

SIMULASI PENGENALAN WAJAH UNTUK MEMBUKA MINIATUR PINTU MENGGUNAKAN METODE *LOCAL BINARY PATTERN (LBP)* DAN *ARDUINO UNO*

Nanda Arief Setiawan¹, Huzeni², Aswandi³

Program Studi Teknik Informatika, Jurusan Teknologi Informasi dan Komputer, Politeknik Negeri Lhokseumawe, Jalan Banda Aceh-Medan Km. 280,3 Buketrata, Lhokseumawe, 24301 PO.BOX 90 Telpon (0645) 42670, 42785 Fax 42785, Indonesia

Email: nanda.arief10@yahoo.com, zaini_pnl@yahoo.co.id, aswandi.pnl.ac.id

Abstrak

Keamanan ruangan yang baik sangat diperlukan untuk menghindari terjadinya pembobolan oleh pihak - pihak yang tidak bertanggung jawab. Seiring perkembangan waktu sistem - sistem keamanan yang telah ada masih memiliki beberapa kelemahan, oleh karena itu diperlukan sistem keamanan tambahan yang sulit untuk dimanipulasi. Salah satunya menggunakan sistem pengenalan wajah. Sistem keamanan ini menggunakan salah satu bagian dari tubuh manusia yaitu wajah sebagai bahan penelitiannya, karena wajah memiliki perbedaan khusus dari masing - masing individu sehingga kecil kemungkinan terjadinya pembobolan dari pihak - pihak yang tidak bertanggung jawab. Metode *Local Binary Pattern (LBP)* merupakan salah satu metode pengenalan wajah yang dapat digunakan sebagai proses untuk sistem keamanan dengan menggunakan pengenalan wajah. Proses *Local Binary Pattern (LBP)* adalah membandingkan nilai piksel tetangga dengan nilai piksel pusat yang disebut sebagai *threshold*. Pada penelitian ini wajah yang di *capture* pada kamera akan di hitung nilai kesamaanya dengan gambar wajah yang telah di *training* dan di simpan ke dalam *database*. Hasil yang di dapat jika terdapat kesamaan yang mendekati maka sistem akan menunjukkan wajah siapa yang memiliki tingkat kesamaan yang mendekati dan pintu akan terbuka jika sistem mengenali wajah tersebut.

Kata kunci : pengenalan wajah, Local Binary Pattern (LBP), pintu

1. Pendahuluan

Keamanan ruangan yang baik sangat diperlukan untuk menghindari terjadinya pembobolan oleh pihak - pihak yang tidak bertanggung jawab. Pada saat ini sudah banyak terdapat kunci - kunci untuk mengamankan suatu ruangan, contohnya adalah kunci mekanik yang terpasang pada pintu, namun kunci mekanik memiliki sistem yang kurang bagus karena kunci mekanik dapat dicuri dan diduplikasi oleh pihak - pihak yang tidak bertanggung jawab.

Untuk mengatasi hal tersebut, sebagai pengganti dari kunci mekanik yaitu menggunakan kunci elektronik, dengan sistem membuka dan menutupnya kunci dikendalikan secara elektrik. Pada saat ini sudah banyak sistem kunci elektronik yang digunakan, beberapa contohnya akses kunci menggunakan kata sandi dari keypad atau keyboard, akses menggunakan kartu *Radio Frekuensi Identification (RFID)*, menggunakan *barcode*, maupun akses menggunakan kata sandi yang ditransmisikan menggunakan gelombang infra merah.

Semua sistem di atas sudah bagus, hanya saja masih terdapat celah - celah yang memungkinkan untuk terjadinya pembobolan dari pihak - pihak yang tidak bertanggung jawab. Karena itu diperlukan sistem

keamanan tambahan lainnya yang lebih sulit untuk dimanipulasi. Salah satu solusinya adalah menggunakan teknologi biometrik.

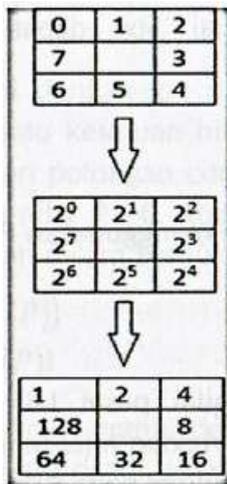
Teknologi biometrik merupakan suatu teknologi yang digunakan untuk menganalisa ciri - ciri fisik yang khas dari tubuh manusia sebagai ukuran yang dapat membedakannya dengan orang lain, contohnya sidik jari, sidik mata, suara, bibir, gigi ataupun wajah. Semua hal tersebut sudah pasti terdapat perbedaan pada masing - masing manusia, sehingga kecil kemungkinan adanya manipulasi dari pihak - pihak yang tidak bertanggung jawab.

2. Metode Penelitian

Operator LBP adalah salah satu analisis tekstur yang telah banyak digunakan dalam berbagai penerapan dan aplikasi. LBP telah terbukti diskriminatif dengan keuntungan utamanya, yaitu variasi untuk perubahan tingkat abu - abu monoton dan efisiensi komputasi, membuat LBP sangat cocok untuk tugas gambar menuntut analisis. Ide untuk menggunakan LBP untuk deskripsi tekstur didukung oleh komposisi pola mikro yang dapat dijelaskan oleh sebuah operator. *Local Binary Pattern (LBP)* didefinisikan sebagai ukuran

tekstur *gray-scale*, berawal dari definisi umum tekstur di daerah sekitar.

Berdasarkan penurunan *LBP*, maka nilai *LBP* dapat diperoleh dengan mengubah nilai - nilai tetangga menjadi 1 atau 0. Nilai 1 akan diperoleh jika piksel tetangga tersebut mempunyai nilai lebih besar atau sama dengan piksel pusat. Jika lebih kecil dari piksel pusat, maka piksel tetangga tersebut diberi nilai 0. Untuk setiap nilai piksel baru tersebut dikalikan dengan 2^P . P mempunyai nilai 0 sampai dengan 7. Nilai 0 dimulai dari pojok kiri atas dan dilanjutkan sesuai dengan arah jarum jam nilainya bertambah 1.



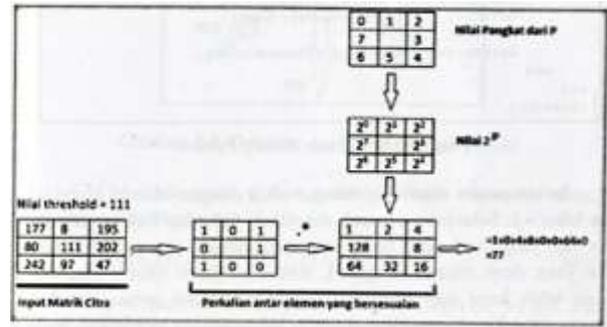
Gambar 1. Proses Pembentukan Nilai Binomial *LBP*

Misalkan potongan citra dengan ukuran 3x3 adalah sebagaimana terlihat pada Gambar 1, maka nilai piksel pusat dianggap sebagai *threshold*, yaitu 111. Tetangga - tetangga piksel pusatnya adalah 177, 8, 195, 202, 47, 97, 242 dan 80.

177	8	195
80	111	202
242	97	47

Gambar 2. Contoh Potongan Citra Ukuran 3x3

Potongan citra sebagaimana pada Gambar 2 selanjutnya diubah menjadi nilai biner dengan mengacu nilai *threshold* yang berada dipusat, yaitu 111. Misalkan nilai 177 akan berubah menjadi 1 karena $177-111 > 0$. Selanjutnya nilai 8 akan berubah menjadi 0, karena $8-111 < 0$. Begitu juga dengan nilai yang lainnya. Hasil pembentukan nilai biner tersebut selanjutnya dikalikan terhadap nilai elemen bobot binomial 2^P yang posisinya bersesuaian. Hasil perkalian kemudian dijumlahkan untuk menggantikan posisi nilai pusat dari potongan citra awal. Proses pembentukan nilai *LBP* selengkapnya dapat dilihat pada Gambar 3.



Gambar 3. Proses Pembentukan Nilai *LBP*

Eigenface merupakan sebuah cara sederhana untuk mengekstrak informasi yang terkandung dalam citra wajah adalah dengan menangkap variasi-variasi penting dalam sekumpulan citra wajah dan menggunakan informasi tersebut untuk mengkodekan dan membandingkan citra wajah. Dalam bahasa matematika, kita harus menemukan komponen-komponen utama dari distribusi citra-citra wajah atau *eigenvector* dan matriks kovarian dari sederetan citra wajah dengan memperlakukan sebuah citra wajah sebagai sebuah titik (atau sebuah vektor) dalam ruang vektor berdimensi sangat tinggi. *Eigenvector* ini kemudian disusun berdasarkan *eigenvalue*-nya. Setiap *eigenvector* mengacu pada suatu nilai variasi diantara citra-citra wajah.

Eigenvector ini bisa dianggap sebagai sederetan ciri yang bersama-sama memberi karakter variasi diantara citra-citra wajah. Setiap titik citra wajah bisa dinyatakan dalam satu atau lebih *eigenvector* sehingga sekumpulan *eigenvector* dapat ditampilkan sebagai sekumpulan wajah. Sekumpulan *eigenvector* yang digunakan inilah yang disebut sebagai *eigenface*. Untuk menemukan *eigenface* dari sebuah database citra wajah, hal penting pertama yang harus dilakukan adalah menentukan vektor rata-rata, vektor deviasi, dan matriks kovarian untuk database tersebut.

Tahapan Perhitungan *Eigenface* :

Langkah pertama adalah menyiapkan data dengan membuat suatu himpunan S yang terdiri dari seluruh training image ($\Gamma_1, \Gamma_2, \dots, \Gamma_M$)

$$S = (\Gamma_1, \Gamma_2, \dots, \Gamma_M) \tag{2.3}$$

Langkah kedua adalah ambil nilai tengah atau *mean* (Ψ)

$$\Psi = \frac{1}{M} \sum_{n=1}^M \Gamma_n \tag{2.4}$$

Langkah ketiga kemudian cari selisih (Φ) antara training image (Γ_i) dengan nilai tengah (Ψ)

$$\Phi = \Gamma_i - \Psi \tag{2.5}$$

Langkah keempat adalah menghitung nilai matriks kovarian (C)

$$C = \frac{1}{M} \sum_{n=1}^M \Phi_n \Phi_n^T = AA^T \quad A = [\Phi_1, \Phi_2, \dots, \Phi_M] \quad (2.6)$$

$$L = A^T A \quad L = \Phi_n \Phi_n^T \quad (2.7)$$

Langkah kelima menghitung *eigenvalue* (λ) dan *eigenvector* (v) dari matriks kovarian (C)

$$Cxvi = \lambda_i xvi \quad (2.8)$$

Langkah keenam, setelah *eigenvector* (v) diperoleh, maka *eigenface* (μ) dapat dicari dengan :

$$\mu_i = \sum_{k=1}^M v_{ik} \Phi_k \quad (2.9)$$

$$l = 1, \dots, M \quad (2.10)$$

Setelah melalui proses ekstraksi ciri dan dihasilkan suatu nilai-nilai parameter tertentu, maka dilanjutkan dengan perhitungan jarak Euclidean. Jarak Euclidean digunakan pada proses pengenalan. Jarak Euclidean antara nilai vector ciri citra uji dan nilai vector citra basisdata dinyatakan oleh:

$$D(A, B) = \sqrt{\sum_{i=1}^n \frac{|A_i - B_i|^2}{A_i}} \quad (2.11)$$

dengan:

D(A,B) = Jarak euclidean antara gambar A dan B

A = Vektor ciri citra masukan

B = Vektor ciri citra basis data

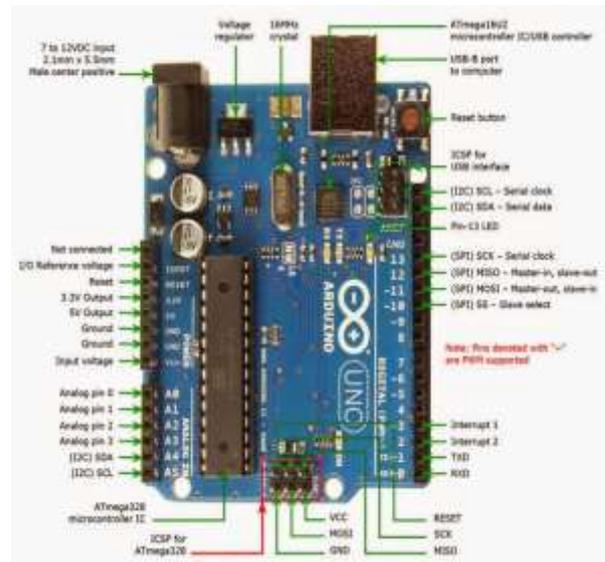
n = Panjang vektor a dan vektor B

Arduino Uno adalah papan sirkuit berbasis mikrokontroler ATmega328. IC (integrated circuit) ini memiliki 14 input/output digital (6 output untuk PWM), 6 analog input, resonator kristal keramik 16 MHz, Koneksi USB, soket adaptor, pin header ICSP, dan tombol reset. Hal inilah yang dibutuhkan untuk mensupport mikrokontroler secara mudah terhubung dengan kabel power USB atau kabel power supply adaptor AC ke DC atau juga battery.

Untuk keunggulan board Arduino Uno Revision 3 antara lain:

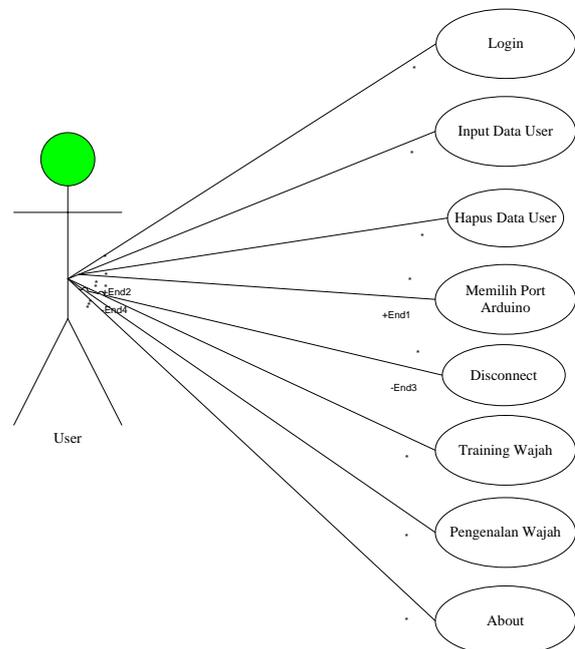
- 1.0 pinout : ditambahkan pin SDA dan SCL di dekat pin AREF dan dua pin lainnya diletakkan dekat tombol RESET, fungsi IOREF melindungi kelebihan tegangan pada papan rangkaian. Keunggulan perlindungan ini akan kompatibel juga dengan dua jenis board yang menggunakan jenis AVR yang beroperasi pada tegangan kerja 5V dan Arduino Due tegangan operasi 3.3V
- Rangkaian RESET yang lebih mantap.

- Penerapan ATmega 16U2 pengganti 8U2.



Gambar 4. Arduino Uno

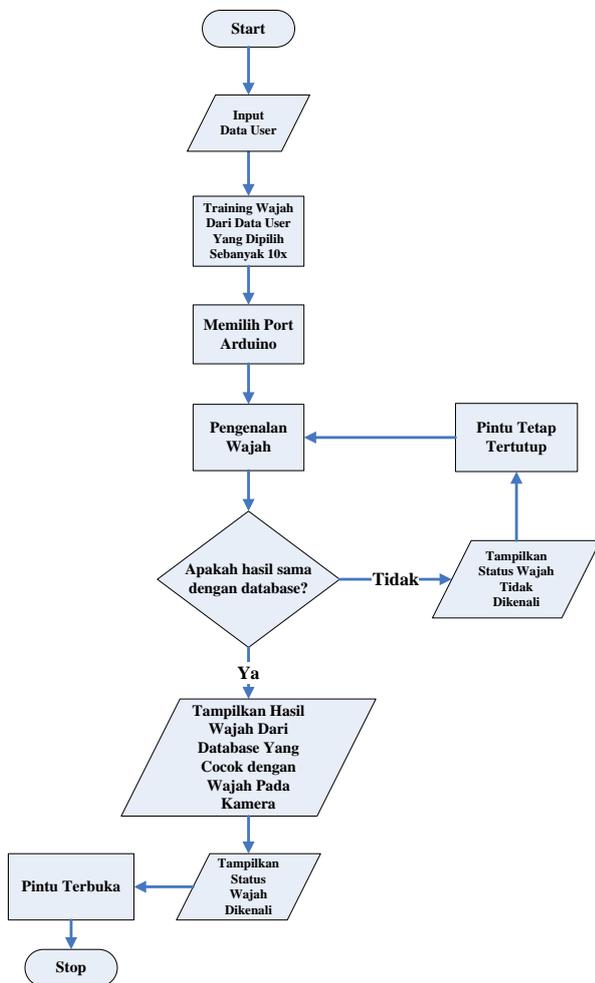
Use case merupakan gambaran skenario dari interaksi antara user dengan sistem. Sebuah diagram use case menggambarkan hubungan antara aktor dan kegiatan yang dapat dilakukannya terhadap sistem.



Gambar 5. Use Case Diagram System

Flowchart adalah diagram yang menunjukkan alir (flow) di dalam sebuah sistem. Selain itu, dapat dikatakan pula sebagai prosedur kerja dari sistem yang sedang dikerjakan secara keseluruhan, yaitu langkah - langkah yang dilakukan dalam sistem dengan tujuan sebagai pengenalan dari sistem tersebut.

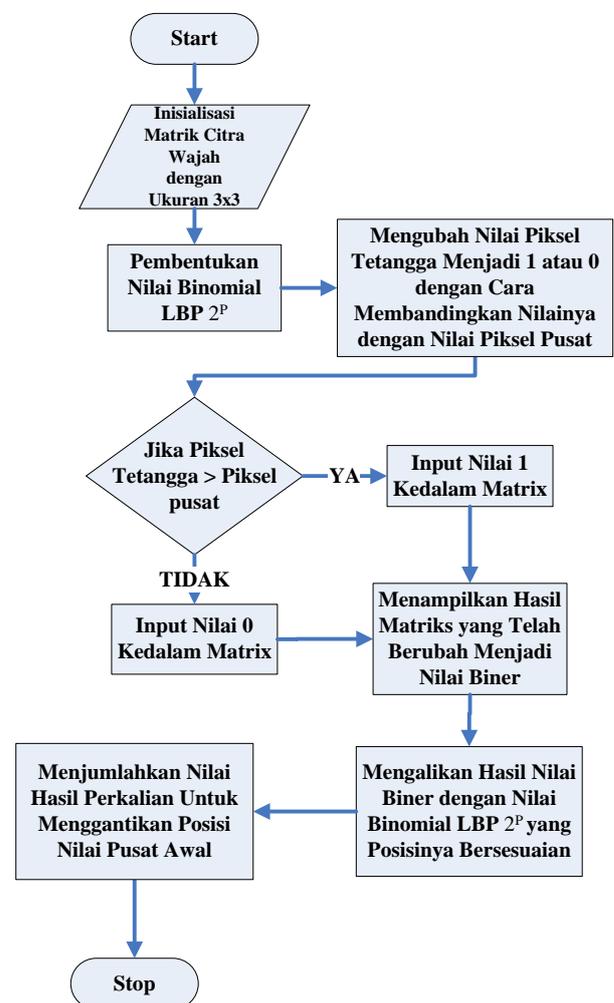
Flowchart di bawah menjelaskan tentang cara kerja sistem secara menyeluruh. Dimulai dengan menginput data *user* berupa *id* dan nama, selanjutnya melakukan proses *training* wajah dari data *user* yang dipilih sebanyak 10 kali, kemudian memilih *Port Arduino Uno* yaitu *COM3* yang terdapat pada *combobox*, selanjutnya melakukan proses pengenalan wajah dengan. Jika hasil nilai wajah yang di-*capture* pada kamera mendekati kesamaan dengan gambar wajah yang terdapat pada database, maka sistem akan menampilkan gambar wajah dari *database* yang cocok dengan wajah yang berada di depan kamera, kemudian status wajah akan menjadi dikenali dan pintu akan terbuka, namun jika tidak terdapat kesamaan, maka sistem tidak akan menampilkan gambar wajah dari *database*, status wajah akan menjadi tidak dikenali dan pintu akan tetap tertutup.



Gambar 6. Flowchart Sistem

Flowchart di bawah menjelaskan tentang cara kerja metode *Local Binary Pattern (LBP)*. Dimulai dengan Menginisialisasi matriks dari citra wajah dengan ukuran 3x3, selanjutnya membentuk nilai binomial $LBP(2^P)$. Dimana P mempunyai nilai 0 sampai dengan 7, nilai 0

dimulai dari pojok kiri atas dan dilanjutkan sesuai dengan arah jarum jam nilainya bertambah 1. Selanjutnya mengubah nilai piksel tetangga menjadi nilai biner 1 atau 0 dengan cara membandingkan nilainya dengan nilai piksel pusat. Nilai piksel pusat disebut juga sebagai *threshold*. Jika piksel tetangga lebih besar dari piksel pusat maka diberi nilai 1, namun jika lebih kecil maka diberi nilai 0. Selanjutnya menampilkan hasil nilai tersebut, selanjutnya mengalikan hasil yang didapat dengan nilai binomial $LBP(2^P)$ dengan posisinya bersesuaian dari mulai pojok kiri atas dan dilanjutkan sesuai dengan arah jarum jam. Setelah selesai, nilai hasil perkalian tersebut dijumlahkan untuk menggantikan posisi nilai pusat awal.



Gambar 7. Flowchart Local Binary Pattern (LBP)

3. Hasil dan Pembahasan

Setelah melakukan penelitian dan perancangan serta untuk mengetahui sejauh mana tingkat keberhasilan dari sistem yang telah rancang, maka perlu dilakukan uji coba program dengan tujuan mengetahui sejauh mana keberhasilan ataupun kegagalan dari sistem

ini dan untuk mengetahui apakah sistem yang dibuat sesuai dengan perancangan yang sudah dilakukan.



Gambar 8. Tampilan Pengenalan Wajah (Wajah Dikenali)

Gambar 8 menjelaskan dimana status wajah sudah dikenali dan pintu masih tertutup dikarenakan sistem masih melakukan proses *counterscan*, setelah status *counterscan* menjadi 0 maka status pintu akan terbuka sampai wajah tidak kenali kembali dan status *counterscan* akan menghitung dari 140 sampai 0 selanjutnya pintu akan kembali tertutup.

Tabel 1. Daftar Uji

No	Nama	Training Wajah	Status Pencahayaan	Menggunakan Atribut	Wajah Dikenali	Status Pengenalan
1	Nanda	10	Normal	Tidak	9	Dikenali
2	Dimas	10	Normal	Tidak	9	Dikenali
3	Rizka	10	Normal	Tidak	8	Dikenali
4	Alvin	10	Normal	Tidak	10	Dikenali
5	Adieth	10	Normal	Tidak	9	Dikenali
6	Nanda	10	Redup	Tidak	0	Tidak Dikenali
7	Alvin	10	Redup	Tidak	0	Tidak Dikenali
8	Adieth	10	Redup	Tidak	0	Tidak Dikenali
9	Nanda	10	Normal	Ya	0	Tidak Dikenali
Jumlah Data Keseluruhan						50
Jumlah Data Yang Dikenali						50
Jumlah Data Yang Error						60

Tabel 1 menjelaskan data - data dari user yang berhasil dan gagal dalam proses pengenalan wajah.

Tabel 2. Akurasi Sistem

Akurasi (%)	$= \frac{\text{jumlah Citra Uji Yang Berhasil}}{\text{jumlah Data Keseluruhan}} \times 100 \%$ $= \frac{45}{50} \times 100 \%$ $= 90 \%$
-------------	--

Berdasarkan Tabel 2 didapatkan tingkat akurasi sistem untuk pengenalan wajah menggunakan metode *Local Binary Pattern (LBP)* yaitu sebesar 90 %.

4. Simpulan

Berdasarkan penelitian yang dilakukan dan pembahasan mengenai sistem simulasi pengenalan wajah untuk membuka miniatur pintu menggunakan metode *Local Binary Pattern (LBP)* dan *Arduino Uno*, maka dapat disimpulkan bahwa :

1. Sistem pengenalan wajah dengan menggunakan *Local Binary Pattern (LBP)* memiliki hasil yang baik, perhitungan yang cepat dan akurat.
2. Proses *training* dan pengenalan wajah yang dilakukan secara *realtime* membuat sistem tersebut mendapatkan hasil yang lebih maksimal.
3. Hasil proses dari *Local Binary Pattern (LBP)* sendiri didapat dengan cara membandingkan nilai piksel pusat dengan nilai piksel tetangganya, kemudian mengalikan dengan nilai binomial 2^p dan terakhir nilai tersebut dijumlahkan untuk menggantikan nilai piksel pusat awal.
4. Terdapat beberapa faktor yang mempengaruhi keberhasilan sistem dalam mendeteksi dan mengenali suatu wajah diantaranya adalah faktor pencahayaan, faktor perubahan penampilan dengan menggunakan atribut yang dapat membuat sistem sulit melakukan proses pendeteksian dan pengenalan wajah, faktor kemiripan nilai dari wajah satu individu dengan individu lainnya, faktor jarak saat pengambilan citra wajah di depan.

Daftar Acuan

[1]. Havaluddin.2011.Memahami Penggunaan UML (Unfied Modelling Language). Jurnal Informatika Mulawarman Vol.6, No.1, Hal : 1- 15.

[2]. Heikkilä, M., Pietikäinen, M., Schmid, C., "Description of interest regions with local binary patterns," Pattern Recognition, vol. 42, no. 3, pp. 425-436, 2009.

- [3]. Jameco. 2002. Arduino Uno Pinout Diagram. (Online)
<http://www.caratekno.com/2015/07/pengertian-arduino-uno-mikrokontroler.html> diakses tanggal 26 Agustus 2016.
- [4]. Liao, S., Law, M. W. K., and Chung, A. C. S., "Dominant local binary patterns for texture classification," IEEE Trans. on Image Processing, vol. 18, no. 5, pp. 1107-1118, 2009.
- [5]. Muntasa, Arif. 2015. PENGENALAN POLA ; Aplikasi untuk Pengenalan Wajah, Analisis Tekstur Obyek, Pengenalan Plat Nomor Kendaraan dan Segmentasi Pembuluh Darah. Yogyakarta : GRAHA ILMU.
- [6]. Pietikäinen, M., Hadid, A., et al, ComputerVision Using Local Binary Patterns, London: Springer, 2010.
- [7] Prawoto, Ihsan. 2015. Pengertian Arduino Uno Mikrokontroler Atmega328. (Online)
<http://www.caratekno.com/2015/07/pengertian-arduino-uno-mikrokontroler.html> diakses tanggal 26 Agustus 2016.
- [8]. Puspitasari, Diah Eka, dkk. 2012. Pengenalan Wajah Menggunakan Metode Principal Component Analysis (PCA) Untuk Aplikasi Sistem Keamanan Rumah. Universitas Diponegoro.
- [9]. Sulistyorini, Prastuti.2009. Pemodelan Visual Dengan Menggunakan UML dan Rational Rose. Jurnal Teknologi Informatika DINAMIK Vol.14,No.1,Hal:23-29.
- [10]. Tyas. 2013. Algoritma Eigenface. (Online)
<http://informatika.web.id/algoritma-eigenface.htm> diakses tanggal 05 September 2016.