



Available online at : <http://bit.ly/InfoTekJar>

InfoTekJar : Jurnal Nasional Informatika dan Teknologi Jaringan

ISSN (Print) 2540-7597 | ISSN (Online) 2540-7600



Aplikasi Simulasi Deteksi Lokasi Parkir Kosong Menggunakan Ekstraksi Ciri Objek

Sayuti Rahman, Haida Dafitri

Program Studi Teknik Informatika Universitas Harapan Medan, Jln HM Joni No 70, Medan, 20217, Indonesia

KEYWORDS

Ekstraksi Ciri, Visi Komputer, Aplikasi Parkir

CORRESPONDENCE

Phone: +6281262847435

E-mail: masay.ram@gmail.com

A B S T R A C T

Parkir merupakan bagian yang penting baik bagi masyarakat, pengelola parkir dan negara. Bagian penting dalam parkir adalah aspek kebutuhan parkir, desain area parkir, pengawasan parkir dan pengendalian parkir. Negara telah berupaya untuk mengendalikan parkir melalui aturan-aturan yang telah diberlakukan pada pengelolaan parkir.

penelitian ini merancang miniatur parkir beserta kendaraan mobil yang dilengkapi dengan kamera, mobil adalah miniatur mobil yang memasuki area parkir. Jika terdapat mobil maka terlihat pada aplikasi android atau browser bahwa parkir telah terisi dan aplikasi akan menunjukkan area parkir yang masih kosong. Dari hasil pengujian aplikasi mampu mendeteksi parkir dengan baik sehingga pengguna dapat melihat informasi melalui browser atau smartphone mereka.

PENDAHULUAN

Simulasi parkir ini diperlukan untuk proses penelitian sebuah sistem parkir yang akan dikembangkan. Sehingga dapat mendeteksi keadaan lokasi parkir, yang mana nantinya akan membantu pengendara menemukan lokasi parkir yang kosong. Menurut Badan Pusat Statistik bahwa kendaraan bermotor meningkat secara drastis pada tahun 2010 berkisar 76 Juta berkembang menjadi 138 Juta pada tahun 2017. Meningkatnya kendaraan mengakibatkan kebutuhan lahan parkir semakin luas, dengan keterbatasan area perparkiran maka perlu adanya aplikasi yang mengarahkan dan memberi informasi tentang keadaan lokasi parkir.

Pada penelitian sebelumnya dilakukan pengambilan informasi lokasi parkir kosong pada video realtime menggunakan metode background subtraction, namun hasil deteksi sangat dipengaruhi oleh perubahan cahaya [1]. Oleh karena itu pada penelitian ini menggunakan ekstraksi ciri untuk mengenali lokasi parkir kosong.

Untuk memudahkan penelitian maka dalam penelitian ini akan merancang miniatur parkir beserta kendaraan mobil yang dilengkapi dengan kamera, mobil adalah miniatur mobil yang memasuki area parkir. Jika terdapat mobil maka terlihat pada aplikasi android bahwa parkir telah terisi dan aplikasi akan menunjukkan area parkir yang masih kosong.

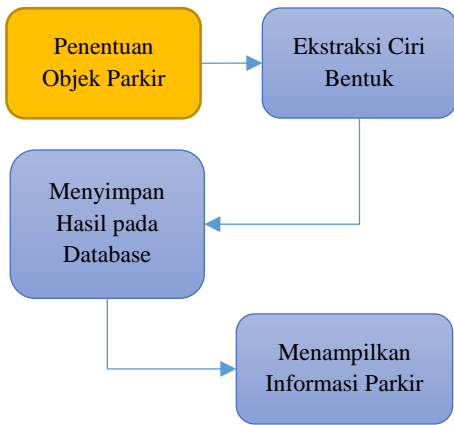
Pembacaan area parkir dengan kamera real time yang dilengkapi dengan metode ekstraksi fitur parkir yang dirancang dengan Bahasa

pemrograman matlab. hasil dari deteksi keadaan parkir dikirim ke database. Database dibaca oleh aplikasi android secara real time sebagai interface bagi pengguna.

Otomatisasi dalam parkir telah banyak diteliti dengan berbagai cara. Diantaranya, dengan membaca smart card untuk proses masuk dan keluar parkir [2]. Namun masih minim penelitian untuk memandu pengguna sampai pada lokasi yang tersedia. Deteksi parkir kosong ini juga sebagai penunjuk pengunjung yang dapat dilihat dalam aplikasi mobile dan browser.

METODOLOGI PENELITIAN

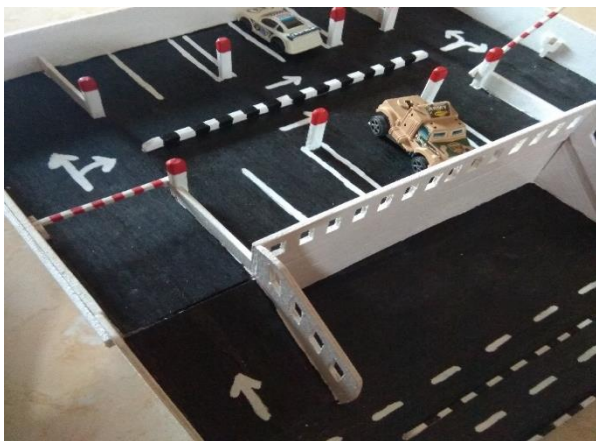
Kerangka kerja ini merupakan langkah-langkah yang akan dilakukan dalam penyelesaian masalah yang akan dibahas. Adapun kerangka kerja penelitian ini dapat digambarkan pada gambar 1 berikut :



Gambar 1 Tahap Penelitian

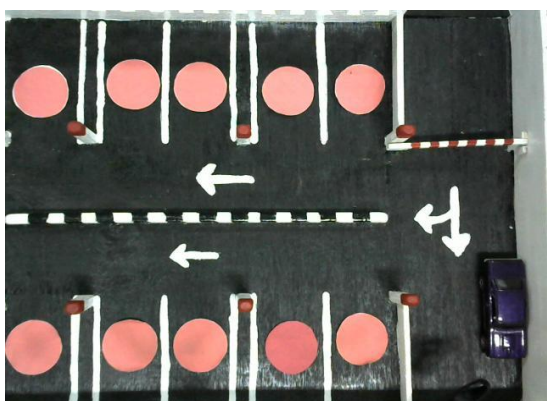
Penentuan Objek Parkir

Miniature parkir yang didesain menyerupai lokasi parkir asli. Miniature ini dilengkapi dengan kamera untuk melihat keadaan parkir secara real-time. Agar komputer mampu mengenali keadaan parkir maka perlu ada objek penanda parkir saat kosong. Pada penelitian ini menggunakan objek lingkaran agar tidak menyerupai bentuk lainnya. Objek lingkaran ini akan tampak saat parkir kosong dan ketika mobil terparkir di atasnya maka objek ini tidak terlihat. Berikut gambar miniatur parkir seperti terlihat pada gambar 2.



Gambar 2 Rancangan Miniatur Parkir

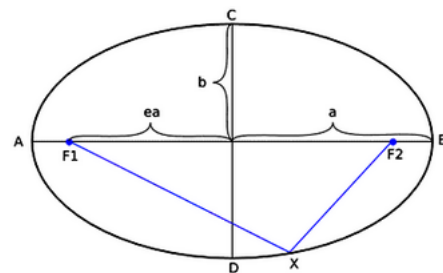
Berikut miniature yang telah diberi objek lingkaran sebagai penanda parkir kosong seperti terlihat pada gambar 3 berikut.



Gambar 3 Miniatur Parkir setelah diberi Objek Lingkaran

Ekstraksi Ciri

Bentuk merupakan gambaran dari suatu objek dalam posisi, orientasi dan ukuran. Ciri bentuk dalam suatu citra sangat esensial untuk segmentasi citra karena dapat mendeteksi objek. Untuk membedakan objek dengan objek lainnya dapat menggunakan parameter *eccentricity*. *Eccentricity* merupakan jarak perbandingan antara *foci ellipsis minor* dengan *foci ellipsis major*. *Eccentricity* memiliki jarak nilai antara 0 hingga 1. Objek yang berbentuk memanjang atau menyerupai bentuk garis lurus, nilai *eccentricity*-nya adalah, sedangkan nilai yang berbentuk lingkaran, nilai *eccentricity*-nya bernilai 0. Perhitungan *eccentricity* dapat diilustrasikan pada gambar 4.



Gambar 4 Perhitungan *eccentricity*

Perhitungan *eccentricity* disajikan pada persamaan 1

$$Metric = \sqrt{1 - \frac{b^2}{a^2}} \dots\dots\dots(1)$$

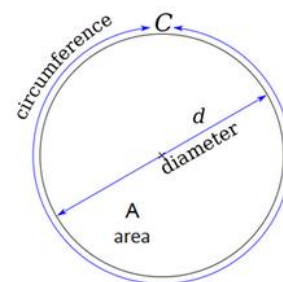
Dimana :

e = *Eccentricity*

a = *major axis*

b = *minor axis*

Dimana *a* menunjukkan *major axis* (jari-jari terpanjang) dari suatu citra dengan ukuran piksel, sedangkan *b* menunjukkan *minor axis* (jari-jari terpendek) dari suatu citra dengan ukuran piksel. Paramater lain yang bisa digunakan untuk membedakan suatu objek yaitu *matric*. *Matric* adalah nilai perbandingan antara luas dan keliling objek yang disebut dengan ukuran kebulatan (*roundness*). *Matric* memiliki rentang nilai antara 0 dan 1. Objek yang berbentuk garis lurus, nilai *matric*-nya mendekati 0, sedangkan objek yang berbentuk lingkaran, nilai *matric*-nya mendekati 1. Perhitungan *matric* diilustrasikan pada gambar 5.



Gambar 5 Perhitungan *matric*

Perhitungan *matric* disajikan pada persamaan 2

$$M = \frac{4\pi + A}{C^2} \dots\dots\dots (2)$$

Dimana:

$M = Metric$

$A = Area$

$C = Circumference$

Dengan C yaitu keliling sebuah citra dengan ukuran piksel, dan A adalah luas area citra dengan satuan piksel. Untuk membedakan ukuran objek yang satu dengan objek lainnya dapat menggunakan parameter luas (*area*) dan keliling (*parameter*). Luas yang disebut juga dengan *area* merupakan banyaknya piksel yang menyusun suatu objek. Sedangkan keliling yang disebut *parameter* adalah banyaknya piksel yang mengelilingi objek [3].

Proses mendeteksi objek lingkaran dengan menghitung nilai *area*, *perimeter*, *matric* dan *eccentricity*. Pada contoh perhitungan akan menggunakan radius 5". Perhitungan pertama mencari nilai *perimeter* dengan rumus $C = 2\pi \cdot radius$

$$C = 2\pi \cdot radius$$

$$C = 2\pi \cdot 5$$

$$C = \pi \cdot 10$$

$$C = 31.4$$

Kemudian mencari nilai *area* pada objek lingkaran dengan rumus

$$A = \pi \cdot r^2$$

$$A = \pi \cdot r^2$$

$$A = \pi \cdot (5)^2$$

$$A = \pi \cdot 25$$

$$A = 78.5$$

Setelah mendapat nilai *perimeter* dan *area*, selanjutnya dilakukan perhitungan *metric* untuk menentukan nilai lingkaran.

$$M = \frac{4\pi + A}{C^2}$$

$$M = \frac{4\pi + 78.5}{31.4^2}$$

$$M = \frac{984.46}{985.96}$$

$$M = 1.0$$

Matric memiliki rentang nilai antara 0 dan 1. Objek yang berbentuk garis lurus, nilai *matric*-nya mendekati 0, sedangkan objek yang berbentuk lingkaran, nilai *matric*-nya mendekati 1. Maka nilai *matric* pada persamaan di atas adalah objek berbentuk lingkaran.

Perhitungan selanjutnya mencari nilai *eccentricity* untuk mengenali objek lingkaran. Menghitung *eccentricity* dengan rumus $e = \frac{c}{a}$. Untuk mendapatkan nilai *c* dilakukan proses sebagai berikut

$$a = 3$$

$$b = 2$$

$$c^2 = a^2 - b^2$$

$$c^2 = 3^2 - 2^2$$

$$c^2 = 9 - 4$$

$$c^2 = 5$$

$$c = \sqrt{5}$$

Proses selanjutnya mencari nilai *e*

$$e = \frac{\sqrt{5}}{5}$$

$$e = 0.44$$

Eccentricity memiliki jarak nilai antara 0 hingga 1. Objek yang berbentuk memanjang atau menyerupai bentuk garis lurus, nilai *eccentricity*-nya adalah, sedangkan nilai yang berbentuk lingkaran, nilai *eccentricity*-nya bernilai 0. Maka objek dengan perhitungan *eccentricity* adalah berbentuk lingkaran.

Menyimpan Hasil Pada Database

Hasil deteksi lingkaran yang dibuat sebagai penanda. Dalam penelitian ini menyesuaikan objek lingkaran yang dibuat maka objek lingkaran yang dianggap sebagai deteksi parkir kosong adalah dengan nilai *eccentricity*-nya ≥ 0.4 dan area ≥ 2000 pixel. Hal ini digunakan agar tidak mendeteksi lingkaran yang tidak diinginkan seperti noise saat pengambilan video real-time. Hasil deteksi disimpan dalam database yang auto update setiap ada perubahan kondisi parkir.

Menampilkan Informasi Parkir

Pada tahap ini hanya menampilkan informasi keadaan parkir kosong atau telah terisi penuh. Data yang telah tersimpan ditampilkan melalui aplikasi android atau browser pada halaman parkir.say-community.com. Aplikasi atau halaman tersebut hanya dapat diakses untuk menampilkan informasi dengan tepat hanya jika proses di ekstraksi ciri diaktifkan.

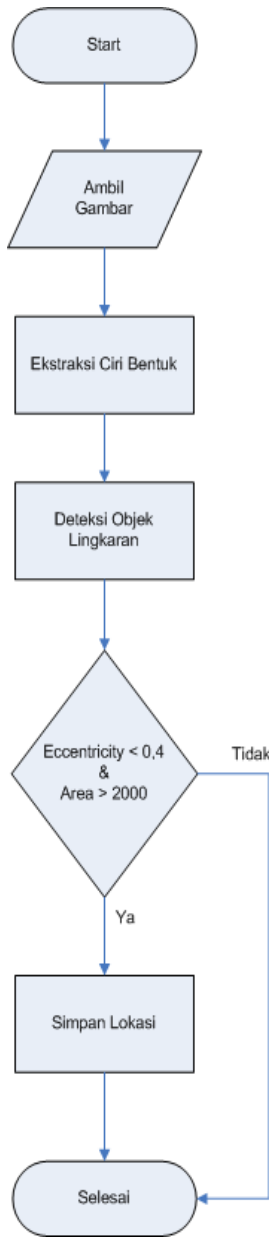
HASIL DAN PEMBAHASAN

Flowchart Ekstraksi Ciri Bentuk

Ekstraksi ciri bentuk untuk mengenali objek lingkaran pada lokasi parkir terdapat dua tahap. Antara lain:

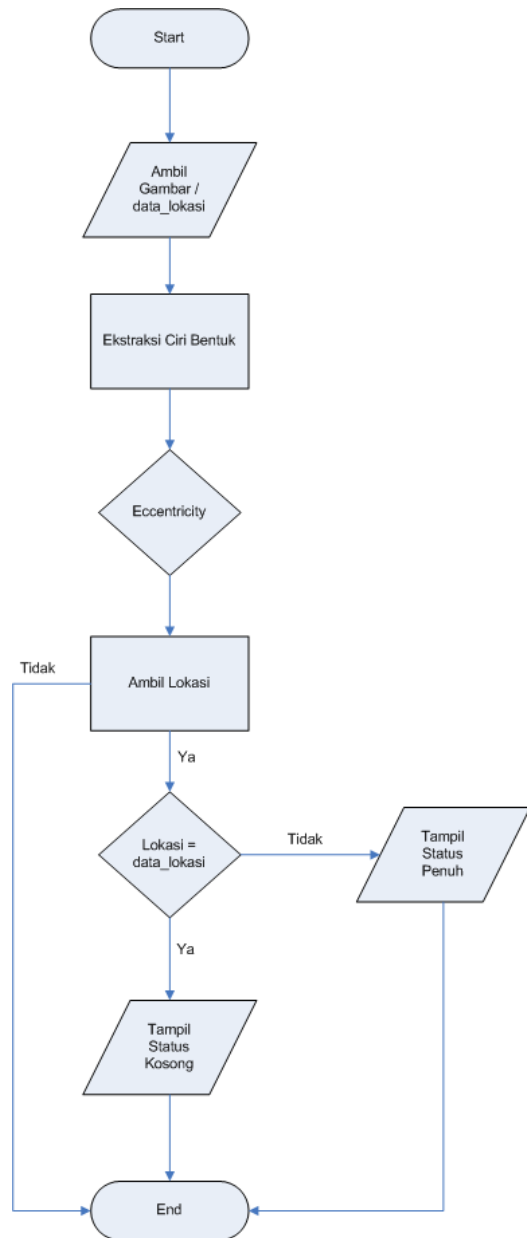
1. Inisialisasi Parkir
 Inisialisasi digunakan untuk mencari objek lingkaran sehingga didapatkan posisi lingkaran dalam area pixel. Dalam hal ini untuk mendapatkan 10 titik lokasi parkir. Objek lingkaran dengan *eccentricity*-nya lebih kecil 0.4 dan besar area ≥ 2000 pixel ditandai dan disimpan sebagai titik lokasi parkir yang dapat diisi dengan kendaraan roda 4. Pemilihan area ≥ 2000 ini berdasarkan objek lingkaran yang dibuat sebagai

penanda dan menghilangkan lingkaran yang terdeteksi lebih kecil yang mungkin terjadi bila ada noise. Berikut gambar proses inialisasi seperti terlihat pada gambar 6.



Gambar 6 Proses Inialisasi

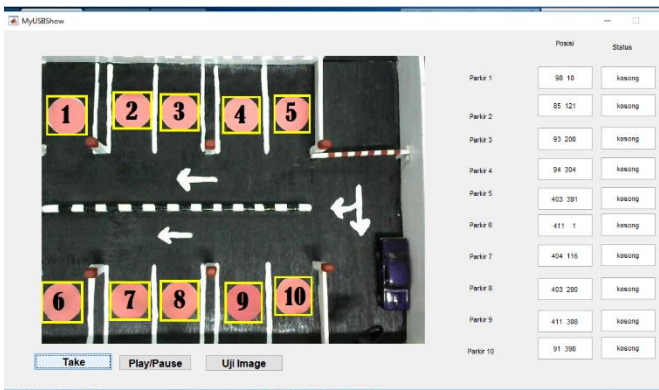
2. Uji/Proses Deteksi parkir kosong atau penuh
 Pada tahap ini dilakukan hal yang sama pada saat inialisasi, setelah dapat lokasi kordinat kemudian dibandingkan dengan lokasi saat inialisasi. Jika terdapat kordinat yang hilang saat proses ini berarti ditemukan bahwa terdapat parkir yang sudah penuh atau terisi. Namun jika jumlah 10 kordinat tetap sama maka parkir benar-benar masi kosong. Berikut gambar flowcart ekstraksi ciri bentuk untuk lingkaran seperti terlihat pada gambar 7.



Gambar 7 Pengujian Lokasi Parkir

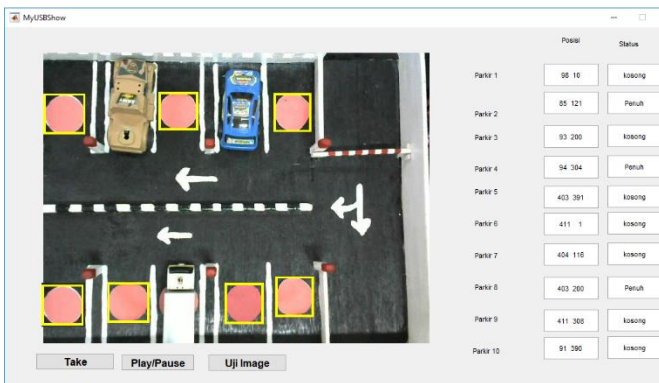
Implementasi Sistem

Simulasi parkir dilakukan dengan dua tahap yaitu tahap inialisasi dan tahap pengujian. Tahap inialisasi untuk mendeteksi lokasi parkir. Kamera diatur agar terlihat seluruh lokasi parkir yang telah diberi tanda lingkaran. Dengan metode ekstraksi ciri objek dideteksi posisi lingkaran pada pixel. Pada gambar 4.5 dapat dilihat lokasi parkir terdapat 10 titik lokasi yang ditandai dengan urutan 1 sampai 10 dimulai dari kiri atas hingga kanan bawah secara berurutan. Posisi parkir 1 terdapat pada posisi monitor pada pixel x=95 dan y=10, parkir 2 terdapat pada posisi x=85 dan y=121 dan seterusnya dengan setatus awal diinialisasi bahwa parkir dalam keadaan kosong. Seperti terlihat pada gambar 8 berikut:



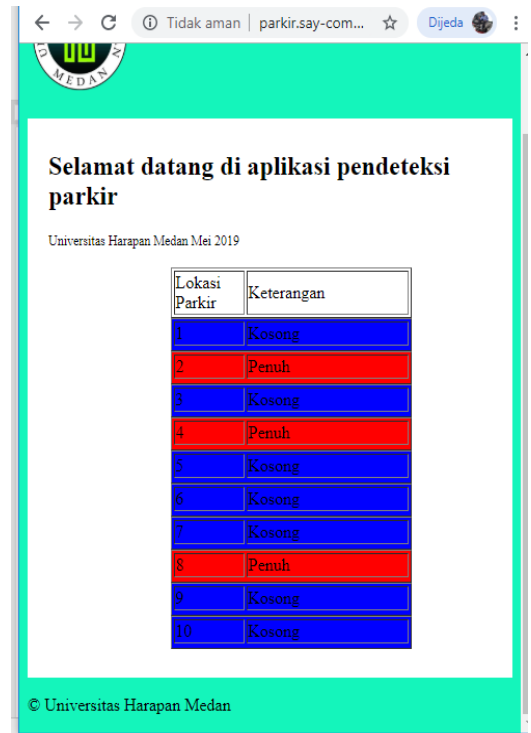
Gambar 8 Penerapan Inisialisai Parkir

Setelah diinisialisasi kemudian tekan “play” untuk menjalankan kamera secara realtime untuk dapat dilihat keadaan parkir sesaca realtime pula, atau menggunakan tombol “Uji Image” untuk mengcapture satu buah gambar yang mempunyai fungsi yang sama dalam mendeteksi keberadaan parkir yang terisi atau kosong. Saat tombol “play” di klik maka video realtime hasil dari kamera dibaca tiap frame-nya. Kemudian dibandingkan dengan posisi saat tahap inisialisasi. Berikut hasil deteksi parkir kosong dapat dilihat pada gambar 9.



Gambar 9 Deteksi Parkir Kosong

Hasil deteksi parkir kosong ini kemudian disimpan kedalam sebuah database, yang akan auto update selama aplikasi parkir dijalankan. User dapat melihat keadaan atau informasi parkir melalui halaman parkir.say-community.com seperti terlihat pada gambar 10 berikut:



Gambar 10 Tampilan Informasi parkir kosong

Pengujian

Pengujian pada awalnya dilakukan dengan menggunakan simulator matlab dan raspberry Pi. Namun pada saat implementasi mengakibatkan aplikasi error karena tidak mampu memproses data realtime yang diambil dari kamera. Oleh karena itu pengujian dilakukan menggunakan Laptop dengan CPU intel core i5 dan Memori RAM 4GB. Hasil pengujian dengan meletakkan mobil pada lokasi tiap parkir dapat dikenali dengan baik seperti tabel 1 berikut.

Tabel 1. Uji Parkir saat satu lokasi parkir diisi

No	Posisi Parkir yang Terisi	Keterangan
1	Parkir 1	Dikenali/Sukses
2	Parkir 2	Dikenali/Sukses
3	Parkir 3	Dikenali/Sukses
4	Parkir 4	Dikenali/Sukses
5	Parkir 5	Dikenali/Sukses
6	Parkir 6	Dikenali/Sukses
7	Parkir 7	Dikenali/Sukses
8	Parkir 8	Dikenali/Sukses
9	Parkir 9	Dikenali/Sukses
10	Parkir 10	Dikenali/Sukses

Kemudian dilakukan pengujian dengan jumlah mobil yang bervariasi saat parkir di lokasi parkir. Seluruh pola pengujian dapat dikenali dengan baik.

KESIMPULAN DAN SARAN

Hasil dari perancangan dan implementasi pembahasan sebelumnya maka dapat disimpulkan antara lain:

1. Objek lingkaran untuk penanda parkir kosong dapat dideteksi dengan baik sehingga dapat mengenali keadaan parkir kosong atau sedang terisi.
2. Simulasi parkir dapat berjalan dengan baik menggunakan komputer core i5 dan RAM 4 GB.
3. Pengguna dapat melihat keadaan parkir melalui aplikasi browser atau android sehingga dapat menentukan lokasi tujuan saat memasuki lokasi parkir.
4. Menghemat waktu pencarian parkir bagi pengguna.
5. Menggunakan raspberry Pi dengan simulator matlab gagal dilakukan dikarenakan proses yang begitu besar secara realtime.

Adapun saran untuk pengembangan berikutnya adalah:

1. Diharapkan aplikasi nantinya dilengkapi dengan pemesanan lokasi parkir saat pengguna berada didekat lokasi. Pada saat parkir dipesan maka pintu atau palang hanya dapat dibuka oleh pemesan.
2. Melakukan percobaan ulang agar dapat diproses dengan raspberry pi dengan Bahasa pemrograman yang lebih ringan penggunaan memori.

DAFTAR PUSTAKA

- [1] S. Rahman, R. Siregar, F. damayanti dan M. Meilani, "Deteksi Lokasi Parkir Kosong Pada Video Real-Time Menggunakan Metode Background Subtraction," *Prosiding Seminar Nasional Era Industri (SNEI) 4.0*, vol. 1, no. 1, pp. 328-335, 2018.
- [2] H. Singh, C. Anand, V. Kumar dan A. Sharma, "Automated Parking System With Bluetooth Access," *International Journal Of Engineering And*, pp. 5773-5775, 2014.
- [3] U. Athiyah, I. Muhimmah dan E. Marfianti, "Ekstraksi Ciri Polip dan Pendarahan Berdasarkan Citra Endoskopi Kolorektal," *Jurnal Informatika: Jurnal Pengembangan IT Poltek Tegal*, p. 81-85, 2018.
- [4] _, *Pedoman Perencanaan dan Pengoperasian Fasilitas Parkir*, Jakarta: Direktorat Jendral Perhubungan Darat, 1998.
- [5] S. Rahman dan U. Sahira, "PENGENALAN IRIS MATA MENGGUNAKAN METODE TEMPLATE MATCHING CORRELATION," *JURNAL TEKNOLOGI INFORMASI*, pp. 95-102, 2019.
- [6] A. P. N, T. Sutojo dan Muljono, *Pengolahan Citra Digital*, Yogyakarta, 2017.
- [7] T. M. Diansyah dan E. Ilyanda, "Rancangan Media Penyimpanan Berbasis Mikrokontroler Menggunakan Raspberry Pi sebagai Mini Server Portabel," *Jurnal Teknologi dan Ilmu Komputer Prima (JUTIKOMP)*, pp. 123-128, 2018.
- [8] A. Z. Hasibuan, M. . S. Asih dan H. Harahap, "Penerapan QR Code dan Vigenere Cipher Dalam Sistem Pelaporan Juru Parkir Ilegal," *Query: Journal of Information Systems*, 2019.