

PERANCANGAN DETEKSI PENGENALAN WAJAH MENGUNAKAN METODE *LOCAL BINARY PATTERN* (LBP)

Mina Ismurahayu¹, Anwar Nasihin²

¹²STMIK BANDUNG

SEKOLAH TINGGI MANAJEMEN DAN INFORMATIKA BANDUNG

JL. Cikutra No. 113, Bandung 40124, INDONESIA

¹ismurahayu@gmail.com, ²anwarnasihin555@gmail.com

Abstrak

pengenalan wajah pada *smartphone android* merupakan sebuah sistem identifikasi pribadi yang menggunakan karakteristik fisik dari seseorang untuk mendeteksi secara otomatis identitasnya melalui wajah. Pada teknologi deteksi pengenalan wajah ini akan mengidentifikasi wajah manusia melalui sebuah gambar digital dengan cara mencocokkan atau membandingkan dengan wajah yang sebelumnya telah disimpan dan dilatih didalam data penyimpanan. Permasalahan utama yang dihadapi dalam melakukan perancangan deteksi pengenalan wajah yaitu bagaimana perangkat android bisa mendeteksi dan mengenali objek wajah.

Metode perancangan untuk deteksi pengenalan wajah berbasis android yang diusulkan adalah metode *local binari pattern* (LBP) yang pada prinsipnya dengan menggunakan metode ini yaitu untuk mendapatkan ciri dari setiap citra wajah dengan cara merepresentasikan sebuah citra abu-abu kedalam bilangan biner. Metode yang dikembangkan adalah dengan melakukan modifikasi fungsi heuristik algoritma *Ant-Miner*.

Dari hasil pengujian diperoleh bahwa performansi deteksi pengenalan wajah dengan menggunakan metode LBP pada citra wajah seseorang memiliki tingkat akurasi yang baik dengan rata-rata 96,7%, dengan percobaan 5 kali pengujian dan dengan 30 jumlah pengujian.

Selain itu pada deteksi pengenalan wajah ini juga dapat dimanfaatkan sebagai kerangka uji untuk deteksi pengenalan wajah berbasis android.

Kata Kunci : deteksi, pengenalan wajah, *local binary pattern*, android

Abstract

Face recognition on an Android smartphone is a personal identification system that uses the physical characteristics of a person to automatically detect their identity through faces. In this face recognition detection technology will identify human faces through a digital image by matching or comparing with faces that have previously been stored and trained in data storage. The main problem faced in designing face recognition detection is how android devices can detect and recognize face objects.

The proposed design method for Android-based face recognition detection is the local binary pattern (LBP) method, which in principle uses this method to get the characteristics of each face image by representing a gray image into binary numbers. The method developed is by modifying the heuristic function of the Ant-Miner algorithm.

From the test results obtained that the performance of facial recognition detection using the LBP method on a person's face image has a good level of accuracy with an average of 96.7%, with 5 trials and 30 trials.

In addition, this face recognition detection can also be used as a test framework for Android-based face recognition detection.

Keywords: detection, face recognition, local binary pattern, android

1. PENDAHULUAN

1.1. Latar Belakang

Mengenali wajah dalam sebuah gambar atau foto, bagi manusia merupakan hal yang mudah. Tetapi tidak demikian untuk *smartphone android*, agar dapat mengenali wajah diperlukan perlakuan khusus sehingga ketika diberi input sebuah gambar atau foto, *smartphone android* dapat mendeteksi apakah dalam gambar tersebut terdapat sebuah gambar wajah atau tidak dan mengenali wajah tersebut. Pada teknologi deteksi pengenalan wajah ini akan mengidentifikasi wajah manusia melalui sebuah gambar digital dengan cara mencocokkan atau membandingkan dengan wajah yang sebelumnya telah disimpan dan dilatih di dalam data penyimpanan. Pada laporan skripsi ini penulis menganalisis kinerja *face recognition* menggunakan metode *Local Binary Pattern* untuk mengidentifikasi manusia berdasarkan pola wajahnya yang di ambil melalui kamera *smartphone android*.

Telah banyak metode metode pengenalan wajah yang dikembangkan. Salahsatunya adalah metode *Local Binary Pattern* (LBP). Untuk mendapatkan ciri dari setiap wajah maka digunakan metode LBP. LBP merepresentasikan sebuah citra abu-abu kedalam bilangan biner. Dengan menggunakan LBP akan didapatkan sebuah nilai dari setiap *pixel* dengan cara membandingkan nilai *pixel* tersebut dengan nilai *pixel* tetangga-tetangganya. Kemudian nilai baru yang didapatkan dari setiap *pixel* itu akan dijadikan sebuah *histogram* baru yang akan dijadikan ciri dari setiap wajah.

1.2. Rumusan Masalah

Berikut rumusan masalah pada penelitian skripsi ini adalah :

1. Bagaimana membuat program untuk dapat mendeteksi dan mengenali wajah ?
2. Bagaimana menerapkan algoritma LBP untuk deteksi citra wajah?

1.3. Tujuan Penelitian

Tujuan penelitian laporan skripsi ini adalah :

1. Memanfaatkan sistem deteksi dan pengenalan wajah menggunakan metode *Local Binary Pattern*
2. Membangun aplikasi deteksi pengenalan wajah berbasis android

1.4. Metodologi Penelitian

Metodologi penelitian yang digunakan untuk pengumpulan data implementasi yaitu metode observasi, metode wawancara (*interview*), metode kearsipan, dan metode kepustakaan (*library*). Sedangkan metodologi penelitian yang digunakan adalah *local binary pattern*.

2. PEMBAHASAN

2.1. Pembahasan Metode Yang Diususlkan

Local Binary Pattern merupakan deskriptor untuk mengklarifikasi gambar berdasarkan tekstur gambar. Jadi sebuah gambar yang berukuran 3x3, dimana nilai biner pada pusat gambarnya dibandingkan dengan nilai sekelilingnya. Jika intensitas piksel tengah lebih besar dari pada biner pusat maka nilai yang ditetapkan 1, Jika lebih kecil maka 0.

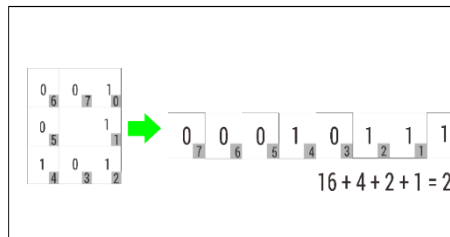
5	8	1
5	4	1
3	7	2

1	1	0
1		0
0	1	0

Gambar 2.1. Langkah pertama dalam membangun *Local Binary Pattern*

perbandingan nilai biner piksel pada pusat gambar dengan 8 nilai piksel disekelilingnya.

Selanjutnya menghitung nilai *Local Binary Pattern* untuk piksel yang ditengah mulai dari piksel disekelilingnya dengan cara *clockwise* (searah jarum jam) atau *counter-clockwise* (berbalik arah jarum jam) dengan syarat harus konsisten. Misal 3x3 berarti ada 8 tes biner. Kemudian hasil uji biner disimpan dalam array 8 bit yang diubah menjadi desimal.



Gambar 2.2. Pengambilan 8 bit biner lingkungan sekitar dari piksel tengah

Pengambilan 8 bit biner lingkungan sekitar dari piksel tengah menjadi representasi atau mewakili desimal nilai tersebut akan disimpan dalam array output *Local Binary Pattern* 2D, kemudian dapat divisualisasikan bisa disebut sebagai proses thresholding yaitu mengumpulkan biner dan menyimpan nilai desimal pada keluaran array *Local Binary Pattern* diulang untuk setiap piksel pada gambar yang diinput. Nilai desimal 8 bit dari *Local Binary Pattern* (LBP) dapat dinyatakan dalam persamaan (2) berikut:

$$P = s - 1 \quad p$$

$$LBPP, R(xc, yc) = \sum s (g - \mu)^{2^p}$$

$$P=0$$

$$\text{Dimana : } s(X) = \begin{cases} 1 & X \geq 0 \\ 0 & X < 0 \end{cases}$$

keterangan:
x dan y : Koordinat pusat piksel

ketetanggaaan^c :
p : Circular sampling point
P : Banyaknya sampling point

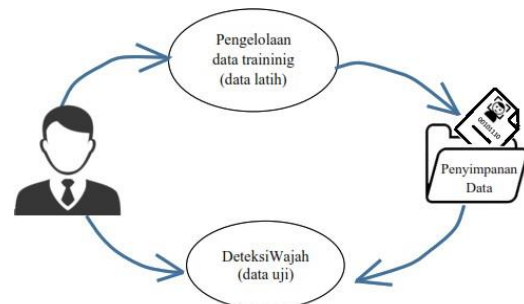
g^p : Nilai keabuan dari p
μ : Nilai rata-rata piksel

ketetanggaaan S : Sign (kode biner)

Pada data uji juga dilakukan proses ekstraksi citra sama seperti yang dilakukan pada data latih, hanya saja pada proses pengujian dilakukan pengklasifikasian data uji dengan cara membandingkan jarak perbedaan nilai LBP dari citra pada data uji dengan seluruh nilai LBP dari citra pada data

latih untuk memperoleh hasil pengenalan wajah.

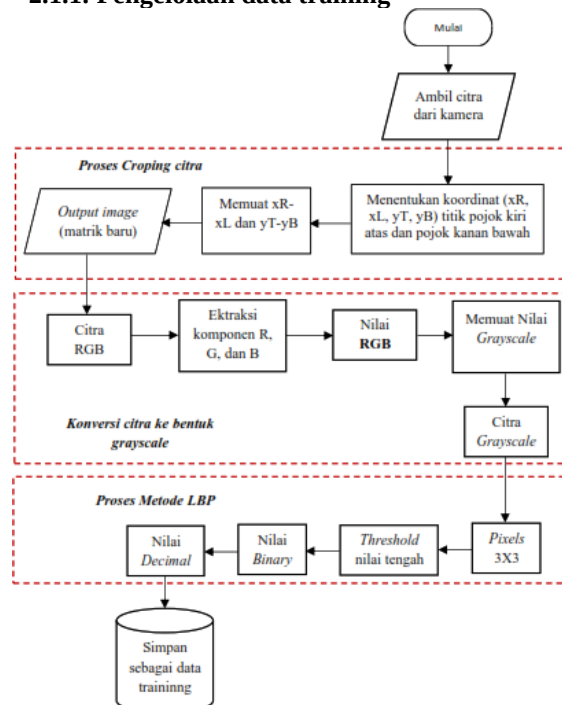
2.2. Gambaran Umum Sistem



Gambar 3.1. Arsitektur deteksi pengenalan wajah

Gambar 3.1. merupakan arsitektur sistem perangkat lunak yang akan dibangun. Pada gambar ditunjukkan fokus pengerjaan pada perangkat lunak yang akan dibangun. Sistem yang dibangun meliputi pendeteksian wajah dari kamera *smartphone*, proses pencocokan wajah di sisi *smartphone*, hingga pengiriman status verifikasi kehadiran ke database server. Pada gambar ini ditunjukkan tahapan *input*, proses, dan *output* pada fokus pengerjaan skripsi ini. Tahapannya adalah sebagai berikut:

2.1.1. Pengelolaan data training

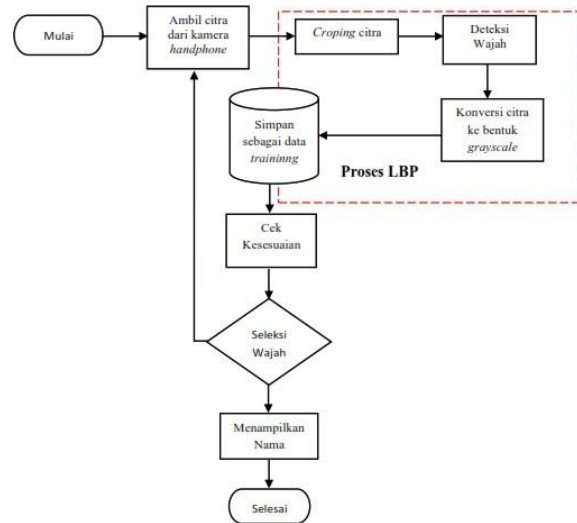


Gambar 3.2. Pengelolaan proses data Training

Pada tahap ini, melakukan proses pendaftaran (data training) data wajah dengan cara:

1. Mengambil citra wajah dari kamera *handphone*
2. Selanjutnya proses *cropping* citra wajah:
 - a. Menentukan *rectangle* sesuai wajah dengan koordinat titik pojok kiri atas dan pojok kanan bawah.
 - b. Dengan Memuat hasil dari $xR-xL = w'$ dan $yT-yB = h'$ yang kemudian menjadi matrikbaru atau *image* baru.
3. Deteksi wajah berdasarkan hasil *cropping*
4. Selanjutnya konversi citra RGB ke bentuk *grayscale*
 - a. Dari hasil *cropping* tadi mempunyai nilai masing-masing dari RGB
 - b. Nilai RGB dikonversi ke *grayscale* dengan membagi nilai RGB dibagi 3
5. Lalu konversi *grayscale* ke biner dengan metode *LBP* dengan tujuan untuk mengetahui ciri dari setiap citra, yaitu dengan cara :
 - a. Dibuat *pixel* menjadi 3x3
 - b. Selanjutnya pada proses *threshold* dengan membandingkan nilai ketetanggan dengan nilai tengah, jika hasilnya $>$ dari nilai tengah maka diberi nilai 1 dan jika hasilnya $<$ dari nilai tengah maka diberi nilai 0.
 - c. Selanjutnya susun menjadi angka biner 8 bit searah jarum jam (*clockwise*) atau berlawanan arah jarum jam (*counter-clockwise*)
 - d. Selanjutnya konversi nilai biner menjadi bilangan desimal.
6. Selanjutnya disimpan kedalam data penyimpanan sebagai data *training*, sebagai acuan data uji nanti.

2.1.2. Identifikasi wajah



Gambar 3.3. Proses identifikasi wajah

Pada tahap ini awalnya sama dengan pengelolaan data *training* hanya saja pada tahap identifikasi ini membandingkan atau mencocokkan data yang di uji (secara *real time*) dengan data *training* yang disimpan di penyimpanan.

- 1) Ambil citra wajah dari kamera *handphone*
 - 2) Selanjutnya *cropping* citra wajah
 - 3) Selanjutnya deteksi wajah secara *real time*
 - 4) Lalu konversi citra ke bentuk *grayscale* (proses *LBP*)
 - 5) Selanjutnya proses pencocokan data dengan data *training* (data uji) yang sudah tersimpan dipenyimpanan
 - 6) Cek kesesuaian data latih dengan data uji yang di uji secara langsung
 - 7) Seleksi wajah apakah sesuai dengan data *training* (data latih), kalau tidak sesuai maka ulangi lagi ke tahap pengambilan citra sampai data benar-bener sama (terdeteksi)
 - 8) Kalau sesuai maka akan menampilkan nama.
- Pada dasarnya *output* dari aplikasi sistem deteksi pengenalan wajah ini adalah pengiriman status pengguna berhasil terdeteksi oleh sistem.

2.1.3. Pengelolaan Data Training

1. Proses Cropping

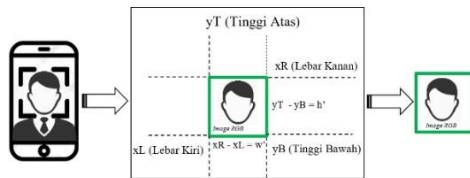
pada pengenalan bisa disebut juga sebagai proses pelatihan, yaitu meliputi pengambilan citra wajah dengan kamera *handphone*, dengan memotong bagian wajah yang dideteksi dengan *rectangular* pada persamaan:

Cropping

$$\begin{aligned} w' &= xR - xL \\ h' &= yT - yB \end{aligned}$$

Dimana:

xL = Lebar Kiri
 xR = Lebar Kanan
 yT = Tinggi Atas
 yB = Tinggi Bawah
 w' = lebar hasil
 h' = tinggi hasil

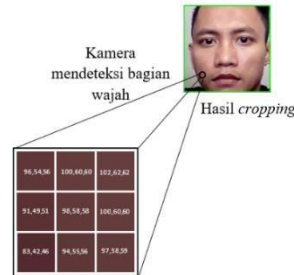


Gambar 3.4. Proses cropping citra wajah

proses *cropping* citra dengan cara pengambilan area tertentu yang akan diamati (*area of interest*) dalam citra melalui kamera *smartphone*, yang pada intinya memotong lebar dan tinggi dari citra wajah dengan *rectangular* mendeteksi bagian wajah dan di *kropping* dengan cara lebar kanan (xR) *background* bagian wajah dikurangi lebar kiri (xL) *background* bagian wajah untuk menentukan lebar (w'), sedangkan untuk tinggi dengan cara mengurangi tinggi atas (yT) dengan tinggi bawah (yB), yang bertujuan untuk mempermudah penganalisaan citra dan memperkecil ukuran penyimpanan citra.

2.1.4. Konversi RGB ke Grayscale

Selanjutnya deteksi bagian wajah yang sudah di *cropping* untuk pengubahan citra RGB menjadi citra aras keabuan dimana dari setiap RGB ini mempunyai nilai masing-masing. Contohnya seperti



Gambar 3.5. Proses konversi RGB ke grayscale

Pada Gambar 3.5 dapat dinyatakan dalam bentuk matrik dengan 3x3 yang ditunjukkan pada Tabel 3.3 berikut ini:

Tabel 3.3 Nilai RGB

RBG (x,y)	Pix1	Pix2	Pix3	...
Pix1 R	96	100	102	...
G	54	60	62	...
B	56	60	62	...
Pix2 R	91	98	100	...
G	49	58	60	...
B	51	58	60	...
Pix3 R	83	94	97	...
G	42	55	58	...
B	46	56	59	...
...	...	...	...	...

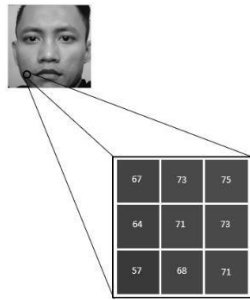
Setelah mendapatkan nilai piksel langkah selanjutnya adalah merubah nilai RGB tersebut ke nilai *grayscale* dengan mencari nilai rata-ratanya, yang dapat dilihat pada Persamaan sebagai berikut:

$$\text{Grayscale} = \frac{R+G+B}{3}$$

Tabel 3.4 Nilai Grayscale

Grayscale	Pixel 1	Pixel 2	Pixel 3	...
Pixel 1	67	73	75	...
Pixel 2	64	71	73	...
Pixel 3	57	68	71	...
...	...	...	...	...

Sehingga setelah di konversi ke *grayscale* hasilnya seperti gambar 3.6 dibawah :



Gambar 3.6. nilai grayscale yang dihasilkan

2.1.5. Konversi Grayscale ke Biner

Setelah mendapatkan nilai grayscale langkah selanjutnya yaitu tahap LBP

$$LBP(xc, yc) = \sum_{n=0} S(in - ic) 2^n(1)$$

$$S(x) = \begin{cases} 1, & \text{if } x \geq 0 \\ 0, & \text{if } x < 0 \end{cases} \quad (2)$$

Dimana:

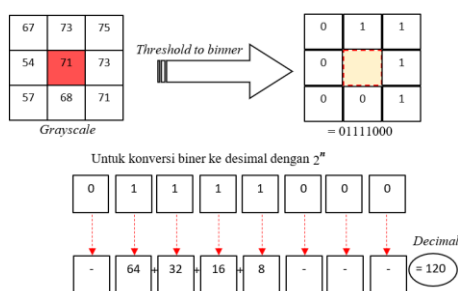
xc = Lebar piksel

yc = Tinggi piksel

$S(x)$ = Fungsi dari *thresholding*

in = Piksel tetangga dari piksel tengah

ic = Piksel tengah



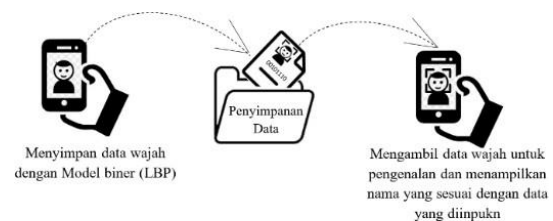
Gambar 3.7 Ilustrasi LBP

Gambar 3.7 merupakan ilustrasi dari proses LBP. Proses pertama adalah

melakukan pengurangan piksel ketetanggaan dengan piksel tengah (1). Selanjutnya hasil pengurangan di-*threshold* menggunakan persamaan (2), jika hasilnya > dari nilai tengah maka diberi nilai 1 dan jika hasilnya < dari nilai tengah maka diberi nilai 0. Setelah itu, nilai biner piksel ketetanggaan akan disusun berlawanan arah jarum jam atau searah

jarum jam dan 8 bit biner tersebut dikonversi ke dalam nilai *decimal* untuk menggantikan nilai piksel tengah ic . Adapun kelebihan dari LBP adalah mudah diimplementasikan dan tingkat komputasinya lebih rendah sehingga tidak membutuhkan waktu yang lama dalam ekstraksi fitur. (Biglari, 2014).

2.1.6. Penyimpanan Data Training Dan Deteksi Wajah



Gambar 3.8 tahap identifikasi wajah

Ini adalah tahap terakhir dalam pengenalan wajah yaitu tahap identifikasi wajah yang mencocokkan wajah dengan data yang disimpan di dalam penyimpanan, apabila wajah teridentifikasi maka akan menampilkan nama sesuai yang *diinputkan* apabila belum terdeteksi maka posisi kamera harus benar-benar lurus dengan wajah yang diidentifikasi.

Tabel 5.2. pengujian hasil kerja

Pengujian	Jumlah pengujian	Jumlah deteksi pengenalan	Selisih
1.	5	5	0
2.	6	5	1
3.	7	7	0
4.	6	5	1
5.	6	6	0
Rata-rata	30	28	96,7%

$$\frac{\text{Jumlah pengujian} + \text{jumlah deteksi pengenalan}}{\text{Jumlah pengujian} + \text{jumlah deteksi pengenalan} + \text{selisih}}$$

X 100

$$\frac{30 + 28}{30 + 28 + 3} \times 100 = 96,7\%$$

Berdasarkan data pada tabel 5.2 diatas, diperoleh akurasi sebagai berikut:

$$\frac{30 + 28}{30 + 28 + 3} \times 100 = 96,7\%$$

Dengan sebanyak 30 jumlah pengujian, yang terdeteksi hanya 28 wajah yang terdeteksi dengan

selisih 3 yang tidak terdeteksi, karena intensitas cahaya dan bukan wajah pun terdeteksi. Tingkat akurasi pada deteksi pengenalan wajah yaitu **96,7%**, dari perhitungan tingkat akurasi diatas dilakukan 5 kali pengujian.

Kesimpulan

Berdasarkan analisis, perancangan, implementasi dan pengujian yang telah dikemukakan sebelumnya maka dapat diambil kesimpulan dari hasil perancangan aplikasi deteksi pengenalan wajah adalah sebagai berikut:

1. Hasil dari aplikasi ini yaitu sebagai kerangka uji untuk deteksi pengenalan wajah pada *smartphone android*.
2. Dengan metode LBP pada citra wajah seseorang memiliki tingkat akurasi yang baik dengan rata-rata 96,7%, dengan percobaan 5 kali pengujian dengan 30 jumlah pengujian.

Saran

1. Aplikasi ini lebih disarankan untuk pengembangan presensi seminar yang di adakan di kampus.
2. Aplikasi yang ditujukan untuk memudahkan dalam pengembangan presensi dengan sistem biometrik.
3. Dengan penggunaan aplikasi ini diharapkan dapat menjadi fasilitas dalam membangun dan mengembangkan sebuah sistem presensi dengan *computer vision* berbasis android

DAFTAR PUSTAKA

- [1] Ardy Dwi Caesar yanto (2012). *Local Binary Pattern* (LBP) : Identifikasi citra wajah tampak samping
- [2] Akariman, Q., Jati, A., N., dan Novianty, A., 2015, Pengenalan Wajah Pada Perangkat Android Menggunakan Algoritma *Local Binary Pattern* (LBP), Vol.2, No.2, ISSN : 2355-9365
- [3] Achmad Hidayatno, dan Teguh Prakoso. transient, vol. 7, no. 1, maret 2018, issn: 2302-9927, 129
- [4] Fandiansyah, Sari, J., Y., dan Ningrum, I., P., 2017, Pengenalan Wajah Menggunakan Metode *Linear Discriminant Analysis* dan *K Nearest Neighbour*, ISSN : 2085-4552
- [5] Muhammad Bagas Gigih Yuda Prasetyo (2015). *Local Binary Pattern* (LBP) : Identifikasi citra wajah dan plat nomor
- [6] Nur Widya Pratiwi, Fauziah, Septi Andryana, Aris Gunaryati Program Studi Informatika, Universitas Nasional, Jurnal String Vol. 3 No.1 Agustus 2018, DETEKSI WAJAH MENGGUNAKAN HIDDEN MARKOV MODEL (HMM) BERBASIS MATLAB
- [7] PENGENALAN WAJAH MENGGUNAKAN *LOCAL BINARY PATTERN* (LBP) DAN *SUPPORT VECTOR MACHINE* (SVM),
- [8] Pengenalan Ekspresi Wajah Menggunakan *Local Binary Pattern* (LBP). Tjokorda Agung Budi W, S.T, M.T Fakultas Informatika Telkom University
- [9] Syakira Nurina Shaputri (2015). *Local Binary Pattern* dan *Fuzzy Logic* : Citra lovebird