

Pendeteksian Ruang Kosong Parkir di dalam Ruangan

Nunik Afriliana¹, Rosalina², Regina Valeria³

¹ Fakultas Teknik & Informatika: Informatika, Universitas Multimedia Nusantara, Tangerang, Indonesia

^{2,3} Faculty of Computing, President University, Jababeka Education Park, Cikarang Baru, Bekasi

nunik@umn.ac.id

rosalina@president.ac.id

reginavaleria96@gmail.com

Diterima 6 Mei 2018

Disetujui 25 Juni 2018

Abstrak— Menemukan tempat parkir kosong di tempat parkir dalam ruangan seperti pusat perbelanjaan menjadi kesulitan banyak pengemudi, terutama saat jam sibuk di kota-kota besar. Dalam makalah ini, sebuah sistem deteksi kekosongan tempat parkir dalam ruangan diusulkan, dengan menggunakan sistem kamera yang melibatkan OpenCV untuk mempercepat waktu dalam mencari tempat parkir bagi pengemudi kendaraan dengan memberi mereka informasi lokasi dan tempat parkir. Sistem ini menggunakan metode deteksi objek statis, yaitu *Haar-Like Cascade Classifier* yang dikombinasikan dengan *Hough Line Detection* untuk mengidentifikasi area parkir kosong dari gambar parkir yang diambil secara *real time* melalui kamera IP atau kamera USB. Sistem ini dirancang untuk disematkan dengan sistem manajemen parkir sebuah bangunan sebagai alat yang menyediakan tempat parkir untuk membantu pengemudi kendaraan memasuki area parkir.

Index Terms—*Haar-Like Cascade Classifier, Hough Line Detection, Sistem Manajemen Parkir*

I. PENDAHULUAN

Meningkatnya penambahan volume kendaraan berdampak pada kebutuhan akan sarana parkir. Banyak pengemudi menghadapi kesulitan dalam menemukan tempat parkir yang tersedia pada saat berkunjung ke suatu tempat. Bahkan ketika hampir semua pusat perbelanjaan besar dan tempat-tempat umum lainnya memiliki gedung parkir bertingkat tersendiri, semakin sulit untuk menemukan tempat parkir yang tersedia terutama pada jam sibuk dimana bangunan parkir dalam ruangan hampir selalu penuh. Sebagian besar waktu, pengemudi harus mengitari bangunan parkir mobil untuk menemukan tempat kosong untuk memarkirkan mobil mereka.

Informasi mengenai lahan kosong parkir baik di pusat perbelanjaan maupun gedung perkantoran masih sangat minim, dimana umumnya sistem parkir yang digunakan masih manual.

Sistem deteksi kekosongan tempat parkir yang sudah ada menggunakan sensor ultrasonik dengan indikator LED terpasang di atas setiap lot di area

parkir keseluruhan. Jumlah besar perangkat keras yang dibutuhkan cukup mahal dan pengguna masih harus mengelilingi tempat parkir sambil mencoba melihat indikator tempat kosong. Pendekatan alternatif dalam menemukan tempat parkir kosong menggunakan kamera pengawas dengan algoritma penglihatan komputer untuk mendeteksi apakah terdapat mobil di area parkir kosong. Selanjutnya, sistem tersebut memberitahukan tempat mana yang tersedia dan menginformasikannya ke pengendara yang masuk. Fokus utamanya adalah menghemat banyak waktu dan usaha dengan memberikan informasi tentang ketersediaan tempat parkir.

Menemukan ruang kosong untuk memarkir mobil di area parkir dalam ruangan sering menjadi masalah, terdapat beberapa penelitian tentang teknik penentuan posisi mobil seperti yang tercantum dalam referensi [2,3,4,5,6,7,8, 11].

Liu [3] mengusulkan navigasi dalam ruangan berdasarkan smartphone, sistem ini dapat menemukan tempat parkir gratis dengan menerima data dari sensor yang terletak di dalam tempat parkir mobil.

Deteksi parkir kosong berbasis video disajikan di [4], proyek ini dievaluasi oleh beberapa kombinasi ekstraktor fitur dan algoritma pembelajaran mesin. Sementara Ichihashi et al [2] mengusulkan pendekatan untuk mengklasifikasikan ruang parkir kosong tunggal berdasarkan logika fuzzy.

Tulisan ini bertujuan untuk membangun dan menerapkan sistem yang akan membantu mengurangi waktu yang dibutuhkan oleh pengemudi dalam mencari tempat parkir kosong dan menghindari perjalanan yang tidak perlu melalui tempat parkir yang penuh.

Sistem ini dirancang untuk memberi informasi tentang lokasi yang tepat pada sebuah tempat parkir kosong. Lokasi harus diperoleh dengan sistem cerdas penglihatan komputer dan informasinya akan diarahkan ke pengemudi. Penelitian ini bertujuan untuk membuat dan menerapkan sistem yang akan membantu mengurangi waktu yang dibutuhkan

pengemudi dalam mencari tempat parkir kosong dan menghindari perjalanan yang tidak perlu melalui tempat parkir yang penuh.

Tujuan dari penelitian ini adalah untuk membuat sebuah aplikasi yang mengimplementasikan algoritma untuk melakukan fitur ekstraksi dan menggunakan *cascade classifier* untuk membangun deteksi objek. Hal ini dilakukan dengan tujuan membuat aplikasi *web* yang bisa menemukan tempat parkir kosong di suatu daerah.

Agar aplikasi ini bisa berfungsi dengan baik, lingkungan harus memenuhi persyaratan area parkir standar yang terdiri dari sederet slot parkir linier dengan garis yang memisahkan masing-masing slot.

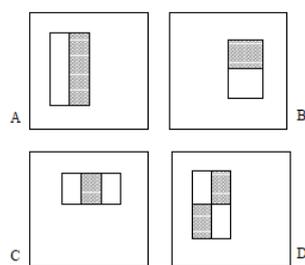
Keterbatasan lainnya adalah:

1. Keakuratan deteksi objek tergantung pada kualitas kamera.
2. Aplikasi ini hanya memproses gambar dari kamera statis sehingga area jangkauannya terbatas.
3. Implementasi deteksi objek hanya akan bekerja untuk mobil.

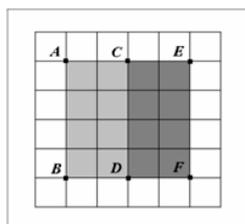
II. STUDI PUSTAKA

2.1 Haar-Like Feature

Pada tahun 2001, Paul Viola dan Michael Jones mempublikasikan pendekatan deteksi wajah yang melibatkan fitur mirip Haar [9]. Fitur mirip-Haar terdiri dari daerah persegi panjang yang berdekatan di lokasi tertentu di jendela deteksi, di mana persegi panjang mewakili subbagian objek. Intinya adalah mengidentifikasi objek berdasarkan nilai fitur dan bukan pada nilai piksel individual dari objek gambar, sehingga menghasilkan eksekusi yang lebih cepat.



Gambar 1. Contoh fitur persegi panjang yang ditampilkan relatif terhadap jendela pendeteksian terlampir [8].



Gambar 2. Perhitungan nilai fitur Haar

Nilai Fitur (ABFE) = jumlah piksel di bawah persegi (CDFE) - jumlah piksel di bawah persegi (ABCD) (1)

Gambar 2. menunjukkan contoh fitur *Haar-like*. Penghitungan nilai fitur dilakukan menggunakan rumus (1), di mana jumlah piksel di bawah area gelap dikurangi dengan jumlah piksel di area terang. Jika nilai fitur ABFE berada di atas ambang batas (disetel selama pembelajaran), maka area di bawahnya dikatakan memenuhi persyaratan fitur *Haar*.

2.2 Canny Edge Detection

Deteksi tepi dilakukan umumnya untuk secara signifikan mengurangi jumlah data dalam suatu gambar, sambil tetap menjaga properti struktural untuk pemrosesan gambar lebih lanjut. Deteksi Tepi Canny dikembangkan oleh John F. Canny pada tahun 1986, dan telah menjadi salah satu metode deteksi tepi standar yang digunakan dalam penelitian [11].

Algoritma berjalan dalam 5 langkah terpisah:

- Smoothing: Terapkan filter Gaussian ke gambar untuk menghilangkan noise.
- Mencari gradien: Tepi-tepinya harus ditandai di mana gradien gambar memiliki besaran besar.
- Penekanan yang tidak maksimal: Konversi tepi yang kabur dari gradien menjadi tajam.
- Ambang ganda: Bedakan antara tepi benar yang potensial dan piksel tepi lainnya yang disebabkan oleh suara atau variasi warna.
- Pelacakan tepi dengan histeresis: Tentukan tepi akhir dengan menekan semua sisi yang tidak terhubung ke tepi (kuat) tertentu.

2.3 Hough Line Transform

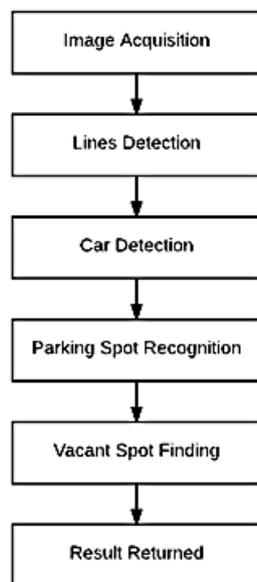
Hough Line Transform adalah algoritma untuk menemukan titik-titik yang sejajar dalam gambar yang menciptakan garis lurus. Suatu garis dapat direpresentasikan sebagai $\rho = x \cos \theta + y \sin \theta$ di mana ρ adalah jarak dari asal ke garis, dan θ adalah sudut yang dibentuk oleh garis jarak dan sumbu horizontal ini. Dalam mencari garis, *Hough Transform* berfungsi dengan menghitung semua kemungkinan kombinasi parameter untuk setiap piksel yang merupakan bagian dari garis. Setiap garis yang mungkin dari *pixel* pertama diwakili dengan kurva, dan titik dengan kurva persimpangan paling mengidentifikasi garis imajiner dalam gambar aslinya.

III. METODOLOGI PENELITIAN

Proses pencari tempat parkir kosong terdiri dari beberapa urutan langkah seperti yang ditunjukkan pada Gambar 3.

Pertama, program akan mengaktifkan kamera, menangkap gambar dan menyimpannya ke folder sumber daya di server. Gambar kemudian akan dimuat ke dalam program sebagai tipe data Matriks.

Kedua, sistem akan melakukan deteksi objek untuk menemukan lokasi garis-garis pada gambar, yang akan menunjukkan partisi area parkir. Bagian pencarian baris terdiri dari beberapa langkah, yang mengkonversi ke *grayscale*, *blurring*, deteksi tepi, dan operasi *HoughLine*. Konversi warna dari RGB ke abu-abu dilakukan untuk menormalkan nilai warna di setiap piksel. *Gaussian blur* digunakan untuk menghaluskan objek gambar. Citra biner dihasilkan oleh deteksi tepi *Canny*, di mana *HoughLine* di OpenCV dapat diterapkan untuk mendeteksi garis.



Gambar 3. Urutan proses yang dilakukan oleh sistem

Ketiga, sistem akan melakukan deteksi objek pada gambar, untuk menemukan lokasi objek mobil di dalamnya. Untuk deteksi objek, program menggunakan *Haar Cascade Classifier* dengan OpenCV. File pengklasifikasi kaskade yang digunakan dalam makalah ini dibuat oleh Andrews Sobral [1] yang diterbitkan di GitHub. *Dataset* dan program Python untuk pelatihan *Haar-Cascade* termasuk dalam repositori publik. Berdasarkan dokumentasi resminya, *Haar-cascade "cars.xml"* dilatih menggunakan 526 gambar mobil dengan ukuran 360 x 240 piksel. Gambar-gambar itu diambil dari *dataset* Mobil yang diusulkan oleh Brad Philip dan Paul Updike yang diambil dari jalan raya di California selatan [1].



Gambar 4. Dataset Mobil

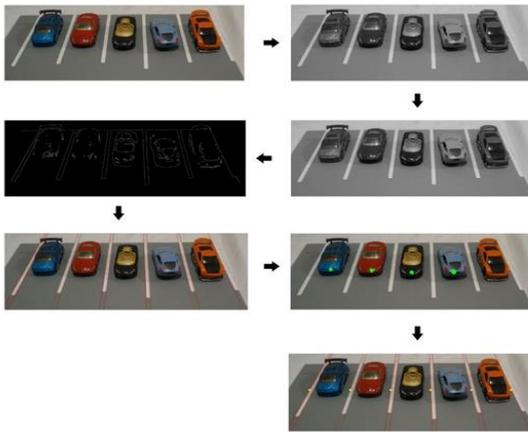
Dalam sistem ini, deteksi garis dilakukan terlebih dahulu, dan deteksi mobil datang setelah itu. Namun, kedua langkah ini dapat dipertukarkan, yang berarti bahwa urutan eksekusi tidak akan mempengaruhi hasilnya, selama hasilnya disimpan dalam variabel masing-masing.

Deteksi garis belum secara sempurna mendeteksi satu garis yang sebenarnya di dalam gambar oleh karena itu hasilnya tidak dapat langsung digunakan dan dihitung sebagai representasi sebenarnya dari garis parkir. Trik untuk menyelesaikan masalah ini adalah dengan memvalidasi garis untuk memperkirakan jarak minimum setiap baris sehingga dapat dianggap sebagai satu garis perwakilan. Dengan demikian, langkah keempat adalah pengenalan tempat parkir, di mana program akan melakukan validasi garis dan mengidentifikasi jumlah slot yang terdeteksi.

Pada langkah kelima, program akan membandingkan lokasi setiap mobil yang terdeteksi dengan slot untuk menemukan semua tempat kosong. Terakhir, sistem akan menampilkan hasilnya kepada pengguna.

Program ini memiliki satu servlet. Java servlet adalah program Java yang memperluas kemampuan server. Servlet bekerja dengan menangani permintaan HTTP dan respons dari browser dan memprosesnya. Dalam metode *doGet*, servlet pada dasarnya menerima permintaan, melakukan pekerjaan yang diperlukan, dan kemudian meneruskan permintaan ke halaman tujuan untuk pengguna.

Perpustakaan OpenCV tersedia di Java, didistribusikan dalam file JAR dan ekstensi .dll nya. Untuk dapat menggunakan perpustakaan, file OpenCV harus tersedia di mesin server dan jalur perlu ditentukan dalam IDE. Visualisasi proses ditunjukkan pada Gambar 5.

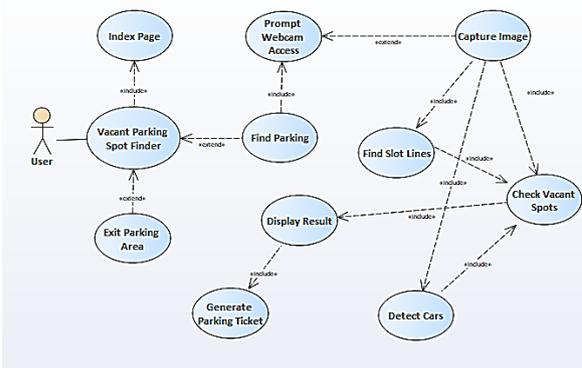


Gambar 5. Alur Proses dari Menemukan Ruang Parkir Kosong

IV. HASIL PENELITIAN

Sistem ini akan melakukan deteksi area dan deteksi objek untuk menemukan contoh objek dari kelas "mobil" dan untuk menemukan di mana ruang parkir kosong dan tersedia.

Ada beberapa fungsi Vacant Parking Spot Finder: (1) pengguna dapat menangkap gambar area parkir, (2) memperoleh data tentang berapa banyak mobil yang menempati area parkir, (3) mendapatkan informasi tempat parkir kosong, dan (4) menghasilkan tiket parkir. Diagram *use-case* dari sistem ini ditunjukkan pada Gambar 6.



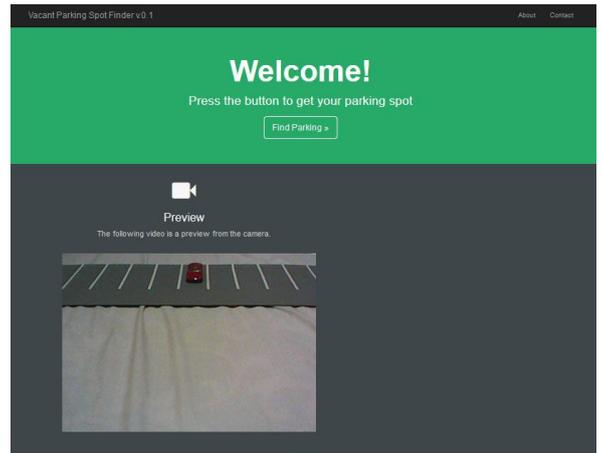
Gambar 6. Diagram Use-Case Vacant Parking Spot Finder

1. Inisiasi Pencarian Lahan Parkir

Ketika pengguna mengklik "Cari Parkir" di tombol menu, sistem akan memeriksa ketersediaan *webcam*. Jika *webcam* tidak tersedia, maka akan ada pemberitahuan kepada pengguna dan tombol untuk menangkap menjadi dinonaktifkan. Jika *webcam* tersedia, *webcam* akan diaktifkan dan ada kotak yang akan menampilkan video langsung dari *webcam*.

Inisialisasi aplikasi juga termasuk pengembangan file pengkodean kaskade Haar seperti yang akan

digunakan untuk deteksi. Fitur seperti Haar [8] adalah filter tepi yang dihitung menggunakan gambar integral.



Gambar 7. Halaman Utama Vacant Parking Spot Finder

2. Persiapan Pencarian Lahan Parkir

Aplikasi ini memiliki beberapa persiapan sebelum sistem mulai melakukan deteksi mobil untuk sistem parkir. Sebelum aplikasi melakukan deteksi mobil, pengguna harus menekan tombol untuk menangkap gambar area parkir. Setelah itu, beberapa pengolahan gambar akan dilakukan untuk mengidentifikasi area parkir, dimana area tersebut dibagi menjadi beberapa area yang ditandai dengan garis putih untuk memisahkan tempat parkir.

Sistem ini mengimplementasikan *Hough Lines Transform* untuk deteksi garis. Untuk menggunakan *Hough Lines Transform*, gambar yang diproses harus biner. Oleh karena itu, gambar warna asli diubah menjadi *grayscale*, sehingga setiap tepinya terdeteksi. Konversi dari gambar RGB ke abu-abu dilakukan dengan fungsi *cvtColor* OpenCV. Sedangkan deteksi tepi yang diimplementasikan dalam sistem ini adalah algoritma deteksi tepi Canny dengan OpenCV.

Untuk membiarkan program mengenali rangkaian garis yang terdeteksi sebagai kolom slot parkir, pertama-tama informasi lokasi mobil yang terdeteksi diproses dan rata-rata *y* dihitung untuk melihat di titik mana sepanjang garis yang terdeteksi dapat digunakan untuk perbandingan dengan lokasi mobil. Rata-rata dilakukan untuk menemukan nilai optimal dari koordinat *y* di mana kolom slot parkir dan mobil berada.

```
public static int findAverageY(List<Point> cars) {
    int sum = 0;
    for (int i = 0; i < cars.size(); i++) {
        sum += cars.get(i).y;
    }
    int avg = sum / cars.size();
    return avg;
}
```

Gambar 8. Fungsi Mencari Rata-Rata Y

Berdasarkan nilai y yang diperoleh dari `findAverageY`, program akan melintasi vektor yang berisi informasi (r, θ) dari garis yang terdeteksi untuk menghitung titik akhir berdasarkan informasi tersebut. Setelah titik akhir dari garis ditemukan, `findXifYKnown` digunakan untuk menentukan satu titik di sepanjang garis di mana koordinat y sama dengan rata-rata y yang diperoleh dari `findAverageY`.

```
public static int findXifYKnown(int y, Point start, Point
end) {
    int x = (int) (start.x + ((y - start.y) * (end.x -
start.x) / (end.y - start.y)));
    return x;
}
```

Gambar 9. Fungsi Mencari X jika Y diketahui

Untuk setiap baris memiliki dua set titik akhir Titik A (x_A, y_A) dan Titik B (x_B, y_B) , adalah mungkin untuk menentukan Titik C baru di suatu tempat antara A dan B yang memenuhi $y = y_{Avg}$.

$$(y_B - y_A) / (x_B - x_A) = (y_C - y_A) / (x_C - x_A) \quad (1.1)$$

sehingga,

$$x_C = x_A + ((y_C - y_A) * (x_B - x_A) / (y_B - y_A)) \quad (1.2)$$

ArrayList of Point pertama diurutkan berdasarkan koordinat poin x , karena langkah selanjutnya akan membutuhkan daftar poin yang akan disortir mulai dari paling kiri ke kanan.

Dapat diamati bahwa hasil deteksi ini tidak sempurna. Diperlukan untuk menghilangkan kelebihan hasil menggunakan `validateLine`. Validasi baris dilakukan dengan memeriksa jarak dari satu baris yang terdeteksi ke baris berikutnya. Jika celah antara garis dan baris berikutnya ke kiri lebih besar dari minimum yang diperlukan, garis dianggap sah, dan disimpan ke dalam daftar `GarisLini` valid.

3. Proses Pencarian Lahan Parkir

Ada beberapa proses menemukan tempat parkir yang kosong saat melakukan deteksi objek. Aplikasi ini akan membaca *file* yang dilatih dengan *cascade*, dan menggunakannya dalam memindai gambar dan menemukan contoh mobil di dalamnya. Untuk setiap tempat parkir, kemunculan mobil akan menunjukkan bahwa tempat tersebut ditempati. Setelah semua tempat parkir diperiksa, aplikasi akan menemukan tempat kosong dan memberikan hasilnya kepada pengguna.

Ada beberapa aturan yang harus diikuti pengguna untuk mendapatkan hasil terbaik dalam proses deteksi mobil:

- Area parkir harus memiliki pencahayaan yang cukup untuk mendapatkan gambar yang layak.
- Mobil-mobil tidak boleh menutupi garis perbatasan yang membagi area parkir.

Untuk mendeteksi mobil, sistem ini menggunakan OpenCV. OpenCV dilengkapi dengan detektor *Haar-*

cascade yang dapat digunakan untuk menemukan objek dalam gambar. Fungsi `detectCars` menerima jalur gambar, memuat file pengklasifikasi kaskade yang telah ditentukan, dan memanggil fungsi `detectMultiScale`.

Fungsi `detectMultiScale` dari OpenCV mengembalikan matriks persegi panjang yang dapat diakses sebagai larik untuk mendapatkan koordinat sudut kotak pembatas yang berisi objek yang terdeteksi. Namun, ketika benar-benar ditarik, kotak pembatas biasanya terlihat lebih besar dari ukuran sebenarnya dari mobil, dan itu tidak diinginkan untuk proses selanjutnya untuk memeriksa slot parkir, karena koordinat mobil dan garis harus dibandingkan. Oleh karena itu, `findCenter` digunakan untuk menentukan koordinat pusat mobil. Pusat mobil ditunjukkan dengan warna hijau seperti yang ditunjukkan pada Gambar 9.

Pencari tempat parkir kosong yang diusulkan dalam makalah ini telah diuji. Ada tiga faktor utama untuk mengukur akurasi deteksi parkir saat melakukan pengujian. Pertama adalah kamera *web*, kamera harus mampu menghasilkan gambar yang layak, dengan resolusi setidaknya 640x480 piksel. Kamera *web* harus ditempatkan di lokasi sedemikian rupa sehingga dapat menangkap objek yang harus dimasukkan, yang dapat bervariasi sesuai dengan panjang fokus kamera.

Dalam pengujian ini kamera ditempatkan sekitar 1 meter dari objek. Kedua adalah cahaya, intensitas cahaya sangat mempengaruhi keakuratan pendeteksian, karena intensitas cahaya akan sangat mempengaruhi gambar yang diambil. Evaluasi mencakup beberapa variasi kondisi pencahayaan yang mungkin dari area parkir. Ketiga adalah objek, yaitu area parkir dan mobil, area parkir harus memiliki slot dibagi dengan garis, dan mobil tidak boleh mencakup lebih dari setengah dari garis.

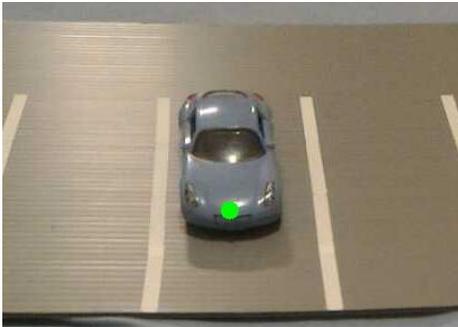
Kasus uji untuk pengujian ketahanan sistem ini didasarkan pada beberapa ruang lingkup dan batasan untuk mengevaluasi kinerja aplikasi. Sistem ini tergantung pada lingkungan sekitarnya, terutama pencahayaan dan kamera. Dengan demikian, sistem ini juga akan dievaluasi oleh kondisi eksternal seperti intensitas cahaya yang berbeda dan jumlah slot parkir yang terlibat, yang akan dijelaskan pada Tabel 1.

Tabel 1. Batasan Pengujian

No.	Batasan	Keterangan	Hasil
1	Cahaya	Terlalu terang atau terlalu gelap	Garis parkir tidak terdeteksi dengan jelas, mobil tidak terdeteksi
2	Jarak	Jarak mobil terlalu dekat dan terlalu jauh dari webcam	Mobil tidak terdeteksi

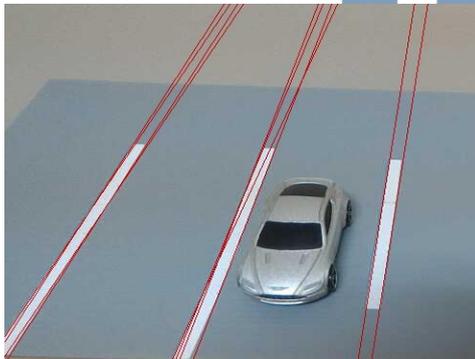
3	Objek selain mobil	Objek lain berada di lahan parkir	Program masih mendeteksi bahwa lahan parkir kosong
---	--------------------	-----------------------------------	--

Hasil simulasi deteksi mobil dapat dilihat pada Gambar 9. Titik hijau menunjukkan pusat area yang terdeteksi sebagai mobil.



Gambar 10. Hasil Deteksi Mobil

Tingkat akurasi deteksi mobil pada foto yang diambil dengan IP Camera (920px x 720px) lebih tinggi daripada tingkat akurasi deteksi mobil pada foto yang diambil dengan kamera web VGA (640px x 360px).



Gambar 11. Hasil Deteksi Garis

Hasil dari simulasi deteksi garis dapat dilihat pada Gambar 10. Garis merah menunjukkan elemen-elemen gambar yang terdeteksi sebagai garis.

Hasil akurasi pengujian sistem terlihat pada Tabel 2, dimana 72,7% sistem dapat mengenali ruang parkir kosong dan 27,3% sistem gagal mengenai ruang parkir kosong. Kegagalan sistem ini salah satunya dikarenakan kualitas kamera, kualitas cahaya dan jarak antar mobil.

Tabel 2. Akurasi Sistem

Jumlah Percobaan	Jumlah Deteksi	
	Benar	Salah
63	62.7%	37.3%

V. KESIMPULAN

Penelitian ini bertujuan untuk menyediakan sistem untuk mendeteksi mobil di tempat parkir tertentu dan secara otomatis mengenali ruang sebagai deretan slot parkir, oleh karena itu tempat parkir kosong dapat ditemukan dengan tepat. Tujuan utama dari penelitian ini telah berhasil diraih. Pengguna dapat menggunakan proyek ini untuk menemukan tempat parkir kosong tanpa benar-benar memiliki informasi visual dari area parkir.

Metode deteksi mobil yang digunakan dalam makalah ini adalah dengan menggunakan *Haar-Like Cascade Classifier* dengan detektor *built-in* dari OpenCV. Deteksi objek bekerja dengan *file cars.xml* oleh Andrews Sobral dari GitHub. Dapat disimpulkan bahwa *file classifier* cukup terlatih dan oleh karena itu dapat digunakan untuk deteksi mobil.

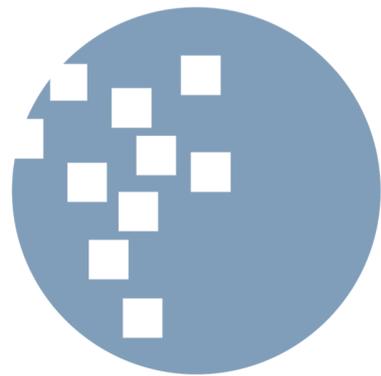
Kualitas gambar sangat mempengaruhi akurasi detektor objek; Oleh karena itu, kualitas kamera, kondisi objek dan lingkungan harus diperhatikan. Untuk tingkat tertentu, aplikasi ini kuat terhadap perubahan ukuran mobil dalam gambar, yang merupakan kasus ketika jarak kamera ke objek berubah.

Di masa depan, Deteksi Tempat Parkir Kosong akan memiliki peningkatan antarmuka pengguna, peningkatan deteksi mobil menggunakan beberapa file pengklasifikasi kaskade, deteksi objek waktu nyata, dan kemampuan untuk memproses input dari beberapa kamera untuk menutupi area yang lebih besar dan / atau menambah kemampuan gerakan ke kamera.

DAFTAR PUSTAKA

- [1] A. Sobral, "GitHub," 3 December 2016. [Online]. Available: https://github.com/andrewsobral/vehicle_detection_haarascades. [Accessed 17 September 2017]
- [2] Ichihashi, H., Katada, T., Fujiyoshi, M., Notsu, A. and Honda, K. "Improvement in the performance of camera based vehicle detector for parking lot". Proceedings of the IEEE International Conference on Fuzzy Systems. Pages 1-7, 2010
- [3] Liu J., Chen R., Chen Y., Pei L., Chen L. iParking: An Intelligent Indoor Location-Based Smartphone Parking Service. Sensors. 2012;12:14612–14629.
- [4] Marc tschentscher and marchel neuhaue, "Video Based Parking Space Detection", Institute for Neural Computation, Ruhr Universitat Bochum. 2012.
- [5] R. Yusnita, Fariza Norbaya, and Norazwinawati Basharuddin. "Intelligent Parking Space Detection System Based on Image Processing" International Journal of Innovation, Management and Technology, Vol. 3, No. 3, June 2012.
- [6] Sayanti Banerjee, Pallavi Choudekar and M. K. Muju. "Real time car parking system using image processing," IEEE, pp. 99-103, 2011.
- [7] Shen-En Shih and Wen-Hsiang Tsai. "A Convenient Vision-Based System for Automatic Detection of Parking Spaces in Indoor Parking Lots Using Wide-Angle Cameras". IEEE Transactions on Vehicular Technology, Volume: 63, Issue: 6, July 2014.
- [8] Viola, P. and Jones, M. "Rapid object detection using a boosted cascade of simple features". Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Vol. 1, Pages 511-518, 2001

-
- [9] W. N. B. W. ISMAIL, "OBJECT DETECTION SYSTEM USING HAAR-CLASSIFIER"
- [10] Y. H. H. Z. a. W. S. F. Mai, "A hierarchical approach for fast and robust ellipse extraction," Pattern Recognition, vol. 41, no. 8, pp. 2512-2524, 2008.
- [11] Z. Bin, J. Dalin, W. Fang, and W. Tingting, A design of parking space detector based on video image, The Ninth International Conference On Electronic Measurement & Instruments ICEMI 2009.



UMN