

Optimalisasi Sistem Parkir Menggunakan YOLO: Penerapan pada Lingkungan Urban Padat

Ilham Ramadhan¹; Paulus Harsadi²

^{1,2}Program Studi Informatika, STMIK Sinar Nusantara

Jl. KH. Samanhudi No.84-86, Laweyan, Surakarta, Jawa Tengah

¹ilham@sinus.ac.id; ²paulusharsadi@sinus.ac.id

Abstract—The concept of modern housing based on vertical housing in urban areas is one of the characteristics of dense urban environments. The concept of parking vehicles is done by gathering them in one place, which can be a large area or basement. Problems that often occur are crowding and difficulty finding a parking space, especially for car users because the space is narrow and limited. The aim of this research is to optimize the parking system using the You Look Only Once (YOLO) method as a parking management detection system. The method used for system development is Waterfall, and the programming language used is Python. At the system testing stage, the methods used are Image Processing testing to test system functionality and User Acceptance Test (UAT) to test system validity. The results obtained from this research are the application of the You Look Only Once (YOLO) information system for parking management which includes car detection, parking spaces, and available land as well as real-time display on the website. This research has been tested using the Image Processing method which shows that the entire system can function as expected. The User Acceptance Test results obtained an average of 96.25%, including in the very good category.

Abstrak—Konsep hunian modern berbasis hunian vertical di perkotaan merupakan salah satu ciri lingkungan urban padat. Konsep parkir kendaraan dilakukan dengan mengumpulkan pada satu tempat bisa berupa lahan luas atau basement. Permasalahan yang sering terjadi adalah kepadatan dan kesulitan mencari tempat parkir terutama bagi pengguna mobil karena tempatnya yang sempit dan terbatas. Tujuan dari penelitian ini adalah melakukan optimalisasi sistem parkir dengan menggunakan metode You Look Only Once (YOLO) sebagai sistem deteksi manajemen parkir. Metode yang digunakan untuk pengembangan sistem adalah Waterfall, dan bahasa pemrograman yang digunakan adalah Python. Pada tahap pengujian sistem, metode yang digunakan adalah pengujian Image Processing untuk menguji fungsionalitas sistem dan User Acceptance Test (UAT) untuk menguji validitas sistem. Hasil yang diperoleh dari penelitian ini adalah penerapan sistem informasi You Look Only Once (YOLO) untuk manajemen parkir yang meliputi deteksi mobil, tempat parkir dan lahan yang tersedia serta tampilan secara real time pada website. Penelitian ini telah diuji menggunakan metode Image Processing yang menunjukkan bahwa keseluruhan sistem dapat berfungsi sesuai dengan yang diharapkan. Dan hasil pengujian User Acceptance Test didapatkan rata-rata sebesar 96,25% termasuk dalam kategori sangat baik.

Kata Kunci —Sistem informasi parkir, Optimasi Parkir, Deteksi Objek, User Acceptance Test (UAT), Teknologi real time, You Look Only Once (YOLO)

I. PENDAHULUAN

Lahan parkir merupakan area yang digunakan untuk menempatkan kendaraan secara sementara dan non-bergerak. Sebagai bagian integral dari sistem transportasi, pengaturan lahan parkir dapat memengaruhi kinerja keseluruhan jaringan transportasi. Dikarenakan sifatnya yang sementara, optimalisasi penempatan diperlukan untuk memaksimalkan pemanfaatan lahan parkir. Pengelolaan lahan parkir di area terbuka seringkali dihadapkan pada tantangan utama, seperti menghitung jumlah kendaraan terparkir dan mengidentifikasi area parkir kosong, yang memerlukan waktu dan usaha yang signifikan [1].

Pada konsep hunian modern sekarang yaitu berupa hunian bersusun menghadapi tantangan signifikan dalam manajemen parkir. Keterbatasan lahan parkir, baik karena lokasi terbatas maupun prioritas pembangunan, menjadi masalah utama. Hal ini menyulitkan penghuni dalam mencari tempat parkir yang tersedia dikarenakan kepadatan parkir dan kesulitan menemukan tempat parkir yang tersedia.

Kepadatan parkir terjadi ketika jumlah kendaraan mencari tempat parkir melebihi kapasitas yang tersedia. Hal ini dapat menyebabkan masalah seperti parkir tidak resmi, penghalangan akses, dan ketidaknyamanan di area parkir.

Waktu yang dihabiskan untuk mencari tempat parkir yang tersedia tidak hanya mengganggu, tetapi juga mengurangi efisiensi waktu pengguna. Pengemudi yang harus mengelilingi area parkir untuk menemukan tempat yang sesuai mengalami frustrasi dan membuang waktu yang seharusnya digunakan untuk tujuan lain. Masalah ini juga dapat menyebabkan kemacetan lalu lintas di sekitar lahan parkir, merugikan pengemudi juga lingkungan yang ada disekitarnya.

Berdasarkan penjelasan latar belakang yang telah diuraikan, penulis ingin mengembangkan sistem Deteksi dan Penghitungan Ruang Parkir Otomatis menggunakan teknologi OpenCV dan Metode YOLO untuk mengatasi masalah kepadatan dan waktu pencarian tempat parkir. Keputusan ini didasari oleh kemampuan YOLO dalam mendeteksi objek secara akurat dan real-time, bahkan dalam

kondisi lingkungan yang beragam dan mampu mendeteksi objek dengan tingkat akurasi yang lebih tinggi dibandingkan metode sejenis seperti *Faster R-CNN*.

Integrasi dengan OpenCV memberikan dukungan pada pengolahan citra yang kuat, dan keduanya merupakan teknologi yang terus berkembang. Inisiatif ini diharapkan menjadi solusi integral untuk meningkatkan manajemen parkir pada lingkungan urban padat, dengan potensi untuk mengoptimalkan penggunaan lahan parkir, dan mengurangi kemacetan lalu lintas. Pengembangan aplikasi ini juga mengikuti *Waterfall*, memastikan pendekatan yang terstruktur dan efisien mulai dari tahapan perencanaan hingga pada tahapan implementasi program.

II. TINJAUAN PUSTAKA

2.1 Sistem Informasi

Sistem informasi adalah hasil pengolahan klasifikasi dan interpretasi dari suatu data sehingga dapat digunakan dalam membantu proses pengambilan keputusan. Nilai karakteristik suatu informasi mencakup intensitas dan lamanya kejutan yang diberikan kepada penerimanya, dan fakta bahwa itu sementara dan berhubungan dengan waktu [2].

2.2 Parkir

Parkir didefinisikan sebagai area atau lokasi yang dikelola, dimiliki, atau disediakan oleh pemerintah daerah sesuai dengan persyaratan dalam pasal 127 huruf e. Tempat parkir seharusnya memenuhi standar layak dan memiliki luas lahan yang cukup, memungkinkan pemanfaatan yang optimal sehingga tata letaknya teratur dan tidak menyebabkan kemacetan [3].

2.3 Metode Waterfall

Dalam pengembangan perangkat lunak, metode air terjun, juga disebut metode Waterfall, merupakan model siklus hidup klasik atau "Linear Sequential Model". Model ini, yang diperkenalkan oleh Winston Royce sekitar tahun 1970, mengadopsi pendekatan berurutan dan sistematis. Saat ini waterfall masih merupakan salah satu Teknik pengembangan rekayasa perangkat lunak yang paling populer, meskipun sudah dianggap kuno oleh beberapa pengembang. Metode pengembangan ini bersifat linear, dimulai dari proses awal berupa perencanaan hingga pemeliharaan, dan setiap tahap harus dilaksanakan secara berurutan setelah menyelesaikan tahap sebelumnya terlebih dahulu. Setelah tahapan tersebut selesai dilaksanakan, tidak mungkin untuk kembali atau mengulangi tahap sebelumnya [4].

2.4 OpenCV

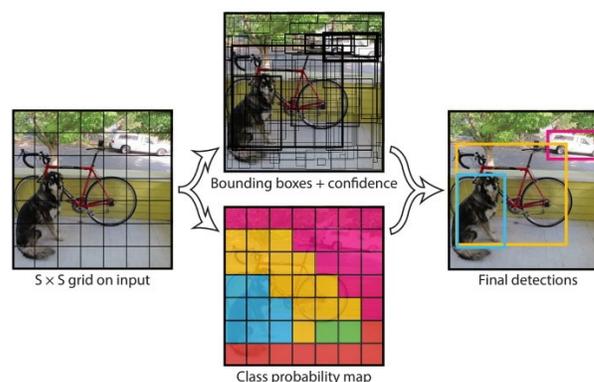
OpenCV (Open-Source Computer Vision) merupakan sebuah komponen pre-existing perangkat lunak bersifat open-source yang berfokus pada pembelajaran mesin dan computer vision. Tujuan utama pembangunan OpenCV adalah menyediakan infrastruktur yang luas dalam pembuatan/pengembangan suatu aplikasi computer vision dan mempercepat kemampuan sebuah pembelajaran mesin dalam memahami lingkungan, terutama dalam produk komersial [5].

2.5 Image Processing

Image Processing adalah disiplin ilmu yang fokus pada manipulasi dan analisis citra digital atau visual untuk menghasilkan informasi yang berguna. Melibatkan aplikasi berbagai operasi dan teknik, seperti pembersihan noise, peningkatan kontras, deteksi objek, dan pengenalan pola, pemrosesan citra digunakan dalam berbagai konteks seperti pengolahan medis, analisis citra satelit, pengenalan wajah, dan kendali kualitas industri. Dengan menerapkan algoritma dan teknik tertentu, pemrosesan citra memungkinkan ekstraksi informasi yang relevan atau modifikasi citra sesuai dengan kebutuhan spesifik [6].

2.6 You Look Only Once

Dalam teknik pendeteksian objek, salah satu pendekatan yang digunakan adalah algoritma YOLO (You Only Look Once). YOLO merupakan jaringan saraf yang secara langsung bekerja pada suatu citra, membaginya ke dalam daerah, dan meramalkan bounding box beserta probabilitas untuk setiap daerah tersebut. Dengan menghitung probabilitas pada setiap bounding box, YOLO dapat mengklasifikasikan apakah suatu daerah tersebut mengandung objek atau tidak. YOLO memiliki kemampuan pengenalan objek secara real-time dengan kecepatan 45 frame per detik, yang menjadikannya unggul dibandingkan dengan metode pengenalan pola lainnya [7].



Gambar 1. Algoritma YOLO

Algoritma "You Only Look Once" (YOLO) berbeda dengan pendekatan deteksi objek lainnya. Pada (Gambar 1. Algoritma YOLO) YOLO membagi citra menjadi beberapa daerah dan secara simultan memprediksi bounding boxes dan probabilitas untuk setiap daerah tersebut. Meskipun lebih cepat, YOLO cenderung memiliki lebih banyak kesalahan dalam lokalisasi objek karena tidak melibatkan tahap deteksi proposal sebelumnya. Selain itu, algoritma ini mengalami kesulitan dalam mendeteksi objek berukuran kecil dan yang berdekatan [8].

2.7 User Acceptance Test

Proses User Acceptance Testing (UAT) dilakukan oleh pengguna akhir dari sistem yang diuji, terutama karyawan atau karyawan perusahaan yang secara langsung melakukan interaksi. Setelah melalui tahap pengujian sistem, tujuan dari UAT adalah untuk memastikan bahwa fungsi sistem berjalan sesuai dengan kebutuhan dan tujuan fungsionalnya [9].

2.8 Penelitian Terkait

Penelitian tentang Aplikasi Penghitung Jumlah Mobil yang digunakan untuk Mengetahui Ketersediaan Lahan Parkir dengan basis metode YOLO versi 4 dan Bahasa pemrograman Python, untuk membantu pemilihan lahan parkir dengan memantau ketersediaan tempat parkir, memudahkan pengguna untuk mengetahui apakah area parkir kosong atau terisi. [5].

Penelitian dengan judul “Pendeteksian dan Pengenalan Jenis Mobil Menggunakan Algoritma You Only Look Once dan Convolutional Neural Network” untuk memberikan kontribusi pada pengembangan teknologi pendeteksian objek, khususnya dalam konteks identifikasi jenis mobil, dengan mengintegrasikan YOLO dan CNN sebagai pendekatan yang komprehensif [10].

Penelitian tentang “Penandaan Otomatis Tempat Parkir Menggunakan YOLO untuk Mendeteksi Ketersediaan Tempat Parkir Mobil pada video cctv” yang bertujuan untuk memberikan solusi terhadap sulitnya menemukan tempat parkir, terutama pada jam sibuk, dengan menyediakan penandaan otomatis mengenai ketersediaan tempat parkir [11]

Penelitian tentang “Deteksi Jenis Mobil Menggunakan Metode YOLO dan Faster R-CNN” yang bertujuan untuk meningkatkan akurasi kebenaran identifikasi jenis mobil dimana YOLO menunjukkan akurasi prediksi yang lebih rendah dibandingkan dengan Faster R-CNN. Meskipun demikian, YOLO mampu memberikan prediksi yang lebih akurat daripada Faster R-CNN dalam beberapa kasus [12],[13].

Penelitian tentang “Aplikasi Penghitung Kapasitas Ruang Parkir pada Lahan Parkir Kosong Menggunakan Library OpenCV pada Bahasa Pemrograman Python” membangun sebuah aplikasi dengan memanfaatkan Library OpenCV dalam melakukan perhitungan adanya lahan parkir kosong pada suatu area parkir terbuka dengan bahasa pemrograman Python [1].

III. METODE PENELITIAN

Tahap yang wajib dilalui oleh seorang peneliti adalah metode penelitian. Tahapan metode ini diperlukan untuk mendapatkan data yang tepat dan akurat, dan beberapa di antaranya adalah sebagai berikut..

3.1 Analisis Sistem

1. Data Primer

Data primer diperoleh melalui dua pendekatan utama yaitu, observasi langsung untuk mendapatkan pemahaman mendalam tentang pola parkir yang terjadi, mengidentifikasi potensi kendala yang mungkin timbul, dan menggali kebutuhan untuk mengetahui dalam pengelolaan parkir dan wawancara untuk mendapatkan pandangan langsung dari beberapa subjek terkait implementasi sistem deteksi dan manajemen parkir berbasis cctv dengan algoritma YOLO. Data langsung dari sumber cctv merupakan data utama dalam perhitungan algoritma ini yang lebih efektif dan efisien dalam perhitungannya menggunakan OpenCV berbasis desktop [14],[15].

2. Data Sekunder

Data sekunder diperoleh dari proses pengumpulan data dengan cara menggali pengetahuan dari berbagai sumber seperti buku, artikel, jurnal, dan materi yang ditemukan di internet yang relevan dengan topik penelitian.

3.2 Desain Sistem

Sebuah Use Case Diagram menunjukkan fungsi-fungsi utama sebuah sistem dan mendokumentasikan interaksi antara berbagai elemen dalam proyek.

3.3 Implementasi Sistem

Tahapan implementasi sistem, juga dikenal sebagai tahap pengkodean, adalah saat sistem yang dirancang dapat diterapkan pada perangkat lunak dan perangkat keras yang akan digunakan..

3.4 Pengujian

Pengujian sistem merupakan langkah kritis untuk mengevaluasi apakah sistem beroperasi sesuai dengan persyaratan yang telah ditetapkan dan untuk mengukur tingkat kepuasan pengguna.

3.5 Maintenance Sistem

Perawatan sistem pada proyek sistem deteksi dan manajemen parkir berbasis CCTV dengan metode You Only Look Once (YOLO) melibatkan tindakan pemeliharaan.

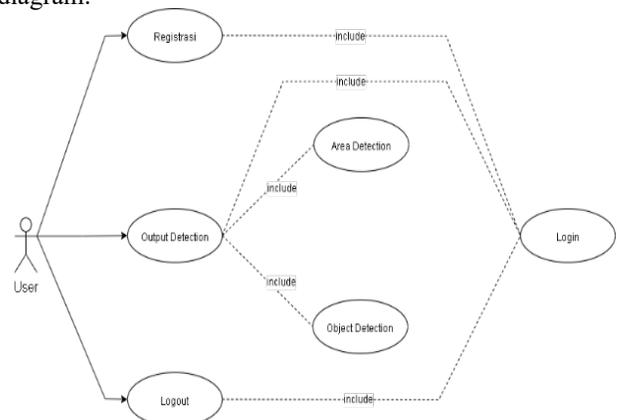
IV. HASIL DAN PEMBAHASAN

4.1 Analisa Sistem

Sistem ini berfungsi untuk mendeteksi keberadaan mobil di area yang telah ditentukan, dan mendeteksi apakah suatu lahan parkir digunakan atau tidak. Selain itu terdapat fitur login dan register yang dimana sistem ini dideploy kedalam sebuah website dan dijalankan secara realtime.

4.2 Desain Sistem

Pada sistem ini user dapat mengakses layanan login, register, logout, dan dashboard untuk melihat deteksi cctv secara realtime yang dapat dilihat pada gambar 2 use case diagram.



Gambar 2. Use Case Diagram

4.3 Implementasi Sistem

Berikut merupakan implementasi dari rancangan yang sudah dijelaskan sebelumnya ke dalam sistem menggunakan

bahasa pemrograman Python dan metode *waterfall*. Penggunaan metode *waterfall* cukup efektif dalam membangun sistem yang tidak begitu kompleks [16].

1. Inisialisasi Model YOLO

Inisialisasi model dilakukan dengan menggunakan model YOLOv8m. Fungsi `detect_parking_spaces` dibuat untuk mendeteksi ruang parkir pada suatu frame gambar. Dalam fungsi ini, dilakukan pembacaan daftar kelas objek dari file "coco.txt" yang kemudian digunakan untuk mendeteksi ruang parkir pada frame seperti pada gambar 12 inisialisasi model YOLO.

```
# Initialize YOLO model
SET model TO YOLO("../YOLO-Weights/yolov8m.pt")

# Function to detect parking spaces and RETURN the availability and empty areas
DEFINE FUNCTION detect_parking_spaces(frame):
    SET my_file TO open("coco.txt", "r")
    SET data TO my_file.read()
    SET class_list TO data.split("\n")
```

Gambar 3 Inisialisasi Model YOLO

2. Pembuatan Marking Area

Pembuatan marking area didefinisikan dengan beberapa variabel, seperti `area1`, `area2`, `area3`, `area4`, dan `area5` yang merepresentasikan koordinat sudut-sudut dari berbagai area dalam gambar. Koordinat ini dapat dimanfaatkan untuk deteksi objek dalam area yang ditentukan seperti pada gambar 13 pembuatan marking area.

```
# Initialize Marking Area

SET area1 TO [(1130, 693), (1177, 783), (1321, 774), (1248, 688)]
SET area2 TO [(1261, 689), (1343, 774), (1464, 760), (1366, 682)]
SET area3 TO [(1379, 681), (1476, 759), (1590, 745), (1485, 671)]
SET area4 TO [(1497, 670), (1597, 743), (1689, 729), (1589, 663)]
SET area5 TO [(1615, 661), (1701, 725), (1780, 711), (1680, 654)]
```

Gambar 4. Marking Area

3. Pemrosesan Awal

Pemrosesan awal disimpan dalam variabel `results`, dan koordinat bounding boxes diambil dari hasil tersebut. Koordinat tersebut kemudian diubah menjadi dataframe menggunakan `pandas`. Meskipun terdapat variabel list yang disiapkan, pada tahap ini, list-list tersebut belum diisi atau dimanipulasi. Kode ini merupakan langkah awal dalam pemrosesan hasil deteksi objek dengan YOLO, dan variabel dataframe `px` dapat digunakan untuk analisis lebih lanjut pada koordinat bounding boxes seperti pada gambar 14

pemrosesan awal.

```
# Saving list result

CALL SET results TO model.predict(frame)
CALL SET a TO results[0].boxes.boxes
CALL SET px TO pd.DataFrame(a).astype("float")
SET list1, list2, list3, list4, list5 TO [], [], [], [], []
```

Gambar 5. Pemrosesan Awal

4. Proses Deteksi

Pada proses deteksi setiap baris diproses untuk mengambil koordinat sudut-sudut bounding box dan informasi kelas objek. Jika objek yang terdeteksi adalah kendaraan ('car'), maka titik tengah objek dihitung dan digunakan untuk menentukan apakah objek tersebut berada di dalam suatu area parkir. Pengujian dilakukan dengan menggunakan fungsi `cv2.pointPolygonTest` untuk menentukan posisi objek dalam poligon yang mewakili area tertentu. Jika hasilnya positif, bounding box objek dan titik tengahnya ditampilkan pada frame gambar, dan informasi kelas objek ditambahkan ke dalam list yang sesuai (misalnya, `list1` untuk `area1`). Proses ini mengimplementasikan deteksi kendaraan di dalam zona-zona parkir pada gambar, dengan penekanan pada informasi geometris untuk memahami lokasi objek dalam suatu area seperti pada gambar 15 proses deteksi.

```
# Process detection

FOR index, row IN px.iterrows():
    SET x1, y1, x2, y2, _ d TO map(row)
    SET c TO class_list[d]

    IF 'car' IN c:
        SET cx, cy TO (x1 + x2) // 2, (y1 + y2) // 2
        CALL SET results1 TO cv2.pointPolygonTest(np.array(area1, np.int32), ((cx, cy)), False)

        IF results1 >= 0:
            CALL cv2.rectangle(frame, (x1, y1), (x2, y2), (0, 255, 0), 2)
            CALL cv2.circle(frame, (cx, cy), 3, (0, 0, 255), -1)
            list1.append(c)
```

Gambar 6. Proses Deteksi

5. Penyimpanan Deteksi

Pada proses ini, variabel '`a1`', '`a2`', '`a3`', '`a4`', dan '`a5`' diinisialisasi untuk menyimpan jumlah objek yang terdeteksi di dalam masing-masing zona parkir (seperti '`list1`', '`list2`', dst.). Kemudian, variabel '`lot`' dihitung sebagai total objek yang terdeteksi di semua zona parkir. Selanjutnya, variabel '`space`' dihitung sebagai selisih antara jumlah zona parkir yang tersedia (dalam hal ini, 5 zona) dan total objek yang terdeteksi ('`lot`') sehingga menghasilkan jumlah ruang parkir yang masih tersedia, memberikan informasi tentang ketersediaan ruang parkir berdasarkan hasil deteksi objek pada setiap zona parkir seperti pada gambar 16 penyimpanan deteksi di area parkir.

```
# Process Calculation

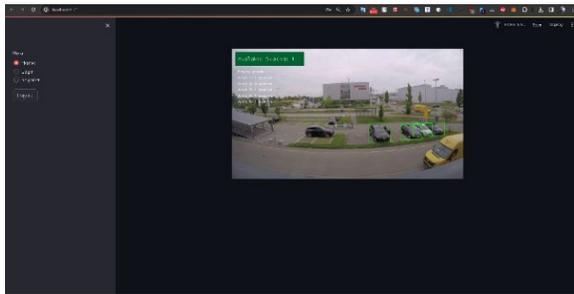
SET a1, a2, a3, a4, a5 TO len(list1), len(list2), len(list3), len(list4), len(list5)
SET lot TO a1 + a2 + a3 + a4 + a5
SET space TO 5 - lot

RETURN space, {'area1': a1, 'area2': a2, 'area3': a3, 'area4': a4, 'area5': a5}
```

Gambar 7. Penyimpanan Deteksi

6. Tampilan Halaman Dashboard

Halaman dashboard menampilkan frame video dari cctv yang dilengkapi dengan deteksi mobil serta informasi lahan parkir tersedia dan kondisi masing masing area parkir seperti pada gambar 23 tampilan halaman dashboard.



Gambar 8. Tampilan Halaman Dashboard

4.4 Pengujian Sistem

Pengujian sistem yang digunakan untuk menguji sistem informasi pendokumentasian data arsip desa adalah pengujian fungsionalitas dan pengujian validitas.

1. Pengujian Fungsionalitas

Pengujian fungsionalitas dilakukan menggunakan metode OpenCV dan image processing yang difokuskan untuk untuk mengetahui input dan output pada sistem. Pengujian ini dilakukan dengan 3 kali running untuk memeriksa fungsional pada sistem deteksi objek dengan menggunakan OpenCV dan image processing model YOLO V8. Hasil dari ketiga running yang telah dilakukan akan dipaparkan dalam tabel 1 hasil uji fungsionalitas.

Tabel 1. Hasil Uji Fungsionalitas

Kasus Dan Hasil Uji		
Skenario	Harapan	Hasil
Mendeteksi mobil pada area dengan tepat	Hanya mendeteksi mobil diarea yang ditetapkan	Berhasil
Mampu menampilkan informasi ketersediaan lahan parkir	Menampilkan ketersediaan lahan parkir secara tepat	Berhasil
Sistem deteksi berjalan secara realtime	Mampu mendeteksi secara realtime	Berhasil

2. Pengujian Validitas

Uji validitas dengan User Acceptance Test

(UAT) adalah proses pengujian sistem implementasi You Only Look Once (YOLO) untuk manajemen parkir. Metode pengujian melibatkan kuesioner kepada pengguna untuk mengevaluasi kesesuaian sistem dengan kebutuhan, kinerja, manfaat, dan kemampuannya mengatasi permasalahan. Pengujian dilakukan dengan menjalankan sistem dan meminta 8 responden untuk mengisi kuesioner yang terdiri dari 8 pertanyaan. Tujuan utama adalah mendapatkan umpan balik dari pengguna terkait keefektifan dan kepuasan mereka terhadap sistem.

Penilaiannya terdiri dalam 5 kategori, yaitu Sangat Setuju dengan skor = 5, Setuju dengan skor = 4, Netral dengan skor = 3, Tidak Setuju dengan skor = 2, dan Sangat Tidak Setuju dengan skor = 1.

Setelah pengujian kuesioner dilakukan maka langkah selanjutnya yaitu menghitung hasil dari kuesioner. Berikut ini rumus perhitungan presentase berdasarkan skor dari skala yang sudah ditentukan.

$$P = \frac{s}{nMax} \times 100\% \tag{1}$$

Keterangan:

P = Nilai presentase

S= Jumlah frekuensi x skor jawaban

nMax= Jumlah responden x skor tertinggi

Berdasarkan rumus diperoleh nilai max adalah 8 x 5 (Sangat Setuju) = 40. Hasil perhitungan kuesioner ditunjukkan pada table 2 hasil perhitungan kuesioner.

Tabel 2. Hasil Perhitungan Kuesioner

P	Jawaban				Jawaban x Skor					+	%	
	S	S	N	T	S	x5	x4	x3	x2			x1
1	6	2	0	0	0	30	8	0	0	0	38	95
2	4	4	0	0	0	20	16	6	0	0	36	90
3	5	3	0	0	0	25	12	0	0	0	37	92,5
4	5	3	0	0	0	25	12	0	0	0	37	92,5
5	8	0	0	0	0	40	0	0	0	0	40	100
6	5	3	0	0	0	25	12	0	0	0	37	92,5
7	6	2	0	0	0	30	8	0	0	0	38	95
8	8	0	0	0	0	40	0	0	0	0	40	100
Rata-Rata P											96,25	

Dari 8 pertanyaan pada kuesioner tersebut didapatkan nilai presentase rata-rata yaitu sebesar 96,25%. Dari hasil rata-rata presentase yang diperoleh, dapat dinyatakan bahwasannya sistem yang telah dibuat termasuk kedalam kategori sangat baik yang didapatkan berdasarkan tabel 3 kategori hasil kuesioner.

Tabel 3. Kategori Hasil Koesioner

Nilai %	Kategori
0% - 20%	Sangat Buruk
20,01% - 40%	Buruk
40,01% - 60%	Cukup
60,01% - 80%	Baik
80,01% - 100%	Sangat Baik

4.5 Maintenance

Maintenance dalam metode Waterfall penting untuk mendeteksi dan memperbaiki kesalahan sistem, termasuk pembaruan model deteksi dan peningkatan algoritma. Pemeliharaan tidak hanya fokus pada perbaikan bug, tetapi juga melibatkan strategi proaktif untuk meningkatkan performa. Dalam implementasi YOLO untuk manajemen parkir, pemeliharaan mencakup pembaruan model, peningkatan algoritma, dan pemeliharaan perangkat keras. Manajemen data, termasuk cadangan dan retensi data. Bab ini krusial untuk memastikan kesuksesan jangka panjang proyek skripsi ini dalam mendukung manajemen parkir di perusahaan tersebut.

V. PENUTUP

5.1 Kesimpulan

Berdasarkan hasil pembahasan dapat disimpulkan bahwa :

1. Permasalahan pada lingkungan urban padat contoh pada hunian bersusun salah satunya adalah lahan parkir, dimana sering terjadi kepadatan dan manajemen parkir yang kurang optimal.
2. Optimalisasi ini membantu mengatasi masalah manajemen parkir pada lingkungan urban padat menggunakan sistem real time dengan metode You Look Only Once (YOLO).
3. Pengujian validitas menggunakan User Acceptance Test (UAT) menunjukkan persentase rata-rata sebesar 96,25%, mengindikasikan bahwa sistem dapat diterima dan layak digunakan oleh pengguna dan dapat menjadi solusi terhadap masalah kepadatan pada lahan parkir.

5.2 Saran

Sistem Implementasi You Look Only Once (YOLO) untuk Manajemen Parkir perlu dikembangkan dengan penambahan fitur booking lahan parkir sesuai dengan posisi yang tersedia, penambahan fitur log untuk menyimpan data parkir dalam kurun waktu tertentu, dan optimalisasi agar sistem dapat berjalan dengan lebih ringan pada saat diimplementasikan.

DAFTAR PUSTAKA

- [1] D. A. Lestari, M. W. Sardjono and M. Mujahidin, "Aplikasi Penghitung Kapasitas Ruang Parkir pada Lahan Parkir Kosong Menggunakan Library OpenCV pada Bahasa Pemrograman Python," *Journal of System and Computer Engineering (JSCE)*, 2023.
- [2] D. S. Malik dan A. Zein, "Perancangan Aplikasi Sistem Informasi Penjualan Online Berbasis Web Menggunakan Metode Personal Extreme Programming Di Toko Surya Gemilang," *Jurnal Ilmu Komputer JIK* Vol. V, p. 54, 2022.
- [3] Elmawati and Y. Tandiri, "Analisis Kebutuhan Parkir Pada Kantor BPJS Di Kota Palopo," *Osfpreprints*, pp. 4-5, 2022.
- [4] A. A. Wahid, "Analisis Metode Waterfall Untuk Pengembangan Sistem Informasi," *Jurnal Ilmu Informatika Dan Manajemen STMIK*, pp. 1-5, 2020.
- [5] G. N. Rizkatama, A. Nugroho and A. F. Suni, "Sistem Cerdas Penghitung Jumlah Mobil untuk Mengetahui Ketersediaan Lahan Parkir Berbasis Python dan YOLO V4," *Edu Komputika Journal*, p. 92, 2021.
- [6] I. Hartono, A. Noertjahyan and L. W. Santos, "Deteksi Masker Wajah Dengan Metode Convolutional Neural Network," *Jurnal Infra*, 2022.
- [7] A. Dalwadi, M. H. Bohara, B. Y. Panchal, I. A. Prabowo, H. Parikh, M. Makawana and D. Soni, "Detecting and Counting People In Dense Crowd," *UGC CARE Listed (Group -I) Journa*, vol. 11, 2022.
- [8] Z. Zou, K. Chen, Z. Shi, Y. Guo and J. Ye, "Object Detection in 20 Years: A Survey," *arxiv*, 2023.
- [9] Y. D. Wijaya dan M. W. Astuti, "Pengujian Blackbox Sistem Informasi Penilaian Kinerja," *Jurnal Digital Teknologi Informasi Volume 4*, p. 23, 2021.
- [10] C. Geraldly and C. Lubis, "Pendeteksian dan Pengenalan Jenis Mobil Menggunakan Algoritma You Only Look Once dan Convolutional Neural Network," *Jurnal Ilmu Komputer dan Sistem Informasi*, 2020.
- [11] E. Tanuwijaya and C. Faticah, "Penandaan Otomatis Tempat Parkir Menggunakan YOLO untuk Mendeteksi Ketersediaan Tempat Parkir Mobil pada Video CCTV," *Jurnal Unublitar*, 2020.
- [12] K. A. Shianto, K. Gunadi and E. Setyati, "Deteksi Jenis Mobil Menggunakan Metode YOLO dan Faster R-CNN," *Jurnal Infra*, 2019.
- [13] P. R. Aningtiyas, A. Sumin dan S. Wirawan, "Pembuatan Aplikasi Deteksi Objek Menggunakan TensorFlow Object Detection API dengan Memanfaatkan SSD MobileNet V2 Sebagai Model Pra - Terlatih," *Jikstik*, 2020.
- [14] M. Leriensyah and A. Kurniawardhani, "Klasifikasi dan Perhitungan Kendaraan untuk Mengetahui Arus Kepadatan Lalu Lintas Menggunakan Metode YOLO," *Automata*, 2020.
- [15] M. R. Pratama, E. P. Bhayangkara and J. M. Ishlah, "Model Aplikasi Model Aplikasi Document Scanner Menggunakan Operator Canny Dan Contour Pada Open Cv Berbasis Desktop," *Jurnal Teknik Informatika (JUTEKIN)*, 2022.
- [16] Rahmat, Ferdy, Faisal Alfarizi, Dimas Maulana, Muhammad Ridwan Ramadhan, and Fuad Nur Hasan. "Implementasi Metode Waterfall Pada Web Company Profile Yayasan Mega Gotong Royong." *Jurnal Informatika Upgris 10*, no. 1 (2024): 5-10.