

**IDENTIFIKASI KODE JARI TANGAN PADA SISTEM OPERASI ANDROID DENGAN METODE EUCLIDEAN DISTANCE UNTUK SISTEM KUNCI PINTU****FINGERPRINT IDENTIFICATION CODE HANDS ON ANDROID OPERATING SYSTEM USING EUCLIDEAN DISTANCE FOR DOOR LOCK SYSTEM**

Santy Fauziyah<sup>1</sup>, Unang Sunarya, ST., MT.<sup>2</sup>, Ratri Dwi Atmaja, ST., MT.<sup>3</sup>

<sup>1,3</sup>Fakultas Teknik Elektro, Telkom University

<sup>2</sup>Fakultas Ilmu Terapan, Telkom University

santy.fauziyah@gmail.com<sup>1</sup>, unangsunarya@telkomuniversity.ac.id<sup>2</sup>,

ratriidwiatmaja@telkomuniversity.ac.id<sup>3</sup>

**Abstrak**

Keamanan adalah kebutuhan vital bagi setiap manusia, baik untuk melindungi dirinya maupun materi yang dimilikinya. Banyaknya produsen yang memproduksi *smartphone* untuk berbagai macam kalangan ekonomi membuat sebagian besar lapisan masyarakat dapat mengenal dan menggunakan teknologi ini. Salah satu kelebihan sistem operasi android yaitu menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri. Dengan memanfaatkan fitur yang ditawarkan oleh sistem operasi android di atas, dalam penelitian ini direalisasikan sistem keamanan kunci pintu berbasis mikrokontroler yang diakses menggunakan citra kode jari tangan yang diambil menggunakan *smartphone* dan kemudian diolah menggunakan algoritma yang telah dirancang dengan metode klasifikasi *Euclidean distance* sehingga menghasilkan keputusan yang digunakan untuk mengeksekusi kunci pintu. Komunikasi antara *smartphone* dan *hardware* dilakukan dengan menggunakan *bluetooth*. Dari hasil penelitian diperoleh tingkat akurasi algoritma secara *non-realtime* adalah 86,11%. Tingkat akurasi sistem rata-rata secara *realtime* adalah 84%. Jarak jangkauan maksimum komunikasi *bluetooth* antara aplikasi dan *hardware* adalah 16,9 meter.

**Kata Kunci :** kunci pintu, Android, mikrokontroler, *Euclidean distance*, *bluetooth*.

**Abstract**

Security is a vital necessity for every human being, either to protect himself as well as its material. Many manufacturers are producing smartphones for various economic circles make the most of society can recognize and use these technologies. One of the advantages of android operating system that provides an open platform for developers to create their own applications. By utilizing the features offered by the Android operating system above, in this study realized the door lock security system based on microcontroller which are accessed using a finger code image taken using a smartphone and then processed using an algorithm that has been designed with the Euclidean distance classification method resulting decisions used for executing lock the door. Communication between the smartphone and the hardware is done by using bluetooth. From the results obtained by the level of accuracy in non - realtime algorithm is 86.11%. The level of

accuracy of the system in real-time average is 84 % . The maximum range of bluetooth communication between the application and the hardware is 16.9 meters .

**Keywords : door locks , Android , microcontroller , Euclidean distance , bluetooth .**

## 1. PENDAHULUAN

Kini penggunaan *smartphone* di kalangan masyarakat bukan lagi merupakan suatu gaya hidup yang mewah. Banyaknya produsen yang memproduksi *smartphone* untuk berbagai macam kalangan ekonomi membuat sebagian besar lapisan masyarakat dapat mengenal dan menggunakan teknologi ini. Dari sekian banyak *smartphone* yang beredar di pasaran, *smartphone* berbasis sistem operasi Android merupakan salah satu *smartphone* yang banyak dikembangkan.

Android adalah sistem operasi untuk telepon seluler yang berbasis Linux. Salah satu kelebihan sistem operasi Android yaitu Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri. Dengan memanfaatkan fitur yang ditawarkan oleh sistem operasi Android, banyak para pencipta aplikasi yang membuat aplikasi dengan menawarkan berbagai macam manfaat dan kemudahan bagi penggunaanya.

Dengan mengacu pada latar belakang di atas, maka dalam penelitian ini dibuat sistem keamanan kunci pintu yang dapat diakses dengan menggunakan aplikasi yang diimplementasikan pada *smartphone* berbasis sistem operasi Android. Sistem yang dibuat terdiri dari *software* dan *hardware*. *Software* berupa aplikasi yang dapat melakukan akuisisi citra yang kemudian diproses untuk mengidentifikasi kode jari tangan, dan *hardware* berupa kunci pintu yang dikendalikan oleh mikrokontroler dan diakses oleh *smartphone* melalui komunikasi *bluetooth*.

## 2. DASAR TEORI

### 2.1. Pengolahan Citra Digital

Pengolahan citra (*image processing*) merupakan proses mengolah piksel-piksel dalam citra digital untuk suatu tujuan tertentu. Beberapa alasan dilakukannya pengolahan citra pada citra digital antara lain yaitu:

- a. Untuk mendapatkan citra asli dari suatu citra yang sudah buruk karena pengaruh derau. Proses pengolahan bertujuan mendapatkan citra yang diperkirakan mendekati citra sesungguhnya.
- b. Untuk memperoleh citra dengan karakteristik tertentu dan cocok secara visual yang dibutuhkan untuk tahap yang lebih lanjut dalam proses analisis citra.

Dalam proses akuisisi, citra yang akan diolah ditransformasikan dalam suatu representasi numerik. Pada proses selanjutnya representasi numerik tersebutlah yang akan diolah secara digital oleh komputer. Pengolahan citra pada umumnya dapat dikelompokkan dalam dua jenis kegiatan, yaitu:

- a. Memperbaiki kualitas citra sesuai kebutuhan
- b. Mengolah informasi yang terdapat pada citra

Bidang aplikasi yang kedua ini sangat erat kaitannya dengan *computer aided analysis* yang umumnya bertujuan untuk mengolah suatu objek citra dengan cara mengekstraksi informasi penting yang terdapat di dalamnya. Dari informasi tersebut dapat dilakukan proses analisis dan

klasifikasi secara cepat memanfaatkan algoritma perhitungan komputer. Dari pengolahan citra diharapkan terbentuk suatu sistem yang dapat memproses citra masukan hingga citra tersebut dapat dikenali cirinya[3].

## 2.2. Metode klasifikasi Euclidean Distance

Klasifikasi adalah proses pengidentifikasian terhadap suatu pola atau objek dalam citra yang diamati, dan mengelompokkannya ke dalam anggota dari suatu kelas pola yang sudah diketahui sebelumnya (sudah ditentukan). Dalam proses klasifikasi, untuk membantu pengambilan keputusan, sudah disediakan input dari training sample yang berisi contoh-contoh pola dengan kategori/kelas objek. Setelah sistem mendapatkan training, ukuran-ukuran karakteristik/ciri objek dapat digunakan untuk membantu proses pengenalan dan pengelompokan untuk sebuah citra. Untuk pengambilan keputusannya ada beberapa cara, misalnya dengan *Gaussian Maximum-Likelihood* atau *Minimum-Distance*. Pengklasifikasian citra dengan basis pengambilan keputusan *Minimum-Distance* dilakukan dengan cara menghitung mean dari masing-masing kelas pada training sample, kemudian tiap piksel pada citra input klasifikasi dihitung jaraknya dengan menggunakan rumus *Euclidean Distance* :

$$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + \dots + (n_1 - n_2)^2} \quad (1)$$

$x_1, y_1, \dots, n_1$  adalah nilai tingkat keabuan untuk sebuah piksel dalam tiap kanal, sedangkan  $x_2, y_2, \dots, n_2$  adalah mean dari masing-masing kelas pada training sample. Piksel yang sedang dihitung akan dimasukkan pada kelas dengan nilai *Euclidean Distance* terkecil[6].

## 2.3. Sistem Operasi Android

Android adalah sistem operasi untuk telepon seluler yang berbasis Linux. Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak. Awalnya, Google Inc. membeli Android Inc., pendatang baru yang membuat peranti lunak untuk ponsel. Kemudian untuk mengembangkan Android, dibentuklah *Open Handset Alliance*, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia. Pada data perilis perdana Android, 5 November 2007, Android bersama *Open Handset Alliance* menyatakan mendukung pengembangan standar terbuka pada perangkat seluler. Di lain pihak, Google merilis kode-kode Android dibawah lisensi Apache, sebuah lisensi perangkat lunak dan standar terbuka perangkat seluler. Di dunia ini terdapat dua jenis distributor sistem operasi Android. Pertama yang mendapat dukungan penuh dari Google atau *Google Mail Service (GMS)* dan kedua adalah yang benar-benar bebas distribusinya tanpa dukungan langsung Google atau dikenal sebagai *Open Handset Distribution (OHD)*. Fitur dan spesifikasi terkini dari OS Android, antara lain adalah *framework* aplikasi, *dalvik virtual machine*, browser terintegrasi, grafik yang dioptimasi, *SQLLite*, *media support*, telepon GSM, *Bluetooth*, EDGE, 3G, WIFI, kamera, GPS, kompas, akselerometer dan lingkungan pengembangan yang sangat kaya[2][4][5].

## 2.4. Mikrokontroler dan Sistem Minimum

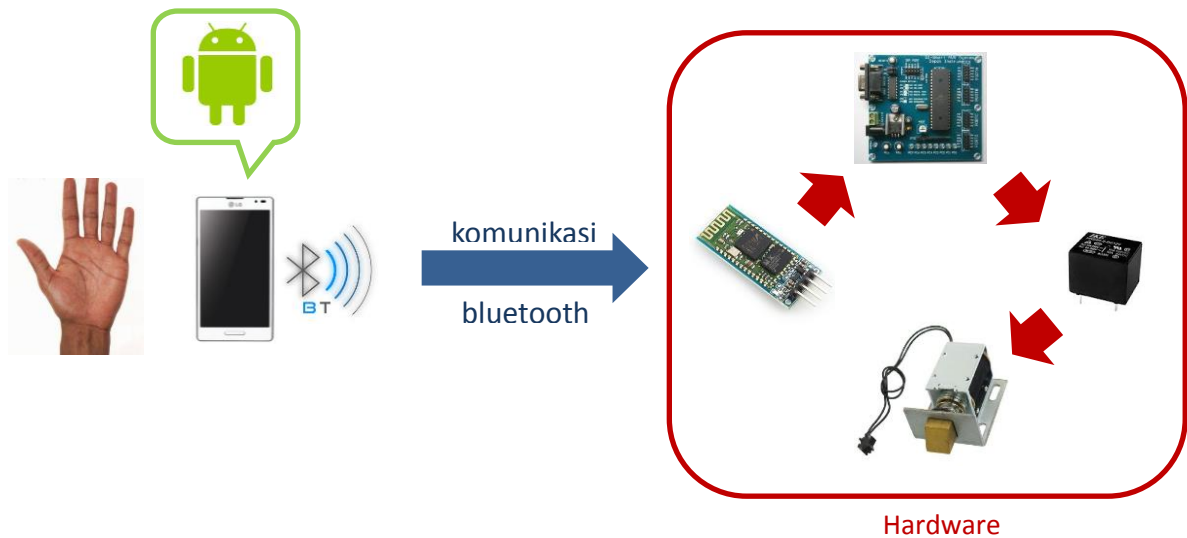
Mikrokontroler adalah sistem komputer yang dikemas dalam sebuah IC. IC tersebut mengandung semua komponen pembentu komputer seperti CPU, RAM, ROM, Port IO. Berbeda dengan PC yang dirancang untuk kegunaan umum (*general purpose*), mikrokontroler digunakan untuk tugas atau fungsi yang khusus (*special puspose*) yaitu mengontrol sistem tertentu[7].

Mikrokontroler sering juga disebut sebagai *Embedded Microcontroller* yang berarti bahwa ia merupakan bagian dari *embedded system* – menjadi satu bagian dari perangkat sistem atau sistem yang lebih besar[1][8].

### 3. PEMBAHASAN

#### 3.1. Blok Diagram Sistem

Sistem yang dibuat dalam penelitian ini secara garis besar terdiri dari *software* dan *hardware*. Blok sistem lebih jelas ditunjukkan oleh Gambar 1 berikut.

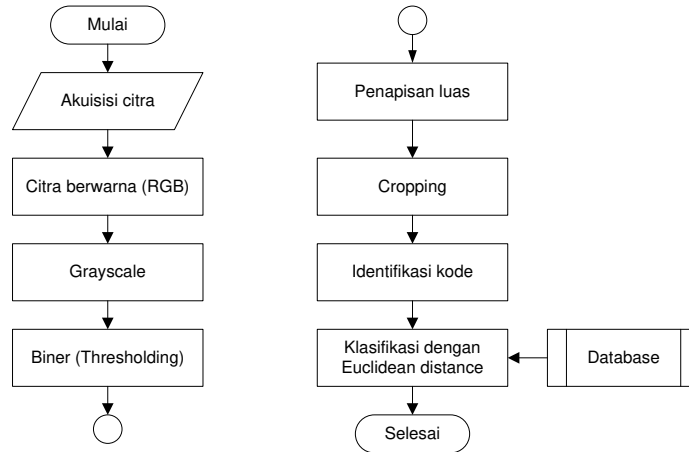


Gambar 1. Blok sistem

*Software* pada penelitian ini berupa aplikasi bernama *Smartlock* yang diimplementasikan pada *smartphone* berbasis Android. Pengambilan citra tangan dilakukan dengan kamera *smartphone* dan kemudian akan diolah (*image processing*) untuk mengidentifikasi kode tangan berdasarkan jarak antar jari sehingga menghasilkan keputusan yang akan digunakan untuk mengeksekusi *hardware*. Komunikasi antara *smartphone* dan *hardware* menggunakan perangkat *bluetooth*. Selanjutnya blok mikrokontroler akan merespon berdasarkan informasi yang diterima dari komunikasi *bluetooth*. Jika citra tangan yang diidentifikasi sesuai dengan citra tangan yang telah disimpan di database, maka kunci akan terbuka. Sebaliknya, jika tidak sesuai maka kunci tidak akan terbuka.

#### 3.2. Perancangan Algoritma

Untuk membuat aplikasi *Smartlock*, maka terlebih dahulu harus dilakukan perancangan algoritma yang akan digunakan. Algoritma harus mampu mengidentifikasi citra tangan sehingga dapat membaca informasi yang dibutuhkan, yaitu kode jari tangan. Perancangan algoritma dilakukan dengan menggunakan perangkat lunak MATLAB R2012. Diperoleh algoritma dari proses perancangan dan optimasi menggunakan 72 citra latih pada MATLAB dengan performansi 88,8889 %. Algoritma yang diperoleh dari proses perancangan kemudian diimplementasikan kedalam aplikasi android dengan bahasa pemrograman Java.



Gambar 2. Diagram Alir Algoritma

### 3.3. Parameter Pengujian Performansi Sistem

Performansi sistem dapat diketahui dengan melakukan pengujian terhadap sistem yang telah dibuat dengan berdasarkan kepada beberapa parameter yaitu sebagai berikut.

#### a. Akurasi Sistem

Akurasi merupakan ukuran ketepatan pada suatu sistem untuk mengenali objek masukan sehingga menghasilkan keluaran yang benar. Secara matematis dapat ditulis sebagai berikut:

$$Akurasi = \frac{jumlah\ data\ benar}{jumlah\ data\ uji} \times 100\% \quad (2)$$

#### b. Jarak Maksimum

Jarak maksimum merupakan parameter untuk menentukan seberapa jauh jarak jangkauan komunikasi *bluetooth* antara aplikasi dan hardware. Jarak maksimum ditentukan dengan melakukan pengujian dengan jarak sebagai variabel yang diubah. Pengujian dilakukan dengan jarak terdekat sampai jarak terjauh sehingga komunikasi *bluetooth* tidak dapat dilakukan.

### 3.4. Pengujian Algoritma

Pengujian dilakukan dengan cara non-real time yaitu menguji algoritma pada aplikasi Android menggunakan 72 citra latih yang digunakan pada proses optimasi yang sebelumnya telah dilakukan dengan menggunakan MATLAB untuk mengetahui apakah algoritma yang diimplementasikan pada aplikasi android telah sesuai dengan algoritma yang dibuat dan dioptimasi menggunakan MATLAB.. Pengujian algoritma menunjukkan bahwa diperoleh 62 citra latih yang diidentifikasi dengan tepat dari 72 citra latih yang diuji, sehingga diperoleh akurasi sebesar 86,1111 %.

### 3.5. Pengujian Sistem

#### a. Pengujian Skenario 1

Pengujian dilakukan dengan skenario yang sama dengan citra latih, namun dilakukan secara realtime. Pengujian dilakukan terhadap 4 *user*, dengan warna pintu hitam dan krem. Masing-

masing pengujian dilakukan pada saat malam (dengan penerangan lampu) dan siang hari dengan menggunakan telapak tangan dan punggung tangan. Setelah melakukan pengujian skenario 1, diperoleh 32 data yang kemudian akan dianalisa dan ditarik kesimpulan. Data hasil pengujian skenario 1 ditunjukkan oleh tabel 1 berikut:

**Tabel 1.** Data hasil pengujian skenario 1

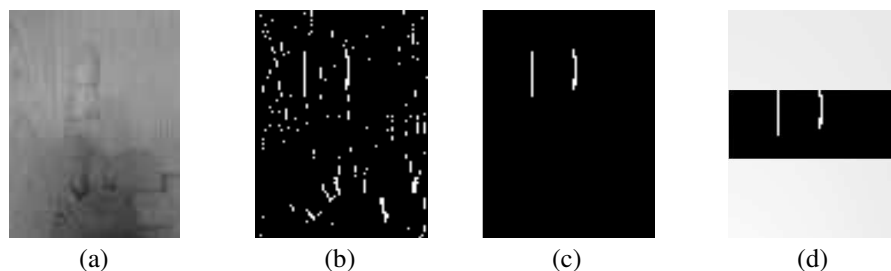
<i>User</i>		Siang		Malam	
		Pintu hitam	Pintu krem	Pintu hitam	Pintu krem
<i>User 1</i>	Telapak tangan	V	X	V	X
	Punggung tangan	V	V	V	X
<i>User 2</i>	Telapak tangan	V	X	V	X
	Punggung tangan	V	V	V	V
<i>User 3</i>	Telapak tangan	V	V	V	X
	Punggung tangan	V	X	V	X
<i>User 4</i>	Telapak tangan	V	X	V	X
	Punggung tangan	V	X	V	X

Keterangan : V = Berhasil  
X = Gagal

Pada tabel 1 data hasil pengujian diatas menunjukkan bahwa semua pengujian terhadap pintu berwarna gelap berhasil sedangkan pengujian terhadap pintu terang hanya berhasil 4 kali dari total pengujian sebanyak 16 kali. Sehingga dapat disimpulkan bahwa akurasi pengujian aplikasi secara real-time adalah sebagai berikut.

$$akurasi = \frac{32 - 12}{32} \times 100 \% = 68,75 \%$$

Seperti yang ditunjukkan tabel 1 diatas, kegagalan terjadi saat pengujian terhadap pintu berwarna terang sedangkan seluruh pengujian terhadap pintu gelap berhasil dilakukan. Hal ini dikarenakan warna pintu yang hampir sama dengan objek sehingga ketika citra RGB diubah menjadi citra grayscale, objek dan *background* memiliki perbedaan nilai derajat keabuan yang kecil. Hal ini membuat objek dan *background* sulit untuk dikelompokkan sehingga citra biner yang dibentuk tidak sempurna dan kode tangan tidak diidentifikasi secara tepat. Threshold yang digunakan untuk mengubah citra grayscale menjadi citra biner adalah 9. Sedangkan sebagian besar dari antara piksel yang bertetangga diatas memiliki selisih nilai derajat keabuan kurang dari 9. Oleh karena itu citra biner yang dihasilkan pun tidak sempurna dan merusak informasi sehingga kode tidak dapat diidentifikasi dengan tepat.



Gambar 3. Identifikasi citra uji  
(a) citra grayscale (b) citra biner (c) citra cleared (d) kode

Pada gambar 3(a) di atas menunjukkan citra hasil proses *grayscale* dari hasil akuisisi dengan kamera, pada gambar 3(b) menunjukkan hasil dari proses pengubahan citra gray ke dalam bentuk citra biner sehingga hanya memiliki dua kemungkinan warna yaitu hitam dan putih, gambar 3(c) merupakan citra biner yang telah difilter untuk menghilangkan *noise* pada citra, dan gambar 3(d) merupakan citra hasil *cropping* untuk dijadikan kode jari tangan sebagai identifikasi.

**b. Pengujian Skenario 2**

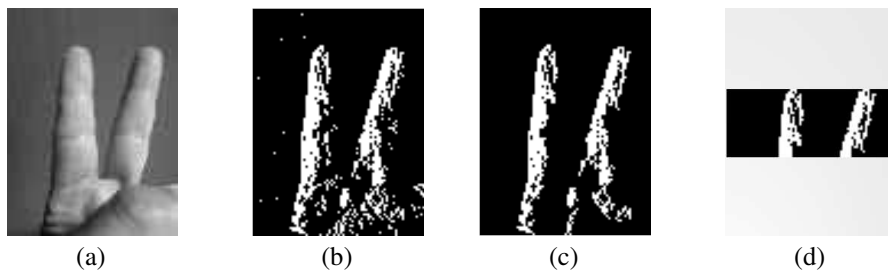
Tabel 2. Data hasil pengujian skenario 2

User		Siang		Malam	
		Pintu coklat	Pintu putih	Pintu coklat	Pintu putih
User 1	Telapak tangan	V	V	V	V
	Punggung tangan	V	V	V	V
User 2	Telapak tangan	V	V	V	V
	Punggung tangan	V	V	V	V
User 3	Telapak tangan	V	V	V	V
	Punggung tangan	V	V	V	V
User 4	Telapak tangan	V	V	V	V
	Punggung tangan	V	V	V	V

Keterangan :    V = Berhasil  
                       X = Gagal

Pengujian dilakukan dengan skenario yang sama pengujian skenario 1 namun dengan warna pintu yang berbeda. Pintu yang digunakan adalah pintu berwarna putih dan coklat. Pengujian dilakukan terhadap 4 *user*. Setiap *user* melakukan pengujian pada malam (dengan penerangan lampu) dan siang hari dengan menggunakan telapak tangan dan punggung tangan. Setelah melakukan pengujian skenario 2, diperoleh 32 data yang kemudian akan dianalisa dan ditarik kesimpulan. Data hasil pengujian skenario 2 ditunjukkan oleh tabel 2.

Berbeda dengan pengujian skenario 1, pengujian skenario 2 menunjukkan keberhasilan sebesar 100 %. Hal ini dipengaruhi oleh warna pintu yang digunakan. Pintu berwarna putih dan berwarna coklat yang digunakan dalam pengujian skenario 2 memiliki warna yang kontras dengan warna kulit orang Indonesia yang mayoritasnya berwarna kuning langsung. Sehingga ketika citra RGB diubah menjadi citra *grayscale*, objek dan *background* memiliki perbedaan nilai derajat keabuan yang cukup tinggi sehingga objek dan *background* dapat dikelompokkan dengan tetap saat proses *thresholding*. Karena selisih nilai derajat keabuan antara objek dan *background* lebih dari *threshold*, maka citra *grayscale* dapat diubah menjadi citra biner dengan sempurna sehingga kode dari citra tersebut dapat diidentifikasi secara tepat.



Gambar 4. Identifikasi citra uji  
 (a) citra *grayscale* (b) citra biner (c) citra cleared (d) kode

Pada gambar 4(a) di atas menunjukkan citra hasil proses *grayscale* dari hasil akuisisi dengan kamera, pada gambar 4(b) menunjukkan hasil dari proses pengubahan citra gray ke dalam bentuk citra biner sehingga hanya memiliki dua kemungkinan warna yaitu hitam dan putih, gambar 4(c) merupakan citra biner yang telah difilter untuk menghilangkan *noise* pada citra, dan gambar 4(d) merupakan citra hasil *cropping* untuk dijadikan kode jari tangan sebagai identifikasi

### **3.6 Pengujian Jarak Jangkauan Komunikasi *Bluetooth***

Pengujian jarak jangkauan komunikasi *bluetooth* dilakukan dengan cara mengakses hardware dengan posisi *user* yang semakin menjauh dari hardware hingga komunikasi tidak dapat dilakukan lagi. Dalam sistem ini, komunikasi *bluetooth* dibagi menjadi 2 tahap. Tahap pertama adalah *connecting*, yaitu saat aplikasi membangun koneksi dengan hardware. Tahap kedua adalah *sending message*, yaitu ketika aplikasi mengirim perintah kepada hardware. Oleh karena itu pengujian jarak jangkauan komunikasi *bluetooth* dilakukan dengan 2 skenario. Skenario 1 dilakukan untuk mengetahui jarak jangkauan maksimum proses *connecting* dan skenario 2 dilakukan untuk mengetahui jarak jangkauan maksimum proses *sending message*.

#### **3.6.1 Pengujian Skenario 1**

Pengujian untuk mengetahui jarak jangkauan maksimum proses *connecting* dilakukan dengan cara mengakses hardware dari jarak tertentu. Koneksi *bluetooth* antara aplikasi dan hardware belum terbangun. Secara perlahan *user* menjauh dari hardware. Dalam setiap perubahan jarak, *user* mencoba membangun koneksi dengan hardware. *User* terus menjauh dari hardware hingga koneksi antara aplikasi dan hardware tidak dapat dibangun.

Hasil pengujian menunjukkan bahwa jarak maksimum dimana *user* dapat membangun koneksi dengan hardware adalah 16,9 meter. Koneksi antara aplikasi dan hardware tidak dapat dibangun lagi setelah *user* melampaui jarak tersebut.

#### **3.6.2 Pengujian Skenario 2**

Pengujian untuk mengetahui jarak jangkauan maksimum proses *sending message* dilakukan dengan cara melakukan perintah kepada hardware melalui aplikasi dengan koneksi *bluetooth* yang telah dibangun sebelumnya. Secara perlahan *user* menjauh dari hardware. Dalam setiap perubahan jarak, *user* mencoba mengirim perintah kepada hardware. *User* terus menjauh dari hardware hingga hardware tidak dapat menerima perintah lagi.

Hasil pengujian menunjukkan bahwa jarak jangkauan maksimum hardware dapat menerima perintah dari aplikasi dengan koneksi *bluetooth* yang telah terbangun sebelumnya adalah 39,1 meter. Setelah *user* melampaui jarak tersebut, hardware tidak dapat menerima perintah yang dikirim oleh *user*.

## **4. Kesimpulan**

Beberapa simpulan yang dapat diperoleh dari hasil perancangan dan analisis diantaranya:

- a. Tingkat akurasi algoritma adalah 86,1111 %. Sedangkan tingkat akurasi sistem secara realtime 68,75 % untuk pintu berwarna hitam dan krem, dan 100% untuk pintu berwarna putih dan coklat.
- b. Akurasi paling tinggi diperoleh ketika aplikasi digunakan terhadap pintu yang memiliki warna kontras dengan tangan *user* sedangkan akurasi paling rendah diperoleh ketika aplikasi digunakan terhadap pintu yang memiliki warna yang hampir sama dengan tangan *user*.



- c. Penggunaan pintu yang berwarna hampir sama dengan tangan *user* membuat citra grayscale diubah menjadi citra biner yang tidak sempurna sehingga informasi rusak dan kode tidak dapat diidentifikasi dengan tepat.
- d. Jarak maksimum *user* dapat mengakses hardware dengan keadaan koneksi *bluetooth* yang belum terbangun adalah 16,9 meter dan jarak maksimum *user* dapat mengakses hardware dengan keadaan koneksi *bluetooth* yang telah terbangun sebelumnya adalah 39,1 meter.

**DAFTAR PUSTAKA**

- [1] Bejo, Agus. 2008. Rahasia Kemudahan Bahasa C dalam Mikrokontroler ATMega8535. Graha Ilmu. Yogyakarta
- [2] Dimarzio. 2008. Android Programmer's Gde. Wiley Publishing Inc. USA.
- [3] Fahmi, S.T, M.Sc., Perancangan Algoritma Pengolahan Citra Mata Menjadi Citra Polar Iris Sebagai Bentuk Antara Sistem Biometrik, Universitas Sumatera Utara, Medan, 2007.
- [4] Husni, M., Radifan, R., Perancangan dan Pembuatan Sistem Informasi Lokasi Friend Finder Berbasis GPS pada sistem Operasi Android, Institut Teknologi Sepuluh Nopember, Surabaya, 2011.
- [5] Meier, Reto. 2010. Professional Android 2 Application development. Wiley Publishing Inc. USA
- [6] Rahman, M.Z., Klasifikasi dan Visualisasi Fusi Citra Berbasis Piksel Untuk Citra Resolusi Berbeda, Universitas Indonesia, Depok, 2005.
- [7] Widyananda, A., Perancangan dan Implementasi Kunci Brankas Otomatis dengan Sidik Jari Berbasis Mikrokontroler, Telkom University, Bandung, 2013.
- [8] Winoto, Ardi. 2008. Mikrokontroler AVR Atmega8/32/16/8535 dan Pemrogramannya dengan Bahasa C pada WinAVR. Penerbit Informatika. Bandung.