

## Implementasi Pengenalan Wajah Pada Sistem Penguncian Rumah dengan Metode Template Matching Menggunakan *Open Source Computer Vision Library (Opencv)*

Husnibes Muchtar<sup>1</sup>, Rizky Apriadi<sup>2</sup>

<sup>1,2)</sup> Teknik Elektro, Universitas Muhammadiyah Jakarta

<sup>1,2)</sup> Jl. Cempaka Putih Tengah 27 no 47 Jakarta 10510

E-mail: rizky.apriadi92@gmail.com

### Abstrak

*Sistem keamanan rumah konvensional saat ini masih tergolong kurang aman dan kurang efisien. Dimana hal ini masih cukup merepotkan para pengguna. Berbekal pengalaman dalam menggunakan sistem keamanan konvensional, penelitian ini diharapkan dapat memberikan kemudahan dalam penggunaannya namun tetap memberikan rasa aman. Penelitian ini menggunakan raspberry pi sebagai otak utama serta sebagai server. Pada sistem ini, raspberry akan diinstal software OpenCV. Pada library opencv terdapat sebuah metode template matching. Metode ini yang akan digunakan pada penelitian ini yang dipadukan penggunaannya dengan sensor kamera dan sensor ultrasonic. Keluaran dari pengolahan citra digital ini akan diimplementasikan pada sistem keamanan penguncian rumah.*

**Kata kunci:** *OpenCV, template matching, kamera, citra digital*

### Abstrak

*Conventional home security systems are still relatively unsafe and inefficient. Where this is still quite troublesome for users of conventional security systems. Armed with experience in using conventional security systems, this research is expected to provide convenience in use but still provide a sense of security. This study uses raspberry pi as the first otaku and as a server. In this system, Raspberry will be installed with OpenCV software. In the OpenCV library there is a template matching method. This method that will be used in this study is combined with the use of camera sensors and ultrasonic sensors. The output of digital image processing will be implemented in a house locking security system.*

**Keywords:** *OpenCV, template matching, camera, digital image*

## 1 PENDAHULUAN

### 1.1 Latar Belakang

Dihunian yang modern saat ini, tak sedikit pula rumah yang memiliki kamera *cctv* untuk mendukung keamanan rumahnya. Namun hal ini tidak cukup untuk menghalau terjadinya pencurian. Dengan kemajuan ilmu teknologi pengolahan citra digital saat ini, semakin banyak pula penggunaannya yang mempermudah serta membantu seseorang dalam kehidupan sehari-hari. Dengan bekal *front camera, smartphone* kini sudah dapat menggunakan sistem penguncian berbasis pengolahan citra digital. Pengguna hanya tinggal mendekati wajah kearah *front camera* maka secara otomatis kamera akan meng-*capture* wajah untuk mencocokkannya dengan data yang sudah tersimpan terlebih dahulu. Dengan melihat kecocokan antar wajah yang ter-*capture* dengan data yang tersimpan maka pengunci layar *smartphone* akan terbuka. Deteksi wajah adalah teknologi komputer yang menentukan lokasi dan

ukuran wajah manusia di gambar (digital) sembarang [1].

Dua sistem ini akan digabungkan menjadi sebuah sistem yang dapat berguna bagi kehidupan modern saat ini. Penerapan ini akan menggunakan teori *template matching* yang dimiliki oleh *platform OpenCV*. Dengan sistem ini maka akan digabungkan dua buah sistem keamanan yaitu keamanan konvensional pada pintu rumah dan keamanan digital menggunakan kamera yang dapat dimonitoring melalui *webbase*. *Raspberry pi* adalah sebuah komputer kecil dan terjangkau yang bisa dipakai untuk belajar pemrograman [2]. *Arduino* adalah sebuah *platform* elektronik yang *open-source* yang berbasis perangkat lunak dan perangkat keras yang mudah digunakan [3]. *OpenCV (Open Source Computer Vision Library)* dirilis di bawah lisensi BSD dan gratis untuk penggunaan akademik dan komersil [4]. Gagasan bahwa internet akan bekerja berdampingan dengan begitu banyak perangkat rumah kita sangat mengejutkan [5]. Pengolahan citra

sangat bermanfaat, diantaranya adalah untuk meningkatkan kualitas citra, menghilangkan cacat pada citra, mengidentifikasi objek, penggabungan dengan bagian citra yang lain [6].

## 2 METODOLOGI

*OpenCV* (Open Source Computer Vision Library) adalah perpustakaan komputer open source visi dan perangkat lunak pembelajaran. *OpenCV* dibangun untuk menyediakan infrastruktur umum untuk aplikasi visi komputer dan untuk mempercepat penggunaan persepsi mesin dalam produk komersial. Menjadi produk berlisensi BSD, *OpenCV* memudahkan bisnis untuk memanfaatkan dan memodifikasi. kode library ini memiliki lebih dari 2500 algoritma yang dapat dioptimalkan, yang mencakup satu set lengkap visi komputer klasik dan state-of-the-art komputer vision dan mesin pembelajaran algoritma. *OpenCV* adalah metode yang paling cepat dan memiliki library paling lengkap untuk komputer vision [7].

### 2.1 Template Matching

*Template matching* adalah sebuah teknik dalam pengolahan citra digital untuk menemukan bagian-bagian terkecil dalam pencocokan gambar dengan *template* gambar. *Template* dalam konteks rekognisi pola menunjuk pada konstruk internal yang jika cocok (*match*) dengan stimulus penginderaan mengantar pada rekognisi suatu objek.

- Cara kerja Template Matching

Untuk dapat bekerja *template matching* membutuhkan dua komponen utama, yaitu:

- a. Sumber gambar : yaitu gambar dimana kita berharap menemukan kecocokan dengan gambar *template*
- b. Gambar *template* : yaitu gambar tambalan yang akan dibandingkan dengan sumber gambar.

### 2.2 Algoritma Pengenalan Wajah Dengan Metode Face-ARG

Metode Face-ARG adalah sebuah metode baru dalam proses pencocokan gambar dari dua wajah. Metode Face-ARG (*Attributed Relational Graph*) mentransformasikan gambar suatu wajah menjadi bentuk suatu graf yang terdiri dari himpunan node dan edge yang saling berhubungan. Metode Face-ARG (*Attributed Relational Graph*) dapat didefinisikan dalam bentuk:

$$face - ARG : g = (V, R, F)$$

Dimana:

$V = \{v_1, \dots, v_n\}$  adalah himpunan node dari suatu *graph*

$R = \{r_{ij} | v_i, v_j \in V, i \neq j\}$  adalah himpunan relasi biner dari vektor-vektor yang terdapat dalam *node*.

$F = \{R_i | i = 1, \dots, N\}$  himpunan dari relasi ruang vektor dari *node*.

Relasi ruang vektor  $R_i = \{r_{ij} | j = 1, \dots, N, j \neq i\}$ , dengan  $i = 1, \dots, N$  merepresentasikan himpunan relasi vektor antara *node*  $V_i$  dengan *node-node* lain didalam  $V$ . Jika terdapat dua wajah yang sama maka relasi ruang vektor yang terdapat dalam bentuk ARG kedua wajah tersebut seharusnya juga sama, dengan membandingkan relasi ruang vektor dari dua gambar wajah maka akan dievaluasi tingkat keamanan dari kedua gambar tersebut. Untuk mendeskripsikan objek secara invarian agar dapat melakukan proses RTS (*Rotasi, Translasi, Skala*) maka digunakan enam tipe ukuran yang berbeda:

$$r_{ij}(1) = \theta, r_{ij}(2) = \theta c, r_{ij}(3) = \theta m$$

$$r_{ij}(4) = DR, r_{ij}(5) = m_{ij}, r_{ij}(6) = m_{ij}$$

dimana  $r_{ij}(1)$  adalah sudut antara dua segment garis  $v_i$  dan  $v_j$ ,  $r_{ij}(2)$  adalah komplemen sudut dari sudut terkecil di dalam vektor,  $r_{ij}(3)$  adalah sudut antara titik tengah dari  $v_i$  terhadap  $v_j$ ,  $r_{ij}(4)$  adalah rasio jarak (*distance ratio/DR*) yang dihitung dengan persamaan:

$$DR = \frac{l1 + l2}{(l3 + l4 + l5 + l6)/4}$$

Dimana  $l(i)$  untuk  $i=1,2,3,\dots,6$

$r_{ij}(5)$  dan  $r_{ij}(6)$  adalah koordinat dari titik tengah vektor  $m_{ij}$ , yang merupakan vektor dari dua titik tengah  $v_i$  dan  $v_j$ . Pola yang diberikan oleh *Face-ARG* berupa koordinat-koordinat yang diberikan pada wajah sehingga membentuk suatu graf yang nantinya akan dinyatakan dalam bentuk vektor garis. Dari graf yang dihasilkan akan diperoleh vektor jarak yang menyatakan hubungan antar *node* yang saling berkorespondensi dalam graf yang dihasilkan oleh masing-masing *image*. Sehingga dapat disimpulkan apabila suatu *image* masuk vektor graf yang dihasilkan juga harus sama dengan vektor yang dihasilkan oleh *image* yang sama yang disimpan dalam basis data, dengan cara membandingkan tingkat kesamaan antar graf yang dihasilkan oleh setiap *image*. Tingkat kesamaan antara dua *Face-ARG* yaitu :

$$S(G^{G1 \leftrightarrow G2}) = \sum_{i=1}^n D Ri . Wi$$

$$= \sum_{i=1}^n Wi . \sum_{j=1}^n , j \neq i p r_{ij}^{G2} - r_{ij}^{G1} . Y_{ij}$$

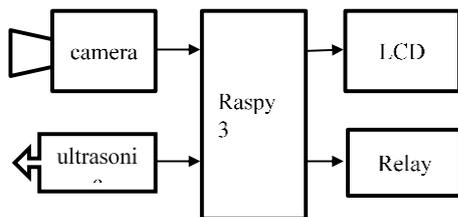
Dimana  $D(R_i)$  adalah fungsi untuk menghitung perbedaan antara relasi ruang vektor  $R_i^{G^1}$  dan  $R_i^{G^2}$ .  $p(r_{ij}^{G^2} - r_{ij}^{G^1})$  adalah probabilitas error pada relasi ruang vektor dan  $\omega_i$  dan  $Y_{ij}$  adalah faktor berat dari relasi biner antara fitur  $v_i$  dan  $v_j$ . Proses pencocokan yang paling baik dan efektif dari kedua wajah yang diuji kesamaannya dapat diidentifikasi dengan menghitung nilai kesamaan tertinggi dari keduanya yang berada di atas nilai threshold. Nilai threshold tersebut didefinisikan sebagai batas pengenalan akhir dari kedua Face-ARG.

$$\text{Face-ID} = \text{arg}_{G_M \in FDB} \max S(G_M^G \leftrightarrow G_T)$$

Maka rumus di atas akan menghasilkan nilai yang paling tinggi dari proses pencocokan antara nilai image yang diinput dengan image di database.

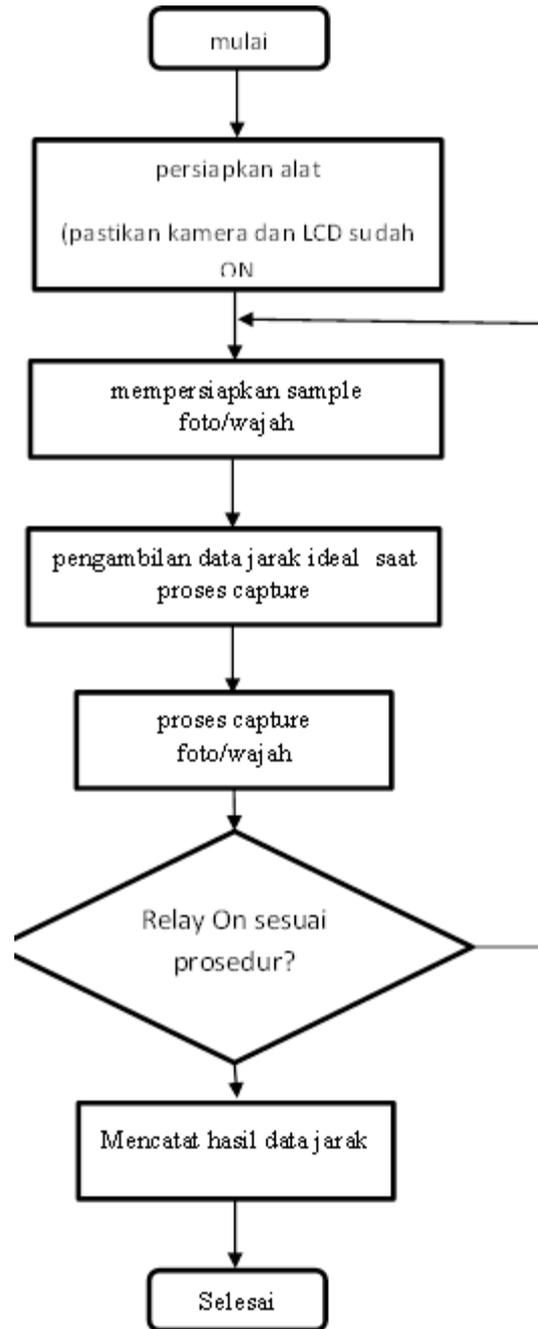
### 2.3 Perancangan Sistem dan Flowchart

Sebelum memulai merancang prototype maka dibutuhkan suatu kriteria yang dijadikan sebuah acuan dasar dalam merancang sistem. Secara garis besar, kriteria kendaliannya adalah dimulai dari start awal yaitu kamera mengambil video secara kontinyu agar dapat dilihat secara live melalui webbase. Kemudian ketika akan memasuki rumah maka akan tersedia sebuah sensor jarak untuk meng-capture wajah. Selain itu terdapat sebuah layar lcd yang berfungsi untuk mengetahui posisi wajah pada kamera. Setelah wajah ter-capture maka akan dicocokkan oleh sistem. Wajah yang diambil memiliki kecocokan dengan wajah yang terdapat pada database maka membuka pintu rumah. sistem kelistrikan rumah akan ON dan akan



Gambar 1 Skema sistem prototype.

Penentuan posisi dan jarak kamera dengan target dan pencahayaan memiliki pengaruh besar dalam pengenalan objek [8]. Penelitian ini dilakukan didalam ruangan dengan penerangan menggunakan lampu LED yang memiliki 400 lumen dan memiliki ukuran lux sebesar 150.



Gambar 2 Flowchart penelitian.

### 3 HASIL DAN PEMBAHASAN

Penelitian ini dilakukan dengan dengan cara mengambil data dari percobaan yang berfungsi untuk mengetahui kinerja dan kekurangan dari sistem. Proses pengujian ini dilakukan untuk mencari tahu adanya kesalahan atau tidak pada setiap blok rangkaian yang digunakan. Dari setiap pengujian tersebut diberikan hasil percobaan dan analisa.

Untuk dapat menilai perbandingan pada setiap nilai threshold maka dapat diperhatikan pada grafik dibawah ini. Dapat dilihat bahwa nilai Treshold 0.6

memiliki kestabilan dalam waktu yang tidak terlalu jauh antar nilai jarak yang diambil. Sehingga nilai threshold 0.6 dijadikan sebagai nilai yang rekomen untuk dijadikan nilai tetap pada program dalam percobaan ini.

Tabel 1 Hasil pengukuran pertama.

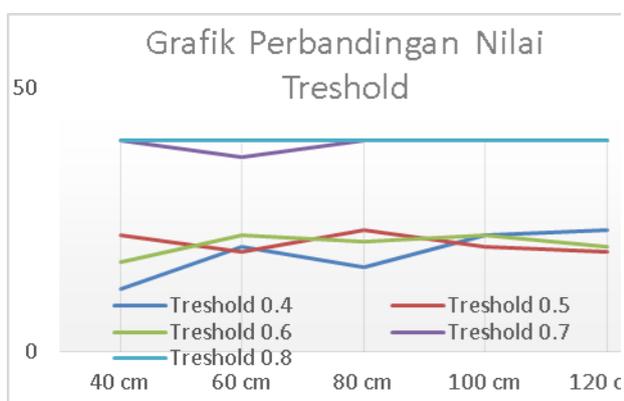
Jarak pengukuran	Waktu pengukuran	
	Threshold 0.4	Threshold 0.5
40cm	12 detik	22 detik
60cm	20 detik	19 detik
80cm	16 detik	23 detik
100cm	22 detik	20 detik
120cm	23 detik	19 detik

Tabel 2 Hasil pengukuran kedua.

Jarak pengukuran	Waktu pengukuran		
	Threshold 0.6	Threshold 0.7	Threshold 0.8
40cm	17 detik	3 menit 43 detik	uncompare
60cm	22 detik	57 detik	uncompare
80cm	21 detik	Uncompare	uncompare
100cm	22 detik	Uncompare	uncompare
120cm	20 detik	Uncompare	uncompare

## DAFTAR PUSTAKA

- [1] M. S. Kalas, "Real Time Face Detection And Tracking Using OpenCV," *Int. J. Soft Comput. Artif. Intell.*, vol. 2, no. 1, pp. 41–44, 2014.
- [2] "Raspberry Pi — Teach, Learn, and Make with Raspberry Pi." [Online]. Available: <https://www.raspberrypi.org/>. [Accessed: 02-Apr-2019].
- [3] "Arduino - Home." [Online]. Available: <https://www.arduino.cc/>. [Accessed: 02-Apr-2019].
- [4] "OpenCV library." [Online]. Available: <https://opencv.org/>. [Accessed: 02-Apr-2019].
- [5] "How The Internet Of Things Will Change Home Security | Articles | Internet of Things." [Online]. Available: <https://channels.theinnovationenterprise.com/articles/how-the-internet-of-things-will-change-home-security>. [Accessed: 02-Apr-2019].
- [6] H. Mulyawan, "Identifikasi Dan Tracking Objek Berbasis Image Processing Secara Real Time," *EPPIS Final Proj.*, 2011.
- [7] K. Mistry and A. Saluja, "An introduction to OpenCV using Python with Ubuntu," *Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol.*, vol. 1, no. 2, pp. 65–68, 2016.
- [8] B. Gupta, A. Chaube, A. Negi, and U. Goel, "Study on Object Detection using Open CV-Python," *Int. J. Comput. Appl.*, vol. 162, no. 8, pp. 17–21, 2017.



Gambar 3 Grafik hasil perbandingan nilai threshold.