung oleh Viola dan Jones banyak diterapkan dalam proses pendeteksian wajah dan fitur pada wajah (Wilson & Fernandez, 2006; Sawada & Samad, 2011; Dabhade & Bewoor, 2012).

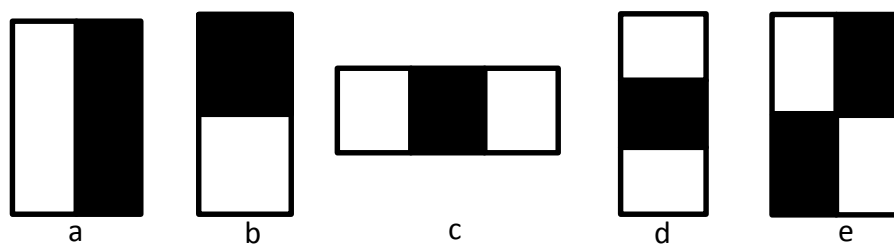
#### 2.1.3.1 HAAR Like-Features

Diusung oleh Viola dan Jones pada tahun 2004, *Haar – like features* menjadi sebuah algoritma yang sering digunakan dalam proses pendeteksian objek pada informasi visual. Hal tersebut dikarenakan performa dari metode tersebut, *less time consumption* dan tingkat pengenalan yang tinggi, sehingga dapat diimplementasikan dalam mengidentifikasi objek secara *real – time*. Kemampuan

Haar – like features dalam pendeteksian objek didukung tiga hal utama dalam metode tersebut: *integral image*, *boosting* untuk *learning classification function*, dan *the attentional cascade* untuk mengurangi *time consumption* dalam proses deteksi (Viola & Jones, 2004).

### 2.1.3.2 Rectangle Features

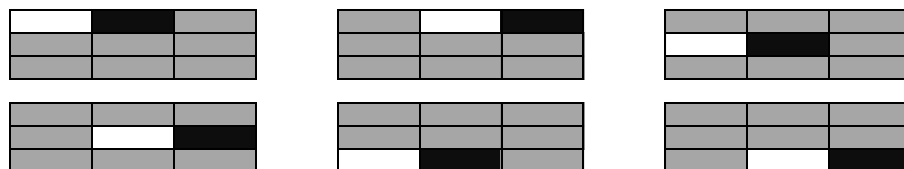
Seperti dijelaskan sebelumnya, bahwa teknik deteksi objek menggunakan *pixel* pada gambar, memerlukan waktu yang cukup lama. Hal ini mendorong para pakar untuk mencari solusi lain dalam teknik mendeteksi objek. Fitur, merupakan solusi dari permasalahan tersebut, *features – based system* jauh lebih cepat dibandingkan *pixel – based system*. Lebih dari itu, *features* dapat meng-*encode adhoc domain knowledge*, dimana hal tersebut sulit dilakukan pada data *train* yang terbatas. Pada penelitian yang dilakukan oleh Viola dan Jones (2004), tiga jenis fitur diimplementasi dalam pendeteksian objek, dua, tiga, dan empat *rectangle features* (Gambar 2.3 (a, b, c, dan e)).



Gambar 2.3. Contoh rectangle features, nilai dari fitur adalah jumlah *pixel* pada bagian putih dikurangi jumlah *pixel* pada bagian hitam. (a) dan (b) merupakan *two – rectangle features*, (c) dan (d) adalah *three – rectangle features*, dan (e) menggambarkan *four – rectangle features*.

**TABEL 2.2. Kemungkinan *rectangle – features* pada area deteksi berdimensi 3 x 3.**

<b>Dimensi</b>	<b>Kemungkinan <i>rectangle – features</i> pada area deteksi</b>
<b><i>Rectangle – features</i> bentuk 2 x 1</b>	
2 x 1	6 penjelasan Gambar 2.4
2 x 2	4
2 x 3	2
<b><i>Rectangle – features</i> bentuk 1 x 2</b>	
1 x 2	6
2 x 2	4
3 x 2	2
<b><i>Rectangle – features</i> bentuk 3 x 1</b>	
3 x 1	3
3 x 2	2
3 x 1	1
<b><i>Rectangle – features</i> bentuk 1 x 3</b>	
1 x 3	3
2 x 3	2
3 x 3	1
<b><i>Rectangle – features</i> bentuk 2 x 2</b>	
2 x 2	4
<b>Total:</b>	<b>40</b>



Gambar 2.4. Kemungkinan *rectangle – features* dengan dimensi 2 x 1 pada area deteksi berdimensi 5 x 5.

Area deteksi berdimensi 3 x 3 mempunyai 40 kemungkinan *rectangle – features* (tabel 2.2), hal ini dapat dikatakan “*overcomplete*”, dengan asumsi dasar bahwa keterangan komplit dapat dikatakan ketika tidak adanya ketergantungan linear di antara basis elemen, selain itu, jumlah dari elemen sama dengan ruang area deteksi (Viola & Jones, 2004), pada kasus ini jumlah elemen ideal adalah 9.

### 2.1.3.3 Integral Image

Seperti dijelaskan sebelumnya bahwa pengaplikasian *rectangle – features* pada metode yang diusung Viola dan Jones untuk mendeteksi sebuah objek menyebabkan *overcomplete*. Namun hal tersebut tidak perlu dikhawatirkan, dikarenakan Viola dan Jones (2004) menggunakan konsep integral *image*, sebuah *intermediate persentation* yang digunakan dalam menghitung nilai dari *rectangle – features*, yang mana hal tersebut membuat proses perhitungan nilai *rectangle – features* menjadi sangat cepat. Nilai dari integral *image* pada titik  $x,y$  merupakan jumlah dari *pixel* bagian atas hingga bagian kiri dari posisi tersebut:

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y') \quad (2.1)$$

dimana  $i(x, y)$  adalah input gambar dan  $ii(x, y)$  adalah integral *image*. Integral *image* pada posisi  $(x, y)$  dapat dihitung menggunakan formula (2.3):

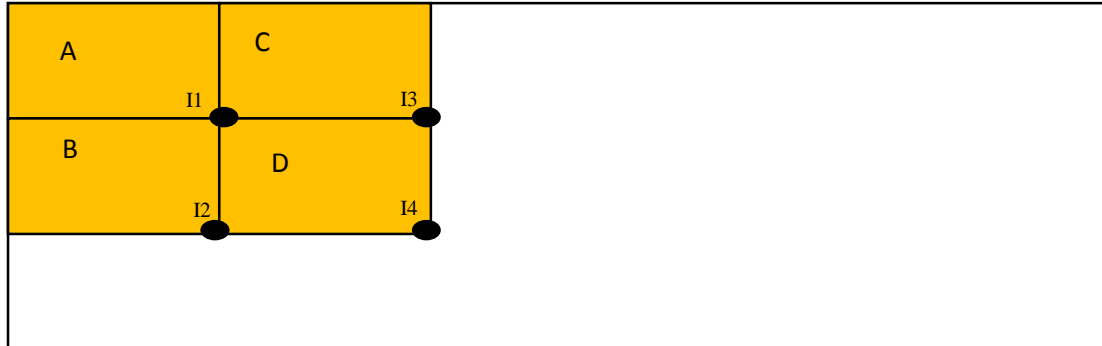
$$s(x, y) = s(x, y - 1) + i(x, y) \quad (2.2)$$

$$ii(x, y) = ii(x - 1, y) + s(x, y) \quad (2.3)$$

dimana  $s(x, y)$  adalah akumulasi dari jumlah nilai *row*,  $s(x, -1) = 0$ , dan  $ii(-1, y) = 0$  (Viola & Jones, 2004).

Berdasarkan teori di atas dapat disimpulkan bahwa nilai integral *image*  $I_4$  adalah jumlah dari pixel pada *rectangle* A, B, C, dan D, sedangkan jumlah dari nilai *source image* pada *rectangle* D adalah  $(I_1 + I_4) - (I_2 + I_3)$ . Dikemukakan oleh Viola dan Jones (2004), perhitungan jumlah setiap *rectangle* dengan integral *image* dapat dilakukan menggunakan empat *array references*,  $I_1$ ,  $I_2$ ,  $I_3$ ,  $I_4$  (Gambar 2.5). Oleh karena itu untuk menghitung *two – rectangle features*

dibutuhkan enam *array references*, delapan *array references* untuk *three – rectangle features*, dan sembilan *array references* untuk *four – rectangle features*.



Gambar 2.5. Integral *image*.

#### 2.1.3.4 Learning Classification Function

Dalam metode yang diusung oleh Viola dan Jones (2004), algoritma AdaBoost digunakan dalam proses *training classifier* dan seleksi *trainingset*. Adaboost merupakan algoritma *supervised learning* (Hu & Maybank, 2008), dimana data *training* harus diberi label sesuai dengan kelas tertentu. Secara umum Adaboost digunakan untuk meningkatkan performa dari algoritma *weak learning* (Viola & Jones, 2004). Dalam proses peningkatan performa yang dilakukan Adaboost terdapat dua proses: (1) pemilihan algoritma *weak learning* secara iteratif dari data, (2) penggabungan kandidat algoritma *weak learning* dan algoritma *strong learning* (Hu & Maybank, 2008).

**Input:** berikan data *train* sebanyak  $N$   $((x_1, y_1), \dots, (x_n, y_n))$ , berikan label 0,1 untuk data sample negatif dan positif, algoritma *weak learning* **WeakLearn**, distribusi  $D$  sebanyak  $N$  example, dan *integer*  $T$ .

**Initialize** weight vector  $w_i^1 = D(i)$  untuk  $i$  sebanyak  $N$

**Loop**  $t = 1, \dots, T$

1. Normalisasi *weights*

$$p^t = \frac{w_i^t}{\sum_{i=1}^n w_i^t}$$

2. Jalankan **WeakLearn** untuk mendapatkan nilai  $h_t$ , dengan  $p^t$  sebagai parameter, output  $h_t: X \rightarrow [1, 0]$ .
3. Hitung *error* dari  $h_t$ ,  $\varepsilon_t = \sum_{i=1}^N p_i^t |h_t(x_i) - y_i|$ .
4. Pilih *classifier*,  $h_t$ , dengan nilai  $\varepsilon_t$  terkecil
5. Set nilai  $\beta_t = \frac{\varepsilon_t}{1-\varepsilon_t}$ .

6. *Update* nilai *weights* vektor untuk terasi selanjutnya:

$$w_i^{t+1} = w_i^t \beta_t^{1-|h_t(x_i)-y_i|}$$

7. **Output**

$$h_f(x) = \begin{cases} 1, & \text{jika } \sum_{t=1}^T (\log 1/\beta_t)(h_t(x)) \geq \frac{1}{2} \sum_{t=1}^T \log 1/\beta_t \\ 0, & \text{sebaliknya} \end{cases}$$

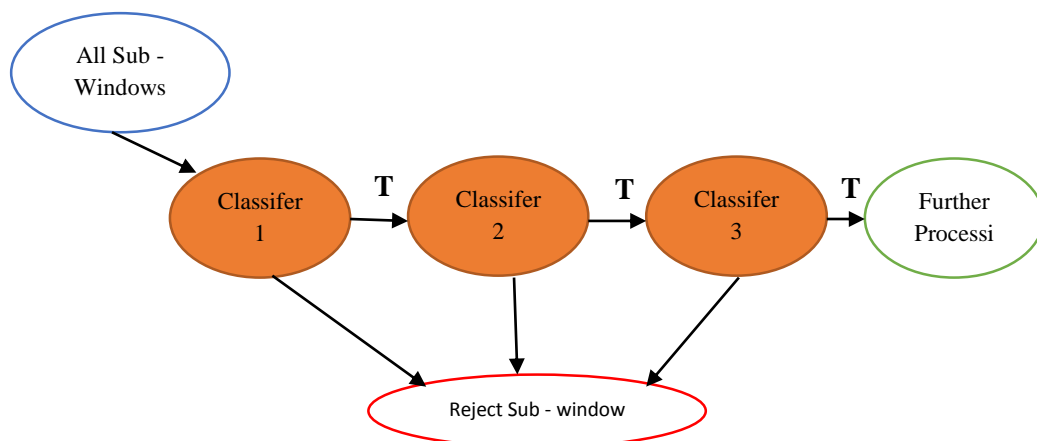
Gambar 2.6. Algoritma AdaBoost. Sumber: (Freund & Schapire, 1997).

Dalam penelitian yang dilakukan oleh Viola dan Jones (2004), algoritma *weak learning* didesain untuk memilih *single rectangle – features* yang memiliki data *train* positif dan negatif yang terpisah secara optimal. AdaBoost menerapkan konsep *boosting* menggunakan sampel, yang biasa digunakan dalam proses *batch*

*learning*. Proses *training* diawali dengan pendefinisian data *train*  $((x_1, y_1), \dots, (x_n, y_n))$ , yang dilanjutkan dengan proses inialisasi *weights*, setelah itu adalah memperbaharui nilai *weights*, dan terakhir adalah klasifikasi algoritma *strong learning* (Gambar 2.6).

### 2.1.3.5 The Attentional Cascade

*Attentional Cascade* ditujukan untuk meningkatkan performa pendeteksian objek pada metode yang diusung oleh Viola dan Jones (2004). Dengan mengadaptasikan konsep *decision tree*, teknik *Attentional Cascade* digunakan dalam mengurangi *false* negatif. Dimulai dari identifikasi oleh *classifier* sederhana terhadap *sub - windows* di tahap pertama, dimana jika pada tahap tersebut menghasilkan keluaran negatif maka tidak akan dilakukan pemrosesan selanjutnya, sedangkan jika keluarannya adalah positif, maka akan dilakukan proses identifikasi oleh *classifier* yang lebih kompleks pada tahap ke selanjutnya (Gambar 2.7).



Gambar 2.7. Sistematis proses *Attentional Cascade*. Sumber: (Viola & Jones, 2004).



Berdasarkan skema *Attentional Cascade* (Gambar 2.7), dapat disimpulkan bahwa algoritma ini bertujuan untuk mengurangi sebanyak mungkin negatif *sub – windows* pada tahap awal, dengan tujuan meningkatkan performa proses deteksi.

#### **2.1.4 Facial Feature Extraction**

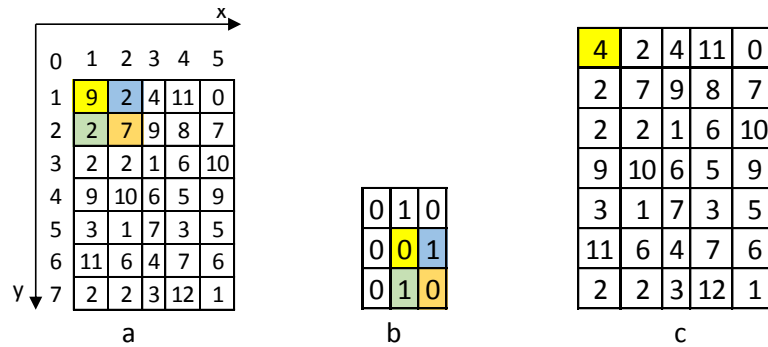
Berbagai macam teknik ekstraksi fitur telah dikembangkan oleh para pakar di bidang *Computer Science*, setiap metode tersebut memiliki kelemahan dan keunggulan masing – masing. Dalam penelitian ini, dipergunakan Gabor *wavelet* dan *Convolution Filter* dalam proses ekstraksi fitur berbasis deteksi tepi, hal ini didasari performa dari metode tersebut yang sudah dibuktikan dalam penelitian yang dilakukan oleh Samad dan Sawada (2011), dari hasil yang didapatkan oleh Samad dan Sawada disimpulkan bahwa fitur yang dihasilkan dari metodenya memiliki informasi yang signifikan, sekalipun telah dilakukan reduksi dimensi pada fitur tersebut. Pada kasus pengenalan emosi dengan menggunakan SVM dan 40 fitur vektor, hasil pengenalan yang didapatkan hingga 91.7%.

##### **2.1.4.1 Convolution Filter**

*Convolution filter* digunakan untuk memodifikasi karakteristik *spatial frequency* dari sebuah citra (Acharya & K. Ray, 2005). Teknik *convolution filter* banyak diterapkan pada proses deteksi tepi, *sharpen*, dan *blur*. *Convolution filter* terdiri dari tiga bagian, *kernel*  $\kappa(x)$ , *source image*  $I(x)$ , dan *filtered image*  $o(x)$ .

$$o(x) = I(x) * \kappa(x) \tag{2.4}$$

*Convolution filter* dilakukan dengan mengubah nilai *pixel source image* pada titik (x ,y) dengan nilai dari hasil perkalian antara *kernel* dan *pixel* (x,y) beserta *neighbouring pixel* dari *pixel* (x,y).



Gambar 2.8. (a) *source image*, (b) *kernel*, (c) *filtered image*. Nilai *pixel* dari *filtered image* pada posisi (1,1) = (9\*0)+(2\*1)+(7\*0)+(2\*1).

#### 2.1.4.2 Gabor Wavelet

Dinyatakan oleh Marcelja (1980) dan Daugman (1985) bahwa sel sederhana dalam *visual cortex* mamalia dapat dimodelkan dengan fungsi Gabor. Gabor *wavelet* merupakan sebuah linear *filter* yang sering digunakan sebagai deteksi tepi, dan ekstraksi fitur dalam *Image Processing* (Sawada & Samad, 2011). Gabor *wavelet*  $\psi_{\mu,\nu}(z)$  dapat diformulasikan sebagai berikut (C. Liu, 2004).

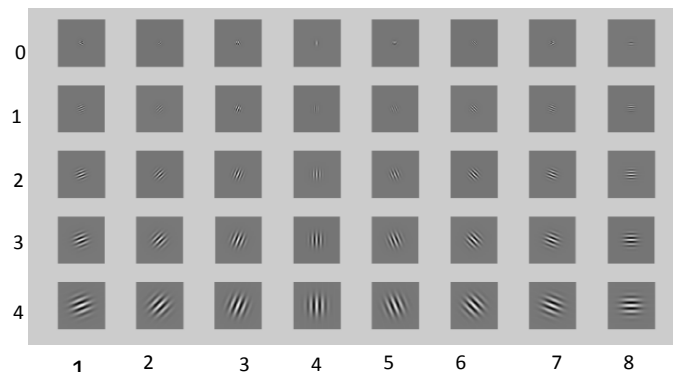
$$\psi_{\mu,\nu}(z) = \frac{\|k_{\mu,\nu}\|^2}{\sigma^2} e^{-\frac{\|k_{\mu,\nu}\|^2 \|z\|^2}{2\sigma^2}} [e^{ik_{\mu,\nu}z} - e^{-\frac{\sigma^2}{2}}] \quad (2.5)$$

Dimana  $z = (x, y)$ ,  $\mu$  dan  $\nu$  melambangkan orientasi dan skala dari Gabor *wavelet*,  $\| \cdot \|$  melambangkan operator *norm*, sedangkan  $k_{\mu,\nu}$  melambangkan vektor *wave*, yang didefinisikan sebagai berikut,

$$k_{\mu,\nu} = k_{\nu} e^{i\phi_{\mu}} \quad (2.6)$$

dimana  $k_v$  adalah  $k_{max}/f^v$ ,  $\phi_u = \pi\mu/8$  jika delapan orientasi dipilih (Zhou & Wei, 2006),  $i$  melambangkan angka imajiner, dan  $\sigma$  merupakan standar deviasi dari fungsi Gaussian pada sumbu x dan y.  $k_{max}$  adalah frekuensi maksimum, dan  $f$  adalah jarak factor dalam *frequency domain*.

Zhou dan Wei menjelaskan bahwa pada banyak kasus penerapan Gabor *wavelet* diterapkan lima skala,  $v = \{0, \dots, 4\}$  dan delapan orientasi,  $\mu = \{1, \dots, 8\}$  (Gambar 2.9). Pada pengekstraksian fitur digunakan orientasi horizontal,  $\mu = 4$ , dikarenakan Gabor *wavelet* dengan orientasi horisontal menghasilkan *discriminative Gabor features* yang lebih baik dibandingkan dengan orientasi lainnya (Sawada & Samad, 2011).

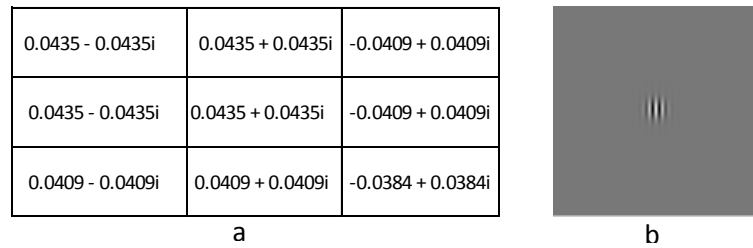


Gambar 2.9. Gabor *wavelet* dengan tujuh orientasi dan lima skala.

Pada penerapannya dalam ekstraksi fitur, Gabor *wavelet* berperan sebagai *kernel* (Gambar 2.10(a)) dalam *convolution filter*, proses tersebut dapat diformulasikan sebagai berikut:

$$\mathbf{o}_k(z) = I(z) * \psi_{\mu,v}(z) \quad (2.7)$$

dimana  $\mathbf{o}_k(z)$  merupakan hasil dari *convolution filter*,  $*$  melambangkan operator *convolution*,  $I(z)$  adalah *gray level distribution* dari sebuah citra,  $\psi_{\mu,v}$  merepresentasikan sebuah *family kernel* Gabor Wavelet (Zhu, Vai, & Mak, 2004).



Gambar 2.10. (a) *kernel complex* dan (b) visualisasi *kernel Gabor wavelet* dengan ukuran  $3 \times 3$ ,  $k_{max} = \pi/2$ ,  $\mu = 3$ ,  $\nu = 0$ ,  $\sigma = 2\pi$ , dan  $f = \sqrt{2}$ .

### 2.1.5 Principal Component Analysis

*Principal Component Analysis* (PCA) merupakan teknik yang biasa digunakan dalam menemukan pola pada data dengan dimensi yang besar. Selain itu, PCA juga digunakan dalam mereduksi dimensi *dataset* dengan tujuan untuk meningkatkan akurasi dengan mengurangi penyimpanan dan kompleksitas komputasi (Acharya & K. Ray, 2005). PCA banyak diaplikasikan dalam permasalahan di bidang *Image Processing*, *Signal Processing*, *Text Analysis*, dan *Retrieval* untuk *Text Mining*. Dalam *Image Processing*, PCA digunakan dalam kompresi, klasifikasi, dan seleksi atau ekstraksi fitur. Konsep PCA sendiri didasarkan pada asumsi terhadap terjadinya korelasi diantar satu fitur dengan fitur lainnya yang cukup sering.

Secara umum, terdapat empat tahapan dalam proses perhitungan PCA dari sebuah citra:

1. Perhitungan matriks kovarians dari *dataset*.
2. Hitung eigenvalues dari matriks kovarians.
3. Hitung eigenvectors dari *eigenvalues* yang paling signifikan.
4. Pembentukan fitur vektor.

Perhitungan matriks kovarians dari *dataset*

Asumsikan bahwa terdapat sebuah citra  $M$  berdimensi  $2 \times 2$ :

$$\begin{bmatrix} 212 & 165 \\ 231 & 217 \end{bmatrix}$$

Kovarians  $A$  dari  $M$  adalah:

$$\begin{bmatrix} 0.1805 & 0.4940 \\ 0.4940 & 1.3520 \end{bmatrix}$$

Hitung *eigenvalues*  $\lambda$  matriks kovarians

$$\begin{bmatrix} -0.0000 \\ 1.5325 \end{bmatrix}$$

Hitung *eigenvector* berdasarkan *eigenvalues* (*eigenvalue*  $\neq 0$ )

*Eigenvector* untuk *eigenvalue* 1.5325

$$(A - \lambda I) \underline{v} = \underline{0}$$

$$\begin{bmatrix} 0.1805 & 0.4940 \\ 0.4940 & 1.3520 \end{bmatrix} - \begin{bmatrix} 1.5325 & 0 \\ 0 & 1.5325 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

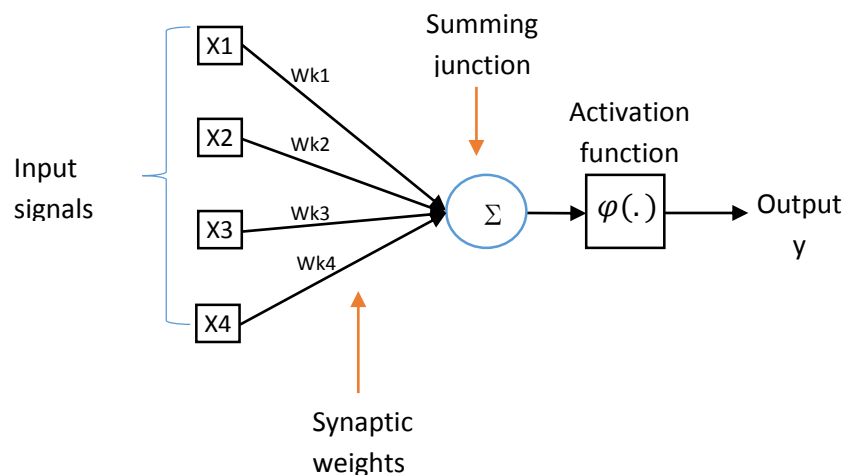
$$\begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} 0.3432 \\ 0.9393 \end{bmatrix}$$

*Final data*

$$\text{Final data} = \text{selected eigenvector} \times \text{adjusted data.}$$

### 2.1.6 Artificial Neural Network

Diusung oleh Warren dan Pitts (Warren & Pitts, 1943), *Artificial Neural Network* (ANN) merupakan sebuah model matematika dari cara kerja *neuron* pada otak. ANN disusun dari *nodes* yang terhubung secara langsung satu sama lainnya. ANN terdiri dari tiga elemen dasar (Haykin, 2009): *synapses*, *adder*, dan *activation function* (Gambar 2.11).



Gambar 2.11. Model neuron.

*Synapse* berperan sebagai penghubung setiap *node* pada *neural networks*. Setiap *synapse* memiliki sebuah *numeric weight*, yang bernilai negatif ataupun positif. *Adder* memiliki fungsi sebagai *linear combiner input signals* dan *weight*. Sedangkan *activation function* berfungsi sebagai *limiter amplitude range* dari nilai *output*.

Secara matematika, model *neuron* (Gambar 2.10) dapat digambarkan dengan persamaan.

$$v_k = \sum_{j=1}^m w_{kj} x_j \quad (2.8)$$

dan

$$y = \varphi(v_k) \quad (2.9)$$

Dimana  $x_j$  adalah *input signals*,  $w_{kj}$  adalah *synaptic weights*,  $v_k$  adalah *linear combiner output*,  $\varphi$  adalah *activation function*, dan  $y$  merupakan keluaran dari *neuron*.

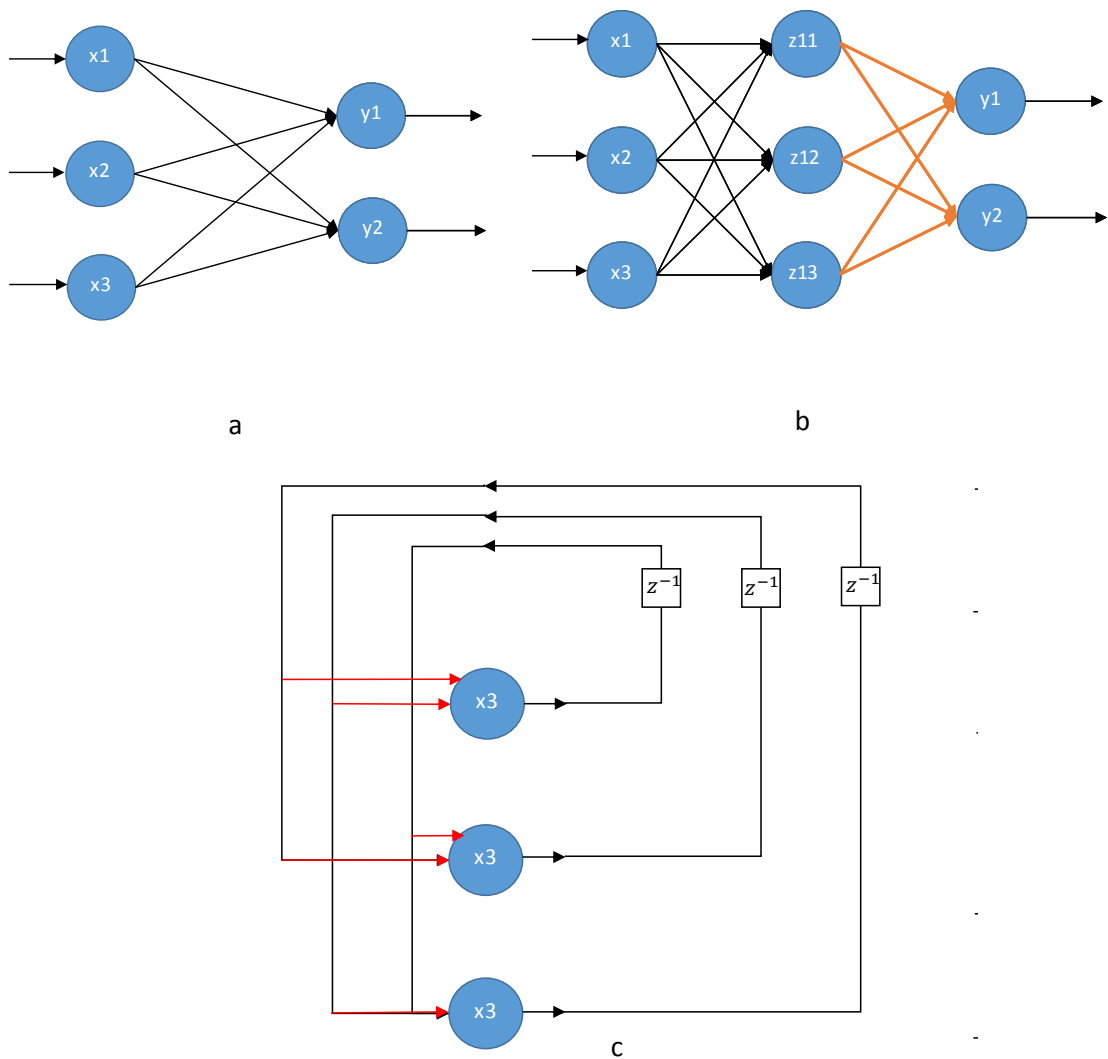
### 2.1.7 Network Architecture

Dalam *Artificial Neural Network* terdapat tiga jenis dasar arsitektur *network*: (1) *Single – Layer Feedforward Networks* (SLFN), (2) *Multilayer Feedforward Networks* (MLN), dan (3) *Recurrent Networks* (Haykin, 2009).

*Single – Layer Feedforward Networks* (Gambar 2.12 (a)), atau disebut juga *perceptrons* (Russell & Norvig, 2010), adalah arsitektur *network* paling sederhana, terdiri dari *input* dan *output layers*, SLFN umumnya digunakan dalam menyelesaikan masalah linear. Dikatakan *single layer* dikarenakan pada SLFN proses komputasi hanya terdapat pada satu *layer*, yaitu *output layer*.

Berbeda dengan SLFN, dimana hanya terdapat *input* dan *output layers*, pada *Multilayer Layer Networks* (MLN) (Gambar 2.12 (b)) terdapat pula satu atau lebih *hidden layers*, tidak hanya itu, MLN juga dapat menyelesaikan permasalahan non – linear. Setiap *layer* pada MLN terhubung langsung dengan *layer* selanjutnya, dimana *input* pada *hidden layer* satu merupakan *output* dari *input layer*, *input hidden layer* dua adalah *output* dari *hidden layer* satu, dan seterusnya (Haykin, 2009).

Arsitektur *Network* yang terakhir adalah *Recurrent Networks* (Gambar 2.12 (c)), berbeda dengan yang lainnya, *recurrent neural networks* memiliki *feedback loop*, yang mana *output* dari *neuron* dikembalikan ke *input signals neuron* tersebut. Pada *recurrent neural networks*, respons terhadap *input* yang diberikan bergantung dari *initial state input* tersebut, yang mungkin memiliki hubungan dengan *input* sebelumnya. Oleh karena itu *recurrent neural networks* memiliki kemampuan *short – term memory* seperti otak.



Gambar 2.12. (a) *Single Layer Neural Networks Model*, (b) *Multi Layer Neural Networks Model*, (c) *Recurrent Networks Model*.

### 2.1.7.1 Learning Method

Terdapat tiga jenis metode *learning* yang digunakan dalam ANN: (1) *Unsupervised learning*, kategorisasi terhadap *input* yang diberikan tanpa adanya kelas tertentu yang berasosiasi dengan *input* tersebut. *Unsupervised learning* sering digunakan dalam proses *clustering*, ekstraksi fitur, dan deteksi kemiripan (Jang, Sun, & Mizutani, 1997). (2) *Supervised learning*, proses klasifikasi sesuai dengan contoh *desired output* (Haykin, 2009). Pada proses *learning* menggunakan



*supervised learning*, nilai parameter pada *networks* akan menyesuaikan sesuai dengan pengaruh dari *training vector* dan *error signal*. Perubahan akan terjadi secara iteratif hingga nilai dari *error signal* sesuai dengan nilai yang diharapkan.

(3) *Reinforcement learning* merupakan proses *learning trial* dan *error*, dimana *network* beradaptasi sesuai dengan hasil evaluasi yang diberikan, *punishment* atau *reward*, melalui interaksi berkelanjutan dengan *environment*.

### 2.1.7.2 Extreme Learning Machine

*Extreme Learning Machine* (ELM) merupakan metode *supervised learning* untuk SLFNs. Diusung oleh Huang, et al. (2004), ELM memiliki *time consumption* yang jauh lebih cepat pada proses *training*, dibandingkan dengan algoritma konvensional untuk SLFNs, hal ini dikarenakan ELM tidak melakukan iteratif *tuning weights*. Selain itu ELM memiliki performa generalisasi yang lebih baik untuk *feedforward neural network*, hal ini dikarenakan ELM dapat mencapai *training error* hingga 0%. Lebih dari itu, ELM juga memiliki kemampuan untuk mencapai *smallest norm of weights*.

Untuk sampel  $(x_i, t_i)$  sebanyak  $N$ , dimana  $x_i = [x_1, x_2, \dots, x_{in}]^T \in R^n$  dan  $t_i = [t_1, t_2, \dots, t_{im}]^T \in R^n$ , SLFNs dengan  $\tilde{N}$  *hidden neuron* dan *activation function*  $g(x)$ , dapat diformulasikan secara matematis dengan

$$\sum_{i=1}^{\tilde{N}} \beta_i g(w_i \cdot x_j + b_i) = o_j, j = 1, \dots, N \quad (2.10)$$

dimana  $w_i = [w_{i1}, w_{i2}, \dots, w_{in}]^T$  adalah *weight vector* yang menghubungkan *hidden neurons* ke  $-i$  dengan *input neuron*, dan  $\beta_i = [\beta_{i1}, \beta_{i2}, \dots, \beta_{im}]^T$  adalah *weight vector* yang menghubungkan *hidden neurons* ke  $-i$  dan *output neuron*,

sedangkan  $b_i$  adalah *threshold hidden neuron* ke  $i$  (Huang, Zhu, & Siew, 2004). Huang, et al. (2004) menyatakan bahwa ELM mampu untuk mencapai *zero error* mempunyai arti  $\sum_{j=1}^n \|o_j - t_j\| = 0$ , dimana  $o_j$  dan  $t_j$  adalah *actual* dan *desired output*. Secara matematis  $t_j$  dapat digambarkan dengan formula,

$$\sum_{i=1}^{\tilde{N}} \beta_i g(w_i \cdot x_j + b_i) = t_j, j = 1, \dots, N \quad (2.11)$$

atau

$$\mathbf{H}\beta = \mathbf{T} \quad (2.12)$$

dimana

$$H(w_1, \dots, w_{\tilde{N}}, b_1, \dots, b_{\tilde{N}}, x_1, \dots, x_N) = \begin{bmatrix} g(w_1 \cdot x_1 + b_1) & \dots & g(w_{\tilde{N}} \cdot x_1 + b_{\tilde{N}}) \\ \vdots & \dots & \vdots \\ g(w_1 \cdot x_N + b_1) & \dots & g(w_{\tilde{N}} \cdot x_N + b_{\tilde{N}}) \end{bmatrix}_{N \times \tilde{N}} \quad (2.13)$$

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_{\tilde{N}}^T \end{bmatrix}_{\tilde{N} \times M} \quad \text{dan} \quad \mathbf{T} = \begin{bmatrix} t_1^T \\ \vdots \\ t_N^T \end{bmatrix}_{N \times M} \quad (2.14)$$

dimana  $\mathbf{H}$  adalah *hidden layer output*,  $\beta$  adalah *weight* yang menghubungkan *hidden* dan *output layer*, dan  $\mathbf{T}$  adalah *output* dari *output layer*. Secara singkat algoritma ELM dapat dituliskan pada Gambar 2.13.

Untuk *trainingset*  $N = \{(x_i, t_i) | x_i \in R^n, t_i \in R^n, i = 1, \dots, N\}$ , *activation function*  $g(x)$  dan *hidden neuron* sebanyak  $\tilde{N}$ .

1. **Assign** arbitrary number untuk  $w_i$  dan  $b_i, i = 1, \dots, \tilde{N}$ .
2. **Calculate** hidden layer output matriks  $\mathbf{H}$ .
3. **Calculate** output weight  $\beta$ :

$$\beta = \mathbf{H}^\dagger \mathbf{T}$$

dimana  $\mathbf{H}$ ,  $\beta$ , dan  $\mathbf{T}$  di formulasikan dalam formula 2.13 dan 2.14.

Gambar 2.13. ELM algorithm. Sumber: (Huang, Zhu, & Siew, 2004).

Mempertimbangkan jumlah *dataset* yang dipergunakan dalam penelitian ini, maka sebuah fungsi optimal diaplikasikan dalam menghitung *weight output* ELM, dengan tujuan untuk meningkatkan performa *learning* (Huang, Zhou, Ding, & Zhang, 2012).

$$\beta = \left( \frac{\mathbf{I}}{C} + \mathbf{H}^T \mathbf{H} \right)^{-1} \mathbf{H}^T \mathbf{T} \quad (2.15)$$

Dimana  $\mathbf{I}$  adalah matriks identitas, dan  $C$  adalah konstanta yang ditetapkan oleh pengguna. Guang Huang-Bin Te al. (2012) menyatakan bahwa ELM adalah sebuah *universal classifier* yang dapat diaplikasikan baik dalam klasifikasi dan regresi. Dalam kasus klasifikasi, ELM cocok digunakan untuk kasus biner dan *multiclass*. Dalam kasus *multiclass* dengan jumlah *datasets training* yang besar, terdapat dua opsi fungsi *output*. Fungsi pertama adalah fungsi dengan *singlet-output node*.

$$f(x) = \text{sign} \left( h(x) \left( \frac{\mathbf{I}}{C} + \mathbf{H}^T \mathbf{H} \right)^{-1} \mathbf{H}^T \mathbf{H} \right) \quad (2.16)$$

Di sisi lain saat *multiclass nodes* diterapkan, label class diprediksi berdasarkan index node dengan nilai *output* tertinggi.

### 2.1.8 Root Mean Squared Error

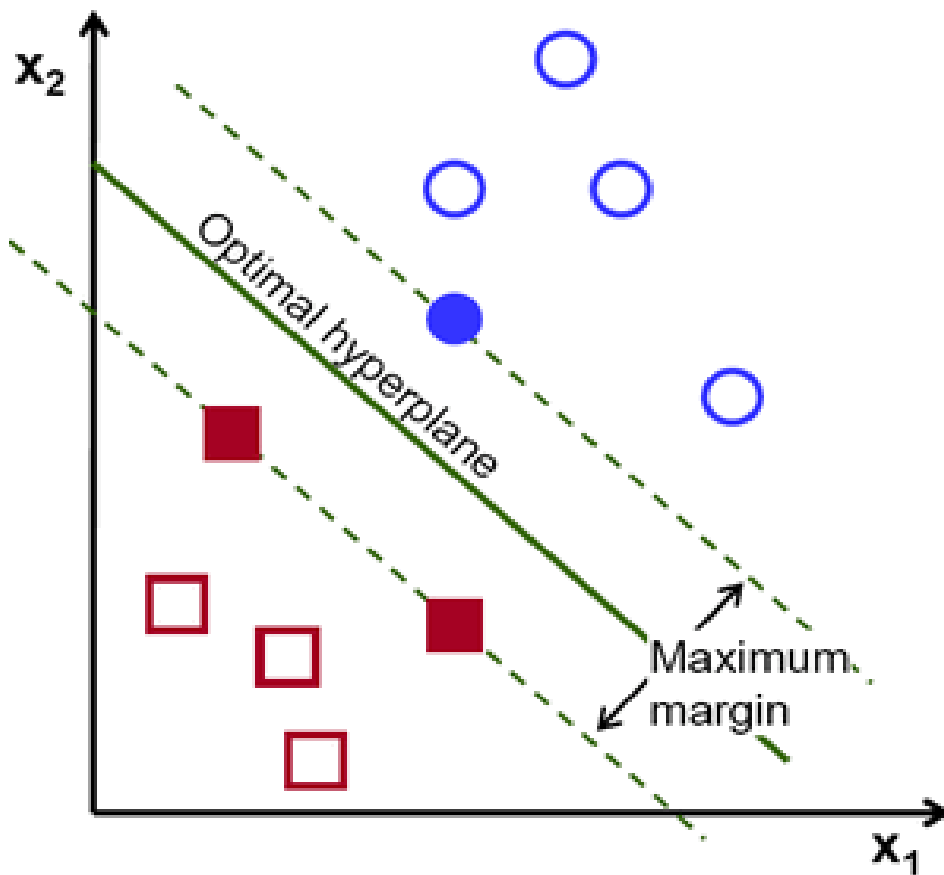
*Root Mean Squared Error* (RMSE), standar deviasi *error*, merupakan sebuah ukuran perbedaan antara *predicted-output* dan *actual-output* pada sebuah model (Kenney & Keeping, 1962).

$$e = \text{predicted output} - \text{actual output} \quad (2.17)$$

$$\text{RMS error} = \sqrt{\frac{e_1^2 + \dots + e_n^2}{n}} \quad (2.18)$$

### 2.1.9 Support Vector Machine

*Support Vector Machine* (SVM) (Cortes & Vapnik, 1995) merupakan *classifier* biner yang menerapkan konsep *hyperplane* dalam proses klasifikasinya. SVM tidak hanya dapat digunakan untuk klasifikasi, tetapi juga regresi (Theodoridis & Koutroumbas, 2009). Dalam penerapannya, walaupun konsep awal SVM sebagai *classifier* biner, SVM juga dapat digunakan untuk melakukan klasifikasi pada kasus *multiple classes* dengan menggunakan teknik *multiple* klasifikasi biner. Dalam proses *training*, SVM memisahkan *labeled* vector fitur dengan *hyperplane*. Yang selanjutnya menerapkan metode *constrained-optimization* untuk menentukan maksimum margin dari dua kelas yang berbeda (Gambar 2.14).



Gambar 2.14. Ilustrasi SVM. Sumber: (Liu, Yuen, & Torralba, 2011).

Secara umum, algoritma untuk SVM dengan menggunakan *kernel trick* dapat diformulasikan sebagai berikut:

<b>Algoritma 2 SVM dengan <i>kernel trick</i> (Jumlah kelas = 2)</b>
<p>Untuk <i>trainingset</i> <math>\tilde{N} = \{\{x_i, t_i\}, x_i \in R^n, t_i \in R^n, i, \dots, N\}</math>, <i>weight vector</i> <math>\mathbf{w}_j = \{w_{1j}, \dots, w_{pj}\}</math>, <i>bias</i> <math>b_j = \{b_{1j}, \dots, w_{pj}\}, j, \dots, P</math>, <math>K(x, z)</math> adalah <i>kernel function</i>, dan kelas <math>\omega_i, i, \dots, 2</math>.</p> <ul style="list-style-type: none"> <li>▪ Set <math>\alpha_i = 0, i = 1, 2, \dots, N</math></li> <li>▪ Repeat <ul style="list-style-type: none"> <li>• count_misclas = 0</li> <li>• For <math>i = 1</math> to <math>N</math> <ul style="list-style-type: none"> <li>◦ If <math>y_i(\sum_{j=1}^N \alpha_j y_j \mathbf{K}(x_i, x_j) + \sum_j \alpha_j y_j) \leq 0</math> then <ul style="list-style-type: none"> <li>– <math>\alpha_i = \alpha_i + 1</math></li> <li>– count_misclas = count_misclas + 1</li> </ul> </li> </ul> </li> <li>• End {for}</li> </ul> </li> <li>▪ Until count_misclas = 0</li> <li>▪ Set <math>x_i \rightarrow \omega_1(\omega_2)</math> if <math>g(x) = (\sum_{j=1}^N \alpha_j y_j \mathbf{K}(x_i, x_j) + \sum_j \alpha_j y_j) &gt; (&lt;) 0</math></li> </ul>

Gambar 2.15. Algoritma SVM. Sumber: (Theodoridis & Koutroumbas, 2009).

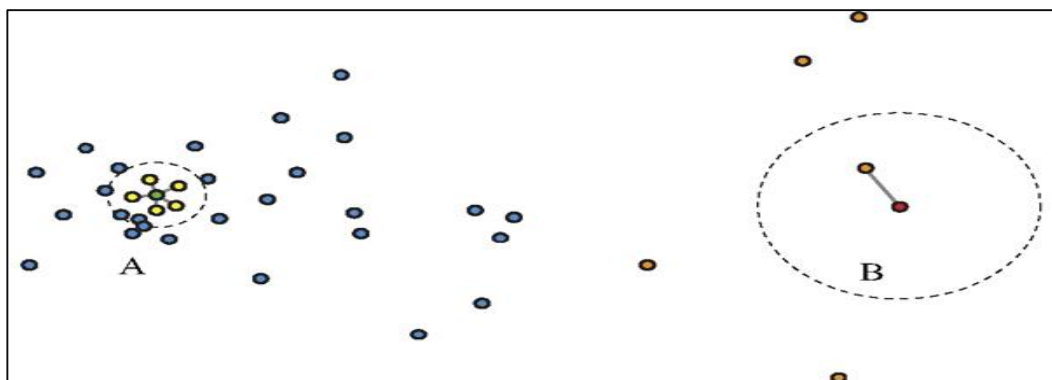
### 2.1.10 k-Nearest Neighbour

*k-Nearest Neighbour* (k-NN) merupakan sebuah *nonparametric classifier*. Di usulkan oleh T. Cover and P. Hart (1967), k-NN merupakan metode *supervised learning* yang memiliki performa yang menjanjikan (Theodoridis &

Koutroumbas, 2009). Seperti kebanyakan metode *learning* lainnya, terdapat dua tahap dalam proses klasifikasi k-NN: (1) *training* dan (2) *testing*.

Dalam proses *training* k-NN tidak dibutuhkan proses penyesuaian parameter seperti yang dilakukan *Back-propagation*. Pada tahap ini, vektor fitur yang telah diberi label dikelompokkan berdasarkan label masing-masing. Oleh karena itu, proses training dalam k-NN dapat dikatakan hanya membutuhkan waktu 0 detik.

Namun, pada proses *testing*, waktu yang dibutuhkan dalam menentukan kelas sebuah *unknown* vektor fitur dapat memakan waktu yang cukup lama, bergantung pada  $k$  yang ditentukan, jumlah vektor fitur dalam dataset, dan elemen dalam vektor fitur. Proses awal dalam melakukan klasifikasi oleh k-NN adalah menentukan nilai  $k$ , jumlah poin terdekat yang diharapkan, tahap selanjutnya adalah melakukan perhitungan jarak *unknown* vektor fitur dan vektor fitur pada *dataset*, dalam proses ini dapat digunakan Euclidian *distance* untuk menghitung jarak antara fitur. Selanjutnya akan dipilih poin yang memiliki jarak terdekat dengan *unknown* vektor fitur. Kelas *unknown* fitur akan ditentukan berdasarkan, probabilitas *labeled* vektor fitur dalam *hypersphere* yang telah ditetapkan (Gambar 2.16).



Gambar 2.16. Ilustrasi k-NN. Sumber: (Liu, Yuen, & Torralba, 2011).

### **2.1.11 Metode Penelitian Implementasi**

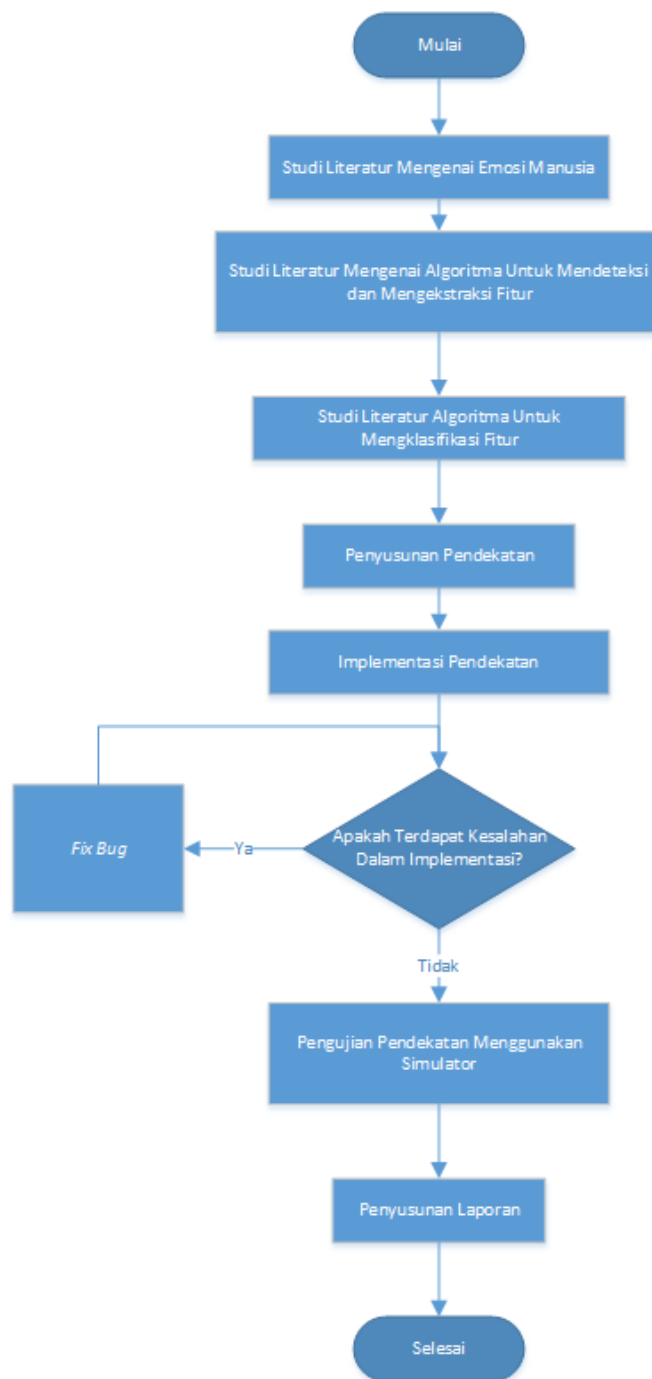
Metode penelitian implementasi merupakan sebuah metode penelitian yang berisi mengaplikasikan solusi baru yang diusulkan oleh peneliti (Berndtsson, Hansson, & Olsson, 2007). Metode penelitian implementasi banyak diterapkan pada penelitian dibidang Ilmu Komputer dan Sistem Informasi. Tujuan dari penggunaan metode penelitian implementasi adalah untuk mendemonstrasikan keunggulan dari solusi yang diusulkan oleh peneliti. Dalam metode penelitian ini, solusi baru diterapkan dalam kondisi tertentu, dan dibandingkan dengan solusi lain yang sudah ada untuk membandingkan keunggulan dan kelemahan dari solusi yang diusulkan.

## **2.2 Kerangka Berpikir**

Dalam penyusunan skripsi pendekatan pengenalan emosi manusia, hal pertama yang perlu dilakukan adalah melakukan studi literatur mengenai emosi manusia. Dalam tahap studi literatur, peneliti mengumpulkan informasi berkaitan dengan emosi manusia, definisi dari emosi manusia, jenis emosi manusia, dan respon manusia terhadap emosi yang sedang dirasakan. Hasil dari proses studi literatur mengenai emosi manusia adalah penetapan variabel yang digunakan untuk mengenali emosi manusia, dan jenis emosi manusia yang akan diestimasi.

Langkah selanjutnya adalah penentuan algoritma untuk mendeteksi dan mengekstraksi fitur. Pada tahap ini peneliti melakukan studi literatur mengenai metode pendeteksi dan pengeksrak fitur, peneliti melakukan perbandingan antara hasil satu metode dan metode lainnya. Setelah menetapkan algoritma untuk mendeteksi dan mengkestrak fitur pada wajah, dilakukan studi literature mengenai algoritma untuk mengklasifikasi fitur. Pada tahap selanjutnya, pendekatan disusun

dengan menggunakan algoritma - algoritma yang telah dipilih sebelumnya. Setelah penetapan variabel dan penyusunan pendekatan, dilakukan pengaplikasian pendekatan menggunakan bahasa pemrograman C++ dan Library OpenCV.



Gambar 2.17. Kerangka berpikir dalam penyusunan skripsi pendekatan pengenalan emosi manusia.



Setelah pendekatan diimplementasi dalam sebuah simulator, maka dilakukan proses uji coba terhadap simulator. Tujuan diadakannya pengujian simulator yaitu untuk mengetahui apakah terdapat kesalahan pada implementasi pendekatan yang diusung. Dalam tahap ini, peneliti mendebug simulator. Jika terdapat kesalahan dalam proses implementasi, maka dilakukan proses *fix bug*.

Jika sudah tidak terdapat permasalahan dalam pengimplementasian pendekatan yang diusung, maka dilakukan proses pengujian terhadap pendekatan yang diusung menggunakan *open database*, dan dilakukan juga perbandingan antara pendekatan yang diusung dengan algoritma lain. Dalam tahap ini peneliti menentukan apakah pendekatan yang diusung memiliki performa yang cukup tinggi atau tidak. Tahap selanjutnya dalam penelitian ini adalah penyusunan laporan hasil penelitian (Gambar 2.17).

### **2.3 Hipotesis Penelitian**

Didasari teori – teori yang dijelaskan sebelumnya, dapat ditarik hipotesis bahwa pengenalan emosi manusia berdasarkan ekspresi pada wajah menggunakan pendekatan yang diusung akan memiliki performa yang baik, *less time consumption* saat proses *training*, dan generalisasi performa yang tinggi saat proses pengenalan.

## **BAB III**

### **METODOLOGI PENELITIAN**

#### **3.1 Tempat dan Waktu Penelitian**

Penelitian ini akan dilakukan di Lab Multimedia Teknik Elektro, Universitas Negeri Jakarta. Dimulai dari bulan Desember 2013 hingga Juni 2014.

#### **3.2 Metode Penelitian**

Metode yang digunakan dalam penelitian ini adalah implementasi. Pendekatan akan diimplementasi dalam sebuah simulator, yang dibuat menggunakan OpenCV (Itseez, 2014) dengan bahasa pemrograman C++. Dalam fase uji coba akan dilakukan dua kali pengujian dengan kasus yang berbeda, kasus personalisasi dan generalisasi wajah. Pendekatan akan diuji terhadap 10 subjek yang mengekspresikan enam emosi dasar dan posisi netral, yang diambil dari *Feedtum database* (Wallhoff, 2006). Dalam kasus personalisasi wajah, digunakan *dataset* dari subjek yang sama, 1-10, pada fase *training* dan *testing*. Sedangkan pada generalisasi wajah, digunakan metode *10-folds cross validasi*, dimana terdapat sepuluh proses *testing* yang menggunakan subjek yang berbeda dalam proses *training* dan *testing*.

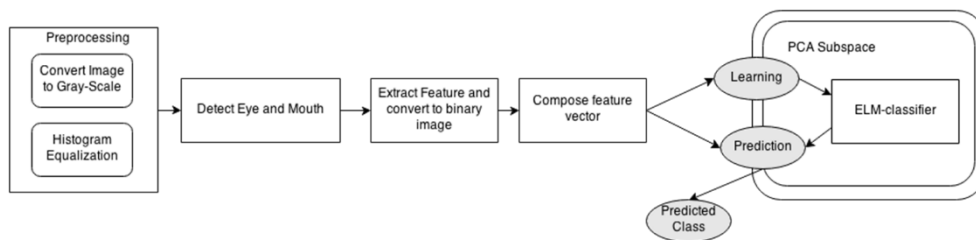
#### **3.3 Rancangan Penelitian**

##### **3.3.1 Rancangan Pendekatan**

Secara general pendekatan yang diusung terdiri dari dua proses, ekstraksi dan klasifikasi fitur. Dalam penelitian ini Gabor *wavelet* digunakan dalam proses

ekstraksi fitur, sedangkan dalam fase klasifikasi fitur, diaplikasikan SLFNs dengan ELM.

Pada tahap *pre-processing*, dilakukan konversi *grayscale* dan ekualisasi histogram pada citra. Proses selanjutnya adalah pendeteksian mata, alis mata, dan mulut yang dilakukan menggunakan *HAAR-like features*. Sebelum melakukan pendeteksian ketiga komponen wajah tersebut, dilakukan pendeteksian wajah terlebih dahulu. Setelah mata, alis mata, dan mulut terdeteksi, pemfilteran menggunakan Gabor *wavelet* dilakukan untuk mendeteksi tepi pada ketiga komponen tersebut. Reduksi dimensi fitur menggunakan PCA dilakukan dengan tujuan untuk mengurangi *time-consumption* pada proses klasifikasi fitur, 132 fitur dipertahankan. Setelah dilakukan reduksi dimensi pada fitur, fitur diklasifikasi menggunakan SLFNs dengan ELM.

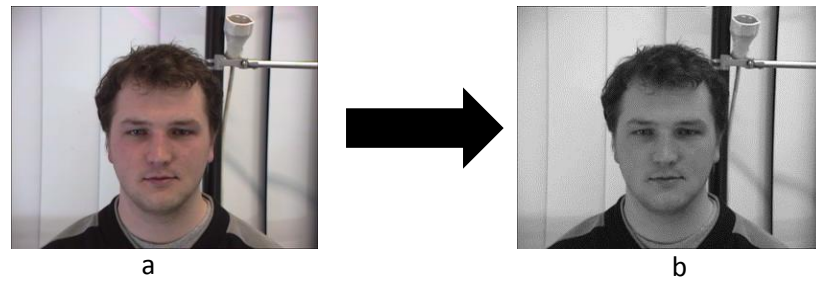


Gambar 3.1. Sistem *Human Emotion Recognition*.

### 3.3.2 Ekstraksi Fitur

#### 3.3.3 Konversi Citra ke *Gray-Scale*

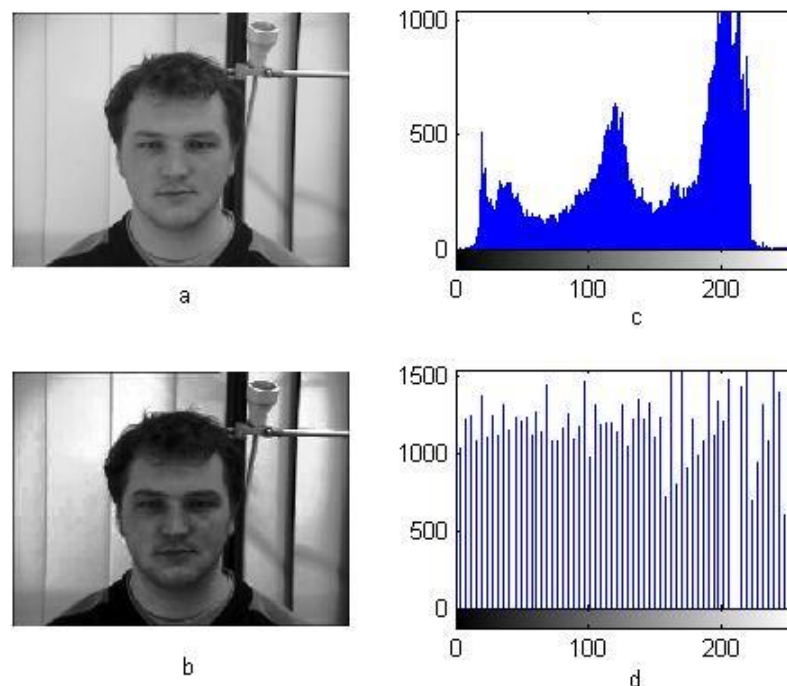
Konversi citra ke *gray-scale* dilakukan dengan cara mengubah nilai setiap komponen *pixel* pada citra ke dalam nilai 1- 255. Konversi citra ke dalam *gray-scale* adalah tahap yang dibutuhkan sebelum melakukan pendeteksian fitur menggunakan *HAAR-classifier*.



Gambar 3.2 (a) citra input, (b) hasil konversi citra ke dalam *gray-scale*.

### 3.3.4 *Equalization Histogram*

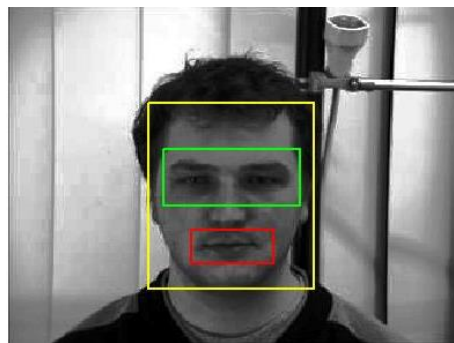
Dengan tujuan untuk meningkatkan performa dalam proses pendeteksian fitur *region* dilakukan ekualisasi histogram pada citra. Ekualisasi histogram meningkatkan kontras sebuah gambar dan membuat sebuah citra memiliki kesamaan nilai histogram (R. C. Gonzales & R. E. Woods, 1992).



Gambar 3.3. (a) *gray-scale* dari *input* gambar, (b) citra setelah ekualisasi histogram, (c) histogram-plot dari citra (a), (d) histogram-plot dari citra (b).

### 3.3.5 Deteksi Fitur

Pendekatan yang diusung menerapkan *HAAR-like features* dalam proses pendeteksian fitur *region*, mata, alis mata, dan mulut. *HAAR-like features* memiliki keunggulan di dalam akurasi mendeteksi objek, dan *time-consumption* yang rendah. Dalam penelitian ini, proses pendeteksian mata, alis mata, dan mulut dibagi dalam dua tahap. Tahap pertama adalah pendeteksian wajah, dan tahap kedua adalah pendeteksian mata, alis mata, dan mulut pada wajah.



Gambar 3.4. Deteksi wajah, daerah mata dan mulut menggunakan *HAAR-like features*.

### 3.3.6 Cropping Mata dan Mulut

Setelah dilakukan pendeteksian bagian mata dan mulut, *cropping* terhadap area tersebut dilakukan. Proses *cropping* dilakukan dengan menyeragamkan tinggi area pada setiap citra *input*. Mata di-*crop* dengan tinggi *25 pixels* dan lebar *84 pixels*, sedangkan mulut dengan tinggi *20 pixels* dan lebar *48 pixels*.



Gambar 3.5. (a) Area mulut, (b) area mata.

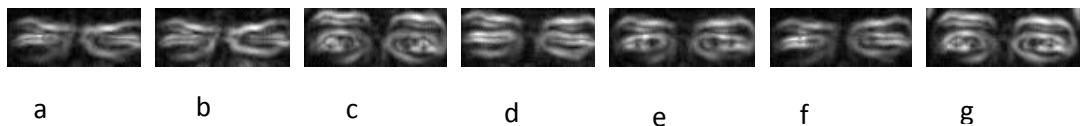
### 3.3.7 Deteksi Tepi Menggunakan Gabor Wavelet

Tepi pada daerah mata dan mulut dideteksi menggunakan teknik *convolution filter* dengan fungsi Gabor sebagai *kernel*. Dalam mendeteksi tepi pada daerah mata, parameter Gabor *wavelet* diset sebagai berikut:  $\nu = 2$ ,  $\mu = \{1, \dots, 8\}$ ,  $\sigma = \pi/2.5$ ,  $k_{max} = \pi/2$ . Filterisasi proses menggunakan Gabor *filter bank* dengan delapan orientasi (Gambar 3.6). *Edge* yang dihasilkan merupakan nilai *magnitude* dari proses filtrasi.



Gambar 3.6. Gabor *filter bank* dengan delapan orientasi untuk mendeteksi tepi pada daerah mata.

Filtrasi proses menghasilkan tepi yang tipis dan kasat mata, kerut pada daerah tersebut pun terlihat jelas.



Gambar 3.7. Hasil *filtering* pada daerah mata menggunakan Gabor *wavelet* untuk enam emosi dasar dan posisi netral, (a) marah, (b) jijik, (c) takut, (d) senang, (e) netral, (f) sedih, (g) terkejut.

Sedangkan dalam mendeteksi tepi pada daerah mulut, parameter Gabor diset sebagai berikut:  $\nu = 2$ ,  $\mu = \{1, \dots, 8\}$ ,  $\sigma = \pi/3$ ,  $k_{max} = \pi/2$  (Gambar 3.8). Perubahan nilai  $\sigma$  menghasilkan tepi yang lebih tipis pada mulut (Gambar 3.6).



Gambar 3.8. Gabor *filter bank* dengan delapan orientasi untuk mendeteksi tepi pada mulut.



Gambar 3.9. Hasil *filtering* pada daerah mulut menggunakan Gabor *wavelet* untuk enam emosi dasar dan posisi netral, (a) marah, (b) jijik, (c) takut, (d) senang, (e) netral, (f) sedih, (g) terkejut.

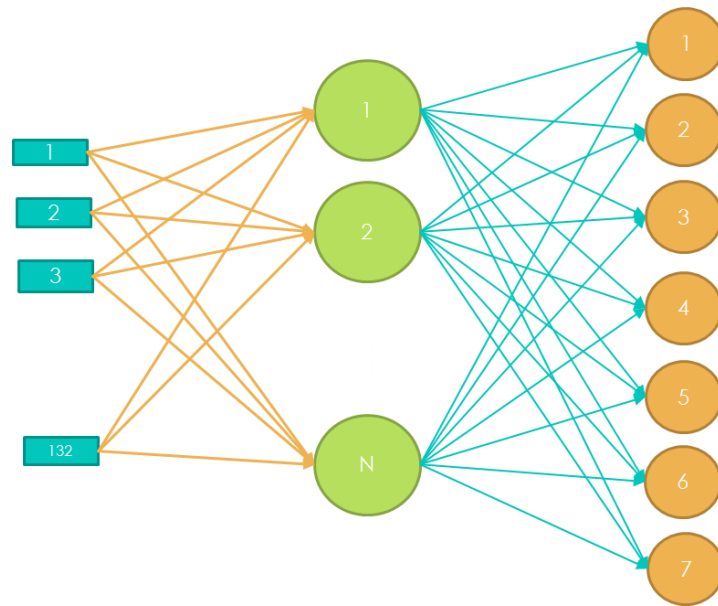
### 3.3.8 Reduksi Fitur *Space*

Langkah terakhir dalam proses ekstraksi fitur adalah reduksi dimensi fitur. Fitur pada mata dan mulut digabungkan dalam satu buah vektor fitur, yang selanjutnya direduksi menggunakan PCA, 132 fitur elemen dipertahakan dalam penelitian ini. Sebelum dilakukan proses reduksi dimensi, nilai fitur diubah ke dalam skala 0 – 1, menggunakan *threshold* sebesar 0.25.

### 3.3.9 Klasifikasi Fitur

Seperti telah dijelaskan sebelumnya bahwa dalam pendekatan yang diusung pada penelitian ini, proses pengenalan emosi manusia dibagi dalam dua tahap, ekstraksi dan klasifikasi fitur. Pada proses klasifikasi fitur digunakan SLFNs dengan ELM sebagai algoritma *learning*.

Arsitektur SLFNs (Gambar 3.10) pada pendekatan terdiri dari: 132 *nodes* pada *input layer*, N *input nodes* pada *hidden layer*, dan tujuh *nodes* pada *output layer*. Jumlah *nodes* pada *input layer* disesuaikan dengan dimensi vektor *input*, yaitu 132. Pada *hidden layer* jumlah *node* ditetapkan sebanyak N, dimana N adalah nilai dari *input* pengguna. Sedangkan tujuh *nodes* pada *output layer* merupakan jumlah kelas yang akan diprediksi dalam pendekatan: (1) marah, (2) takut, (3) jijik, (4) senang, (5) netral, (6) sedih, dan (7) terkejut.



Gambar 3.10. Arsitektur SLFNs yang dipergunakan dalam pendekatan.

### 3.4 Instrumen Penelitian

#### 3.4.1 Perangkat Keras

Dalam penelitian ini digunakan sebuah laptop dengan spesifikasi sebagai berikut:

1. Inter (R) Core(TM) i5 CPU, M 450 @ 2.40GHz 2.40GHz.
2. 4096 MB RAM.
3. VGA: 2736 MB.

#### 3.4.2 Perangkat Lunak

Perangkat lunak yang digunakan dalam penelitian ini meliputi:

1. Windows 7 Ultimate 64 bit.
2. Visual Studio 2012.
3. OpenCV.

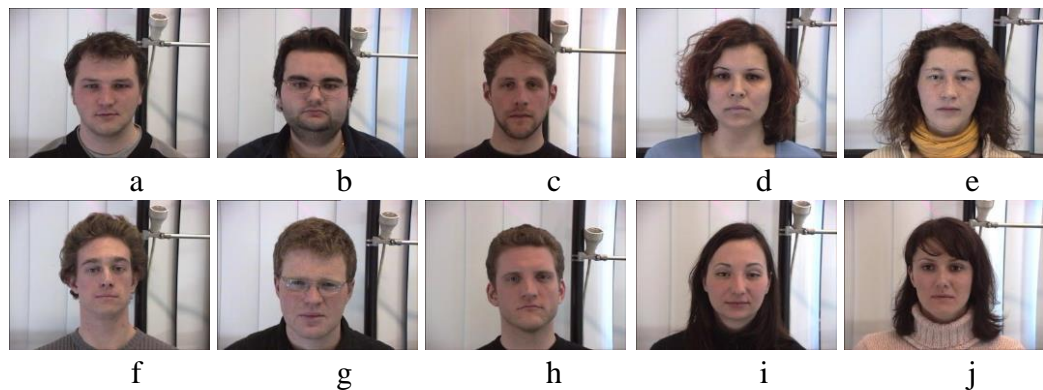


### 3.5 Perancangan Uji Coba

Dalam tahap pengujian, pendekatan diuji dalam dua buah kasus, personalisasi dan generalisasi wajah. Data yang digunakan dalam proses ini adalah serangkaian foto yang diambil dari wajah sepuluh subjek mengekspresikan keenam emosi dasar dan dalam posisi netral. Dalam kasus personalisasi wajah, subjek pada fase *training* dan *testing* merupakan subjek yang sama, 1 – 10 (Gambar 3.11). Data yang digunakan dalam proses *training* berjumlah 560 foto, dan 934 foto pada *testing*. Dari setiap subjek diambil 56 foto untuk proses *training*, dan 93 - 94 foto untuk *testing*. Sedangkan dalam kasus generalisasi wajah, digunakan 10-*folds cross* validasi; data saat proses *training* berjumlah 504 foto dan 56 foto pada *testing* (Tabel 3.1). Selain itu, nilai  $C$  dan jumlah *nodes* pada *hidden layer*  $\tilde{L}$  diset pada nilai  $\{5, 10, 20, 25, 30, 40, 45\}$  dan  $\{100, 200, 400, 600, 800, 1000, 1500\}$  pada kasus personalisasi. Untuk mengevaluasi performa dari ELM, komparasi ELM dengan SVM dan k-NN dilakukan.

**TABEL 3.1. Detail dataset.**

Kasus	#train	Training Subject	#test	Testing Subject	#features	#class
Personalisasi	560	1-10	934	1-10	132	7
Generalisasi	@504	Cross validation	@56	Cross validation	132	7



Gambar 3.11. Subjek 1 – 10, gambar (a – j).

### 3.6 Prosedur Penelitian

1. Implementasi pendekatan pada sebuah simulator yang dibuat menggunakan bahasa pemrograman C++ dan library OpenCV.
2. Pemilihan subjek secara random dari Feedtum (Wallhoff, 2006) *database*.
3. Pengujian pendekatan pada dua buah kasus, generalis dan personalisasi wajah. Pada personalisasi wajah, subjek yang digunakan dalam fase *training* dan *testing* merupakan subjek yang sama, 1 – 10. Sedangkan pada generalisasi, digunakan metode *10-folds cross* validasi.
4. Baik pada kasus personalisasi wajah, nilai  $C$  dan jumlah *nodes* pada *hidden layer*  $\tilde{L}$  diset pada nilai  $\{2^5, 2^{10}, 2^{20}, 2^{25}, 2^{30}, 2^{40}, 2^{45}\}$  dan  $\{100, 200, 400, 600, 800, 1000, 1500\}$ . Sedangkan dalam kasus generalisasi wajah, nilai  $C$  dan  $\tilde{L}$  diset pada nilai  $2^{25}$  dan 1500. Dan untuk SVM, nilai 32 diset untuk  $C$  pada kedua kasus.
5. Analisis performa pendekatan, pengujian pendekatan memfokuskan pada tingkat tingkat pengenalan pendekatan dan *squared error*.
6. Menyimpulkan hasil dari penelitian berdasarkan performa pendekatan.

## BAB IV

### HASIL PENELITIAN DAN PEMBAHASAN

#### 4.1 Hasil Penelitian

##### 4.1.1 Hasil Pengujian Pendekatan

Seperti dijelaskan dalam BAB III, dalam penelitian ini pendekatan yang diusung diuji dalam dua buah kasus personalisasi dan generalisasi wajah. Pendekatan yang diusung diuji menggunakan open database Feedtum (Wallhoff, 2006).

##### 4.1.1.1 Hasil Pengujian Kasus Personalisasi Wajah

Dalam kasus personalisasi wajah, subjek yang digunakan pada saat fase *training* dan *testing* adalah subjek yang sama, subjek 1- 10. Dari hasil pengujian didapatkan bahwa pada kasus ini, ELM mencapai tingkat pengenalan tertinggi, 93.36%, saat nilai  $C = 2^{25}$  dan  $\tilde{L} = 1500$ . Dalam kasus ini, nilai  $\tilde{L}$  mempengaruhi performa dari ELM, secara umum semakin tinggi nilai dari  $\tilde{L}$ , semakin tinggi performa ELM. Selain itu, ELM juga memperoleh *squared – error* yang rendah.

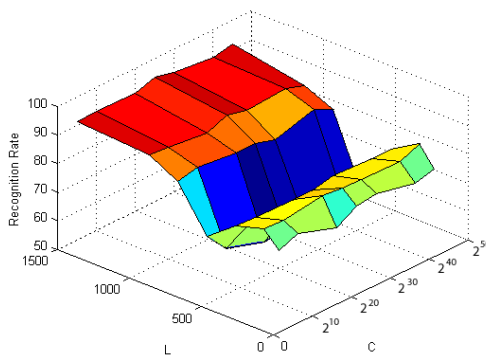
Dibandingkan dengan k-NN dan SVM, performa dari ELM lebih unggul. SVM dengan 406 *support vectors* mempunyai waktu *training* 568.42 lebih cepat dibandingkan ELM (perlu diperhatikan bahwa proses perhitungan inversi sebuah matriks pada OpenCV memerlukan waktu yang cukup lama ketika bekerja dengan matriks berskala besar). Namun, tingkat pengenalan SVM lebih rendah dibanding dengan ELM. k-NN memiliki tingkat pengenalan lebih tinggi dibanding SVM, namun dibandingkan dengan ELM, performa dari k-NN tidaklah lebih baik.

**TABEL 4.1. Hasil pengujian ELM, k-NN, dan SVM pada kasus personalisasi wajah.**

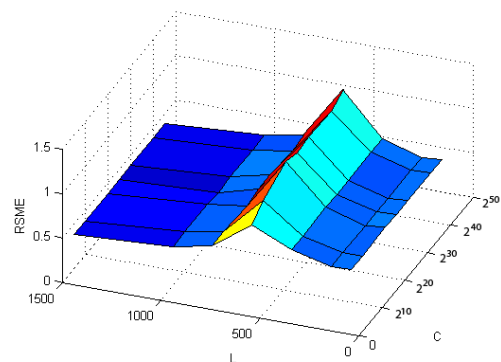
Metode	Testing Result		Training Time
	RMSE	Recognition Rate	
ELM	0.378	93.36%	108.57
k-NN	0.361	87.20%	-
SVM	0.343	85.8%	0.191

**TABEL 4.2. Performa pendekatan pada kasus personalisasi wajah, nilai  $C = 2^{25}$  dan  $\tilde{L} = 1500$ .**

		Actual Class						
		Marah	Jijik	Takut	Senang	Netral	Sedih	Terkejut
Predicted Class	Marah	<b>18.75%</b>	21.25%	15.00%	15.00%	7.50%	6.25%	16.25%
	Jijik	10.00%	<b>23.75%</b>	12.50%	23.75%	3.75%	8.75%	17.50%
	Takut	20.00%	11.25%	<b>17.50%</b>	0.00%	13.75%	11.25%	26.25%
	Senang	2.50%	15.00%	5.00%	<b>57.50%</b>	0.00%	0.00%	20.00%
	Netral	17.50%	15.00%	13.75%	1.25%	<b>25.00%</b>	5.00%	22.50%
	Sedih	16.25%	20.00%	15.00%	11.25%	11.25%	<b>15.00%</b>	11.25%
	Terkejut	10.44%	9.98%	9.22%	8.03%	1.42%	5.53%	<b>55.38%</b>



A



b

Gambar 4.1. Hasil pengujian pendekatan pada kasus I, (a) Tingkat pengenalan, (b) RMSE dari pendekatan.



## 4.2 Pembahasan

### 4.2.1 Pembahasan Hasil Pengujian Pendekatan

Dalam penelitian ini dihasilkan sebuah pendekatan untuk mengenali enam emosi dasar pada manusia. Pendekatan yang diusung terdiri dari enam tahap:

1. Perubahan citra ke dalam *gray-scale*, komponen *pixel* pada citra diubah ke dalam rekanan 1-255.
2. Tahap kedua adalah ekualisasi histogram, tingkat kontras pada citra ditingkatkan untuk memaksimalkan hasil dalam proses pendeteksian mulut dan mata pada wajah.
3. Pendeteksian mulut dan mata pada wajah menggunakan HAAR-*classifier*, mata dan mulut dideteksi setelah proses deteksi terhadap wajah selesai dilakukan.
4. Ekstraksi fitur menggunakan Gabor *wavelet*, dalam proses ini digunakan delapan *kernel* untuk membentuk sebuah *wavelet* pada proses deteksi tepi.
5. Reduksi dimensi menggunakan PCA, dimensi fitur direduksi menggunakan PCA. Sebelum proses ini dilakukan, nilai citra diubah dalam *binary-scale*.
6. Klasifikasi fitur menggunakan *Extreme Learning Machine* (ELM), fitur diklasifikasi menggunakan SLFNs dengan ELM.

Pendekatan yang diusung diuji dalam dua kasus: personalisasi dan generalisasi wajah menggunakan *open database* Feedtum (Wallhoff, 2006). Berdasarkan hasil pengujian yang dijelaskan pada sesi 4.1, diketahui bahwa

pendekatan yang diusung memiliki performa yang tinggi diimplementasi pada kasus personalisasi wajah, tingkat pengenalan pendekatan yang diusung mencapai 93.36%, secara umum tingkat pengenalan dari ELM meningkat seiring peningkatan nilai  $\tilde{L}$ . Selain itu, jika dibandingkan dengan k-NN dan SVM yang masing – masing memiliki tingkat pengenalan 87.20% dan 85.8%, ELM memiliki tingkat pengenalan yang lebih tinggi. Namun, untuk kasus generalisasi wajah pendekatan yang diusung tidak memiliki performa yang cukup tinggi, dikarenakan tingkat pengenalan yang diusung hanya mencapai 30.41%. Tidak hanya ELM, tetapi k-NN dan SVM juga memiliki tingkat pengenalan yang rendah, 25.41% dan 14.06%.

## BAB V

### KESIMPULAN DAN SARAN

#### 5.1 Kesimpulan

Penelitian dalam pengenalan emosi manusia telah mendapatkan banyak perhatian dari para peneliti di bidang Ilmu Komputer. Secara umum, metode pengenalan terdiri dari dua tahap, ekstraksi dan klasifikasi fitur. Dalam pendekatan yang diusung, Gabor *wavelet* digunakan sebagai metode ekstraksi fitur. Tepi yang dihasilkan oleh Gabor *wavelet* tipis dan dapat dilihat secara kasat mata. Selain itu, tepi menunjukkan pola yang unik pada setiap pengekspresian emosi. Dalam tahap klasifikasi, pendekatan menggunakan SLFNs dengan ELM sebagai algoritma *learning* untuk mengklasifikasi fitur. Metode diuji dalam dua kasus, personalisasi dan generalisasi wajah, menggunakan *open database* Feedtum (Wallhoff, 2006).

Dalam kasus personalisasi wajah, tingkat pengenalan pendekatan mencapai 93.36%. Tingkat pengenalan meningkat seiring dengan peningkatan nilai  $\tilde{L}$  pada ELM. Lebih dari itu, pendekatan tidak hanya mencapai performa yang tinggi, tetapi juga memperoleh *squared – error* yang rendah. Dibandingkan dengan SVM dan k-NN yang masing – masing memiliki tingkat pengenalan 87.20% dan 85.8%, ELM memiliki performa yang lebih tinggi. Namun pada kasus generalisasi wajah, performa dari pendekatan yang diusulkan tidak cukup tinggi. Lebih dari itu, pada kasus ini, tidak satupun dari metode yang diuji memiliki tingkat pengenalan lebih dari 31%. Meskipun performa dari pendekatan yang diusung tidak cukup tinggi



diuji dalam kasus generalisasi wajah, pendekatan yang diusung memiliki performa yang tinggi pada kasus personalisasi wajah.

## 5.2 Saran

Seperti dijelaskan sebelumnya bahwa metode yang diusung dalam penelitian ini tidak memiliki performa yang cukup tinggi dalam kasus generalisasi wajah. Dari hasil pengujian dan perbandingan dengan metode lain, dapat diketahui secara jelas bahwa terdapat isu dalam proses seleksi fitur. Dikarenakan, pendekatan yang diusulkan menggunakan informasi tepi yang terdapat pada mata dan mulut secara langsung. Atau secara umum, dapat dikatakan bahwa informasi personal, *uniqueness* atribut pada mata dan mulut, setiap subjek mempengaruhi tingkat generalisasi dari pendekatan yang diusulkan.

Kami berharap dimasa yang akan datang, dapat melakukan penelitian lebih lanjut untuk menyelesaikan permasalahan diatas. Pendekatan Geometri dan *Context Free Grammar* dapat dipertimbangkan sebagai solusi untuk mengekstrak fitur wajah tanpa membawa serta informasi personal.

## DAFTAR PUSTAKA

- Acharya, T., & K. Ray, A. (2005). *Image Processing, Principles and Applications*. United States of America: A Wiley-Interscience Publication.
- Berndtsson, M., Hansson, J., & Olsson, B. (2007). *Thesis Projects: A Guide for Students in Computer Science and Information Systems (Second Edition)*. London: Springer-Verlag.
- Black, M. J., & Yacoob, Y. (1995). Recognizing facial expressions under rigid. *Proceedings of the IEEE International Workshop on Automatic*, 12-17.
- Bradski, G. (1998). Computer vision face tracking for use in a perceptual user interface. *Intel Technology Journal*.
- C, J., & Hager . (2003). *Human Face*. Dipetik 10 22, 2013, dari Introduction To Visage Demonstration Applet: <http://face-and-emotion.com/dataface/visage/visage.jsp>
- C. Liu. (2004). Gabor-based kernel PCA with fractional power. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26, 572–581.
- Cortes, & Vapnik. (1995). Support-vector networks. *Machine Learning*, 20, 273-297.
- Daniel, D. H., & Kim, H. (1996, 20 4). *Modeling Six Universal Emotions*. Dipetik 10 20, 2013, dari Animating Human Facial Expressions: <http://www.nbb.cornell.edu/neurobio/land/oldstudentprojects/cs490-95to96/hjkim/deidrell.html>
- Darwin, C. (1998). *The expression of emotion in man and animals*. New York. Oxford: Oxford University Press.
- Daugman, J. (1985). Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *JOSA A*, 2, 1160–1169.
- Ekman, P., & Friesen, V. W. (1967). Head and Body Cues in The Judgment of Emotion: A Reformulation. *Perceptual and Motor Skills*, 711-724.
- Freund, Y., & Schapire, R. E. (1997). A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *Journal of Computer and System Sciences*, 119-139.
- Gu, H., Su, G., & Du, C. (2003). Feature Points Extraction from Faces . *Image and Vision Computing NZ*, 154-158.

- Haykin, S. (2009). *Neural Networks and Learning Machines*. Pearson Education: New Jersey.
- Hu, W., & Maybank, S. (2008). AdaBoost-Based Algorithm for Network Intrusion Detection. *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART B: CYBERNETICS*, 577-583.
- Huang, G. B., Zhu, Q. Y., & Siew, C. K. (2004). Extreme Learning Machine: A New Learning Scheme of Feedforward Neural Networks. *Neurocomputing*, 489-501.
- Huang, G.-B., Zhou, H., Ding, X., & Zhang, R. (2012). Extreme Learning Machine for Regression and Multiclass Classification. *IEEE TRANSACTION ON SYSTEMS, MAN, AND CYBERNETICS-PART B: CYBERNETICS*, 42, 513-528.
- Ioannou, S. V., Raozaiou, A. T., Tzouvaras, V. A., Mailis, T. P., Karpouzis, C. K., & Kollias, S. D. (2005). Emotion Recognition Through Facial Expression Analysis Based on a Neurouzzy Network. *Neural Networks*, 423-435.
- Istiany, A., Yusro, M., Nasution, N., Amalia, R., & Muksin. (2012). *Buku Pedoman Skripsi / Komprehensif/ Karya Inovatif (S1)*. Jakarta: Fakultas Teknik Universitas Negeri Jakarta.
- Itseez. (2014, 09 12). *Open Computer Vision*. Diambil kembali dari Open Computer Vision: <http://opencv.org/>
- James, W. (1884). What is an Emotion? *Mind*, 9, 188-205.
- Jang, J.-S., Sun, C.-T., & Mizutani, E. (1997). Unsupervised Learning and Other Neural Networks. Dalam J.-S. Jang, & E. Mizutani, *Neuro - Fuzzy and Soft Computing* (hal. 301-328). United States of America: Prentice - Hall.
- Kenney, J., & Keeping, E. (1962). Root Mean Square. Dalam J. F. Kenney, & E. S. Keeping, *Mathematics of Statistics, Pt. 1, 3rd ed* (hal. 59-60). NJ: Van Nostrand: Princeton. Diambil kembali dari MathWorld--A Wolfram Web Resource.
- Liu, C., & Wechsler, H. (2003). Independent Component Analysis of Gabor Features for Face Recognition. *IEEE Trans. Neural Networks*, 919-928.
- Liu, C., Yuen, J., & Torralba, A. (2011). Nonparametric scene parsing via label transfer. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* (hal. 2368-2382). IEEE.
- Lou, W. (2011). *Human Face to Face Morphing Project*. Diambil kembali dari Vimeo: <http://vimeo.com/27800910>

- Ma, J., Yan, J., & Cole, R. (2002). CU Animate Tools for Enabling Conversations with Animated Characters. *ICSLP-2002: Inter. Conf. on Spoken Language Processing*, (hal. 197-200). Denver.
- Marcelja, S. (1980). A Theory for Multiresolution Signal Decomposition: The Wavelet Representation. *IEEE Trans, Pattern Analysis and Machine Intelligence*, 70, 674-693.
- Matsumoto, D. (2009). *The Origin of Universal Human Emotions*. San Francisco.
- Newen, A., & Barlassina, L. (2013). The Role of Bodily Perception in Emotion: In Defense of an Impure Somatic Theory. *Philosophy and Phenomenological Research*.
- Quraishi, M. I., Choudhury, J. P., De, M., & Chakraborty, P. (2012). A framework for the Recognition of Human Emotion using Soft Computing models. *Internal Journal of Computer Application*, 0975-8887.
- R. C. Gonzales, & R. E. Woods. (1992). *Digital Image Processing*. MA: Addison Wesley.
- Russell, S., & Norvig, P. (2010). *Artificial Intelligence A Modern Approach*. New Jersey: Pearson Education.
- Sarode, N., & Bhatia, S. (2010). Facial Expression Recognition. *International Journal on Computer Science and Engineering*, 02, 1552-1557.
- Sawada, H., & Samad, R. (2011). Edge-Based Facial Feature Extraction Using Gabor Wavelet and Convolution Filters. *Machine Vision Applications*, 13-15.
- Sears, A., & Jacko, J. (2009). *Human Computer Interaction-Fundamentals*. Broken Sound Parkway NW: Taylor & Francis Group.
- Shan, C., Gong, S., & McOwan, P. (2009). Facial expression recognition based on Local Binary Patterns - A comprehensive study. *Image and Vision Computing*, 803-816.
- T.Cover, & P.Hart. (1967). Nearest neighbor pattern classification. *Information Theory, IEEE Transaction on*, 13, 21-27.
- Theodoridis, S., & Koutroumbas, K. (2009). *Pattern Recognition*. United States of America: Elsevier Inc.
- Uar, A., Demir, Y., & Gzeli, C. (2014). A new facial expression recognition based on curvelet transform and online sequential extreme learning machine initialized with spherical clustering. *Neural Computing and Applications*, 1-12.

- Viola, P., & Jones, M. (2004). Rapid Object Detection using a Boosted Cascade of Simple Features. *ACCEPTED CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION*.
- Wallhoff, F. (2006). Facial Expressions and Emotion Database. Diambil kembali dari <http://www.mmk.ei.tum.de/~waf/fgnet/feedtum.html>
- Warren, S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4), 115-133.
- Wilson, P. I., & Fernandez, J. (2006, April 21). FACIAL FEATURE DETECTION USING HAAR CLASSIFIERS. 2006. Consortium for Computing Sciences in Colleges.
- Zhou, M., & Wei, H. (2006). Face Verification Using Gabor Wavelets and AdaBoost. *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, 404 - 407 .
- Zhu, J., Vai, M. I., & Mak, P. U. (2004). *A New Enhanced Nearest Feature Space (ENFS) Classifier for Gabor Wavelets Features-Based Face Recognition*. Berlin Heidelberg: Springer-Verlag.

**LAMPIRAN 1: HASIL PENGUJIAN PENDEKATAN PADA KASUS  
PERSONALISASI WAJAH.**

No	C	L	RMSE	<i>Recognition Rate</i>
1	2 <sup>5</sup>	100	0.545	0.744
2	2 <sup>10</sup>	100	0.538	0.770
3	2 <sup>20</sup>	100	0.563	0.723
4	2 <sup>25</sup>	100	0.547	0.767
5	2 <sup>30</sup>	100	0.550	0.777
6	2 <sup>40</sup>	100	0.552	0.745
7	2 <sup>45</sup>	100	0.543	0.762
8	2 <sup>5</sup>	200	0.534	0.800
9	2 <sup>10</sup>	200	0.543	0.798
10	2 <sup>20</sup>	200	0.534	0.815
11	2 <sup>25</sup>	200	0.537	0.809
12	2 <sup>30</sup>	200	0.527	0.804
13	2 <sup>40</sup>	200	0.543	0.801
14	2 <sup>45</sup>	200	0.523	0.831
15	2 <sup>5</sup>	400	0.657	0.769
16	2 <sup>10</sup>	400	0.647	0.765
17	2 <sup>20</sup>	400	0.663	0.763
18	2 <sup>25</sup>	400	0.650	0.763
19	2 <sup>30</sup>	400	0.653	0.777
20	2 <sup>40</sup>	400	0.659	0.770
21	2 <sup>45</sup>	400	0.658	0.771
22	2 <sup>5</sup>	600	0.845	0.693
23	2 <sup>10</sup>	600	0.975	0.617
24	2 <sup>20</sup>	600	1.094	0.573
25	2 <sup>25</sup>	600	1.042	0.586
26	2 <sup>30</sup>	600	1.067	0.592
27	2 <sup>40</sup>	600	1.040	0.596
28	2 <sup>45</sup>	600	1.132	0.578
29	2 <sup>5</sup>	800	0.535	0.845
30	2 <sup>10</sup>	800	0.540	0.862
31	2 <sup>20</sup>	800	0.517	0.846
32	2 <sup>25</sup>	800	0.550	0.845
33	2 <sup>30</sup>	800	0.557	0.824
34	2 <sup>40</sup>	800	0.519	0.864
35	2 <sup>45</sup>	800	0.536	0.830

No	C	L	RMSE	<i>Recognition Rate</i>
36	$2^5$	1000	0.448	0.898
37	$2^{10}$	1000	0.461	0.890
38	$2^{20}$	1000	0.463	0.876
39	$2^{25}$	1000	0.432	0.904
40	$2^{30}$	1000	0.454	0.887
41	$2^{40}$	1000	0.448	0.898
42	$2^{45}$	1000	0.462	0.872
43	$2^5$	1500	0.382	0.913
44	$2^{10}$	1500	0.375	0.918
45	$2^{20}$	1500	0.382	0.912
46	$2^{25}$	1500	0.378	0.933
47	$2^{30}$	1500	0.384	0.918
48	$2^{40}$	1500	0.388	0.906
49	$2^{45}$	1500	0.385	0.918

## TENTANG PENULIS



Prasetia Utama Putra, lahir di Bogor pada tanggal 01 Desember 1992. Dengan orang tua bernama, Agus Susilo dan Sri Yuhana.

Sejarah pendidikan yang ditempuh oleh penulis:

(1). 1998 – 2004 menempuh pendidikan dasar di SDN 07 Cibinong, Bogor. (2). 2004 – 2007 menempuh pendidikan menengah pertama di SMPN 1 Cibinong, Bogor. (3). 2007 – 2010 menempuh pendidikan menengah atas di SMKN 1 Cibinong, Bogor.

Penulis memiliki minat pada bidang *Computer Vision*. Selain itu penulis juga aktif dalam bidang business aplikasi dan kegiatan *volunteer*. Perkenalan penulis dengan bidang *Computer Vision* dimulai pada semester 04, disaat penulis menempuh pendidikan di Universitas Negeri Jakarta. Saat ini penulis aktif dalam mempelajari *machine learning* dan interpretasi *visual*.