

IMPLEMENTASI ALGORITMA *LOCAL BINARY PATTERN* PADA SISTEM KEAMANAN PARKIR BERBASIS RASPBERRY PI

LOCAL BINARY PATTERN ALGORITHM IMPLEMENTATION IN PARKING SECURITY SYSTEM BASED RASPBERRY PI

M.Bagas Gigih Yuda Prasetyo¹, Astri Novianty², Agung Nugroho Jati³

¹²³Prodi S1 Sistem Komputer, Fakultas Teknik Elektro, Universitas Telkom

¹bagaze@students.telkomuniversity.ac.id, ²astrinov@telkomuniversity.ac.id,
³agungnj@telkomuniversity.ac.id

Abstrak

Sistem pengenalan biometrik merupakan sistem pengenalan pola yang menggunakan karakteristik fisiologis atau karakteristik perilaku untuk mengenali identitas seseorang. Wajah merupakan salah satu karakteristik fisiologis yang paling berpotensi digunakan dalam sistem pengenalan biometrik.

Penelitian ini bertujuan dan fokus dalam pengimplementasian pengolahan citra digital dengan merancang suatu sistem mampu untuk mengenal (*recognition*) wajah seseorang. Sistem ini dirancang dengan bahasa *Python* menggunakan software *IDLE(python GUI)*. Selain itu sistem ini juga mengimplementasikan pengolahan citra digital yang digunakan untuk mengenal (*recognition*) plat nomor menggunakan algoritma *openALPR*. Sistem pengenalan wajah menggunakan algoritma LBP (*Local Binary Patern*) pada citra latih dan uji. Dan sistem ini menggunakan klasifikasi *euclidean distance* untuk mencocokkan ciri dari citra latih dengan citra uji.

Dari simulasi sistem yang sudah dilakukan, didapatkan akurasi tertinggi dari sistem yaitu 93,33%. Kondisi tersebut didapat dengan menggunakan nilai *threshold* sebesar 1,5, artinya apabila *euclidean distance* dari ciri citra uji dengan data ciri citra latih diatas 1,5 maka sistem tidak mengenali citra uji tersebut. Berdasarkan hasil pengujian, jarak terbaik antara kamera dengan objek adalah 1 meter, kamera dapat diletakkan dengan sudut 45⁰,90⁰,135⁰, dan dengan nilai *crop* terbaik 92 x 112. Rata-rata waktu komputasi dari sistem ini adalah 2,098571429 detik, dengan rata-rata *CPU Usage* 30,08571% yang menandakan bahwa sistem ini tidak menghabiskan *resource memory*.

Abstract

Biometric recognition system is a pattern recognition system that uses physiological characteristics or behavioral characteristics to identify a person's identity. Face is one of the most potential physiological characteristics used in biometric recognition system.

The aim of this research and focus in the implementation of digital image processing to design a system capable to recognize someone's face. This system designed with the Python language using software *IDLE (Python GUI)*. In addition the system also implements digital image processing is used to recognize the plate number using *openALPR* algorithm. Face recognition system uses an algorithm LBP (*Local Binary Patern*) in the training and test images. And the system uses a classification *euclidean distance* to match the characteristics of the image practiced with test images.

From simulation systems already done, get the highest accuracy of the system is 93.33%. The condition was obtained by using a threshold value of 1.5, meaning that if euclidean distance of the characteristics of test images with the data train characteristic image above 1.5 then the system does not recognize the test images. Based on test results, the best distance between the camera and the object is 1 meter, the camera can be placed at an angle 45,90,135, and with the best normalization value 92 x 112. Average computing time of this system is 2.098571429 seconds, with an average *CPU Usage* 30.08571%, which indicates that this system does not spend memory resource.

1. Pendahuluan

Salah satu penunjang keamanan yang sering digunakan pada suatu sistem tempat parkir adalah menggunakan karcis parkir. Namun seiring berjalannya waktu penggunaan karcis parkir dinilai kurang begitu efektif, banyak masalah yang timbul dengan sistem ini, antara lain : kertas parkir yang hilang dan masalah global warming (karna penggunaan kertas skala besar setiap harinya). Selain menggunakan kertas parkir, banyak perusahaan yang telah menggunakan sistem ID Card / Member , namun hal ini juga memiliki kekurangan, yaitu tentang kartu ID yang hilang dan urusan kepemilikan kartu.

Perkembangan teknologi komputer saat ini sangat pesat. Komputer pertama di dunia memiliki ukuran yang sangat besar bahkan sebesar ruangan. Tetapi saat ini ukuran dari sebuah komputer sudah sangat kecil bahkan sudah bisa dibawa kemana-mana. Salah satu jenis komputer yang sangat kecil dan murah adalah *Raspberry Pi*. *Raspberry Pi* (juga dikenal sebagai RasPi) adalah sebuah SBC (*Single Board Computer*) seukuran kartu kredit yang dikembangkan oleh Yayasan Raspberry Pi di Inggris (UK). Secara spesifikasi, *Raspberry Pi* menggunakan arsitektur berbasis ARM dengan *System On Chip (SoC)*. *Raspberry Pi* dapat dijalankan menggunakan sistem operasi berbasis *open source* seperti *GNU/Linux*.

Tujuan dari perancangan penelitian ini adalah untuk merancang dan mengimplementasikan algoritma PCA pada *Raspberry Pi* dalam melakukan proses pengolahan citra. Selain itu untuk mengukur performansi *Raspberry Pi* ketika melakukan pengolahan citra.

Metode yang digunakan dalam penelitian ini adalah antara lain Studi Literatur, Pengumpulan data, Studi pengembangan aplikasi, Implementasi program aplikasi, Analisis performansi, dan Pengambilan kesimpulan

2. Dasar Teori

2.1 Ekstraksi Ciri

Ekstraksi ciri merupakan suatu proses untuk mendapatkan ciri dari sebuah objek atau citra. Untuk mendapatkan ciri dari suatu citra memerlukan suatu metode khusus. Salah satu metode yang digunakan dalam ekstraksi ciri sebuah citra adalah *Local Binary Patern (LBP)*.

2.1.1 Local Binary Patern

Metode LBP sering disebut dengan eigenface atau eigenvector yang diterapkan pada wajah. Hal ini disebabkan ekstraksi ciri yang digunakan untuk mencirikan suatu objek merupakan hasil proses persamaan karakteristik yaitu eigenvector. Jadi, dalam proses ekstraksi fitur menggunakan LBP terdapat proses untuk menghasilkan eigenvector terlebih dahulu.

2.1.2 Eigenface^[3]

Konsep *Eigenface* adalah informasi wajah diproyeksikan ke dalam ruang multidimensional vektor orthogonal yang disebut ruang wajah (*face space*). Faktanya, kumpulan vektor orthogonal yang merepresentasikan *face space* merupakan eigenvektor dari matriks kovarian citra wajah pada *training set*.

2.2 K-Nearest Neighbor (k-NN)^[4]

Algoritma *k-nearest neighbor* (k-NN) adalah sebuah metode untuk melakukan klasifikasi terhadap objek berdasarkan data pembelajaran yang jaraknya paling dekat dengan objek tersebut.

Data pembelajaran diproyeksikan ke ruang berdimensi banyak, dimana masing-masing dimensi merepresentasikan fitur dari data. Ruang ini dibagi menjadi bagian-bagian berdasarkan klasifikasi data pembelajaran. Sebuah titik pada ruang ini ditandai kelas *c* jika kelas *c* merupakan klasifikasi yang paling banyak ditemui pada *k* buah tetangga terdekat titik tersebut. Dekat atau jauhnya tetangga biasanya dihitung berdasarkan jarak *Euclidean*, *Cosine*, *Cityblock*, *Correlation* dan sudut antar vektor fitur.

$$d(\mathbf{a}, \mathbf{b}) = \sqrt{\sum_{i=1}^n (a_i - b_i)^2}$$

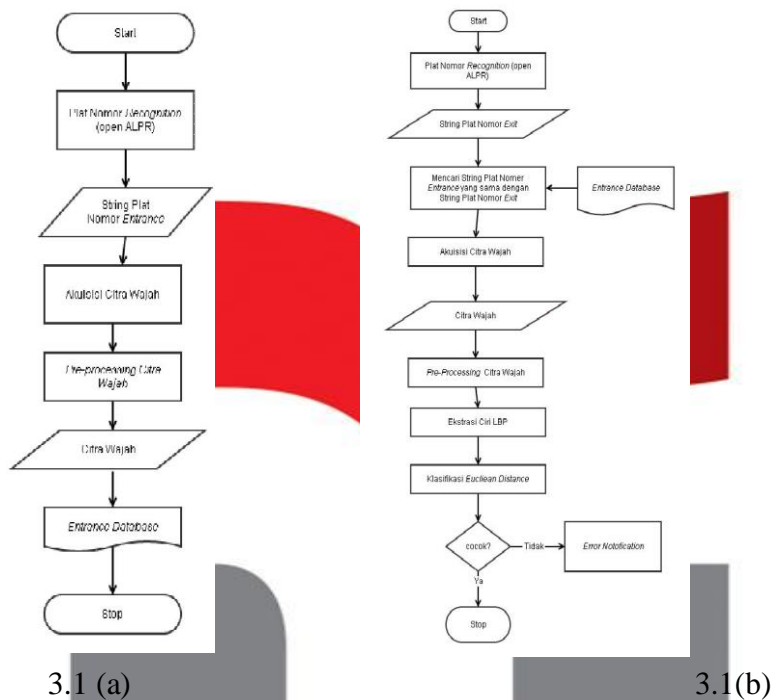
(2.1)

3. Perancangan Sistem

Secara umum pemodelan sistem dilakukan dalam 2 tahap, yaitu proses saat pintu masuk dan proses pada saat pintu keluar yang akan dilakukan dengan bahasa pemrograman python.

Proses input data merupakan proses awal sistem yang dilakukan secara *realtime*. Pada bagian ini, data input berupa citra plat nomor dan citra wajah dari masing-masing individu yang nanti akan kita kenali. Semua ini diperlukan untuk pembentukan identitas yang selanjutnya tersimpan dalam *database*.

Pada penelitian ini akan disimulasikan sistem pengenalan plat nomor dengan menggunakan *open ALPR* dan lebih fokus kepada sistem pengenalan wajah dengan menggunakan metode *Local Binary Patern (LBP)*. Untuk metode pengenalan wajah ini akan diuji akurasi dengan menggunakan metode klasifikasi *euclidean distance*. Secara umum sistem dari penelitian ini dapat digambarkan pada diagram alir pada Gambar 3.1 dan Gambar 3.2.



Gambar 3.1(a) Diagram Alir Sistem Pintu Masuk , Gambar 3.2(b) Diagram Alir Sistem Pintu Keluar

3.1 Diagram Alir Sistem

3.1.1 Plat Nomor Recognition

ALPR dibagi dalam dua langkah utama, yaitu *plate detection* dan *plate recognition*. *Plate detection* digunakan untuk mendeteksi lokasi plat yang terdeteksi dalam foto, segmentasi plat termasuk pada proses kedua, yaitu *plate recognition* yang menggunakan algoritma OCR untuk menentukan karakter alfanumerik di plat. Langkah *face detection* terdiri dari tiga bagian, yaitu akuisisi citra plat nomor, *pre-processing*, dan lokalisasi. Sedangkan pada *plate recognition* terdapat proses analisis komponen terhubung, segmentasi, dan karakter *recognition*.

3.1.2 Akuisisi Citra Wajah

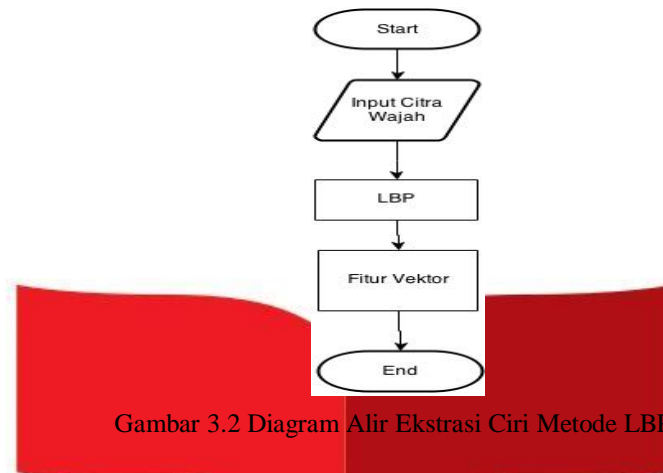
Proses akuisisi citra merupakan proses pengambilan citra dengan menggunakan kamera *webcam*. Pada sistem ini, proses akuisisi citra dilakukan secara langsung (*realtime*). Proses pengambilan citra ini dilakukan di pintu masuk dan pintu keluar. Pada pintu masuk akan dilakukan pengambilan lima citra untuk setiap wajah, dan nantinya citra tersebut akan tersimpan di dalam *Entrance Database* dengan filename sesuai dengan string plat nomor. Pada pintu keluar, sistem akuisisi citra dilakukan dengan pengambilan satu citra wajah saja. Hasil keluaran dari proses akuisisi citra ini akan menjadi masukan di proses *pre-processing*

3.1.3 Pre-Processing Citra Wajah

Tahapan ini dilakukan untuk mempersiapkan citra agar proses ekstraksi ciri yang akan dilakukan setelahnya berlangsung optimal. Pada tahapan ini kualitas citra yang telah diakuisisi diperbaiki dan data-data yang tidak dibutuhkan saat ekstraksi ciri akan dibuang.

3.1.4 Ekstrasi Ciri LBP

Ekstrasi ciri merupakan proses pengambilan ciri dari sebuah citra yang menggambarkan karakteristik dari suatu objek, dan digunakan sebagai pembeda antara karakter yang satu dengan karakter yang lainnya.



Gambar 3.2 Diagram Alir Ekstrasi Ciri Metode LBP

Ada dua hal penting pada implementasi algoritma LBP ini, yaitu proses *enrollment*/registrasi dan pengenalan. Pada tahap *enrollment*, beberapa citra *training* akan diambil nilai tertentu atau *feature extraction* yaitu proses *thresholding* dengan membandingkan *pixel* tetangga dengan *pixel* pusatnya. Setelah itu, dihitung nilai binernya dengan cara *clockwise* atau *counterclockwise*. Dengan memasukkan nilai *weighted* pada setiap blok *pixel* maka akan didapatkan nilai desimal dari citra tersebut. Pada proses ini, akan menghasilkan fitur vektor (*feature vector*). Selanjutnya adalah proses pengenalan. Tahap ini adalah membandingkan nilai vektor dari citra *training* dengan citra asli yang akan dibandingkan.

Pada penelitian ini citra yang digunakan mempunyai ukuran 92x112 maka akan dibagi kedalam blok 3x3 agar menghasilkan 8 bit *pixel*. Dari blok LBP 3x3, didapat nilai *pixel* pusat lalu dibandingkan/*thresholding* dengan *pixel* tetangganya. Didapat bahwa proses LBP 3x3 menghasilkan nilai desimal 241 yang akan dibandingkan pada proses klasifikasi

3.1.5 Klasifikasi *Euclidean Distance*

Pada penelitian ini klasifikasi sistem yang digunakan adalah *Euclidean Distance*. *Euclidean Distance* adalah metode pengukuran kemiripan yang sederhana.

3.1.6 *Error Notification*

Pada penelitian ini, sistem akan memberikan *Error Notification* apabila nilai dari hasil klasifikasi lebih dan sama dengan satu, maka sistem secara otomatis akan memberikan *error notification*. Untuk nilai dari hasil klasifikasi yang kurang dari satu, sistem akan menyatakan benar dan menghentikan proses file tersebut.

3.2 Performansi Sistem

Setelah melakukan seluruh proses, selanjutnya dilakukan pengujian sistem sistem. Pengujian ini dilakukan untuk mengetahui performansi sistem sehingga dapat diketahui kelebihan dan kekurangan sistem. Performansi sistem diukur berdasarkan parameter sebagai berikut :

1. Akurasi Sistem

Akurasi merupakan ukuran ketepatan sistem dalam mengenali masukan yang diberikan sehingga menghasilkan keluaran yang benar. Secara matematis dapat dituliskan seperti pada persamaan di bawah ini

$$\text{Akurasi} = \frac{\text{Jumlah data benar}}{\text{Jumlah data keseluruhan}} \times 100\%$$

2. Waktu Komputasi

Waktu komputasi adalah waktu yang dibutuhkan sistem untuk melakukan suatu proses. Pada sistem ini, waktu komputasi dihitung dengan menggunakan waktu selesai dikurangi waktu mulai, sehingga akan didapatkan waktu komputasi sistem.

4. Pengujian Sistem dan Analisis

4.1 Perbandingan Tingkat Akurasi dengan Treshold yang Berbeda

Tabel 4.1 Hasil Akumulasi Pengujian Treshold

| No | Treshold | Banyak Pengujian | Data Benar | Data Salah | Akurasi |
|----|----------|------------------|------------|------------|---------|
| 1 | 1 | 30 | 1 | 29 | 3,33% |
| 2 | 1,1 | 30 | 1 | 29 | 3,33% |
| 3 | 1,2 | 30 | 3 | 27 | 10% |
| 4 | 1,3 | 30 | 14 | 16 | 46,66% |
| 5 | 1,4 | 30 | 27 | 3 | 90% |
| 6 | 1,5 | 30 | 28 | 2 | 93,33% |
| 6 | 1,6 | 30 | 24 | 6 | 80% |

Dilihat dari tabel pengujian diatas bahwa treshold paling optimal didapat di angka 1,5 yang memiliki akurasi 93,33%. Sehingga dapat disimpulkan bahwa sistem ini di tiap pengujiannya akan lebih optimal apabila menggunakan treshold 1,5.

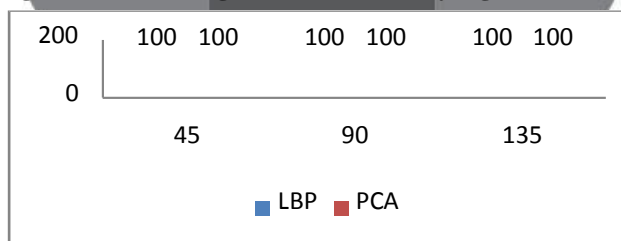
4.2 Perbandingan Tingkat Akurasi dengan Penempatan Kamera yang Berbeda

Tabel 4.2 Hasil pengujian Tingkat Akurasi dengan Penempatan Kamera yang Berbeda

| No | Jarak (meter) | Banyak Percobaan | Data Benar | Data Salah | Akurasi |
|----|---------------|------------------|------------|------------|---------|
| 1 | 1 | 5 | 3 | 2 | 60% |
| 2 | 1,5 | 5 | 1 | 4 | 20% |
| 3 | 2 | 5 | 0 | 0 | 0% |
| 4 | 2,5 | 5 | 0 | 0 | 0% |

Dilihat dari tabel pengujian di atas, maka jarak optimal antara objek dan kamera adalah sejauh satu meter, dengan akurasi 60%.

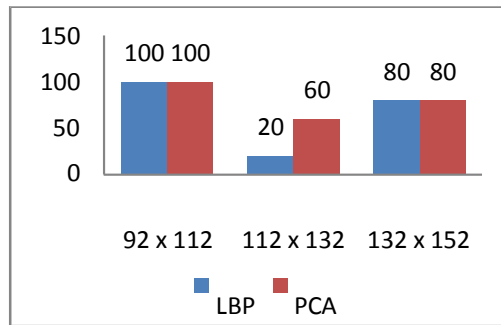
4.3 Perbandingan Tingkat Akurasi dengan Sudut Kamera yang Berbeda



Gambar 4.1 Grafik Perbandingan Tingkat Akurasi dengan Sudut Kamera yang Berbeda

Dari tabel hasil pengujian di atas dapat dilihat bahwa kamera dapat diletakkan pada sudut 45°, 90°, 135° karna dari hasil pengujian akurasi dari tiap sudut menunjukkan hasil 100%.

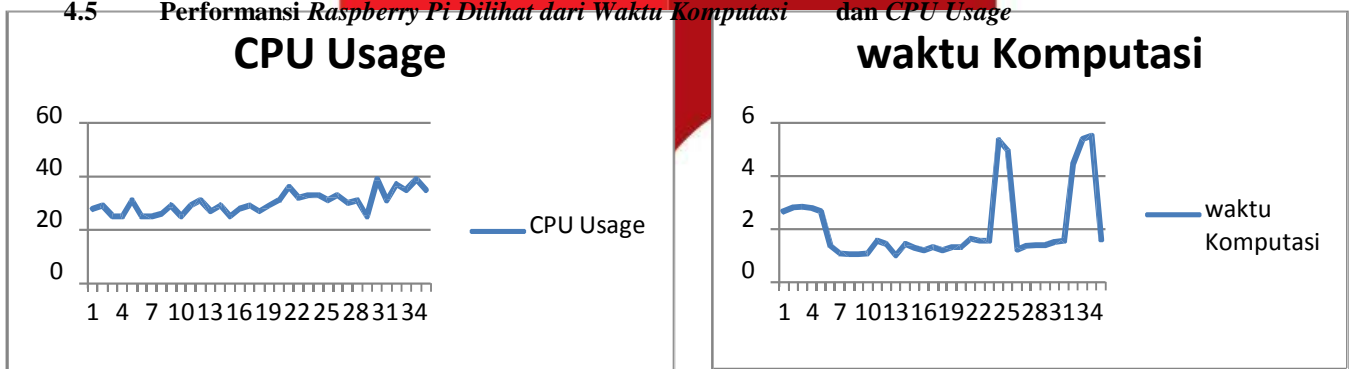
4.4 Perbandingan Tingkat akurasi dengan Ukuran Crop yang Berbeda



Gambar 4.2 Grafik Perbandingan Tingkat Akurasi dengan Ukuran Crop yang Berbeda

Tingkat akurasi tertinggi terdapat pada saat ukuran *crop* yang diberikan adalah sebesar 92x112. Hal ini disebabkan karena pada saat ukuran *crop* 112x132 dan 132x152, bagian lain selain wajah ikut masuk, seperti helm dan mungkin *noise* lainnya yang masuk oada citra *crop* nantinya sehingga mengganggu ciri dari wajah dan menurunkan tingkat akurasi. Maka dari itu, ukuran *crop* yang ditetapkan untuk sistem ini adalah 92x112.

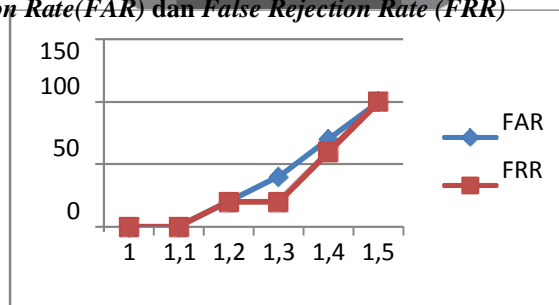
4.5 Performansi Raspberry Pi Dilihat dari Waktu Komputasi dan CPU Usage



Gambar 4.3 Grafik Performansi Raspberry Pi Dilihat dari Waktu Komputasi dan CPU Usage

Dari hasil percobaan di atas dapat dilihat bahwa rata-rata waktu komputasi dari dua skenario percobaan adalah 2,098571429 detik , hal ini menunjukkan bahwa dalam satu waktu proses sistem ini tidak membutuhkan waktu yang lama, yaitu rata-rata hanya 1,9 detik saja. Dan untuk rata-rata *CPU Usage* nya adalah sebesar 31,48571429% , hal ini menunjukkan bahwa dalam satu waktu proses sistem ini tidak begitu memberatkan sistem dari *Raspberry Pi* itu sendiri, karna hanya memakai rata-rata 30,08571%.

4.6 Nilai False Acception Rate(FAR) dan False Rejection Rate (FRR)



Gambar 4.4 Grafik FAR dan FRR

dapat kita lihat bahwa grafik FRR dan FAR berpotongan pada saat nilai *threshold* 0,4. Itu berarti sistem akan optimal apabila ditetapkan nilai *threshold* sebesar 0,4, yaitu pada saat nilai FRR dan FAR memiliki nilai yang hampir sama. Nilai FRR yang didapatkan pada saat *threshold* sebesar 0,4 adalah sebesar 100%, dan nilai

FAR-nya adalah sebesar 100%. Hal ini membuktikan pada skenario pengujian pertama dimana menghasilkan treshold optimal 1,5 ,dan setelah pengujian treshold dengan menggunakan sistem yang lengkap, kembali mendapatkan treshold optimal 1,5. Maka dari itu, ukuran treshold yang ditetapkan untuk sistem ini adalah 1,5.

5. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dari pengujian dan analisis yang telah dilakukan pada penelitian ini dapat diambil beberapa kesimpulan sebagai berikut :

1. Nilai akurasi paling tinggi berhasil didapatkan pada saat pengujian treshold sebesar 1,5 yaitu 93,33%
2. Jarak paling optimal antara objek dan penempatan kamera yang didapat dari hasil pengujian , yaitu sepanjang satu meter.
3. Kamera dapat ditempatkan pada sudut 45° , 90° , 135° dengan objek.
4. Ukuran *crop* yang paling memberikan nilai akurasi paling tinggi adalah ukuran *crop* 92x112
5. Rata-rata waktu komputasi untuk menjalankan satu sistem adalah 2,098571429 detik dan rata-rata *CPU Usage* dari sistem ini adalah 30.08571%.
6. Nilai *threshold* yang optimal dari suatu sistem didapatkan apabila nilai dari FRR dan FAR yang dihasilkan hampir sama besar. Nilai *threshold* yang paling optimal untuk sistem ini adalah nilai treshold sebesar 1,5. Nilai treshold ini memiliki nilai FRR sebesar 100%, dan nilai FAR-nya adalah sebesar 100%

5.2 Saran

Adapun saran untuk pengembangan penelitian selanjutnya adalah :

1. Menggunakan modul *Raspberry Pi* type 2 karena memiliki RAM yang lebih besar yaitu 1 GB, sehingga waktu komputasi lebih cepat.
2. Agar dicoba menggunakan pemrograman parrarel, agar kinerja semakin cepat
3. Mencoba ekstraksi ciri lainnya yang mungkin bisa lebih cocok dan meningkatkan akurasi untuk pengenalan wajah.
4. Mencoba algoritma yang mungkin bisa lebih cocok dan meningkatkan akurasi untuk pengenalan plat nomor

DAFTAR PUSTAKA

- [1] Nasution, N. M. 2010. *Desain Dan Implementasi Sistem Penerjemah Bahasa Isyarat Berbasis Webcam Dengan Metode Linear Discriminant Analysis*. Bandung: Institut Teknologi Telkom.
- [2] Muntasa, A., & Purnomo, M. H. 2010. *Konsep Pengolahan Citra Digital dan Ekstraksi fitur*. Yogyakarta: Graha Ilmu.
- [3] Kania Dara Pradini. 111080217. 2012. *Pengenalan Wajah Manusia dengan Metode Spectral Eigenface pada Citra Hyper Spectral untuk Verifikasi Identitas Pribadi*. IT Telkom.
- [4] Laura Sánchez López. 2010. *Local Binary Patterns applied to Face Detection and Recognition*. Barcelona : Universitat Politècnica De Catalunya.
- [5] T.Ojala, M.Pietikäinen & T.Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with Local Binary Patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971-987, July 2002.
- [6] T.Ahonen, A.Hadid & M.Pietikäinen, Face Description with Local Binary Patterns: Application to Face Recognition. Draft, 5th June 2006.
- [7] Arief Trifianto Nurichsan. 1101100100. 2014. *Analisis dan Perancangan Sistem Pengenalan Iris Mata Manusia Berbasis Java*. Telkom University
- [8] Jinesh,K.J. 2014. *Advance License Plate Recognition System For Car Parking*. Thrissur, India: Jyothi Engineering College