

Implementasi *Connected Component Labeling* untuk Deteksi Objek Penghalang Bagi Penyandang Tunanetra Berbasis Raspberry Pi

Ida Yusnilawati¹, Fitri Utamingrum², Mochammad Hannats Hanafi Ichsan³

Program Studi Teknik Komputer, Fakultas Ilmu Komputer, Universitas Brawijaya
Email: ¹idayusnilawati@gmail.com, ²f3_ningrum@ub.ac.id, ³hanas.hanafi@ub.ac.id

Abstrak

Tunanetra adalah suatu keadaan dimana kedua indera pengelihatannya tidak berfungsi untuk menerima informasi layaknya orang awas, untuk itu membutuhkan alat bantu seperti tongkat untuk menjalankan aktifitas sehari-hari. Namun tongkat juga masih memiliki kekurangan yaitu hanya dapat digunakan untuk meraba objek atau halangan dengan jangkauan yang terbatas. Salah satu teknologi yang memungkinkan tunanetra dalam menjalankan aktifitas sehari-hari yaitu memanfaatkan *computer vision* untuk pengolahan citra digital yang dapat mendeteksi sebuah objek penghalang saat Tunanetra berjalan di dalam ruangan. Sistem ini menggunakan kamera webcam sebagai sensor yang dipasangkan di depan dada pengguna pada ketinggian 110cm dan kemiringan kamera sebesar 41° sehingga dapat mengambil citra di depan pengguna sampai dengan 125cm. Proses deteksi objek penghalang ini dilakukan dalam beberapa tahap yaitu *resize* citra, *cropping*, kemudian *thresholding*. Proses *thresholding* ini memanfaatkan nilai dari citra RGB lantai. Untuk mendapatkan suatu blob pada citra menggunakan *connected component labeling* 4 *connectify* yang digunakan untuk melabeli piksel. Piksel yang telah terlabeli tersebut akan dianalisis untuk dapat mendeteksi objek penghalang. Dari percobaan yang telah dilakukan sistem mampu mendeteksi objek penghalang dengan akurasi sebesar 91,66%. Hasil pengujian akurasi integrasi sistem dengan *hardware* yaitu sebesar 98,33% dan rata-rata waktu komputasi sistem yaitu sebesar 166,15 ms

Kata kunci: *tunanetra, computer vision, thresholding, connected component labeling*

Abstract

Blind is a condition where the both senses of sight do not work to receive information like the alert person, that's why it needs auxiliary tool like stick to carry out daily activities. However, the stick still has a deficiency too that can only be used to touch objects or obstacles with a limited range. One of technologies that enables blind people in carrying out daily activities is to use computer vision for processing of digital image that can detect a barrier object when a blind person walks in the room. This system uses a webcam camera as a sensor attached in front of the user's chest at a height of 110cm and a camera tilt of 41°, so that it can take the image in front of the user up to 125cm. The detection process of this barrier object is done in several steps, such as resizing the image, cropping, then thresholding. This thresholding process utilizes values from the RGB image of floor. To get a blob in the image uses connected component labeling 4 connectify used to label pixels. Pixels that have been labeled will be analyzed to be able to detect barrier object. From the study that has been done by the system, it can detect barrier object with accuracy of 91,66%. The result of study for accuracy of system integration with hardware is 98.33%, and the average time of system computing is 166.15 ms.

Keywords: *blind, computer vision, thresholding, connected component labeling*

1. PENDAHULUAN

Tunanetra adalah suatu keadaan, dimana seseorang kehilangan pengelihatannya karena kedua indera pengelihatannya tidak berfungsi untuk menerima informasi dalam kegiatan sehari-hari

seperti orang awas (Harimukhti & Dewi, 2014). Pada tahun 2017, *World Health Organization* (WHO) menuliskan sebanyak 81% gangguan pengelihatannya terjadi pada orang-orang berusia diatas 50 tahun keatas. Terdapat 253 juta orang di seluruh dunia mengalami tuna netra. 36 juta mengalami kebutaan dan 217 juta memiliki

gangguan pengelihatian sedang hingga berat. (Silvia P, 2017)

Dalam kehidupan sehari-hari penyandang tunanetra mengalami kesulitan dalam beraktifitas, salah satunya yaitu berjalan. Karena memiliki keterbatasan dalam pengeliatian kebanyakan tunanetra menggunakan tongkat sebagai pengganti indera pengelihatian dan menggunakan indera pendengaran sebagai respon dari tongkat tersebut (Al Kadafi & Utaminungrum, 2017). Namun tongkat masih memiliki kekurangan yaitu hanya dapat digunakan untuk meraba objek atau halangan dengan jangkauan yang terbatas. Untuk mengatasi masalah tersebut, berbagai kalangan dan berbagai bidang mencoba mengajukan beberapa solusi seperti penerapan teknologi sebagai pengganti indera pengelihatian.

Salah satu teknologi yang dapat dilakukan untuk mengatasi masalah tersebut adalah memanfaatkan *computer vision*. Menurut Wahyudi, *Computer Vision* merupakan cabang ilmu Pengolahan Citra Digital yang memungkinkan komputer dapat melihat seperti manusia, sehingga dapat mengambil keputusan, melakukan aksi, dan mengenali suatu Objek (Wahyudi & Kartowisastro, 2011).

Oleh karena itu, penulis mencoba membangun sebuah sistem yang memanfaatkan *computer vision* untuk membantu penyandang tunanetra dalam melakukan aktifitas berjalan. Pada penelitian ini, penerapan *computer vision* untuk mengidentifikasi objek berupa halangan dengan memanfaatkan keadaan lantai. Halangan yang dimaksud adalah objek seperti orang, *box*, meja atau benda lain yang berada di depan penyandang tunanetra ketika berjalan yang dapat dikategorikan sebagai halangan. Secara garis besar prinsip kerja sistem ini dapat mendeteksi objek yang dianggap sebagai penghalang menggunakan sensor kamera yang dipasang depan dada pengguna setinggi 110 cm dengan kemiringan kamera sebesar 41° sehingga dapat mengambil citra di depan pengguna samapi dengan 125 cm. Tahap awal yaitu menghitung nilai *min-max*, *median*, *mean* citra RGB pada lantai. Kemudian hasil citra RGB pada lantai akan digunakan sebagai batas *threshold*. Proses *thresholding* pada penelitian ini digunakan untuk membedakan *background* dan *foreground*. selanjutnya yaitu mendeteksi objek menggunakan algoritma *Connected Component Labelling*.

Algoritma *Connected Component Labelling* pada penelitian ini adalah untuk deteksi blob.

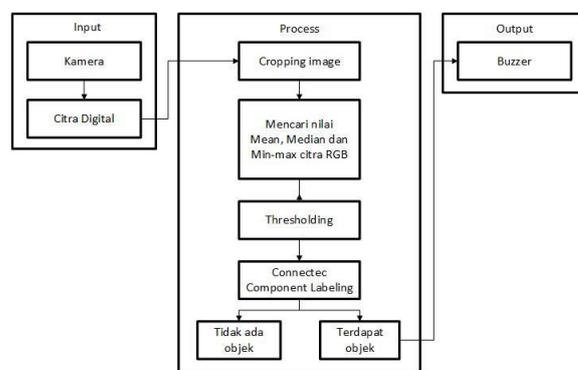
Connected Component Labeling merupakan algoritme dasar dalam pengolahan citra digital yang secara umum digunakan dalam proses yang berhubungan dengan deteksi objek (Schwenk, 2015). Penggunaan metode *Connected Component Labeling* sendiri sudah sering dipakai untuk deteksi blob/objek. Pada penelitian ini blob yang telah terdeteksi selanjutnya dianalisis untuk mendeteksi suatu halangan yang terdapat pada citra. Selanjutnya sistem akan mengeluarkan output berupa suara dari buzzer yang akan digunakan bagi penyandang tunanetra sebagai pertanda keadaan jalan yang ada di depannya.

2. PERANCANGAN DAN IMPLEMENTASI

2.1 Gambaran Umum Sistem

Secara umum sistem ini terbagi menjadi tiga bagian yaitu *input*, proses dan *output*. Pada Gambar 1 yaitu *input* pada sistem ini yaitu kamera webcam yang digunakan untuk mengambil citra digital. Kamera akan menangkap citra digital *video* dan mengirimkannya ke Raspberry Pi untuk dilakukan pemrosesan. Langkah pertama, video akan dipotong (*cropping*) pada bagian tertentu sesuai dengan ukuran sebesar 150 x 150 piksel. Selanjutnya, citra digital yang sudah melalui proses *cropping* kemudian dihitung nilai *mean*, *median* dan *min-max*nya untuk dijadikan sebagai batas pada proses *thresholding*.

Pada proses *thresholding*, terjadi pemisahan *background* dan *foreground* yang kemudian dilanjutkan dengan deteksi objek menggunakan *connected component labeling*. Sistem akan memberikan output berupa suara melalui *buzzer* ketika sistem berhasil mendeteksi objek yang ada di depan pengguna



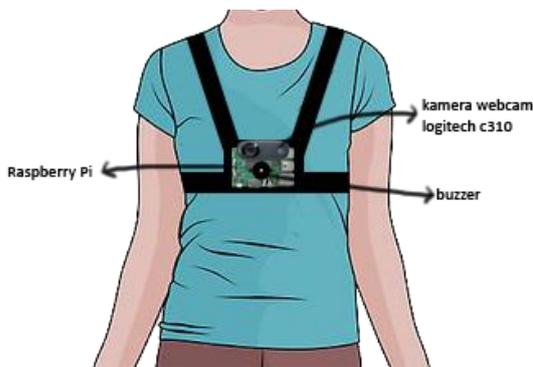
Gambar 1. Blok Diagram Sistem

2.2 Perancangan Sistem

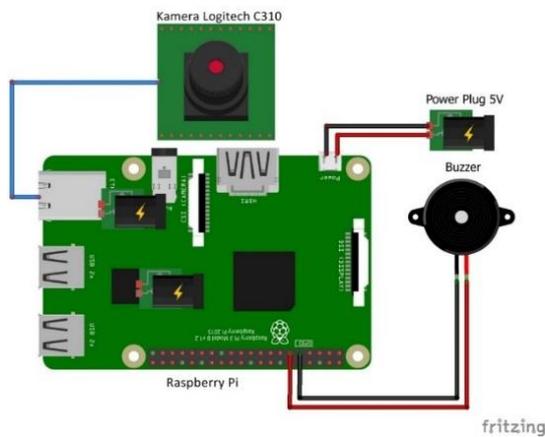
Perancangan alat untuk mendeteksi objek

penghalang perlu diperhatikan peletakan kamera dan sudut kamera agar dapat mengambil citra dengan maksimal. Desain alat ini membutuhkan *strap belt chest* untuk memasang Raspberry Pi, kamera dan buzzer. *Strap belt chest* ini akan digunakan pada dada depan pengguna setinggi 110 cm dan kemiringan kamera sebesar 41. Perancangan alat dan peletakan sensor dapat dilihat pada Gambar 2.

Pada Gambar 3 merupakan skematik rangkaian sistem. Kamera logitech C310 yang digunakan sebagai sensor untuk mengambil citra gambar dipasang pada Raspberry Pi menggunakan kabel USB. Terdapat *buzzer* aktif sebagai indikator peringatan berupa alarm ketika sistem mendeteksi objek penghalang. *Buzzer* dipasang pada raspberry pi pada pin GPIO. Serta memerlukan daya yang didapatkan dari powerbank yang dipangkan pada USB OTG.



Gambar 2. Desain Prototype Sistem



Gambar 3. Skematik Rangkaian Sistem

Tabel 1. Konfigurasi Pin Kamera dengan Raspberry Pi

Kamera Logitech C310	Raspberry Pi
USB	Slot USB
USB	Slot USB

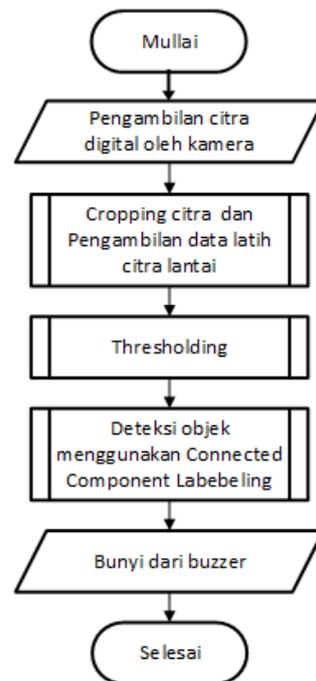
Tabel 2. Konfigurasi Pin Buzzer dengan Raspberry

Pi	
Buzzer	Raspberry Pi
VCC	GND pin 9
GND	GPIO 11

Pada tabel 1 dapat dilihat bahwa kamera logitech C310 dihubungkan pada Raspberry Pi menggunakan kabel USB pada slot USB Raspberry Pi.

Dari tabel 2 untuk dapat mengaktifkan buzzer dipasang pada mikrokontroler Raspberry Pi melalui pin GPIO. VCC pada buzzer dipasang pada pin GND 9 pada Raspberry Pi, sedangkan GND pada buzzer dipasang pada pin GPIO 11 pada Raspberry Pi

Dalam perancangan proses utama dilakukan beberapa proses yaitu, proses pengambilan citra digital oleh kamera dan inialisasi *variabel* dan *library* yang dibutuhkan terlebih dahulu. Kemudian citra digital tersebut dilakukan pemotongan (*cropping*) pada bagian bawah citra video dengan ukuran 150 x 150 piksel guna mempercepat waktu komputasi. Selanjutnya dilakukan perhitungan nilai *mean*, *median*, dan *min-max* pada citra RGB lantai pada citra yang sudah dipotong. Nilai-nilai tersebut akan menjadi batas nilai *threshold* pada proses *thresholding*. Proses tersebut yaitu pemisahan *background* dan *foreground*.



Gambar 4. Diagram Alir Perancangan Proses Utama

Setelah mendapatkan hasil dari proses

thresholding, selanjutnya akan dilakukan proses pelabelan menggunakan *connected component labeling*. Piksel yang telah terlabeli akan dihitung nilai rata-rata piksel labelnya. Setelah mendapatkan nilai rata-rata piksel label kemudian sistem akan menentukan objek yang dianggap sebagai penghalang ketika nilai dari piksel objek tersebut melebihi nilai rata-rata piksel label, namun objek yang memiliki nilai piksel dibawah rata-rata nilai piksel akan diabaikan. Sistem akan memberikan keluaran berupa bunyi dari *buzzer* ketika sistem dapat berhasil mendeteksi objek penghalang. Perancangan perangkat lunak sistem ini dapat dilihat berupa diagram alir pada Gambar 4.

2.3 Implementasi Sistem

Dalam mengimplementasikan alat deteksi objek penghalang ini ditunjukkan pada Gambar 5 dan 6. Pada Gambar 5 merupakan implementasi alat deteksi objek penghalang dari depan dan belakang. Alat ini dipasang di depan dada pengguna dengan ketinggian 110 cm menggunakan *strap belt chest*. Gambar 6 merupakan desain alat dari jarak dekat. Terdapat Raspberry Pi, kamera webcam dan buzzer



Gambar 5 Implementasi Alat Deteksi Objek Penghalang

(a) Implementasi alat deteksi objek penghalang tampak depan; (b) implementasi alat deteksi objek penghalang tampak belakang.



Gambar 6. Implementasi Alat Deteksi Objek Penghalang dari Dekat

3. PENGUJIAN DAN ANALISIS

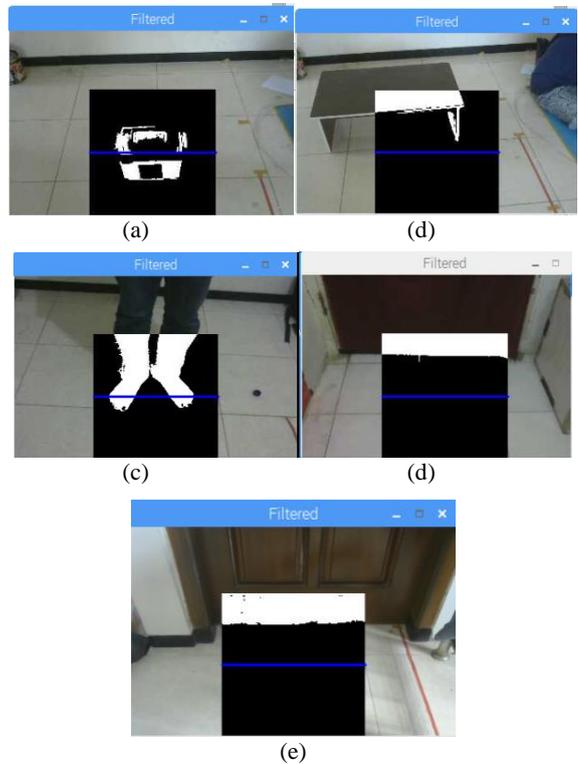
3.1. Pengujian Nilai *Threshold* pada Berbagai Waktu dan Jarak yang Berbeda-beda

Pengujian ini dilakukan untuk mengetahui nilai *threshold* yang memiliki akurasi paling baik dalam mendeteksi objek penghalang. Terdapat tiga nilai *threshold* yaitu nilai *mean*, *median* dan *min-max*. Pengujian nilai *threshold* ini dilakukan pada 5 objek berbeda yaitu orang, *box*, tempat sampah, meja dan pintu. Pada pengujian ini, objek berada pada jarak tertentu yaitu 50 cm, 75 cm, 100 cm dan 125 cm di depan pengguna.

Pada Gambar 7 merupakan contoh hasil pengujian pada pagi hari dengan jarak 75 cm dan 100 cm

3.1.1 Theshold dengan nilai *mean*

Dari data hasil perhitungan RGB batas nilai yang digunakan pada *threshold mean* adalah $B_{min} = 71.9$, $B_{max} = 202.6$, $G_{min} = 80.2$ dan $G_{max} = 198.2$, $R_{min} = 75.0$ dan $R_{max} = 193.4$. Hasil pengujian parameter nilai *threshold* dengan nilai *mean* mendapatkan nilai akurasi rata-rata sebesar 88.51%.



Gambar 7 Contoh Hasil Pengujian Objek pada Pagi Hari

(a) Objek *box*; (b) Objek meja; (c) Objek orang; (d) Objek dinding; (e) Objek Pintu

3.1.1 Theshold dengan nilai median

Dari data hasil perhitungan RGB batas nilai yang digunakan pada *threshold median* adalah Bmin = 71.0, Bmax = 179.0, Gmin = 78.0 dan Gmax = 194.5, Rmin = 75.0 dan Rmax = 190.0. Hasil pengujian parameter nilai *threshold median* memiliki nilai akurasi sebesar 83,43%.

3.1.1 Theshold dengan nilai min-max

Dari data hasil perhitungan RGB batas nilai yang digunakan pada *threshold mean* adalah Bmin = 40.0, Bmax = 255.0, Gmin = 49.0 dan Gmax = 255.0, Rmin = 48.0 dan Rmax = 252.0. Hasil pengujian parameter nilai *threshold* dengan nilai *min-max* memiliki nilai akurasi sebesar 62,25%

3.2. Pengujian Akurasi Sistem dalam Mendeteksi Objek menggunakan *Connected Component Labeling*

Tabel 3. Hasil Pengujian Deteksi Objek pada Kondisi Pagi, Siang dan Malam

Objek	Jarak (cm)	Hasil Deteksi			Akurasi (%)
		Pagi	Siang	Malam	
Box	50 cm	yes	yes	yes	100
Meja		yes	yes	yes	100
Orang		yes	yes	yes	100
Dinding		yes	yes	yes	100
Pintu		yes	no	no	66,7
Box	75 cm	yes	yes	yes	100
Meja		yes	yes	yes	100
Orang		yes	yes	yes	100
Dinding		yes	yes	yes	100
Pintu		yes	yes	yes	100
Box	100 cm	yes	yes	yes	100
Meja		yes	yes	yes	100
Orang		yes	yes	yes	100
Dinding		yes	yes	yes	100
Pintu		yes	yes	yes	100
Box	125 cm	yes	yes	yes	100
Meja		yes	yes	yes	100
Orang		yes	yes	no	33,3
Dinding		yes	yes	yes	100
Pintu		yes	no	yes	33,3
Rata-rata akurasi hasil deteksi					91,66

Pengujian ini dilakukan untuk mengetahui berapa persen tingkat akurasi sistem dalam mendeteksi objek halangan. Pengujian ini menguji kesesuaian hasil deteksi objek oleh *connected component labeling* dengan output sistem. Pengujian ini dilakukan pada 5 objek berbeda yaitu orang, *box*, tempat sampah, meja dan pintu. Pada pengujian ini, objek berada pada jarak tertentu yaitu 50 cm, 75 cm, 100 cm dan 125 cm di depan pengguna. Pada tabel 3 dapat dilihat bahwa system dalam mendeteksi objek penghalang memiliki akurasi sebesar 91,66%.

3.3 Pengujian Integrasi *Software* dan *Hardware*

Pengujian ini dilakukan untuk mengetahui berapa persen tingkat akurasi integrasi sistem dalam mendeteksi objek halangan dengan *hardware*. Pengujian ini menguji kesesuaian hasil deteksi objek dengan output sistem. Pengujian ini dilakukan pada 5 objek berbeda yaitu orang, *box*, tempat sampah, meja dan pintu. Pada pengujian ini, objek berada pada jarak tertentu yaitu 50 cm, 75 cm, 100 cm dan 125 cm di depan pengguna pada waktu pagi, siang dan Malam

Akurasi presentase rata-rata pada pagi hari dan siang hari mencapai 100% namun akurasi pada malam hari yaitu sebesar 95%. Dari hasil pengujian integrasi sistem *software* dan *hardware* pada waktu pagi, siang dan malam memiliki rata-rata akurasi yaitu 98,33%

3.4 Pengujian Rata-rata Waktu Komputasi

Pengujian ini dilakukan untuk mengetahui berapa lama waktu yang dibutuhkan sistem untuk menyelesaikan proses hingga mendapatkan hasil keluaran. Tabel 4 adalah hasil pengujian waktu komputasi deteksi objek penghalang. Dari pengujian yang telah dilakukan, nilai rata-rata waktu komputasi yaitu 166,15 ms.

Tabel 4. Hasil Pengujian Waktu Komputasi

Jarak	Waktu Komputasi (ms)
Rata-rata waktu komputasi pada jarak 50 cm	163,8
Rata-rata waktu komputasi pada jarak 75 cm	166,6
Rata-rata waktu komputasi pada jarak 100 cm	171
Rata-rata waktu komputasi pada jarak 125 cm	167,2
Rata-rata waktu komputasi	166,15 ms

4. KESIMPULAN

Berdasarkan pengujian yang telah dilakukan, terdapat beberapa kesimpulan yaitu:

1. Sistem dapat mendeteksi objek penghalang dengan menggunakan beberapa parameter threshold yaitu parameter nilai threshold mean, median, dan min-max pada jarak yang berbeda-beda menghasilkan nilai akurasi yang berbeda-beda. Dari ketiga batas threshold tersebut, threshold mean memiliki akurasi terbaik yaitu 88,51%. Sedangkan akurasi terburuk yaitu threshold min-max yang memiliki akurasi sebesar 66,25% sedangkan akurasi threshold median yaitu 84,47%.
2. Akurasi sistem dalam mendeteksi objek penghalang ini dapat dilihat pada hasil pengujian. Pengujian dilakukan di dalam ruangan dengan pada waktu pagi siang dan malam memiliki akurasi sebesar 91,66% tetapi dalam sistem ini masih memiliki kekurangan yaitu masih mengaggap cahaya pantulan, bayangan dan objek yang berwarna mirip dengan warna lantai akan tidak dapat terdeteksi. Hal ini dikarenakan sistem masih bergantung pada komponen RGB dari citra.
3. Akurasi integrasi sistem software dengan hardware yang diujikan pada waktu pagi, siang dan malam yaitu sebesar 98,33% hal ini dipengaruhi oleh noise seperti cahaya lampu yang memantul pada lantai atau bayangan pada lantai saat sistem digunakan
4. Waktu komputasi pada pengujian deteksi objek penghalang ini memiliki nilai yang berbeda-beda. Rata-rata waktu komputasi pada sistem deteksi objek penghalang ini yaitu 166,15 ms.

5. DAFTAR PUSTAKA

- Al Kadafi, A. J. & Utaminungrum, F., 2017. Deteksi Objek Penghalang Secara Real-Time Berbasis Mobile Bagi Penyandang Tunanetra Menggunakan Analisis Blob. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, Volume 2, pp. 423 - 432.
- Harimukhti, M. T. & Dewi, K. S., 2014. Ekplorasi Kesejahteraan Psikologis Individu Dewasa Awal Penyandang Tunanetra. *Jurnal Psikologi Undip*, Volume 13, pp. 64-77.
- Schwenk, K. H. F., 2015. *Connected Component Labelling Algorithm for very complex*

and High Resolution Images on an FPGA platform. s.l.:s.n.

- Silvia P, M., 2017. *World Health Organization*. [Online] Available at: <http://www.who.int/news-room/factsheets/detail/blindness-and-visual-impairment> [Diakses 27 May 2018].

- Wahyudi, D. A. & Kartowisastro, I. H., 2011. Menghitung Kecepatan Menggunakan Computer Vision. *Jurnal Computer Engineering Binus University*, Volume 19, p. 101.