

# Sistem Deteksi Lubang pada Pedestrian dengan Teknik Pengolahan Citra

Indrabayu<sup>1\*</sup>, Ahmad Rifaldi<sup>1</sup>, Intan Sari Areni<sup>2</sup>, Ingrid Nurtanio<sup>1</sup>, Anugrayani Bustamin<sup>1</sup>

<sup>1</sup>Departemen Teknik Informatika, Fakultas Teknik, Universitas Hasanuddin

<sup>2</sup>Departemen Teknik Elektro, Fakultas Teknik, Universitas Hasanuddin

Jl. Poros Malino km. 6, Bontomarannu, Kabupaten Gowa, Sulawesi Selatan 92171

\*Email: indrabayu@unhas.ac.id

DOI: 10.25042/jpe.112019.04

## Abstrak

Kawasan pedestrian di Indonesia masih tergolong belum optimal dalam memfasilitasi penggunaannya, dalam hal ini para pejalan kaki. Tidak sedikit dari kawasan tersebut masih ditemui jalanan pedestrian yang berlubang, hal ini bisa saja mencelakai pejalan kaki terutama pada penyandang tunanetra. Untuk itu dilakukan penelitian untuk membuat sistem yang dapat mendeteksi dan mengestimasi jarak lubang dengan pengolahan citra menggunakan mono kamera yang dapat membantu penyandang tunanetra. Metode yang digunakan untuk mendeteksi lubang adalah metode Threshold + Blob Analysis dan metode HSV. Hasil yang diperoleh menunjukkan tingkat akurasi pendeteksian lubang dengan menggunakan metode Threshold+Blob Analysis lebih baik dari metode HSV. Tingkat akurasi rata-rata Threshold+Blob Analysis sebesar 88,91%, sedangkan untuk metode HSV sebesar 86,82%.

## Abstract

*Pothole Detection System on Pedestrian using Image Processing Techniques.* The pedestrian areas in Indonesia are still far from optimal in facilitating the users or the pedestrians. Potholed pedestrian areas are found in many parts of the street. This issue can harm pedestrians, especially blind people. For this reason, research has been carried out to create a system that can detect and estimate hole distances by processing images using mono cameras that can help blind people. The methods used to detect holes are the Threshold + Blob Analysis method and the HSV method. The obtained results indicate the level of accuracy of hole detection using the Threshold + Blob Analysis method is better than the HSV method. The average accuracy level of Threshold + Blob Analysis is 88.91%, while for the HSV method is 86.82%.

**Kata Kunci:** Blob analysis, deteksi lubang, HSV, tuna netra

## 1. Pendahuluan

Lubang di jalan merupakan suatu hal yang dihindari, bukan hanya bagi pengendara motor atau mobil, lubang juga dapat mencelakakan pejalan kaki, khususnya pejalan kaki yang memiliki keterbatasan fisik dalam hal ini penyandang tunanetra. Lubang dapat terbentuk karena berbagai hal, salah satu penyebabnya ialah air yang menggenang [1].

Alat bantu tradisional yang biasa digunakan oleh penyandang tuna netra ialah tongkat pemandu dan anjing penuntun, kelemahannya ialah keterampilan dalam penggunaan alat dan diperlukan waktu untuk melatih anjing penuntun tersebut [2].

Berbagai penelitian sebelumnya memberikan solusi dengan membuat sebuah tongkat pintar yang dapat membantu kebutuhan tunanetra dalam mobilitasnya, penelitian yang dilakukan oleh Rao pada tahun 2016

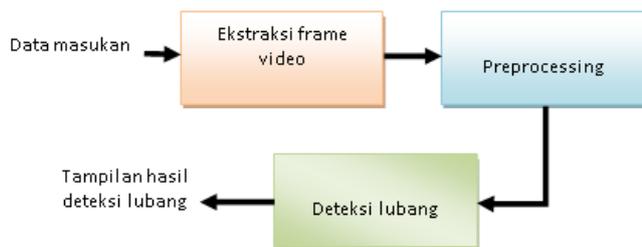
menggunakan laser dan mono kamera dalam membuat sistem yang dapat mendeteksi dan mengenali lubang ataupun retakan pada jalan [3]. Penelitian oleh Shurti Dambhare dan Prof. A. Sakhare menggunakan kamera stereo, proximity sensor, ultrasonic sensor dan modul GPS, dalam menghasilkan sebuah tongkat pintar untuk penyandang tunanetra [2].

Penelitian-penelitian yang telah dilakukan sebelumnya masih fokus pada pendeteksian lubang dan estimasi jarak lubang menggunakan lebih dari satu sensor atau tambahan perangkat lain. Oleh karena itu, pada penelitian ini dibuat sebuah sistem yang hanya menggunakan mono kamera untuk mendeteksi sekaligus menghitung estimasi jarak lubang berdasarkan teknik pengolahan citra. Metode yang digunakan yaitu Thresholding + Blob Analysis yang akan dibandingkan dengan metode HSV.



## 2. Metodologi Penelitian

Alur penelitian diperlihatkan pada Gambar 1. Data masukan pada penelitian ini adalah data video dengan resolusi 1920 x 1080 dengan format file mp4. Jumlah data masukan sebanyak 12 data video. Data masukan selanjutnya di ekstrak menjadi *frame-frame* lalu dilakukan *pre-processing* pada hasil ekstraksi. Untuk mendeteksi lubang digunakan dua metode pengolahan citra, yaitu *Thresholding* + *Blob Analysis* dan *HSV*. Unjuk kerja dari kedua metode akan dibandingkan dengan menggunakan software Matlab.



Gambar 1. Alur Penelitian

### 2.1. Ekstraksi Frame

Pada tahap ini, video di ekstrak menjadi beberapa frame yang akan diolah satu persatu sampai frame terakhir dalam video. Tahapan algoritma untuk ekstraksi frame diawali dengan melakukan inisialisasi terhadap jumlah frame video masukan.

### 2.2. Preprocessing

Pada bagian ini akan dilakukan beberapa tahapan, yaitu:

- *Cropping*  
Video yang telah di ekstrak menjadi frame-frame akan di-*crop* agar citra yang tidak perlu untuk di proses akan dibuang. Hal yang dibutuhkan dalam melakukan *cropping* adalah adanya koordinat awal, panjang dan lebar area *crop*.
- *Resize*  
Pada tahap ini citra di *resize* ke ukuran 640 x 480 piksel agar proses komputasi bisa lebih cepat.

### 2.3. Metode Thresholding

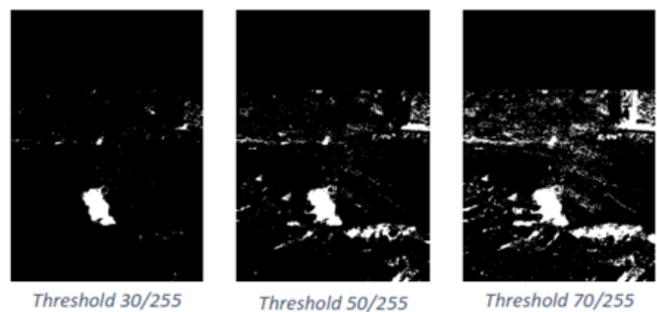
Metode *Thresholding* merupakan metode pengolahan citra untuk memetakan piksel yang memenuhi syarat ambang batas dan dipetakan

ke nilai piksel yang dikehendaki. Pada penelitian ini digunakan metode *thresholding* tunggal seperti diperlihatkan pada persamaan (1). Piksel – piksel yang nilai intensitasnya dibawah 30 diubah menjadi hitam (nilai intensitas 0), sedangkan piksel-piksel yang nilai intensitasnya di atas 30 diubah menjadi warna putih (nilai intensitas = 255) [4].

$$f_0(x, y) = \begin{cases} 0, & f_1(x, y) < 30 \\ 255, & f_1(x, y) \geq 30 \end{cases} \quad (1)$$

dimana  $f_0(x, y)$  adalah citra hasil *threshold*.

Perbandingan hasil dari beberapa *threshold* diperlihatkan pada Gambar 2. Hasil *threshold* dengan nilai 30/255 lebih bersih dari *noise* dibandingkan dengan nilai *threshold* lainnya.



Gambar 2. Hasil perbandingan nilai *threshold*

### 2.4. Blob Analysis

Pada pengolahan citra yang menggunakan segmentasi *foreground*, analisis *blob* merupakan teknik yang digunakan untuk menyatakan luas area piksel dari suatu image yang menjadi fokus deteksi [5]. Berdasarkan fungsi tersebut, pada penelitian ini digunakan metode *blob* yang nantinya digunakan sebagai filter berdasarkan luas area untuk mengeleminasi objek-objek yang bukan lubang namun terdeteksi sebagai lubang pada sistem. Untuk menentukan nilai *blob*, ada beberapa hal yang harus diketahui untuk menghasilkan sebuah *blob* yang optimal. Penentuan luas *blob* pada setiap objek pada proses segmentasi *foreground* perlu dianalisis karena nilai *blob* pada tiap objek akan berbeda. Hal ini dipengaruhi oleh fitur objek seperti ukuran, jenis, dan teknik pengambilan data video [6].

Nilai *blob* minimal yang digunakan pada penelitian ini sebesar 2000 dan nilai *blob* maksimalnya sebesar 25000 jadi sebuah objek

akan dikategorikan sebagai lubang jika luas areanya diantara 2000 dan 25000.

### 2.5. Segmentasi Warna dengan HSV

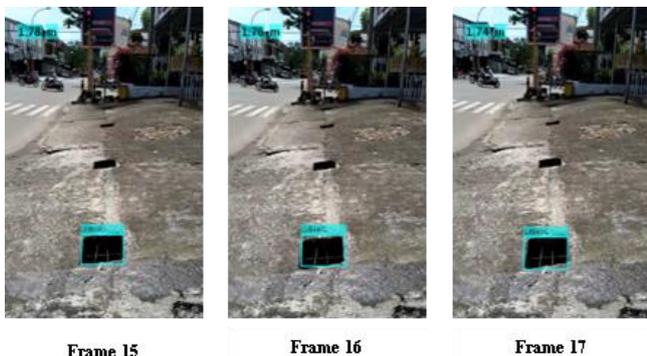
Segmentasi warna merupakan proses segmentasi dengan pendekatan daerah yang bekerja dengan menganalisis nilai warna dari tiap piksel pada citra dan membagi citra tersebut sesuai fitur yang diinginkan. Segmentasi citra dengan deteksi warna HSV menggunakan dasar seleksi warna pada model warna HSV dengan nilai toleransi tertentu [7].

Pada metode segmentasi dengan deteksi warna HSV dilakukan pemilihan sampel piksel sebagai acuan warna untuk membentuk segmen yang diinginkan [8]. Citra digital menggunakan model warna RGB sebagai 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 *adaptive threshold*. Hasil dari proses *threshold* tersebut akan membentuk segmen area dengan warna sesuai toleransi yang diinginkan.

Penelitian ini menggunakan nilai HSV terendah dengan kode HSV(0,0,0) dan nilai HSV tertinggi dengan kode HSV(180,255,30), jadi objek yang masuk dalam warna HSV tersebut yang akan dikategorikan sebagai lubang.

### 3. Hasil

Hasil deteksi lubang pada *frame* 15 hingga *frame* 17 untuk "Data 01" diperlihatkan pada Gambar 3. Lubang mulai terdeteksi ketika memasuki area ROI.



Gambar 3. Hasil deteksi lubang pada "Data 01"

Perbandingan *confusion matrix* hasil deteksi lubang dengan Thresholding + Blob Analysis dan HSV diperlihatkan pada Tabel 1. Untuk nilai akurasi menentukan Hasil perhitungan akurasi dari kedua metode dengan menggunakan *confusion matrix* berdasarkan Persamaan (2).

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (2)$$

dimana:

1. *TP (True Positive)*, merupakan *bounding box* pada objek lubang atau dengan kata lain objek tersebut terklasifikasi benar sebagai lubang;
2. *FP (False Positive)*, merupakan *bounding box* pada objek selain lubang atau dengan kata lain objek tersebut terklasifikasi dengan salah sebagai lubang.
3. *FN (False Negative)*, merupakan objek lubang yang tidak di *bounding box* atau dengan kata lain objek tersebut terklasifikasi dengan benar sebagai bukan lubang.
4. *TN (True Negative)*, merupakan objek selain lubang yang tidak di *bounding box* atau dengan kata lain objek tersebut terklasifikasi dengan benar sebagai bukan lubang.

Tabel 1. Perbandingan hasil deteksi lubang dengan metode Thresholding + Blob Analysis dan HSV

Data	Jumlah frame	Thresholding + Blob Analysis				HSV			
		TP	FP	TN	FN	TP	FP	TN	FN
Data 01	40	40	0	0	0	40	4	0	0
Data 02	40	40	0	0	8	37	22	0	3
Data 03	40	2	38	0	0	47	0	0	19
Data 04	40	36	4	0	6	43	0	0	6
Data 05	40	19	11	0	0	40	0	0	0
Data 06	40	38	2	0	0	40	0	0	0
Data 07	40	0	0	40	0	38	0	0	1
Data 08	40	0	0	40	0	0	0	40	0
Data 09	40	0	0	40	0	0	0	40	0
Data 10	40	0	0	40	0	0	5	35	0
Data 11	40	0	0	40	0	0	0	40	0
Data 12	40	0	0	40	0	0	0	40	0
<b>Total</b>	<b>480</b>	<b>175</b>	<b>55</b>	<b>240</b>	<b>8</b>	<b>285</b>	<b>31</b>	<b>195</b>	<b>29</b>

Setelah melakukan perhitungan terhadap 12 video dengan total *frame* sebanyak 480, diperoleh nilai akurasi untuk metode Thresholding + Blob Analysis sebesar 88,91%. Sedangkan untuk metode HSV diperoleh hasil akurasi sebesar 86,82%. Kesalahan deteksi terjadi akibat adanya objek lain yang berwarna gelap masuk ke dalam area deteksi seperti diperlihatkan pada Gambar 4.



**Gambar 4.** Contoh kesalahan deteksi lubang

#### 4. Kesimpulan

Berdasarkan penelitian yang telah dilakukan, pendeteksian lubang menggunakan metode pengolahan citra telah berhasil dilakukan. Tingkat akurasi dengan metode *Threshold + Blob analysis* diperoleh sebesar 88,91% sedangkan tingkat akurasi HSV sebesar 86,82%. Hasil deteksi memperlihatkan bahwa metode *Threshold + Blob Analysis* lebih baik dari metode HSV. Penelitian ini akan dikembangkan dengan menghitung estimasi jarak lubang dan pengembangan penggunaan *platform* Android.

#### Referensi

- [1] Detik.com. (2016). *Detik News*. Retrieved 12 12, 2016, from <http://news.detik.com/berita-jawa-timur/3139479/banyak-lubang-jalan-terbentuk-akibat-genangan-air-hujan>.
- [2] Shurti Dambhare, P. (2011). Smart Stick for Blind : Obstacle Detection, Artificial vision and Real-time assistance via GPS. *II*, 3.
- [3] Aravinda S. Rao, J. G. (2016). A Vision-Based System to Detect Pothole and Uneven Surfaces for Assisting Blind People. *IEEE ICC 2016*, 6.
- [4] Rahman, S. (2013, Desember). *Lecturers Blog STT-Harapan Medan*. Retrieved Juni 22, 2017, from <http://sayutirahman.sth-medan.ac.id/2013/12/thresholding.html>.
- [5] Zakiyabarsi, F., Niswar, M., & Zainuddin, Z. (2019). Crab Larvae Counter Using Image Processing. *EPI International Journal of Engineering*, 2(2), 127-131.
- [6] Basri (2015). *Metode Gaussian Mixture Models Untuk Optimalisasi Perhitungan Kendaraan Dalam Sistem Transportasi Cerdas*. Makassar.
- [7] Gunanto, S. (2009). Segmentasi warna bagian tubuh manusia pada citra 2D. *Proceeding SENTIA*.
- [8] Giannakopoulos, T. (2008). *Matlab color detection software, Department of Informatics and Telecommunications, University of Athens, Greece*. Retrieved from <http://www.di.uoa.gr/~tyiannak>.