



## Pengembangan Sistem Pendeteksi *Gesture* Angka pada Tangan secara *Realtime* Berbasis *Android*

Pradipta Sekar Ayu Putri W.<sup>✉</sup>, Kurniawan Teguh Martono, Ike Pertiwi Windasari

Departemen Teknik Komputer, Fakultas Teknik, Universitas Diponegoro, Indonesia

### Info Artikel

*Sejarah Artikel:*

Diterima: Mei 2020

Direvisi: Juni 2020

Disetujui: Juni 2020

*Keywords:*

*Gesture, Pengolahan Citra, RAD, Mobile Application, Android.*

### Abstrak

Komunikasi adalah hal penting dasar yang sangat penting. Berdasarkan profil kesehatan tahun 2005 penyandang tuli kongenital di Indonesia mencapai 214.100 orang. Tuli kongenital adalah gangguan pendengaran sejak lahir, yang disebabkan oleh kurangnya kemampuan impuls saraf pada telinga untuk mengubah energi mekanik getaran menjadi energi listrik. Perkembangan teknologi pengolahan citra saat ini sangat berkembang pesat. Tidak sedikit penelitian yang mengembangkan sistem pengolahan citra. Sistem pengolahan citra ini dapat membantu penyandang tuli dalam hal komunikasi. Dengan adanya aplikasi-aplikasi yang dapat menerjemahkan bahasa isyarat ke dalam teks, penyandang tuli akan dimudahkan dalam berinteraksi dengan seseorang yang tidak mengerti bahasa isyarat sekalipun. Untuk itu tujuan dari terciptanya Aplikasi "Pendeteksi *Gesture* Angka" ini adalah sebagai penelitian dasar untuk membangun aplikasi penerjemah bahasa isyarat ke dalam bentuk teks. Platform Android ini dipilih karena sudah tersebar luas dan open source sehingga akan mudah dalam proses penyebarannya.

### Abstract

Communication very important, cause communication is basic of life. Based on health profile in 2005 from Indonesian Ministry of health, the Indonesian congenital deafness reached 214.100 people. The congenital deafness is who get hearing loss since birth, because lack of nerve impulses in the ear to convert the mechanical energy of vibration into electrical energy. The Image Processing technology growing up rapidly now days. Many research interest to develop this system. Cause image processing system can help deaf communication. Using translator applications, which can translate the sign languages to text, the deafness will be facilitated to interacting with other, despite the person can't understand sign languages. The purpose of Numbering *Gesture* Recognition Application exists is to be basic for the next research to built translator sign languages to text. Android platform choosen because this platform already spread and open source, so it will be easily to spread this application.

## PENDAHULUAN

Perkembangan teknologi pengolahan citra digital kerap kali digunakan untuk membantu permasalahan image processing. Tujuan pengolahan citra digital ini untuk memperbaiki kualitas citra agar mudah diinterpretasikan oleh manusia atau mesin komputer. Hal ini dikarenakan teknologi pengolahan citra digital memungkinkan manusia untuk membuat suatu sistem yang dapat mengenali suatu citra digital baik berupa foto maupun gambar bergerak. Dalam pengolahan citra, gambar diolah sedemikian rupa sehingga gambar tersebut dapat digunakan untuk aplikasi lebih lanjut<sup>[1]</sup>. Pengolahan citra digital memiliki beberapa kelebihan, yaitu murah, cepat, dan tidak merusak sampel yang diukur dan mampu mengidentifikasi fisik produk secara objektif<sup>[2]</sup>.

Penelitian sebelumnya dengan menggunakan teknologi citra digital ini banyak yang hanya mengacu pada gambar saja, bukan gambar bergerak yang diambil secara *realtime*<sup>[6]</sup>. Pengolahan objek secara *realtime* diperlukan untuk dapat mengembangkan fungsi dari teknologi ini agar dapat digunakan untuk tujuan komunikasi dan menyampaikan pesan. Sehingga, dalam penelitian ini akan dibahas lebih lanjut mengenai pengaplikasian teknologi pengolahan citra digital dengan metodologi RAD (*Rapid application development*) yang mampu mendeteksi *gesture* angka pada tangan secara *realtime* dengan basis android.

RAD (*Rapid application development*) merupakan metodologi yang menekankan pada siklus pembangunan pendek, singkat, dan cepat. Waktu yang singkat adalah batasan yang penting untuk model ini<sup>[3]</sup>. *Gesture* tangan dijadikan sebagai objek penelitian ini dikarenakan *gesture* merupakan bentuk komunikasi non-verbal yang sangat sederhana yang mampu menyampaikan pesan-pesan tertentu. Penelitian ini dimaksudkan untuk dapat mendeteksi *gesture* tangan yang kemudian masuk ke tahap image processing dan diinterpretasikan ke dalam bentuk teks. Batasan dalam mobile apps yang dibuat dalam penelitian ini adalah input *gesture* angka yang dapat dideteksi terbatas berupa angka 1-5. Dengan adanya mobile apps ini, diharapkan nantinya dengan pengembangan lebih lanjut mampu mendeteksi bahasa isyarat (*gesture*) dan menerjemahkannya, sehingga dapat membantu komunikasi antara penyandang tuli dan orang normal. Selain itu, diharapkan penelitian ini mampu menjadi acuan maupun penambah wawasan yang dapat digunakan untuk penelitian lebih lanjut

## METODE PENELITIAN

### A. Landasan Teori

Aplikasi yang memanfaatkan sistem pengolahan citra digital banyak sekali diminati. Karena dengan adanya pengolahan citra digital ini dapat membantu dalam pengolahan citra agar dapat digitalisasi, yang kemudian data digital tersebut diproses hingga mendapat keluaran yang diinginkan.

Komunikasi adalah suatu dasar dalam kehidupan, tanpa komunikasi manusia kurang dapat menafsirkan situasi yang sedang dihadapi<sup>[4]</sup>. Pada penyandang tuli, tentu saja mengalami gangguan saat berkomunikasi dengan seseorang yang tidak mengerti bahasa isyarat. Padahal, berdasarkan profil kesehatan di tahun 2005 penyandang tuli kongenital di Indonesia sendiri diperkirakan berjumlah 214.100 orang bila jumlah penduduk sebesar 214.100.000 orang dan jumlah ini akan terus meningkat dengan tingginya angka kelahiran sebesar 0,22%<sup>[5]</sup>. Untuk itu banyak penelitian-penelitian yang bertujuan untuk membantu agar penyandang tuli dapat berkomunikasi dengan mudah.

Pada penelitian "Aplikasi Pembelajaran bahasa Isyarat Berbasis Android" bertujuan untuk membangun sebuah aplikasi yang membantu dalam mempelajari bahasa isyarat melalui sebuah video tidak *realtime* pada perangkat android<sup>[6]</sup>.

Bahasa isyarat sendiri merupakan segala sesuatu (gerakan tangan, anggukan kepala, dan sebagainya) yang dipakai sebagai tanda menurut kamus besar bahasa Indonesia. Untuk itu pada penelitian hand tracking menggunakan metode lucas kanade dan kalman filter pada virtual mouse, memiliki tujuan untuk mengikuti gerakan tangan (hand tracking) dari *gesture* yang diberikan yang kemudian *gesture* tersebut digunakan untuk mengoperasikan mouse virtual<sup>[7]</sup>.

Pada penelitian hand *gesture recognition* menggunakan metode segmentasi warna kulit dan centeroid memiliki tujuan untuk mendeteksi *gesture* tangan secara *realtime*<sup>[8]</sup>.

Aplikasi pendeteksi *gesture* angka pada tangan ini merupakan pengembangan dari aplikasi hand tracking<sup>[7]</sup> agar menjadi *realtime*<sup>[8]</sup> yang dibangun untuk perangkat android<sup>[6]</sup>. Tujuan agar aplikasi ini dapat dikembangkan lebih lanjut untuk membantu memecahkan masalah komunikasi penyandang tuli<sup>[5]</sup>.

### B. Gesture

*Gesture* merupakan perilaku non-verbal pada gerakan tangan, dan jari-jari<sup>[9]</sup>. Kombinasi dari bentuk tangan, orientasi dan gerakan tangan

untuk menyampaikan pesan dari seseorang merupakan penjelasan mengenai apa itu *gesture*.

Terdapat dua elemen komunikasi yang sangat penting yaitu komunikasi lisan (verbal) dan *gesture* (non-verbal). *Gesture* merupakan sumber informasi penting, karena dapat mengurangi ambiguitas bahas atau kesalah pahaman dalam berkomunikasi.

### C. Pengolahan Citra Digital

Pengolahan citra digital adalah sebuah disiplin ilmu yang mempelajari tentang teknik-teknik mengolah citra. Citra yang dimaksud disini adalah gambar diam (foto) maupun gambar bergerak (yang berasal dari webcam). Sedangkan digital disini mempunyai maksud bahwa pengolahan citra/gambar dilakukan secara digital menggunakan komputer<sup>[10]</sup>.

Pada pembuatan aplikasi ini terdapat beberapa tahapan yang digunakan yaitu pra pengolahan, segmentasi warna, operasi morfologi, dan ekstrasi kontur.

#### 1. Akuisisi Citra

Langkah awal dalam proses pengolahan citra digital adalah akuisisi citra. Akuisisi citra adalah sebuah tahap untuk mendapatkan citra digital<sup>[11]</sup>. Tujuan akuisisi citra yaitu untuk mendapatkan data yang diperlukan dan memilih metode pengambilan citra digital. Tahap ini dimulai dengan menentukan objek yang akan diambil gambarnya, persiapan alat-alat, dan pencitraan. Pencitraan adalah sebuah kegiatan yang mentransformasikan citra tampak (foto, gambar, lukisan, pemandangan, dan lain-lain) menjadi citra digital. Alat yang digunakan untuk mengambil citra pada aplikasi deteksi *gesture* ini adalah kamera pada mobile device.

#### 2. Pra-Pengolahan

Pra pengolahan adalah sebuah proses untuk menjamin kelancaran pada proses berikutnya. Pada aplikasi ini pra pengolahan yang digunakan yaitu pengkaburan. Pengkaburan yang digunakan yaitu metode gaussian blur. Gaussian blur ini memberikan efek blur pada citra. Gaussian blur ini digunakan untuk memperhalus citra dan menghilangkan noise. Yang membantu dalam pembentukan kontur pada citra.

##### Segmentasi Warna

Segmentasi warna merupakan proses segmentasi dengan pendekatan daerah yang bekerja dengan menganalisis nilai warna dari tiap piksel pada citra dan membagi citra tersebut sesuai dengan fitur yang diinginkan. Segmentasi citra dengan deteksi warna HSV yaitu menggunakan dasar seleksi warna pada model

warna HSV dengan nilai toleransi tertentu menurut Gunanto<sup>[12]</sup>. Pada metode segmentasi dengan deteksi warna HSV dilakukan pemilihan sampel piksel sebagai acuan warna untuk membentuk segmen yang diinginkan. Citra digital menggunakan model warna RGB sebagai standar acuan warna, oleh karena itu proses awal pada metode ini memerlukan konversi model warna RGB ke HSV. Untuk membentuk segmen sesuai dengan warna yang diinginkan maka ditentukan nilai toleransi pada setiap dimensi warna HSV, kemudian nilai toleransi tersebut digunakan dalam perhitungan proses adaptif *threshold*. Hasil dari proses *threshold* tersebut akan membentuk segmen area dengan warna sesuai toleransi yang diinginkan. Berikut ini merupakan proses segmentasi menurut Giannakopoulos<sup>[13]</sup> :

- a. Tentukan citra RGB yang menjadi objek deteksi, nilai warna HSV yang menjadi acuan (hasil proses pelatihan data) dan nilai toleransi HSV yang digunakan.
- b. Transpose citra RGB ke HSV.
- c. Lakukan filter warna pada citra berdasarkan nilai acuan (T) dan nilai toleransi (tol). Dengan x sebagai warna HSV pada piksel yang ada maka warna yang tidak termasuk dalam rentang  $T - tol < x < T + tol$  diberi warna hitam.
- d. Transpose kembali citra ke RGB, tampilkan hasil filter.

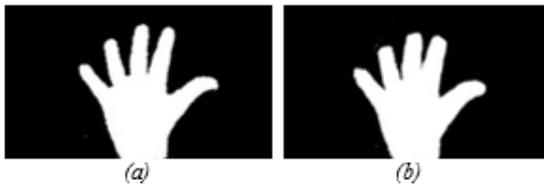
Hasil segmentasi warna tersebut menghasilkan segmen citra yang membentuk suatu blob, yaitu sekumpulan piksel bertetangga yang memiliki nilai tertentu.

#### 3. Operasi Morfologi

Operasi morfologi merupakan teknik atau proses yang digunakan untuk mengolah image atau citra yang didasarkan atas prinsip morfologi matematika<sup>[14]</sup>. Manfaat dari proses morfologi yaitu untuk mengurangi noise secara efektif. Pada aplikasi ini operasi morfologi yang digunakan yaitu erosi yang diikuti oleh dilation. Proses erosi merupakan teknik dasar pada operasi morfologi. Teknik ini digunakan untuk mengetahui struktur elemen objek, yaitu dengan cara mengurangi piksel area foreground. Sedangkan untuk proses dilation hampir sama dengan teknik erosi hanya saja pada teknik ini yaitu menambahkan piksel pada area *foreground*<sup>[15]</sup>.

Operasi morfologi yang digunakan pada pengembangan aplikasi deteksi *gesture* adalah dilasi yang dapat di jelaskan melalui persamaan (1). Dimana, D merupakan himpunan hasil dilasi, A adalah himpunan citra asal dan S merupakan himpunan dari structuring element. Tampilannya seperti pada Gambar 1.

$$D(A,S)=A\oplus S \quad (1)$$



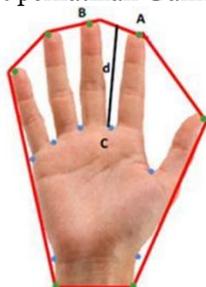
Gambar 1. (a) Citra sebelum proses dilasi, (b)Citra setelah proses dilasi

#### 4. Ekstraksi Kontur

Setelah didapatkan gambar biner yang terdiri warna putih yang bernilai 1 untuk objek, dan warna hitam yang bernilai 0 untuk *background*. Perlu adanya ekstraksi kontur agar informasi bentuk objek dapat diketahui. Pada aplikasi ini menggunakan pendekatan polygon untuk ekstraksi kontur, karena pendekatan ini dapat membantu untuk menemukan bentuk kontur yang dapat di gabungkan dengan informasi lain agar gerakan tangan tertentu dapat terbaca.

#### D. Convex Hull dan Convexity defect

*Convex Hull dan Convexity defect* merupakan algoritma yang digunakan untuk mengetahui posisi hull maupun defect pada sebuah pola<sup>[16]</sup>. Untuk memahami bentuk objek atau kontur yaitu dengan menghitung *hull convex* untuk objek kemudian menghitung *defect*. Kompleksitas bentuk juga dapat dilihat dari defect tersebut. *Convex hull* merupakan himpunan titik-titik yang membentuk sebuah poligon konveks yang melingkupi seluruh himpunan titik tersebut<sup>[16]</sup>. *Defect* sendiri dapat didefinisikan sebagai area terkecil dalam *convex hull* yang mengelilingi tangan. Pencarian defect ini berguna untuk menemukan feature pada tangan, sehingga nantinya dapat dideteksi jari tangan dan diketahui berapa jumlahnya<sup>[14]</sup>. Agar lebih jelas mengenai defect perhatikan Gambar 2.



Gambar 2. Convex hull dan convexity defect

Proses simpul poligon yang dihasilkan oleh *Convex Hull* menggunakan algoritma Graham's Scan<sup>[17]</sup>, yang ditentukan dengan menghitung tiga titik silang  $P1=(x1,y1)$ ,  $P2=(x2,y2)$ , dan  $P3=(x3,y3)$ .

Berikut adalah algoritma Graham's Scan<sup>[17]</sup> :

- a. Inisialisasi list hull.
- b. Atur counter = 2.
- c. Simpan array p[0] dan p[1] ke dalam list hull.
- d. Hitung arah titik p[0], p[1], dan p[2] dengan persamaan:

$$CH = (x_2 - x_1)(y_3 - y_1) - (y_2 - y_1)(x_3 - x_1) \quad (2)$$

- e. Jika  $CH > 0$ , maka p[1] diterima dan p[2] dimasukkan ke dalam list hull. Tambahkan counter +1.
- f. Jika  $CH = 0$ , maka pada garis lurus tambahkan counter +1.
- g. Jika tidak memenuhi kriteria sebelumnya, hapus titik terakhir pada list hull.

Dari sejumlah set titik Hull yang ditemukan dilakukan pengecekan titik kedalaman (*depth point*) dan panjang kedalaman (*depth*). Jika memenuhi kriteria, maka start point pada *convexity defect* diidentifikasi sebagai jari tangan. Antara titik A dan titik B, titik C adalah jarak maksimum dari garis yang menghubungkan titik A dan B. Jarak antara C(x0,y0) dengan garis A(x1,y1) dan B(x2,y2) dihitung menggunakan persamaan(3). Lihat juga pada gambar 2.

$$d = \frac{|(x_2-x_1)(y_2-y_1)-(x_1-x_0)(y_2-y_1)|}{\sqrt{(x_2-x_1)^2-(y_2-y_1)^2}} \quad (3)$$

#### E. Threshold

Dalam bahasa Indonesia, *threshold* memiliki arti ambang batas. Nilai pixel image yang mempunyai rentang interval antara [0,n] yang dicari distribusi tiap gray levelnya. Dari distribusi tersebut akan membentuk suatu histogram yang nantinya akan digunakan untuk mencari nilai dari *threshold*<sup>[18]</sup>.

Nilai ambang batas pada aplikasi pendeteksi *gesture* ini dapat diatur secara manual oleh pengguna yang terletak pada seekbar. Setelah melalui trial and error, nilai ambang batas yang paling ideal adalah antara 0-30000. Dimana ambang batas ini digunakan juga pada aplikasi "Permainan Kertas Gunting Batu" yaitu antara 0-30000. Pada aplikasi permainan kertas gunting batu ini juga terdapat deteksi tangan untuk mengetahui bentuk *gesture* tangan tersebut, apakah membentuk kertas atau gunting atau batu<sup>[19]</sup>. Nilai-nilai tersebut diproses untuk mendapatkan suatu *threshold* dari suatu image. Dimana tujuan dari threshold ini yaitu untuk membagi suatu batas antara *background* dengan objeknya.

#### F. Realtime

*Realtime System* memiliki karakteristik yang unik dimana sistem harus dapat menanggapi suatu peristiwa dalam jangka waktu

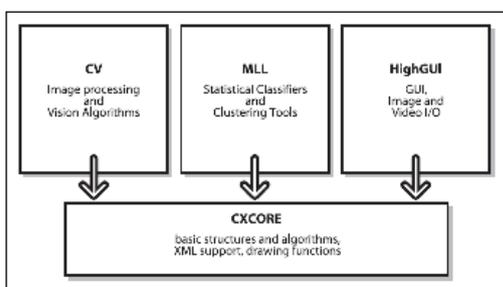
yang singkat. *Realtime System* akan merespon saat stimulasi diberikan. Stimulasi tersebut dapat berupa stimulasi yang periodik maupun stimulasi yang menggunakan mekanisme interupsi.

Untuk aplikasi pendeteksi *gesture* angka pada tangan ini menggunakan stimulasi periodik karena harus berlangsung terus menerus, saat sistem dijalankan.

#### G. OpenCV

OpenCV (*Open Source Computer Vision*) adalah library untuk fungsi pemrograman *realtime* visi komputer<sup>[20]</sup>. OpenCV dapat digunakan pada bahasa pemrograman C, C++, Python, Java dan sebagainya. Library ini juga dapat digunakan pada sistem operasi Windows, Linux, Android, iOS dan Mac OS. Pada OpenCV ini juga memiliki lebih dari 2500 algoritma yang telah dioptimalkan<sup>[21]</sup>. Sudah banyak aplikasi berbasis windows atau linux yang menggunakan Open CV untuk mengembangkan aplikasi pembaca *gesture* namun dalam versi *mobile* masih jarang. Versi OpenCV 3.1.0 untuk sistem operasi android yang digunakan dalam pengembangan aplikasi ini.

Fitur-fitur yang terdapat pada OpenCV antara lain seperti memanipulasi data citra, manipulasi matriks, pemrosesan citra fundamental, kalibrasi kamera, dan juga analisis gerakan<sup>[22]</sup> :



Gambar 3. Struktur OpenCV<sup>[23]</sup>

Struktur dan konten OpenCV<sup>[23]</sup> :

CV : untuk algoritma image processing dan vision.

ML : untuk machine learning library

Highgui : untuk GUI, image dan video I/O.

CXCORE : untuk struktur data, support XML dan fungsi-fungsi grafis.

#### H. Bahasa Pemrograman Java

Java adalah bahasa pemrograman berorientasi objek murni yang dibuat berdasarkan kemampuan-kemampuan terbaik bahasa pemrograman objek sebelumnya (C++, Ada, Simula)<sup>[24]</sup>. Karena Java merupakan bahasa berorientasi objek maka, bahasa pemrograman

Java ini dapat memudahkan pembuatan aplikasi berskala besar. Sebagai bahasa yang beraras tinggi, dan perintah-perintah yang mudah dimengerti oleh manusia, Java memiliki keunggulan yakni bersifat universal<sup>[25]</sup>.

#### I. Sistem Operasi Android

Android digunakan sekarang ini dan sangat terkenal di pasaran smartphone. Android adalah sebuah sistem operasi perangkat mobile berbasis linux yang mencakup sistem operasi, middleware dan aplikasi. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi<sup>[26]</sup>.

Untuk mengembangkan aplikasi berbasis Android diperlukan beberapa environment yang mendukung pengembangan aplikasi android. Environment yang diperlukan yaitu antara lain Java Development Kit (JDK), Android SDKs, dan Android Studio IDE<sup>[27]</sup>.

Secara garis besar arsitektur android berupa application and widgets, application framework, libraries, android run time, dan linux kernel<sup>[28]</sup>. Agar lebih optimal menjalankan aplikasi ini, perangkat android setidaknya memiliki versi 8.0.0.

## HASIL DAN PEMBAHASAN

#### A. Implementasi Aplikasi

Setelah menekan tombol “AYO MULAI” pada halaman intro, selanjutnya akan dialihkan langsung menuju kehalaman pendeteksi atau halaman inti dari aplikasi deteksi *gesture* ini. Dimana *gesture* yang dapat terdeteksi yaitu *gesture* angka 1-5. Untuk tampilan halaman intro pada aplikasi pendeteksi *gesture* ini ditunjukkan oleh Gambar 4.



Gambar 4. Halaman intro aplikasi

Antarmuka halaman pendeteksi akan ditunjukkan pada Gambar 5.



Gambar 5. Antarmuka awal deteksi

Saat berpindah dari halaman intro ke halaman pendeteksi user akan melihat angka 0 pada pojok kanan atas, preview dari kamera belakang perangkat yang digunakan, dan juga

seekbar untuk mengatur *threshold* secara manual. Selanjutnya user harus meletakkan tangannya sampai terlihat pada preview kemudian sentuh layar di area yang terdapat tangan dari user. Fungsi dari menyentuh bagian ini adalah agar sistem mengambil sampel warna dari kulit, untuk selanjutnya diproses. Berikut merupakan tampilan setelah layar disentuh dan user memberikan *gesture* angka 1 pada aplikasi ditunjukkan oleh Gambar 6.



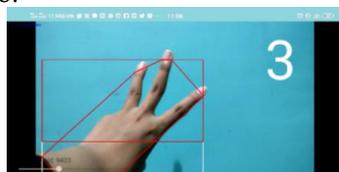
Gambar 6. Tampilan *gesture* angka satu

Berikut adalah tampilan halaman bila diberikan *gesture* angka 2 ditunjukkan oleh gambar 7.



Gambar 7. Tampilan *gesture* angka dua

Berikut adalah tampilan halaman bila diberikan *gesture* angka tiga ditunjukkan oleh Gambar 8.



Gambar 8. Tampilan *gesture* angka tiga

Berikut adalah tampilan halaman bila diberikan *gesture* angka empat ditunjukkan oleh Gambar 9.



Gambar 9. Tampilan *gesture* angka empat

Berikut adalah tampilan halaman bila diberikan *gesture* angka lima ditunjukkan oleh Gambar 10.



Gambar 10. Tampilan *gesture* angka lima

## B. Pengujian Aplikasi

### 1. Pengujian *Black Box*

Tahapan pengujian ini berisi rangkaian pengujian tombol dan fungsi yang terdapat dalam aplikasi. Pengujian ini dilakukan untuk memeriksa fungsi yang terdapat dalam aplikasi, apakah hasilnya telah sesuai dengan yang direncanakan. Pengujian ini juga dilakukan untuk mengetahui apakah masih terdapat kesalahan dalam aplikasi sehingga secepatnya dapat diberi solusi. Pengujian dijalankan pada perangkat seluler Asus Zenfone L1. Berikut tabel-tabel hasil pengujian menggunakan *black-box* yang telah dilakukan.

#### Pengujian Halaman Deteksi

Berikut ini merupakan tabel pengujian pada halaman deteksi yang ditunjukkan pada Tabel 1.

Tabel 1. Pengujian *Black Box* Halaman Deteksi

Nama Pengujian	Bentuk Pengujian	Hasil yang Diharapkan	Hasil Pengujian
Pengujian kamera <i>preview</i>	Menekan tombol "AYO MULAI"	Kamera <i>preview</i> menampilkan gambar yang ditangkap oleh kamera belakang perangkat Tampilan kamera <i>preview</i> akan muncul	Berhasil
Pengujian pengambilan sampel warna	Menyentuh layar	garis merah dan putih yang menunjukkan area yang warnanya dijadikan sampel	Berhasil
Pengujian <i>seekbar</i>	Menggeser <i>point seekbar</i> kekanan atau kekirri	Point <i>seekbar</i> bergeser dan <i>threshold</i> berubah	Berhasil
Pengujian deteksi <i>gesture</i> angka satu	Memberikan <i>gesture</i> angka satu	Teks angka berubah menjadi angka "1"	Berhasil
Pengujian deteksi <i>gesture</i> angka dua	Memberikan <i>gesture</i> angka dua	Teks angka berubah menjadi angka "2"	Berhasil
Pengujian deteksi <i>gesture</i> angka tiga	Memberikan <i>gesture</i> angka satu	Teks angka berubah menjadi angka "3"	Berhasil
Pengujian deteksi <i>gesture</i> angka empat	Memberikan <i>gesture</i> angka satu	Teks angka berubah menjadi angka "4"	Berhasil
Pengujian deteksi <i>gesture</i> angka lima	Memberikan <i>gesture</i> angka satu	Teks angka berubah menjadi angka "5"	Berhasil

## 2. Pengujian Peforma Aplikasi

Pada aplikasi “Pendeteksi *Gesture* Angka” dilakukan pengujian performa aplikasi, yaitu dengan mengitung waktu rata-rata yang dibutuhkan sistem untuk mendeteksi *gesture*. Pengujian ini bertujuan untuk mengetahui performa aplikasi bila berada pada ruangan yang cukup cahaya atau kurang cahaya, selain itu juga menguji dampak pengaturan ambang batas (*threshold*). Untuk ukuran kurang cahaya pada pengujian ini dilakukan di dalam ruangan, dengan satu lampu menyala, sedangkan kondisi cukup cahaya dilakukan didalam ruangan dengan tambahan dua lampu yng diletakkan di dekat perangkat. Pengujian aplikasi ini menggunakan perangkat android Asus Zenfone L1 dengan spesifikasi RAM 3 GB dan versi android 8.0.0. Berikut ini adalah tabel ringkasan dari pengujian performa aplikasi pada ruangan kurang cahaya yang ditunjukkan oleh Tabel 2.

Tabel 2. Ringkasan Pengujian pada Kondisi Kurang Cahaya

Threshold	Gesture	Waktu Deteksi Rata-Rata (ms)
0 - 10000	1	15725
	2	14225
	3	16308,33
	4	8958,33
	5	11533,33
10000 - 20000	1	16083,33
	2	13858,33
	3	12666,67
	4	11308,33
	5	13325
20000 - 30000	1	4658,33
	2	1633,33
	3	9600
	4	5600
	5	8391,67

Berikut adalah tabel ringkasan dari pengujian performa aplikasi pada cahaya yang cukup, ditunjukkan oleh Tabel 3.

Nilai *threshold* yang digunakan adalah 5000 untuk mewakili nilai antara 0 sampai 10000, 15000 untuk mewakili nilai antara 10000 sampai 20000, dan 25000 untuk mewakili nilai 20000 sampai 30000. Semakin tinggi nilai *threshold* maka semakinjauh pula jarak perbedaan warna. Tentu saja berpengaruh pada proses deteksi. Terlihat semakin tinggi nilai *threshold* berpengaruh pada semakin sedikitnya waktu yang dibutuhkan pada proses deteksi. Namun pada cahaya yang kurang, waktu deteksi yang dibutuhkan masih tergolong tinggi jika

dibandingkan pada saat cahaya ruangan cukup terang.

Tabel 3. Ringkasan Pengujian pada Kondisi Cukup Cahaya

Threshold	Gesture	Waktu Deteksi rata-rata (ms)
0-10000	1	1658,33
	2	541,67
	3	1558,33
	4	2825
	5	3541,67
10000 - 20000	1	708,33
	2	383,33
	3	191,67
	4	1575
	5	1016,67
20000 - 30000	1	391,67
	2	941,67
	3	458,33
	4	2625
	5	1683,33

Berdasarkan hasil pengujian performa aplikasi dapat disimpulkan apabila akurasi pada pendeteksian *gesture* dipengaruhi oleh intensitas cahaya dan juga pengaturan *threshold* yang sesuai.

## C. Pembahasan Aplikasi

Hasil pengujian aplikasi menggunakan metode black-box, menunjukkan bahwa Aplikasi “Pendeteksi *Gesture* Angka” secara garis besar sudah berjalan sesuai dengan spesifikasi kebutuhan dan skenario aplikasi. Hal ini ditunjukkan oleh tabel 2 sampai dengan tabel 3 pada tiap fungsi aplikasi dan tombol. Secara fungsional aplikasi ini sudah dapat menghasilkan keluaran sesuai dengan yang diharapkan.

Selain itu akurasi aplikasi “Pendeteksi *Gesture* Angka” dipengaruhi oleh intensitas cahaya pada proses pendeteksian, dan juga pengaturan *threshold*. Hal ini dibuktikan dengan pengujian performasi, pada tabel 4 dan tabel 5.

Seperti dalam batasan masalah, bahwa aplikasi “Pendeteksi *Gesture* Angka” ini akan dijalankan pada sistem operasi Android. Setelah dilakukan pengujian, skenario aplikasi ini berjalan dengan baik dan lancar pada perangkat seluler Asus Zenfone L1 dengan versi Android 8.0.0. Perangkat seluler tersebut telah berhasil dalam distribusi dan implementasi aplikasi secara keseluruhan, fungsi tombol berfungsi dengan baik dan deteksi *gesture* angka dapat dilakukan.

Pembuatan aplikasi “Pendeteksi *Gesture* Angka” ini hanya dimaksudkan untuk penelitian dasar mengenai *gesture recognition* berbasis Android, bukan membahas dampak yang terjadi setelah menggunakan aplikasi ini. Pada aplikasi

ini merupakan pengembangan dari penelitian terdahulu, dimana penelitian tersebut belum mampu untuk mendeteksi *gesture* secara *realtime*.

## SIMPULAN

Dari hasil pengujian dan analisis aplikasi pendeteksi *gesture* angka pada tangan secara *realtime* berbasis Android, dapat disimpulkan bahwa aplikasi ini dibangun untuk mejadi penelitian awal pengembangan aplikasi pendeteksi bahasa isyarat yang membantu penyandang tuli dalam berkomunikasi.

Aplikasi ini berhasil dibangun pada perangkat *mobile* dengan sistem operasi android yang telah terinstal OpenCV Manager.

Pendeteksian *gesture* angka ini berdasarkan jumlah defect yang terdeteksi, sehingga jumlah jari sama dengan jumlah point defect bila point defect yang terdeteksi lebih dari 5 maka teks angka yang muncul tetaplah angka 5.

Berdasarkan hasil pengujian menggunakan metode *Black-box*, seluruh fungsi yang ada pada aplikasi telah berhasil sesuai dengan fungsinya. Untuk akurasi hasil deteksi dipengaruhi oleh intensitas cahaya (optimal pada siang hari diluar ruangan) dan pengaturan *threshold* (ambang batas) hasil dari pengujian performa aplikasi.

## UCAPAN TERIMAKASIH

Puji syukur kehadiran Allah Subhanallahu wa Ta'ala yang telah melimpahkan rahmat dan anugerah-Nya sehingga penulis dapat menyelesaikan penelitian yang berjudul "Pengembangan Sistem Pendeteksi *Gesture* Angka pada Tangan secara *Realtime* Berbasis Android".

Dalam penyusunan laporan Tugas Akhir ini penulis banyak mendapatkan bimbingan, dorongan dan bantuan dari berbagai pihak.

Penulis menyadari bahwa masih banyak kekurangan di dalam penyusunan laporan ini, maka penulis mengharapkan kritik dan saran yang membangun untuk menyempurnakan laporan ini. Semoga laporan ini dapat memberikan manfaat bagi penulis maupun bagi rekan-rekan mahasiswa.

## DAFTAR PUSTAKA

- [1] Fitri Muwardi, dan Abdul Fadlil, "Sistem Pengenalan Bunga Berbasis Pengolahan Citra dan Pengklasifikasi Jarak", Jurnal Ilmu Teknik Elektro Komputer dan Informatika (JITEKI) Vol. 3, No. 2, pp. 124-131, 2017.
- [2] Mas'ud Effendi, Fitriyah, dan Usman Effendi, "Identifikasi Jenis dan Mutu Teh Menggunakan Pengolahan Citra Digital dengan Metode Jaringan Syaraf Tiruan", Jurnal Teknotan Vol. 11 No. 2, pp. 67-76, 2017.
- [3] Edwina Siska Pusparini, Meicsy E.I. Najoan, dan Xaverius B.N. Najoan, "Sistem Informasi Akademik Berbasis Mobile Web Menggunakan Pendekatan Metodologi RAD", Jurnal Teknik Elektro dan Komputer vol.6 no.4, pp. 182 -193, 2017.
- [4] S. Ramadhanty, "Penggunaan Komunikasi Fatis dalam Pengelolaan Hubungan di Tempat Kerja," J. Ilmu Komun., vol. 5, no. 1, pp. 1-12, 2015.
- [5] M. Rahardjo, "Bahasa Sebagai Alat Komunikasi Publik Dan Pembangunan Wacana," Ling. J. Ilmu Bhs. dan Sastra, vol. 2, no. 1, pp. 59-69, 2011.
- [6] J. K. Hikmalansya dan D. Cahyono, "Aplikasi Pembelajaran Bahasa Isyarat Berbasis Android," INFORM, vol. 1, no. 2, pp. 118-124, 2016.
- [7] N. H. M. Konstyono et al., "Hand Tracking Menggunakan Metode Lucas Kanade Dan Kalman Filter Pada Virtual Mouse," 2014.
- [8] W. Kurniawan and A. Harjoko, "Hand *Gesture* Recognition Menggunakan Metode Segmentasi Warna Kulit Dan Center Of Gravity Untuk Translasi Bahasa Isyarat Secara Real-Time," 2012.
- [9] Hendra Purnama, "Seni Bicara dan Bahasa Tubuh", Yogyakarta: Mantra Books, pp. 48, 2014.
- [10] R. D. Kusumanto and A. N. Tompunu, "Pengolahan Citra Digital Untuk Mendeteksi Objek Menggunakan Pengolahan Warna Model Normalisasi RGB," vol. 2011, no. Semantik, 2011.
- [11] Sutoyo, T., E. Mulyanto, V. Suhartono, O.D. Nurhayati, dan Wijanarto, Teori Pengolahan Citra Digital, Penerbit Andi, Yogyakarta, 2009.
- [12] Gunanto, S.G. "Segmentasi Warna Bagian Tubuh Manusia pada Citra 2D". Proceeding Sentia. 2009.
- [13] Giannakopoulos, T. "Matlab Color Detection Software". Greece : Departementof Informatics and Telecommunication University of Athens. 2008.
- [14] R. Hartanto dan M. N. Aji, "Perancangan Awal Antarmuka *Gesture* Tangan Berbasis Visual," vol. 1, no. 1, pp. 36-43, 2012.
- [15] V. M. Tiwari, "University of Manchester Final year project Android Application for Sign Language *Recognition*," 2017.

- [16] Youssef, Menatoallah M., “Hull *Convexity defect* Features For Human Action *Recognition*”, Dissertation, The School of Engineering of the University of Dayton, 2011.
- [17] N. Yuliana, K. Ratri, and R. Wardani, “Metode Convex Hull dan *Convexity defects* untuk Pengenalan Isyarat Tangan,” vol. 11, no. 2, pp. 81–88, 1858.
- [18] M. H. Fauzi et al., “Implementasi *Thresholding* Citra Menggunakan Algoritma Hybrid Optimal Estimation,” 2010.
- [19] Rlgur41, “AndroidVision\_rock\_scissor\_paper”. 2019. [Online]. Available : <https://github.com>
- [20] Corporation, “OpenCV,” Itseez, 2000. [Online]. Available: <http://opencv.org>.
- [21] Lazaro, J. L. Buliali, and B. Amaliah, “Deteksi Jenis Kendaraan di Jalan Menggunakan OpenCV,” vol. 6, no. 2, 2017.
- [22] M. D. Tobi, P. Katolik, and S. Paul, “Rancang Bangun Purwarupa Sistem Pendeteksi Kendaraan,” no. May, 2018.
- [23] B. Susianto, “Deteksi Fitur Gawang Pada Robot Berbasis Hough Transform,” Fak. Teknol. dan Inform. Inst. Bisnis dan Inform. STIKOM Surabaya, 2018.
- [24] S. Noviyanto, “Pemrog. Berorientasi Objek,” Pengenalan Bhs. Pemrograman Java Java, no. Java VM, pp. 1–6, 2016.
- [25] Kadir, “Buku Pertama Belajar Pemrograman Java untuk Pemula,” no. August, 2014.
- [26] Juansyah, “Pembangunan Aplikasi Child Tracker Berbasis Assisted – Global Positioning System ( A-GPS) dengan Platform Android,” J. Ilm. Komput. dan Inform., vol. 1, no. 1, pp. 1–8, 2015.
- [27] N. Smyth, Android Studio Development Essentials, 2.0. eBookFrenzy, 2015
- [28] N. Fauzy, “Kamus Obat Berbasis Android,” Fak. Tek. UMP, 2014.