



Sistem Cerdas Penghitung Jumlah Mobil untuk Mengetahui Ketersediaan Lahan Parkir berbasis *Python* dan *YOLO v4*

Getsa Novandra Rizkatama^{1)✉}, Anan Nugroho²⁾ dan Alfa Faridh Suni²⁾

¹Jurusan Teknik Elektro, Fakultas Teknik, Universitas Negeri Malang, Indonesia

²Jurusan Teknik Elektro, Fakultas Teknik, Universitas Negeri Semarang, Indonesia

Info Artikel

Sejarah Artikel:

Diterima: Juli 2021

Direvisi: Agustus 2021

Disetujui: Desember 2021

Keywords:

Parking Space, Computer

Vision, Python, Object

Detection, YOLO v4

Abstrak

Perkembangan teknologi pada era digital semakin cepat, mulai dari bidang *hardware* maupun *software*. *Software* atau perangkat lunak sering digunakan pada sebuah sistem, baik untuk otomatisasi sistem itu sendiri, maupun digunakan untuk membantu manusia dalam kehidupan sehari-hari. Pencarian sebuah lahan parkir yang tersedia menjadi sebuah permasalahan yang dihadapi oleh manusia pada saat hendak memasuki sebuah lahan parkir. Pembayaran biaya masuk lahan parkir menjadi kendala saat lahan parkir tidak ditemukan. Maka dari itu pengembangan sebuah sistem yang digunakan sebagai pendeteksi ketersediaan lahan parkir menjadi salah satu solusinya. Sistem atau perangkat lunak dikembangkan menggunakan bahasa pemrograman *Python* dan *YOLO v4*. Ketersediaan lahan parkir dihitung dengan membandingkan hasil deteksi mobil dengan ketersediaan lahan parkir. Deteksi dilakukan dengan menggunakan algoritma *YOLO v4* dengan *dataset COCO* yang ditandai dengan *bounding box* dan *confidence level*. Berdasarkan hasil pengujian pada beberapa video, didapatkan tingkat akurasi program sebesar 72,8%.

Abstract

Technological developments in the digital era are getting faster, starting from the hardware and software fields. Software is often used in a system, both for automation of the system itself, or used to help humans in everyday life. The search for an available parking space becomes a problem faced by humans when they want to enter a parking lot. Payment of parking lot entrance fees becomes an obstacle when parking spaces are not found. Therefore, the development of a system that is used to detect the availability of parking spaces is one solution. The system or software is developed using the Python programming language and YOLO v4. The availability of parking space is calculated by comparing the results of car detection with the availability of parking spaces. Detection is carried out using the YOLO v4 algorithm with the COCO dataset marked with a bounding box and confidence level. Based on the test results on several videos, the program accuracy rate is 72.8%.

PENDAHULUAN

Perkembangan teknologi pada era digital sangatlah cepat, terutama pada bidang elektronik seperti *smartphone*, Komputer, Laptop dan lain sebagainya. Hal tersebut tak hanya berdampak pada perkembangan hidup manusia, tetapi juga perkembangan perangkat lunak yang semakin marak dikembangkan, baik untuk membuat sebuah sistem, maupun untuk membantu manusia dalam kehidupan sehari-hari.

Salah satu kegunaan sistem untuk membantu manusia adalah pada saat pemilihan lahan parkir. Sebuah lahan parkir umumnya berisi berbagai jenis kendaraan, baik yang mempunyai roda 2 (sepeda, motor) maupun roda 4 (mobil). Pemilihan lahan parkir dilakukan dengan cara mengamati ketersediaan lahan parkir tersebut, sehingga kendaraan yang digunakan dapat dengan mudah terparkir. Hal tersebut dapat dilakukan apabila kita mengetahui secara langsung lahan parkir yang kosong atau tersedia, sehingga kita harus membayar biaya masuk parkir yang belum tentu dapat digunakan.

Sebuah perangkat lunak, dapat dikembangkan dengan memanfaatkan perangkat keras yang ada, salah satunya adalah komputer atau laptop. Perangkat lunak pada komputer dapat dikembangkan dengan menggunakan berbagai macam bahasa pemrograman, salah satunya adalah bahasa pemrograman *Python*. Dimana bahasa pemrograman python merupakan bahasa pemrograman tingkat tinggi yang dibuat agar mudah dibaca dan diterapkan, serta bersifat *open source*. *Python* juga dikenal sebagai bahasa *script* yang dapat dijalankan pada berbagai sistem operasi, seperti *Mac*, *Windows*, dsb. *Python* juga dapat digunakan dalam pengembangan web, pemrograman numerik, pengembangan game, akses *port serial*, dsb yang menggunakan bahasa C, C++ (Belanja et al., 2020).

Salah satu manfaat dari pemrograman berbasis *Python* adalah sebagai sarana dalam menggabungkan antara sebuah citra dengan sistem yang dapat mendeteksi objek tertentu. Pada penelitian yang dilakukan oleh Alvin Lazaro, Joko Lianto Buliali, dan Bilqis Amaliah dikatakan bahwa perlu memperkaya fitur jenis kendaraan lain pada sebuah file (.XML) yang digunakan untuk mendeteksi objek kendaraan, sehingga program dapat mengenali objek kendaraan jenis baru di masa depan (Lazaro et al., 2017). Melalui penelitian tersebut, maka dapat dikatakan bahwa penggunaan bahasa pemrograman *Python* dapat digunakan sebagai sarana untuk mendeteksi objek, khususnya pada objek kendaraan.

Dalam pengembangan sistem atau perangkat lunak berbasis *Python* diperlukan *library* yang dapat digunakan untuk mendeteksi objek. OpenCV (*Open-Source Computer Vision*) merupakan sebuah *library* perangkat lunak *computer vision* dan *machine learning* yang bersifat *open-source*. OpenCV dibangun untuk menyediakan infrastruktur umum untuk aplikasi *computer vision* dan untuk mempercepat persepsi mesin (*machine perception*) dalam produk komersial. OpenCV berlisensi BSD (*Berkeley Software Distribution*), sehingga memudahkan bagi pelaku bisnis untuk memanfaatkan dan mengubah kode. OpenCV memiliki lebih dari 2500 algoritma optimal, yang mencakup sekumpulan algoritma *computer vision* dan *machine learning* bertipe klasik dan canggih. Algoritma ini dapat digunakan untuk mendeteksi dan mengenali wajah, mengidentifikasi objek, mengklasifikasikan tindakan manusia dalam video, melacak gerakan kamera, melacak objek yang bergerak, dan lain-lain (Gautama et al., 2016).

Deteksi objek atau biasa disebut dengan *object detection* adalah suatu proses yang digunakan untuk menentukan keberadaan objek tertentu dalam suatu citra digital. Proses deteksi dapat dilakukan dengan berbagai macam metode yang pada dasarnya melakukan pembacaan fitur-fitur dari seluruh objek pada citra input. Fitur dari objek pada citra akan dibandingkan dengan fitur dari objek referensi atau *template* lalu dibandingkan dan ditentukan apakah objek yang dideteksi termasuk objek yang ingin dideteksi atau tidak (Nagataries et al., 2012).

Salah satu teknik dalam pendeteksian objek adalah dengan menggunakan algoritma YOLO (*You Only Look Once*). YOLO merupakan *neural network* pada sebuah citra, kemudian membagi citra menjadi daerah dan memprediksi *bounding box* serta probabilitas untuk masing-masing daerah. Probabilitas untuk setiap *bounding box* kemudian dihitung untuk mengklasifikasikan sebagai objek atau bukan. YOLO dapat melakukan sebuah pengenalan objek secara *real-time* dengan kecepatan 45 *frame per second* (Karlina & Indarti, 2019).

Melalui permasalahan untuk mengetahui ketersediaan lahan parkir, peneliti tertarik untuk mengembangkan sebuah perangkat lunak atau sistem untuk mendeteksi jumlah mobil pada sebuah lahan parkir. Dalam pengembangannya, peneliti menggunakan bahasa pemrograman berbasis *Python* dengan *library* OpenCV. Untuk mendeteksi sebuah objek, peneliti menggunakan algoritma YOLO versi 4 dan dataset COCO,

dimana dapat memuat berbagai macam deteksi, sehingga deteksi mobil akan lebih akurat.

METODE PENELITIAN

Deteksi objek merupakan sebuah teknik dalam mendeteksi suatu objek atau benda melalui sebuah citra yang dapat mengenali objek tersebut. Citra yang digunakan pada penelitian ini adalah berupa sebuah video yang diambil melalui kamera *smartphone* dan kamera CCTV, sehingga objek yang dideteksi akan terus bergerak sesuai dengan durasi video yang ada.

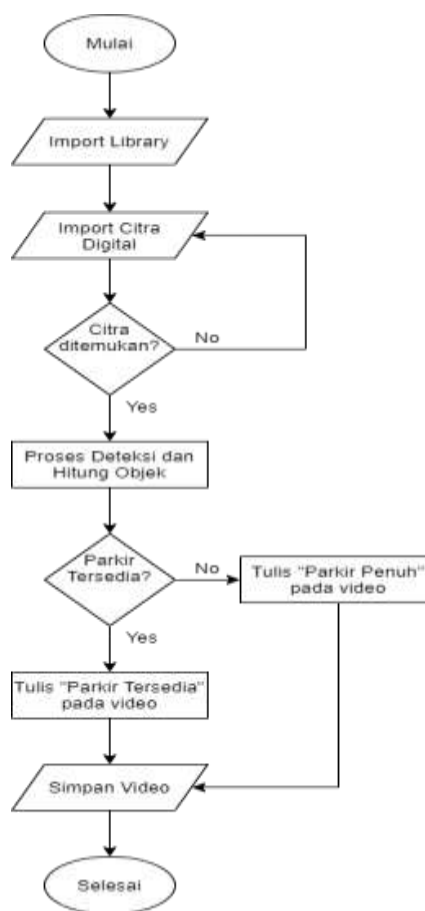
Penelitian sebelumnya dilakukan dengan menggunakan metode SSIM (*Structural Similarity Index Measurement*) dengan cara pengambilan sampel foto slot parkir yang penuh dan slot parkir yang kosong dengan kendaraan roda empat. Kemudian kedua foto tersebut dilakukan perbandingan dengan cara *threshold*. Penggunaan metode tersebut mampu menghasilkan deteksi slot parkir dengan akurasi lebih dari 60% (Mauludy et al., 2020).

Berdasarkan tingkat akurasi pada penelitian sebelumnya, peneliti bertujuan untuk meningkatkan hasil akurasi deteksi slot parkir kendaraan roda empat. Peneliti menggunakan metode YOLO v4 dengan basis COCO *datadset*. Pendeteksian menggunakan video yang telah direkam dan disimpan kembali saat deteksi telah dilakukan.

Alat dan bahan yang digunakan dalam pengembangan ini adalah sebagai berikut:

1. Kamera *Smartphone*
Smartphone yang digunakan adalah merk *xiaomi Redmi Note 8 Pro* dengan menggunakan spesifikasi CPU *Media-Tek Helio G90T*, Kamera 64 MP, f/1.9.
2. Laptop
 Laptop yang digunakan dalam pengembangan ini adalah Laptop *ASUS X455LF* dengan spesifikasi CPU *Intel Core i3 4005U*
3. Internet
 Internet sebagai sumber video yang berasal dari sebuah CCTV pada sebuah lahan parkir.

Dalam pengembangan aplikasi pendeteksi jumlah kendaraan, peneliti menggunakan beberapa tahapan inti yang tersusun dalam sebuah alur aplikasi pada Gambar 1.



Gambar 1. Alur kerja aplikasi penghitung ketersediaan lahan parkir.

Pada Gambar 1, dapat dilihat bahwa terdapat beberapa tahapan yang diperlukan dalam pengembangan aplikasi penghitung ketersediaan lahan parkir. Tahapan yang digunakan antara lain:

1. Tahap *Import Library*
 Pada tahap ini dilakukan sebuah input untuk pengembangan program, yaitu dengan meng-impor *library* OpenCV sebagai *library* utama dalam pengembangan aplikasi. Selain penggunaan *library* Open CV, peneliti juga menggunakan algoritma *Tensorflow* dan *YOLOv4*. *Tensorflow* merupakan sebuah antarmuka untuk mengekspresikan algoritma pembelajaran mesin dan untuk mengeksekusi perintah dengan menggunakan informasi yang dimiliki tentang objek tersebut, sehingga dapat membedakan objek satu dengan objek lainnya (Nurfita, 2018). *Tensorflow* memiliki fitur untuk menjalankan

pelatihan model menggunakan CPU (*Central Processing Unit*) dan pelatihan model GPU (*Graphic Processing Unit*) (Nurfita, 2018). Pada pengembangan ini, peneliti menggunakan model pelatihan CPU.

2. Tahap kedua adalah dengan melakukan input citra digital berupa sebuah video. Video yang diinputkan berjumlah 3 buah dengan durasi video pertama adalah 20 detik, video kedua 20 detik, dan video ketiga dengan durasi 10 detik. Segala video yang diinputkan mempunyai ekstensi *file* (.mp4).
3. Tahap ketiga adalah sebuah tahap kondisi untuk menentukan ketersediaan citra yang dipanggil. Terdapat dua kondisi yang dikembangkan, yaitu kondisi ada dan tidak. Apabila dalam citra terdeteksi, maka program akan menganggapnya sebagai jawaban ada, sehingga program akan melanjutkan sistem kedalam tahapan yang selanjutnya. Namun apabila citra tidak ditemukan, maka program akan menganggapnya sebagai jawaban tidak ada, sehingga pengguna akan diminta untuk memperbaiki inputan atau menginputkan video yang lain.
4. Setelah program berjalan dari tahap ketiga, maka proses selanjutnya program akan melakukan deteksi pada citra yang telah dimasukkan. Deteksi dilakukan hanya pada objek mobil, sehingga objek lain tidak akan ikut terdeteksi. Objek mobil akan dideteksi menggunakan algoritma *YOLO v4* dan *COCO dataset*. *COCO dataset* atau *MSCOCO* digunakan karena pada sebuah *COCO dataset* berisi 91 objek dengan total 2,5 juta label dalam 328.000 gambar (Nugroho & Hidayatullah, 2021). Penggunaan *YOLO v4* pada tahap deteksi, dilakukan dengan berbagai tahap, yaitu (Nazilly et al., 2020):
 1. Membagi citra dalam *region/grid* berukuran *sxs*. *Grid-grid* tersebut bertanggung jawab untuk mendeteksi sebuah objek. Pada setiap *grid* dapat diprediksi dengan memberikan sebuah *bounding box* dengan keterangan nilai *confidence*. Nilai *confidence* menunjukkan seberapa akurat deteksi yang

dilakukan pada tiap *bounding box*. Nilai *confidence* diperoleh melalui persamaan

$$Conf(class) = Pr(class) \times IOU_{Pred}^{Truth} \quad (4)$$

$Pr(class)$ adalah probabilitas objek yang muncul dalam suatu region dan IOU adalah rasio tumpang tindih (*Intersection Over Union*) antara kotak prediksi dan kotak *ground truth*. $Pred$ adalah luas area dalam kotak prediksi, $Truth$ adalah area dalam *ground truth*. Semakin besar IOU, maka akan semakin tinggi tingkat akurasinya.

2. Tahap selanjutnya yaitu tiap *bounding box* memiliki 5 nilai informasi, yaitu x , y , w , h dan c . Nilai x dan y adalah koordinat titik tengah *bounding box* yang terprediksi, nilai w dan h adalah rasio ukuran lebar dan tinggi relatif terhadap *grid*, dan c adalah nilai *confidence bounding box* tersebut.
3. Pada tahap selanjutnya tiap *grid* akan memprediksi nilai *class probability* jika diprediksi terdapat objek didalamnya. Saat pengujian, *YOLO* akan mengkalikan nilai *class probability* dengan nilai *confidence* dari *bounding box*. Sehingga menghasilkan nilai *confidence class* secara spesifik pada tiap *bounding box*.

Dalam melakukan perhitungan mobil, peneliti menghitung jumlah dari mobil yang terdeteksi berdasarkan *bounding box* yang ada, sehingga ketersediaan lahan parkir dapat dihitung dengan cara mengurangi kapasitas lahan parkir dengan jumlah mobil yang tersedia.

5. Tahap selanjutnya adalah menentukan apakah sebuah lahan parkir tersedia atau tidak. Terdapat dua buah kondisi, yaitu kondisi parkir penuh dan tidak. Apabila kondisi parkir penuh, maka sistem akan menuliskan "Parkir Penuh" dengan warna merah pada citra video yang ditampilkan. Namun apabila lahan parkir tersedia, maka sistem akan menuliskan "Parkir Tersedia".

6. Setelah melakukan tahap 5, maka selanjutnya adalah menyimpan video dalam bentuk citra digital, sehingga menghasilkan sebuah citra dengan keterangan jumlah mobil dan keterangan kapasitas parkir yang ada.

HASIL DAN PEMBAHASAN

A. Input Citra

Pengamatan pada penelitian ini dilakukan dengan cara mengamati hasil deteksi pada sebuah input citra digital berupa video dari aplikasi yang dikembangkan. Video yang digunakan berasal dari kamera *smartphone* dan internet yang berisi rekaman kumpulan mobil yang berada pada sebuah lahan parkir. Mobil yang diamati pada video mempunyai keberagaman mulai dari jenis, warna dan posisi tiap mobil.

Pada penelitian ini, video akan dimasukkan kedalam program yang dikembangkan adalah menggunakan format (.mp4). Gambaran video sebelum dilakukan pendeteksian dapat terlihat pada Gambar 2, Gambar 3, dan Gambar 4.



Gambar 2. Tampilan awal video 1



Gambar 3. Tampilan awal video 2



Gambar 4. Tampilan awal video 3

Pada Gambar 2. Pengambilan video dilakukan menggunakan kamera *smartphone* dengan resolusi 1080p / *Full HD*. Dapat terlihat dalam lahan parkir terdapat beberapa mobil dengan jenis yang berbeda mulai dari bentuk, warna dan posisi. Dalam video juga terdapat banyak gangguan seperti pohon, orang, dan lain sebagainya.

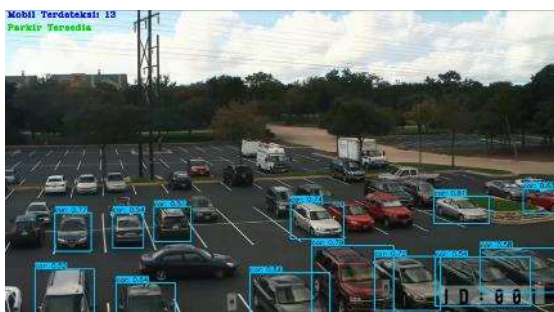
Pada Gambar 3 dan Gambar 4. adalah video yang diambil melalui *internet* yang berasal dari *cctv* sebuah lahan parkir. Mobil yang dideteksi berbeda dari segi bentuk, warna dan posisi.

B. Hasil Deteksi

Deteksi pada video dilakukan dengan menggunakan algoritma *YOLO v4*. Objek yang akan dideteksi hanya berupa sebuah mobil, sehingga objek lain pada video tidak akan dideteksi.



Gambar 5. Hasil deteksi video 1



Gambar 6. Hasil deteksi video 2



Gambar 7. Hasil deteksi video 3

Pada Gambar 5, Gambar 6 dan Gambar 7 terlihat sejumlah objek yang terdeteksi. Objek terdeteksi ditandai dengan sebuah *bounding box* berwarna biru muda dengan disertai klasifikasi dan tingkat *confidence* dari masing – masing deteksi. Pada video hasil deteksi diperlihatkan jumlah dari mobil yang terdeteksi dan status lahan parkir yang ada.

$$Akurasi = \frac{Hasil\ Deteksi}{Fakta\ Jumlah} \times 100\%$$

Tabel 1. Hasil Deteksi Mobil

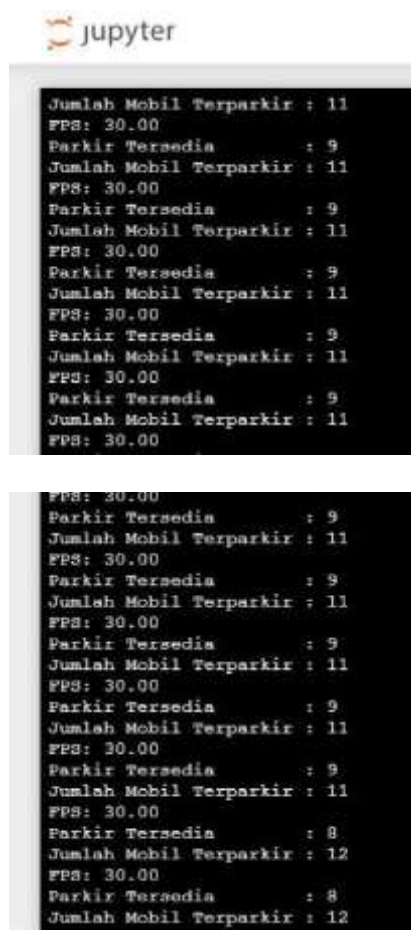
Ket.	Hasil Deteksi	Fakta Jumlah	Akurasi
Video 1	11 Mobil	11 Mobil	100%
Video 2	15 Mobil	30 Mobil	50%
Video 3	11 Mobil	10 Mobil	90%

Pada tabel 1 dapat dilihat bahwa akurasi tertinggi berada pada video 1 dengan tingkat akurasi 100%, sehingga jumlah mobil yang terdeteksi pada video sama dengan hasil deteksi menggunakan mata manusia. Pada tabel 2 hasil deteksi dengan jumlah fakta adalah sebesar 50%, dengan jumlah mobil terdeteksi adalah 15,

sedangkan jumlah mobil yang terlihat menggunakan mata manusia adalah 30 mobil. Pada tabel 3 terlihat hasil deteksi pada video adalah sejumlah 11 mobil dan hasil pengamatan sebenarnya adalah 10 mobil, tetapi deteksi pada Gambar 7 menunjukkan bahwa terdapat mobil diluar lahan parkir yang ikut terdeteksi, sehingga jumlah mobil pada hasil deteksi dikurangi 2, dan menghasilkan tingkat akurasi sebesar 90%.

C. Perhitungan ketersediaan lahan parkir

Perhitungan pada ketersediaan lahan parkir dilakukan dengan cara mengurangi jumlah lahan parkir yang ada dengan mobil yang terdeteksi. Lahan parkir keseluruhan ditentukan dengan cara manual menggunakan sebuah variabel yang berisi data jumlah lahan parkir yang dapat diisi, sehingga dapat dikurangkan dengan jumlah mobil yang terdeteksi.



Gambar 8. Data lahan parkir video 1



Gambar 9. Data lahan parkir video 2



Gambar 10. Data lahan parkir video 3

Gambar 8, Gambar 9 dan Gambar 10 menunjukkan data lahan parkir pada ketiga video. Jumlah mobil terparkir adalah jumlah mobil yang terdeteksi pada video. sedangkan

parkir tersedia merupakan jumlah parkir yang dapat digunakan.

$$Rata - rata akurasi = \frac{\sum Akurasi}{\sum Video}$$

Tabel 2. Data Lahan Parkir Video 1

No.	Ketersediaan Lahan	Jumlah lahan	Akurasi
1	9	20	100%
2	Mobil	Mobil	
3	9	20	100%
4	Mobil	Mobil	
5	8	20	91%
6	Mobil	Mobil	
Rata - rata akurasi			97 %

Tabel 3. Data Lahan Parkir Video 2

No.	Ketersediaan Lahan	Jumlah lahan	Akurasi
1	35	50	50%
2	Mobil	Mobil	
3	36	50	46%
4	Mobil	Mobil	
5	37	50	44%
6	Mobil	Mobil	
7	36	50	46%
8	Mobil	Mobil	
Rata - rata akurasi			46,5 %

Tabel 4. Data Lahan Parkir Video 3

No.	Ketersediaan Lahan	Jumlah lahan	Akurasi
1	5	15	80%
2	Mobil	Mobil	
3	5	15	80%
4	Mobil	Mobil	
5	6	15	70%
6	Mobil	Mobil	
7	6	15	70%
8	Mobil	Mobil	
Rata - rata akurasi			75 %

Pada Tabel 2 dapat terlihat data dari lahan parkir pada video 1, didapatkan tingkat akurasi rata - rata sebesar 97%. Pada Tabel 2 nomor 3 mengalami penurunan tingkat akurasi, dikarenakan terdapat pendeteksian mobil diluar lahan parkir, sehingga hasil akurasi dikurangi. Hal tersebut dapat terjadi dikarenakan pada saat pengambilan citra video tidak berfokus hanya pada lahan parkir, tetapi mengambil gambaran secara luas, termasuk pada jalan diluar lahan parkir.

Pada Tabel 3 dapat terlihat data dari lahan parkir pada video 2, didapatkan tingkat akurasi

rata – rata sebesar 46,5%. Pada nomor 1 didapatkan tingkat akurasi 50%, dikarenakan mobil yang terdeteksi tidak sesuai dengan kondisi pengamatan dengan mata manusia. Pada nomor 2 dan 3 mengalami penurunan tingkat akurasi. Hal tersebut dikarenakan nilai *confidence* pada deteksi objek menurun hingga dibawah 50% atau 0,5, sehingga menyebabkan objek tidak dideteksi sebagai objek yang diinginkan. Sedangkan pada nomor 4, tingkat akurasi kembali naik, dikarenakan nilai *confidence* pada deteksi meningkat melebihi 50% atau 0,5. Tingkat akurasi yang rendah pada deteksi video 2 disebabkan oleh kualitas citra yang kurang baik, dan pengambilan citra yang terlalu luas, sehingga menyebabkan pendeteksian memiliki akurasi yang lemah. Selain hal tersebut, peneliti mensinyalir bahwa terdapat kesamaan pendeteksian warna antara citra mobil dengan citra jalan, sehingga mengakibatkan program mendeteksi sebagai sebuah jalan, bukan sebuah mobil. Hal ini dapat terjadi akibat kamera yang digunakan untuk mengambil citra video kurang dalam melakukan penangkapan warna, sehingga rentang warna dalam citra video kurang luas.

Pada Tabel 4 berisi data lahan parkir pada video 3 dengan mendapatkan akurasi rata – rata sebesar 75%. Angka tersebut didapatkan dengan cara menjumlahkan hasil akurasi tiap deteksi secara dominan dengan jumlah deteksi dominan yang dilakukan. Tingkat akurasi pada nomor 1 dan 2 didapatkan 80%, hal ini dikarenakan pada pendeteksian yang ada pada nomor 1 dan 2, terdapat 10 mobil yang terdeteksi, tetapi 1 diantaranya adalah sebuah mobil yang berada pada luar lahan parkir, sehingga hasil akurasi dikurangkan menjadi 80%. Pada nomor 3 dan 4 didapatkan hasil akurasi 70%, hal ini dikarenakan pada saat pendeteksian, mobil dengan jarak yang jauh dari kamera kehilangan deteksi dengan nilai *confidence* dibawah 50% atau 0,5. Hal ini disebabkan karena objek mobil terlalu jauh dengan kamera, sehingga menyebabkan mobil gagal deteksi.

D. Nilai akhir deteksi

Jumlah akhir deteksi ditentukan melalui perhitungan akhir akurasi pada masing – masing video. Hasil akhir ditentukan dengan menggunakan rumus:

$$NA = \frac{\sum \text{Rata – rata Akurasi}}{\sum \text{Video}}$$

$$NA = \frac{97 + 46,5 + 75}{3} \%$$

$$NA = 72,8 \%$$

Setelah dilakukan perhitungan, didapatkan nilai akhir akurasi deteksi pada program “penghitung jumlah mobil untuk mengetahui kapasitas lahan parkir” adalah sebesar 72,8%

SIMPULAN

Aplikasi atau perangkat lunak “penghitung jumlah mobil untuk mengetahui kapasitas lahan parkir” berjalan dengan baik. Mobil dapat terdeteksi baik pada video 1, video 2, maupun video 3 dengan tingkat akurasi masing-masing video adalah 97%, 46,5% dan 75%. Tingkat akurasi pada keseluruhan hasil deteksi pada 3 video adalah sebesar 72,8%, sehingga program dapat digunakan untuk mendeteksi mobil sebagai penghitung ketersediaan lahan parkir.

Pada pengembangan selanjutnya, dapat menambahkan beberapa kamera untuk mengambil pada sudut pandang tertentu, sehingga citra yang digunakan untuk melakukan deteksi dapat menangkap gambar secara lebih luas. Menambahkan data set yang ada, sehingga dapat mendeteksi lebih banyak kendaraan. Dapat mengaitkan program dengan perangkat elektronik lain, sehingga kinerja sistem dapat lebih maksimal. Dapat menambahkan perangkat elektronik berbasis *wireless* dan daring, sehingga dapat diakses dari lokasi yang jauh.

UCAPAN TERIMAKASIH

Peneliti mengucapkan terima kasih kepada dosen mata kuliah *computer vision* Universitas Negeri Semarang dan semua pihak yang telah mendukung dan membantu terselenggaranya penelitian ini.

DAFTAR PUSTAKA

Andrew, A., Buliali, J. L., & Wijaya, A. Y. (2017). Deteksi Kecepatan Kendaraan Berjalan di Jalan Menggunakan OpenCV. *Jurnal Teknik ITS*, 6(2), 366–371. <https://doi.org/10.12962/j23373539.v6i2.23489>

Belanja, K., Antrian, P., Kasir, D., Elektro, J. T., Sam, U., Manado, R., & Bahu, J. K. (2020). Penerapan Pendeteksian Manusia Dan Objek Dalam Keranjang Belanja Pada Antrian Di Kasir. *Jurnal Teknik Informatika*, 15(2), 101–108. <https://doi.org/10.35793/jti.15.2.2020.29004>

- Fatmawati, I., Utamingrum, F., & Kurniawan, W. (2019). Deteksi Kendaraan Roda Empat Untuk Mendukung Keamanan Berkendara Menggunakan Histogram of Oriented Gradients dan Support Vector Machine Berbasis Raspberry Pi. *3*(2), 1860–1866.
- Gautama, T. K., Hendrik, A., & Hendaya, R. (2016). Pengenalan Objek pada Computer Vision dengan Pencocokan Fitur Menggunakan Algoritma SIFT Studi Kasus: Deteksi Penyakit Kulit Sederhana. *Jurnal Teknik Informatika Dan Sistem Informasi*, *2*(3), 437–450. <https://doi.org/10.28932/jutisi.v2i3.554>
- Han, X., Chang, J., & Wang, K. (2021). Real-time object detection based on YOLO-v2 for tiny vehicle object. *Procedia Computer Science*, *183*, 61–72. <https://doi.org/10.1016/j.procs.2021.02.031>
- Karlina, O. E., & Indarti, D. (2019). Pengenalan Objek Makanan Cepat Saji Pada Video Dan Real Time Webcam Menggunakan Metode You Look Only Once (Yolo). *Jurnal Ilmiah Informatika Komputer*, *24*(3), 199–208. <https://doi.org/10.35760/ik.2019.v24i3.2362>
- Lazaro, A., Buliali, J. L., & Amaliah, B. (2017). Deteksi Jenis Kendaraan di Jalan Menggunakan OpenCV. *Jurnal Teknik ITS*, *6*(2). <https://doi.org/10.12962/j23373539.v6i2.23175>
- Mada, U. G., & Nugroho, H. A. (n.d.). Ultrasound object detection using morphological region-based active contour: an application system Anan Nugroho * Risanuri Hidayat and Johan Debayle. X.
- Nagataries, D., Hardiristanto, S., Purnomo, M. H., & Klasik, A. A. G. (2012). Deteksi Objek pada Citra Digital Menggunakan Algoritma Genetika untuk Studi Kasus Sel Sabit. *Journal of Electrical Engineering*.
- Nazilly, M. L., Rahmat, B., & Puspaningrum, E. Y. (2020). Implementasi Algoritma Yolo (You Only Look Once) Untuk Deteksi Api. *Jurnal Informatika Dan Sistem Informasi*, *1*(1), 81–91.
- Nugroho, A., Hidayat, R., Nugroho, H. A., & Debayle, J. (2020). Combinatorial active contour bilateral filter for ultrasound image segmentation. *Journal of Medical Imaging*, *7*(05), 1–13. <https://doi.org/10.1117/1.jmi.7.5.057003>
- Nugroho, A. M., & Hidayatullah, A. F. (2021). Keterangan Gambar Otomatis Berbahasa Indonesia dengan CNN dan LSTM. *Automata*, *2*(1), 0–3.
- Nugroho, A., Nugroho, H. A., Setiawan, N. A., & Choridah, L. (2016). Internal content classification of ultrasound thyroid nodules based on textural features. *Communications in Science and Technology*, *1*(2), 61–69. <https://doi.org/10.21924/cst.1.2.2016.25>
- Nurfita, R. (2018). Implementasi Deep Learning Berbasis Tensorflow. *Jurnal Emitor*, *18*(01), 22–27.
- Shianto, K. A., Gunadi, K., & Setyati, E. (2019). Deteksi Jenis Mobil Menggunakan Metode YOLO Dan Faster R-CNN. *Jurnal Infra*, *7*(1), 157–163.
- Mauludy, A. T., Khrisne, D. C., & Saputra, K. O. (2020). Rancang Bangun Aplikasi Pencarian Slot Empat Dengan Pendekatan Computer Vision Parkir Kosong Untuk Kendaraan Roda Empat Dengan Pendekatan Computer Vision. *Jurnal SPEKTRUM*, *7*(1), 36–40.