

IMPLEMENTASI SISTEM PARKIR CERDAS DI UNIVERSITAS TELKOM. SUBSISTEM : PENGOLAHAN CITRA DIGITAL DENGAN DETEKSI TEPI CANNY DAN EMBEDDED SYSTEM

IMPLEMENTATION OF SMART PARKING SYSTEM IN TELKOM UNIVERSITY. SUBSYSTEM : DIGITAL IMAGE PROCESSING WITH CANNY EDGE DETECTION AND EMBEDDED SYSTEM

Restu Aulia Ulfah¹, Agus Virgono, Ir., M.T.², Agung Nugroho Jati, S.T., M.T.³

^{1,2,3}Prodi S1 Teknik Komputer, Fakultas Teknik Elektro, Universitas Telkom Bandung

¹raulafh@gmail.com, ²avirgono@telkomuniveristy.ac.id, ³agungnj@telkomuniversity.ac.id

Abstrak

Kendaraan beroda empat membutuhkan *space* lebih luas sehingga pengemudi harus mengelilingi area parkir untuk mendapatkan tempat parkir. Pada Tugas Akhir ini akan diimplementasikan *smart parking system* dengan pendeteksi ketersediaan parkir mobil berbasis pengolahan citra digital dan Raspberry pi. Masukkan data dari kamera *usb* berupa model citra RGB kemudian dilakukan konversi warna menjadi *grayscale* selanjutnya dilakukan *thresholding* untuk menghasilkan keluaran citra biner yang akan dikirim ke *server* melalui LAN. Pengolahan citra pada sistem menggunakan metode *canny edge detetction*. Berdasarkan pengujian yang dilakukan diperoleh rata-rata presentase konsistensi data dalam mendeteksi tanda segitiga sebesar 83.333% dan keberhasilan sistem dalam memastikan informasi yang di peroleh *database* pada *server* sesuai dengan data pada Raspberry pi memiliki presentase sebesar 92%. Waktu komputasi sistem dalam mendeteksi tiga tanda mencapai 46.8 *ms*, deteksi dua tanda selama 36.1 *ms* dan deteksi satu tanda yaitu 33.2 *ms*. Kondisi cahaya yang berbeda mempengaruhi kestabilan sistem dalam mendeteksi tanda. Sistem stabil dalam mendeteksi tanda pada intensitas cahaya dengan rentang nilai sebesar 6000 - 4000 *lux*.

Kata kunci : Pengolahan citra digital, Raspberry pi, *Local Area Network*, *Canny Edge Detetction*, Ketersediaan parkir.

1. Pendahuluan

Berdasarkan data Korps Lalu Lintas Kepolisian Negara Republik Indonesia peningkatan jumlah mobil naik 11% tentunya hal ini akan menimbulkan kendala dalam menemukan tempat parkir yang kosong[1]. Pada tempat parkir kampus biasanya pengemudi harus mencari sendiri area parkir yang kosong, pengemudi parkir harus berputar terlebih dahulu jika lokasi tempat parkir sangat luas. Permasalahan tersebut dapat dihindari apabila pengemudi telah mengetahui kondisi area parkir yang kosong. Melihat permasalahan yang telah dipaparkan maka penulis berniat membuat "*Implementation smart parking system in Telkom University; Subsistem Digital Pengolahan citra and Raspberry pi*". Diharapkan sistem parkir ini dapat membantu pengemudi atau pengguna parkir mobil *outdoor* agar mendapatkan area parkir yang tepat dan tidak harus berputar-putar mencari tempat parkir yang kosong.

Penelitian ini bertujuan untuk mendeteksi tanda pada slot area parkir agar dapat memberikan informasi kepada *user* mengenai ketersediaan slot parkir yang kosong. Kamera *usb* diletakkan pada lokasi parkir dengan ketinggian dan sudut tertentu agar dapat menjangkau slot parkir sesuai dengan display pada monitor yang terhubung pada Raspberry Pi. Pada setiap slot tempat parkir dibuat tanda yang akan menjadi tanda pengambilan citra. Untuk mengetahui sebuah area parkir terdapat slot parkir yang kosong dapat dilakukan dengan meletakkan kamera yang dapat mendeteksi tanda pada slot parkir. Tanda pada slot parkir dapat dideteksi dengan melakukan pengolahan citra digital pada hasil

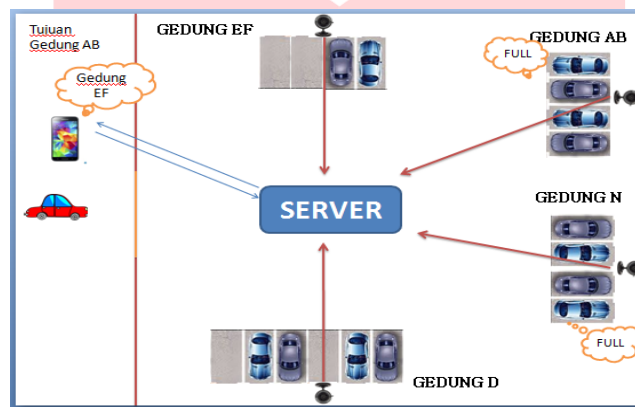
capture kamera yang terhubung Raspberry pi sebagai *embedded system*. Pengiriman data melalui *Local Area Network (LAN)* kemudian *server* mengirimkan informasi slot parkir yang kosong kepada pengemudi melalui *smartphone* berbasis *android* yang telah diinstal aplikasi *smart parking*.

2. Metodologi (Perancangan Sistem)

2.1 Smart Parking System

Smart system berarti sistem yang pintar, pintar memiliki arti mampu melakukan sesuatu dengan baik, teratur, dan rapi sesuai dengan aturan yang berlaku, serta mampu menyerap informasi dengan baik dan cepat sebagai hasil dari pembelajaran yang dilakukan oleh sistem tersebut. *Smart parking system* merupakan bagian atau *part of smart city* yang mengatur dalam hal tata kelola area parkir sehingga lebih teratur dan efisien [8].

2.2 Perancangan Umum



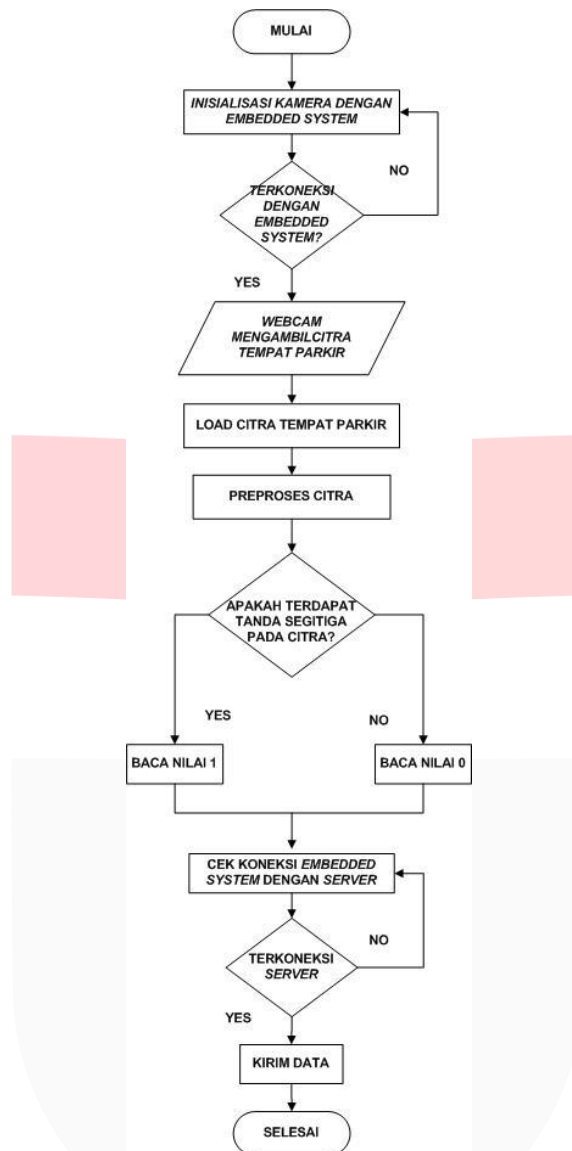
Gambar 1 Deskrip Umum *Smart Parking System*

Kamera *usb* diletakkan pada area parkir dengan ketinggian dan sudut tertentu agar dapat menjangkau slot parkir sesuai dengan display pada monitor yang terhubung pada Raspberry pi. Pada setiap slot tempat parkir dibuat tanda sebagai tanda yang akan dideteksi oleh sistem. Tanda yang akan diberikan pada setiap slot tempat parkir yaitu tanda segitiga. Ketika kamera melakukan *capture* pada area slot parkir maka dilakukan pengolahan citra digital oleh Raspberry pi dan selanjutnya data citra yang telah diproses akan dikirim ke *server* setiap terjadi perubahan data. Apabila kamera digital menangkap citra tanda segitiga pada sebuah slot maka slot pada tempat parkir tersebut kosong, sedangkan jika hanya menangkap sebagian tanda segitiga atau bahkan tidak terlihat maka pada slot tersebut telah terisi. Hasil *output* dari pengolahan citra pada Raspberry pi berupa citra biner yaitu 1 dan 0. Hasil *output* tersebut memberi informasi terisi atau kosong pada suatu slot parkir yang kemudian hasil tersebut dikirim ke *server* menggunakan kabel LAN dan disimpan pada basis data lokal yang berada di *server*. Kecepatan pengiriman data hasil pengolahan citra digital bergantung pada kecepatan mendeteksi tanda segitiga pada hasil *capture image* tempat parkir.

2.3 Perancangan Data Input dan Output

Tabel 1 Data *Input* dan *Output*

No.	Data input	Data output	Keterangan
1.	Citra RGB	1	Terdeteksi tanda
2.	Citra RGB	0	Tidak terdeteksi tanda



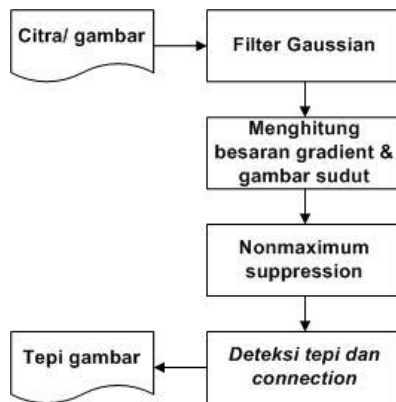
Gambar 2 Skema Perancangan Sistem

2.4 Perancangan *Image Processing*

- Kamera melakukan *capture* image pada slot parkir
- Diperoleh citra RGB kemudian konversi citra *grayscale*
- *Smoothing* citra *grayscale* dengan filter *Gaussian*
- Proses deteksi tepi dengan algoritma *canny*
- Konversi citra *grayscale* ke citra *biner* dengan *thresholding*
- Kirim nilai “1” atau “0” kepada *server* melalui kabel LAN

2.2. *Canny Edge Detetction*

Deteksi tepi adalah alat dasar yang digunakan dalam sebagian besar aplikasi pengolahan citra untuk mendapatkan informasi dari frame sebagai langkah precursor untuk fitur ekstraksi dan segmentasi objek.



Gambar 2 Langkah Algoritma *Canny*[10]

3. Pengujian Sistem

3.1. Hasil Pengujian Konsistensi Data

Tabel 2 Tabel Sample Pengujian Konsistensi Data

No.	Input (citra)	Slot Area	Kondisi	Output Monitor	Keterangan
1	Deteksi 2 tanda	E003	Terdeteksi	Masuk area 3	Konstan
		E002	Terdeteksi	Masuk area 2	
		E001	Terdeteksi	-	
2	Deteksi 2 tanda	E003	Terdeteksi	Masuk area 3	Konstan
		E002	Terdeteksi	Masuk area 2	
		E001	Terdeteksi	-	
3	Deteksi 1 tanda	E003	Tidak Terdeteksi	-	Tidak Konstan
		E002	Tidak Terdeteksi	-	
		E001	Tidak Terdeteksi	-	
4	Deteksi 3 tanda	E003	Terdeteksi	Masuk area 3	Konstan
		E002	Terdeteksi	Masuk area 2	
		E001	Terdeteksi	Masuk area 1	
5	Deteksi 1 tanda	E003	Tidak Terdeteksi	-	Konstan
		E002	Tidak Terdeteksi	-	
		E001	Terdeteksi	Masuk area 1	
Presentase nilai tidak konstan				16.67%	

Berdasarkan pada Tabel di atas dapat dilihat bahwa *capture* kamera berupa citra yang terdapat tanda pada slot area dengan hasil deteksi tanda yang ditampilkan pada monitor memiliki nilai presentase tidak konstan sebesar 16.667% dari pengambilan data 150 kali. Jika terdeteksi tanda maka *output* pada monitor akan menampilkan keterangan masuk area 1, 2 atau 3 sesuai dengan slot parkir mana yang terdeteksi tanda. Dari hasil pengambilan data sebanyak 150 kali terdapat 25 data yang tidak konstan atau tidak stabil, dari hasil pengamatan jumlah data yang tidak konstan berbanding lurus dengan banyaknya jumlah tanda yang dideteksi. Semakin banyak tanda yang dideteksi maka semakin

banyak kemungkinan sistem yang tidak konstan dalam mendeteksi tanda. Sehingga nilai konsistensi pada deteksi data yang dilakukan yaitu sebesar 83.333%.

3.4. Hasil Pengujian Pencahayaan

Tabel 3 Pengujian Pencahayaan

No	Waktu	Luxmeter	Keterangan (150 pengambilan data)
1	Pagi (07.00 – 10.00)	4000 – 6000 lux	46 tidak konsisten
2	Siang (10.00 – 14.00)	7000 – 9000 lux	59 tidak konsisten
	Siang (14.30 – 15.30)	8000 – 6000 lux	40 tidak konsisten
3	Sore (15.30 – 17.30)	6000 – 4000 lux	25 tidak konsisten
4	Malam (18.00 – 21.00)	Minimum 173 lux	48 tidak konsisten

Berdasarkan di atas intensitas cahaya sangat berpengaruh dalam proses pendeteksi tanda pada area slot parkir. Sistem dengan rentang nilai intensitas cahaya sebesar 6000 - 4000 lux memiliki presentase nilai yang tidak konstan sebesar 30.667%. Intensitas cahaya dengan nilai rentang 7000 – 9000 lux, presentase nilai yang tidak konstan sebesar 39.333%. Intensitas cahaya dengan nilai rentang 8000 - 6000 lux, presentase nilai yang tidak konstan sebesar 26.667%. Pada malam hari diperlukan lampu bolam yang dapat menerangi tanda pada slot parkir agar intensitas cahaya mencapai 173 lx dengan presentasi nilai yang tidak konstan sebesar 32%,

3.2. Hasil Pengujian Akurasi Pengiriman Data

Tabel 4 Tabel Sample Pengujian Akurasi Pengiriman Data

No	Input (citra)	Slot Area	Output Raspberry Pi	Output Server	Keterangan
1	Deteksi 3 tanda	E003	Masuk area 3	1	Sesuai
		E002	Masuk area 2	1	
		E001	Masuk area 1	1	
2	Deteksi 1 tanda	E001	-	0	Sesuai
3	Deteksi 1 tanda	E001	Masuk area 1	1	Sesuai
4	Deteksi 1 tanda	E002	-	0	Sesuai
5	Deteksi 2 tanda	E003	-	0	Sesuai
		E001	-	0	
6	Deteksi 1 tanda	E003	Masuk area 3	1	Sesuai
7	Deteksi 2 tanda	E002	Masuk area 2	1	Sesuai
		E001	Masuk area 1	1	
8	Deteksi 3 tanda	E003	Masuk area 3	0	Sesuai
9	Deteksi 1 tanda	E001	-	0	Sesuai
10	Deteksi 1 tanda	E003	Masuk area 3	1	Sesuai

Citra inputan berupa deteksi tiga tanda, dua tanda dan satu tanda slot parkir memiliki *output* yang sesuai dengan *output server*. Data yang ditampilkan pada *server* hanya data slot area parkir yang mengalami perubahan. Jika terdapat data yang tidak terjadi perubahan namun ditampilkan *output* pada

server maka terjadi kesalahan pada sistem. Sistem dikatakan berhasil melakukan pengiriman data jika deteksi tanda pada Raspberry pi dan *server* memiliki nilai atau status yang sama. Pada pengujian ini terjadi nilai eror atau data yang tidak konstan sebesar 8% dan nilai keberhasilan 92% dalam pengiriman data sebanyak 150 kali.

3.3. Hasil Pengujian Waktu Komputasi

Tabel 5 Tabel Pengujian Waktu Komputasi Deteksi 3 Tanda

No	Input	Waktu Komputasi (ms)	Rata – rata waktu komputasi
1	Citra deteksi 3 tanda	46	46.8 ms
2		47	
3		46	
4		50	
5		47	
6		42	
7		45	
8		46	
9		52	
10		47	

Tabel 6 Tabel Pengujian Waktu Komputasi Deteksi 2 Tanda

No	Input	Waktu Komputasi (ms)	Rata – rata waktu komputasi
1	Citra deteksi 2 tanda	36	36.1 ms
2		38	
3		36	
4		36	
5		37	
6		36	
7		34	
8		40	
9		36	
10		32	

Tabel 7 Tabel Pengujian Waktu Komputasi Deteksi 1 Tanda

No	Input	Waktu Komputasi (ms)	Rata – rata waktu komputasi
1	Citra deteksi 1 tanda	32	33.2 ms
2		34	
3		32	
4		32	
5		34	
6		36	
7		34	
8		34	
9		32	
10		32	

Berdasarkan hasil pengujian dapat diketahui waktu komputasi sistem dalam mendeteksi tanda pada area slot parkir. Rata-rata waktu yang diperlukan dalam mengolah citra dengan deteksi tiga tanda yaitu selama 46.8 ms dengan waktu maksimum 52 ms dan waktu minimum 42 ms. Pada deteksi dua tanda diperoleh rata-rata waktu komputasi dalam melakukan pengolahan citra selama 36.1 ms dengan waktu maksimum 40 ms dan waktu minimum 32 ms. Kemudian pada deteksi satu tanda diperoleh rata-rata waktu komputasi yang diperlukan selama 33.2 ms dengan waktu maksimum 32 ms dan waktu minimum 36 ms. Analisa yang diperoleh dari pengujian ini yaitu jumlah tanda yang dideteksi berbanding lurus dengan lamanya waktu komputasi dalam mendeteksi tanda. Semakin banyak jumlah tanda yang dideteksi maka semakin lama waktu yang dibutuhkan untuk melakukan komputasi deteksi tanda.

4. Kesimpulan

Berdasarkan analisis dan pengujian yang dilakukan pada bab sebelumnya, diperoleh beberapa kesimpulan sebagai berikut :

1. *Smart parking system* dengan algoritma pendeteksi tepi *canny* dapat mendeteksi tanda pada slot area parkir *outdoor* dengan nilai keberhasilan kestabilan dalam mendeteksi data sebesar 83.333% dan ketidakstabilan dalam mendeteksi tanda sebesar 16.667% berdasarkan 150 kali pengambilan data.
2. Pengaruh intensitas cahaya terhadap deteksi tanda pada slot area parkir *outdoor* memiliki lima presentase keberhasilan dalam mendeteksi tanda pada kondisi intensitas cahaya yang berbeda yaitu 69.333% pada intensitas cahaya pagi hari, 60% dan 73.333% pada intensitas cahaya siang hari, 83.333% pada intensitas cahaya sore hari, dan 68% pada intensitas cahaya malam hari. Pengaruh yang muncul yaitu kamera akan sulit mendeteksi tanda. Rentang nilai intensitas sebesar 6000 - 4000 *lux* membuat sistem cukup stabil dalam mendeteksi tanda. Sedangkan rentang nilai intensitas sebesar 7000 - 9000 *lux* mengakibatkan sistem tidak stabil. Pencahayaan yang sangat kecil seperti kondisi malam hari dengan nilai 3 *lux* dapat mengakibatkan sistem tidak mampu mendeteksi tanda. Oleh karena itu perlu dipasang lampu bolam dengan nilai intensitas cahaya minimum 173 *lux* agar dapat mendeteksi tanda.
3. Informasi hasil pengolahan citra pada Raspberry pi berhasil dikirim ke *server* melalui kabel LAN. Pengiriman data hanya dilakukan ketika data mengalami perubahan nilai hasil deteksi tanda. Presentase akurasi dalam pengiriman data yaitu 92% berdasarkan 150 kali pengambilan data.
4. Tanda pada slot area parkir berhasil di deteksi dengan waktu komputasi yang beragam pada saat sistem mendeteksi tiga tanda, dua tanda dan satu tanda. Rata-rata waktu komputasi berlangsung selama 46.8 ms pada deteksi tiga tanda, pada deteksi dua tanda diperoleh waktu komputasi dalam melakukan pengolahan citra selama 36.1 ms. Kemudian pada deteksi satu tanda waktu komputasi yang diperlukan selama 33.2 ms.

5. Daftar Pustaka

- [1] Agung Kurniawan, "Jumlah Kendaraan Indonesia Capai 104,211 Juta Unit," April 2014. [Online]. Available : <http://www.tribunnews.com/otomotif/2014/04/15/jumlah-kendaraan-di-indonesia-capai-104211-juta-unit>. [Accessed 28 Oktober 2014].
- [2] Ahmad Saprudin, "Jurnal Membangun Jaringan LAN pada Perusahaan Three Pringsewu Lampung dengan menggunakan Metode OSPF by. Asep," Lampung: Akademi Manajemen Informatika dan Komputer Dian Cipta Cendekiawan Pringsewu, 2013.
- [3] Alvin Antonius, Dedi Triyanto and Ikhwan Ruslianto, "Penerapan Pengolahan Citra dengan Metode Adaptive Motion Detection Algorithm pada Sistem Kamera Keamanan dengan Push Notification ke Smartphone Android," Pontianak: Universitas Tanjungpura, 2015.
- [4] Anoop Mishra and Arshita Dixit, "Embedded Image Capturing & Digital Converting Process using Raspberry pi System interfacing and Comparison of Generation 2 verses Generation 1 models in Raspberry pi" International Journal of Computer Science and Information Technologies (IJCSIT), Vol. 6, no. 2, pp. 1798-1801, April 2015.
- [5] C.Danil, "Deteksi tepi dengan Algoritma Canny," Medan: STMIK IBBI, 2013.
- [6] Decy Nataliana, Iqbal Syamsu and Galih Giantara, "Sistem Monitoring Parkir Mobil menggunakan Sensor Infrared berbasis Raspberry pi," Bandung: Institut Teknologi Nasional, 2014.
- [7] Helmiriawan, "Rancang Bangun dan Analisis Sistem Pemantau Lalu Lintas menggunakan OpenCV dengan Algoritma Canny dan Blob Detetction, " Jakarta: Universitas Indonesia, 2012.
- [8] I Putu Agus Eka Pratama, "Smart City beserta Cloud Computing," Bandung : Informatika, 2014.
- [9] Johannes W. Yodha and Achmad W. Kurniawan, "Perbandingan Penggunaan Deteksi tepi dengan Metode Laplace , Sobel, dan Prewit dan Canny pada Pengenalan Pola," Semarang: Universitas Dian Nuswantoro, 2014.
- [10] Rashmi, Mukesh Kumar and Rohini Saxena, "Algorithm and Technique on Various Edge Detection : A Survey," Signal & Image Processing : An International Journal (SIPIJ), Vol.4, No.3, June 2013.
- [11] Mazid Kamal and Ruri S. Basuki, "Segmentasi Citra Daun Tembakau Berbasis Deteksi tepi menggunakan Algoritma Canny," Semarang: Universitas Dian Nuswantoro, 2011.
- [12] Nanda Septiana, "Implementasi dan Analisis Motion Identification Berupa Gaya Berjalan Manusia serta Aplikasinya dalam Sistem Keamanan Rumah menggunakan Support Vector Machine," Bandung: Telkom University, 2014.
- [13] Eko Wibowo, R.Isnanto, and Ajub Ajulian Zahra, "Perbandingan Kinerja Operator Sobel dan Laplacian of Gaussian (LoG) terhadap Acuan Canny untuk Mendeteksi tepi Citra," Semarang: Universitas Dian Nuswantoro, 2013.
- [14] Ping ZHOU, Wenjun YE, Yaojie XIA and Qi WANG, "An Improved Canny Algorithm for Edge Detection," Journal of Computational Information Systems (JOFCIS), Vol 7, no. 5, pp. 1516-1523, 2011.

