

Simulasi Okupansi Parkir Dinamis Menggunakan Deteksi Ruang Parkir Berbasis *Computer Vision*

¹Yehezkiel Haganta Tarigan, ²Zulfahmi Indra, ³Andly Sofian Hasugian, ⁴Chelsea Febiola
br Sebayang

^{1,2,3,4}Program Studi Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam,
Universitas Negeri Medan, Medan, Indonesia

E-mail: ¹yehezkielhaganta@gmail.com, ²zulfamhi.indra@mhs.unimed.ac.id, ³
dlyhasugian@gmail.com, ⁴chelseaafss7@gmail.com

ABSTRAK

Fasilitas parkir sulit dikelola ketika jumlah kendaraan yang datang bertambah, dan mengetahui kondisi parkir saat ini belum menjelaskan bagaimana okupansi akan berubah pada permintaan yang lebih tinggi. Penelitian ini mengembangkan kerangka kerja yang mengintegrasikan deteksi ruang parkir berbasis YOLO26s dengan simulasi okupansi stokastik. Pada set uji Roboflow, detektor mencapai presisi 93,6%, *recall* 91,8%, dan mAP@50 95,5%, meskipun positif palsu dan negatif palsu yang tersisa masih memengaruhi masukan simulasi. Pada video parkir, median deteksi sepuluh *frame* pertama menghasilkan estimasi kapasitas 174 ruang dan okupansi awal 58,6%. Nilai ini merupakan estimasi karena kapasitas hasil deteksi bervariasi antara 150 dan 194 ruang pada setiap *frame*. Simulasi tiga skenario kedatangan menunjukkan bahwa skenario rendah tetap berada di bawah kapasitas dengan okupansi rata-rata 50,78%, sedangkan skenario sedang dan tinggi mencapai kapasitas penuh pada seluruh pengulangan setelah rata-rata 124,5 dan 30,5 menit. Laju kedatangan yang lebih tinggi dengan demikian mempercepat kejenuhan dan memperpanjang waktu pada kapasitas penuh. Namun, parameter kedatangan dan keberangkatan merupakan asumsi skenario sehingga hasilnya bukan prakiraan permintaan parkir yang sebenarnya. Penelitian selanjutnya perlu menggunakan video yang lebih panjang, memvalidasi kapasitas terhadap data acuan (*ground truth*), dan menambahkan pelacakan kendaraan untuk mengestimasi laju kedatangan dan keberangkatan.

Kata kunci : *okupansi parkir; YOLO; computer vision; simulasi stokastik; deteksi ruang parkir*

ABSTRACT

Parking facilities are difficult to manage when the number of arriving vehicles grows, and knowing the current parking condition does not explain how occupancy will change under higher demand. This study developed a framework that integrates YOLO26s-based parking space detection with a stochastic occupancy simulation. On the Roboflow test set, the detector achieved a precision of 93.6%, recall of 91.8%, and mAP@50 of 95.5%, although remaining false positives and false negatives still affect the simulation input. Applied to the parking video, the median detections of the first ten frames gave an estimated capacity of 174 spaces and an initial occupancy of 58.6%. These values are estimates, as the detected capacity varied between 150 and 194 spaces across frames. The simulation of three arrival scenarios showed that the low scenario stayed below capacity, with a mean occupancy of 50.78%, whereas the moderate and high scenarios reached full capacity in all runs after mean times of 124.5 and 30.5 minutes. A higher arrival rate therefore brings saturation forward and lengthens the time spent at capacity. However, the arrival and departure parameters are scenario assumptions, so the results are not forecasts of real parking demand. Future work should use longer videos, validate capacity against ground truth, and add vehicle tracking to estimate arrival and departure rates.

Keywords : *parking occupancy; YOLO; computer vision; stochastic simulation; parking space detection*

1. PENDAHULUAN

Meningkatnya jumlah kendaraan di wilayah perkotaan telah meningkatkan kebutuhan akan fasilitas parkir dan menimbulkan tantangan dalam mengelola kapasitas parkir yang terbatas. Pengelolaan parkir yang tidak efisien dapat menambah waktu pengemudi mencari ruang kosong dan menurunkan pemanfaatan efektif fasilitas yang ada. Oleh karena itu, informasi okupansi parkir yang akurat penting untuk mendukung pengelolaan parkir dan sistem transportasi cerdas. *Computer vision* menawarkan pendekatan praktis untuk memantau fasilitas parkir melalui citra kamera, sedangkan *deep learning* memungkinkan deteksi dan pelacakan kendaraan secara otomatis (Sharma dkk., 2023).

Perkembangan terbaru *deep learning* telah memperluas penggunaan deteksi objek dan klasifikasi citra untuk menentukan ketersediaan ruang parkir. Yuldashev dkk. (2023) mengembangkan pendekatan berbasis MobileNetV3 yang ditingkatkan untuk deteksi okupansi lahan parkir, sedangkan Sobirin dkk. (2024) meneliti YOLOv8 dan YOLOv5 untuk mendeteksi kendaraan dan ruang parkir yang tersedia dengan mengevaluasi presisi, *recall*, dan *mean Average Precision* (mAP). Studi-studi tersebut menunjukkan bahwa *deep learning* dapat mendukung pemantauan parkir otomatis, meskipun kinerja deteksi bergantung pada dataset, kondisi citra, dan karakteristik model.

Metode berbasis YOLO relevan dengan penelitian ini karena dapat mengidentifikasi objek pada citra dan menghasilkan keluaran yang dapat diolah untuk mengestimasi okupansi parkir. Saputra dkk. (2024) meneliti deteksi kendaraan dan ruang parkir menggunakan YOLOv5 untuk sistem parkir cerdas. Namun, informasi kondisi parkir saat ini tidak dengan sendirinya menjelaskan bagaimana okupansi akan berubah ketika jumlah kendaraan yang datang meningkat. Karena itu, deteksi parkir perlu dilengkapi

metode yang mampu menganalisis perubahan okupansi dari waktu ke waktu.

Okupansi parkir merupakan proses dinamis yang dipengaruhi oleh kedatangan kendaraan, keberangkatan, dan kapasitas fasilitas. Ye dkk. (2024) mengusulkan prediksi okupansi parkir yang memadukan dekomposisi deret waktu, jaringan *long short-term memory*, dan mekanisme *temporal pattern attention* untuk menangkap variasi temporalnya. Meskipun demikian, prediksi berdasarkan okupansi historis dan simulasi pada skenario kedatangan yang dikendalikan eksplisit memiliki tujuan berbeda. Simulasi dapat memisahkan pengaruh laju kedatangan terpilih dan membandingkan seberapa cepat fasilitas parkir mendekati kapasitasnya pada kondisi permintaan yang berbeda.

Simulasi menyediakan lingkungan terkendali untuk mengkaji perilaku sistem parkir tanpa percobaan yang langsung memengaruhi fasilitas yang beroperasi. Mendoza-Silva dkk. (2019) mengembangkan simulator okupansi untuk sistem parkir cerdas dan membahas rancangan serta pertimbangan eksperimennya, yang menjadi landasan relevan untuk merepresentasikan okupansi parkir secara komputasional. Pada penelitian ini, simulasi digunakan untuk menyelidiki bagaimana laju kedatangan kendaraan rendah, sedang, dan tinggi memengaruhi jumlah ruang terisi serta waktu hingga kapasitas penuh.

Penelitian terbaru terus mengkaji peningkatan deteksi okupansi parkir. Pore dan Nemade (2025) mengusulkan pendekatan berbasis *deep learning* untuk mendeteksi okupansi ruang parkir pada dataset PKLot, yang menegaskan relevansi pemantauan okupansi otomatis. Studi-studi tersebut menjadi dasar bagi evaluasi komponen deteksi, sedangkan penelitian ini memperluas analisis ke perilaku dinamis sistem parkir pada kedatangan kendaraan yang terkendali. Penelitian ini bertujuan mengembangkan

kerangka kerja yang mengintegrasikan deteksi ruang parkir berbasis YOLO26s dengan simulasi okupansi stokastik, serta membandingkan skenario kedatangan rendah, sedang, dan tinggi.

2. LANDASAN TEORI

2.1 Deteksi Okupansi Parkir Berbasis *Computer Vision*

Pemantauan okupansi parkir dengan *computer vision* memanfaatkan citra atau video kamera untuk menentukan apakah suatu ruang parkir terisi atau kosong. Pendekatan yang telah dilaporkan meliputi klasifikasi status ruang parkir (Yuldashev dkk., 2023; Pore & Nemade, 2025), deteksi objek untuk menemukan kendaraan dan ruang yang tersedia (Sobirin dkk., 2024; Saputra dkk., 2024), serta pelacakan kendaraan di lingkungan parkir (Sharma dkk., 2023). Kinerja pendekatan tersebut bergantung pada dataset, kondisi citra, dan karakteristik model yang digunakan. Pada penelitian ini, status ruang parkir dirumuskan sebagai deteksi objek dua kelas, yaitu *car* (ruang terisi kendaraan) dan *free* (ruang kosong), sehingga hasil deteksi pada satu citra dapat langsung dikonversi menjadi jumlah ruang terisi dan kapasitas total.

2.2 Detektor Objek YOLO dan Metrik Evaluasinya

You Only Look Once (YOLO) adalah keluarga detektor objek satu tahap yang memprediksi lokasi dan kelas objek dalam satu lintasan jaringan (Redmon dkk., 2016). Penelitian ini menggunakan YOLO26s, varian kecil yang menyeimbangkan akurasi dan biaya komputasi. Keluaran detektor berupa *bounding box* beserta kelas dan skor kepercayaan; deteksi di bawah ambang kepercayaan dibuang. Kinerjanya dinilai dengan presisi, *recall*, *F1-score*, dan mAP. Prediksi dihitung sebagai positif benar (*true positive*, TP) bila cocok dengan kotak acuan berkelas sama, positif palsu (FP) bila tidak ada kotak acuan yang cocok, dan kotak acuan tanpa prediksi yang cocok sebagai negatif palsu (FN).

Kecocokan ditentukan oleh *intersection over union* (IoU). Metrik mAP@50 memakai ambang IoU 0,5, sedangkan mAP@50:95 merata-ratakan mAP pada ambang IoU 0,50–0,95 dengan langkah 0,05 (Ultralytics, t.t.).

2.3 Simulasi Stokastik Okupansi Parkir

Berbeda dengan prediksi berbasis data historis (Ye dkk., 2024), simulasi memungkinkan pengaruh laju kedatangan tertentu dipisahkan dan dibandingkan dalam skenario terkendali tanpa percobaan pada fasilitas yang beroperasi (Mendoza-Silva dkk., 2019). Pada penelitian ini, simulasi berjalan dalam waktu diskret: kedatangan kendaraan per langkah dimodelkan sebagai peubah acak Poisson, sedangkan keberangkatan dimodelkan sebagai peubah acak binomial dengan peluang keberangkatan tetap per langkah, sehingga lama tinggal berdistribusi geometrik. Rasio antara aliran kedatangan dan aliran keberangkatan yang diharapkan pada area parkir penuh disebut faktor beban (*load factor*), yang menentukan apakah okupansi cenderung berada di bawah kapasitas atau jenuh.

3. METODOLOGI

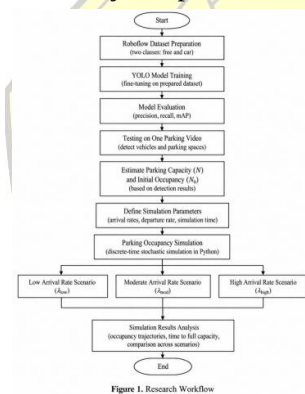
3.1 Desain Penelitian

Penelitian ini menggunakan pendekatan eksperimental kuantitatif untuk menyelidiki deteksi okupansi parkir menggunakan *computer vision* dan menganalisis dinamika okupansi parkir melalui simulasi stokastik. Penelitian terdiri atas dua tahap utama, yaitu deteksi parkir berbasis YOLO dan simulasi okupansi parkir.

Pada tahap pertama, model YOLO yang dilatih pada dataset parkir dari Roboflow dievaluasi menggunakan metrik deteksi objek, yaitu presisi, *recall*, dan *mean Average Precision* (mAP). Model terlatih kemudian diuji pada satu video parkir untuk mendeteksi kendaraan dan ruang parkir yang diklasifikasikan sebagai terisi atau tersedia. Hasil deteksi

diolah untuk mengestimasi kapasitas parkir total dan okupansi awal dengan prosedur penghitungan yang telah ditetapkan.

Pada tahap kedua, estimasi kapasitas dan okupansi awal digunakan untuk menginisialisasi simulasi stokastik waktu diskret yang diimplementasikan dengan Python. Tiga skenario kedatangan kendaraan hipotetis, yaitu laju kedatangan rendah, sedang, dan tinggi, dibandingkan untuk menganalisis perubahan okupansi parkir dari waktu ke waktu. Hasil simulasi dievaluasi berdasarkan lintasan okupansi, waktu hingga kapasitas penuh bila tercapai, dan ukuran okupansi lainnya. Hasil deteksi dan hasil simulasi dianalisis secara terpisah. Alur penelitian secara keseluruhan disajikan pada Gambar 1.

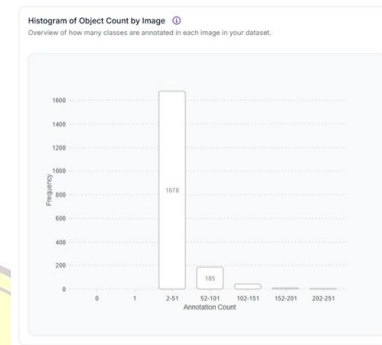


Gambar 1. Alur penelitian

3.2 Dataset dan Persiapan Data

Dataset yang digunakan adalah dataset deteksi objek ruang parkir dari Roboflow dengan nama proyek “Parking” (<https://universe.roboflow.com/threerapongs-workspace/parking-u6fqb>). Dataset ini berisi 1.921 citra dengan 51.240 *bounding box* pada dua kelas: *car*, yaitu ruang parkir yang terisi kendaraan, dan *free*, yaitu ruang parkir yang kosong. Citra sebagian besar diambil dari sudut pandang atas (*top-down*) pada kondisi pencahayaan dan tata letak lahan parkir yang beragam. Setiap citra memuat rata-rata 26,7 objek teranotasi, dengan rentang 2 hingga 250 objek per citra. Seperti ditunjukkan pada Gambar 2, sebanyak 1.678 citra (87,3%) memuat 2–51 objek,

185 citra (9,6%) memuat 52–101 objek, dan 58 citra sisanya (3,0%) memuat lebih dari 101 objek, sehingga dataset mencakup adegan yang jarang maupun padat.



Gambar 2. Histogram jumlah objek teranotasi per citra pada dataset (sumber: Roboflow Dataset Analytics)

Persiapan data dilakukan di Roboflow. Pertama, integritas dataset diperiksa menggunakan modul *Dataset Analytics*, yang melaporkan tidak ada citra tanpa anotasi dan tidak ada contoh kosong (*null*). Seluruh citra beresolusi sama, yaitu 640×640 piksel (sekitar 0,41 MP, rasio aspek persegi), sehingga tidak ada citra yang dibuang karena dimensi yang tidak konsisten. Kedua, langkah penyesuaian ukuran (*resize*, *stretch* ke 640×640 piksel) disertakan pada versi dataset untuk menjamin ukuran masukan yang seragam bagi model YOLO; karena citra sumber sudah persegi pada resolusi tersebut, langkah ini tidak mengubah bentuk isi citra. Ketiga, versi yang digunakan untuk pelatihan (v1) memuat 1.921 citra yang sama dengan dataset sumber, yang menunjukkan bahwa tidak ada augmentasi *offline* untuk memperbanyak data.

Data dibagi menjadi subset pelatihan, validasi, dan uji masing-masing sebanyak 1.345, 384, dan 192 citra. Berdasarkan jumlah instans teranotasi, pembagian tersebut sekitar 70:20:10 (70,5%; 20,4%; dan 9,1%). Subset pelatihan digunakan untuk melatih model, subset validasi untuk memantau pelatihan dan memilih bobot model, dan subset uji hanya untuk evaluasi akhir pada data yang

belum pernah dilihat. Tabel 1 merangkum jumlah instans teranotasi per kelas pada setiap subset.

Tabel 1. Jumlah instans teranotasi per kelas pada setiap subset data

Subset	Car	Free	Total	Porsi car (%)
Pelatihan	22.183	13.929	36.112	61,4
Validasi	6.369	4.078	10.447	61,0
Uji	3.161	1.520	4.681	67,5
Semua subset	31.713	19.527	51.240	61,9

Secara keseluruhan, *car* menyusun 61,9% dari seluruh instans dan *free* 38,1% (rasio sekitar 1,6:1), sehingga kedua kelas terwakili dengan baik pada setiap subset dan ketidakseimbangan kelas tergolong sedang. Proporsi *car* serupa pada subset pelatihan (61,4%) dan validasi (61,0%) tetapi lebih tinggi pada subset uji (67,5%); perbedaan ini diperhatikan dalam menafsirkan hasil uji. Karena kedua kelas berkaitan langsung dengan ruang terisi dan kosong, deteksi pada citra lahan parkir dapat dikonversi menjadi jumlah ruang terisi dan kapasitas total (jumlah deteksi *car* ditambah *free*) yang menentukan kondisi awal simulasi pada Subbab 3.6. Dataset terdiri atas citra statis sehingga hanya digunakan untuk melatih dan mengevaluasi detektor, sedangkan dinamika temporal diperkenalkan secara terpisah melalui simulasi.

3.3 Pengembangan Model

Model deteksi yang digunakan adalah YOLO26s, varian kecil dari keluarga detektor objek satu tahap YOLO26 (Redmon dkk., 2016) yang dirilis di bawah lisensi AGPL-3.0. Varian kecil memberikan keseimbangan antara akurasi deteksi dan biaya komputasi. Tugas dirumuskan sebagai deteksi objek dua kelas pada dataset yang disiapkan pada Subbab 3.2: setiap *bounding box* menandai kendaraan yang parkir (*car*) atau ruang parkir kosong (*free*). Dengan demikian, model memprediksi lokasi dan status okupansi ruang parkir dalam satu lintasan, bukan hanya mendeteksi kendaraan.

Model *di-fine-tune* pada subset pelatihan dataset versi v1, dengan subset validasi digunakan untuk memantau pelatihan. Pelatihan berjalan selama 300 *epoch*, dan *loss* kotak, kelas, dan objek serta mAP validasi dicatat setiap *epoch*. Pelatihan dilakukan pada layanan pelatihan GPU berbasis awan milik Roboflow, dimulai dari bobot terlatih tanpa *checkpoint* khusus, dengan citra masukan beresolusi 640×640 piksel. Konfigurasi dirangkum pada Tabel 2 agar pengaturan pelatihan dapat direproduksi.

Tabel 2. Konfigurasi model dan pelatihan

Parameter	Pengaturan
Model	YOLO26 <i>Object Detection (Small)</i> , YOLO26s; dua kelas (<i>car</i> , <i>free</i>)
Platform	Pelatihan awan Roboflow (GPU); bobot awal terlatih tanpa <i>checkpoint</i> khusus
Dataset	v1; 1.921 citra, <i>stretch resize</i> 640×640 piksel
Ukuran masukan	640×640 piksel
Epoch	300
Ambang kepercayaan	34% (optimal F1)

Deteksi disaring dengan ambang kepercayaan (*confidence threshold*). Ambang ditetapkan sebesar 34%, yaitu nilai tempat *F1-score* mencapai maksimum pada kurva F1–kepercayaan di *metrics explorer* platform. Deteksi pada atau di atas ambang ini digunakan baik untuk evaluasi maupun untuk estimasi okupansi.

Untuk estimasi okupansi, deteksi pada satu citra lahan parkir dihitung per kelas. Jumlah deteksi *car* dianggap sebagai jumlah ruang terisi, dan kapasitas total adalah jumlah deteksi *car* dan *free*, $C = N_{car} + N_{free}$. Jumlah terisi awal $N_0 = N_{car}$ dan okupansi awal $O_0 = (N_0 / C) \times 100\%$ kemudian menentukan keadaan awal simulasi pada Subbab 3.6. Terakhir, model dievaluasi pada subset uji menggunakan metrik pada Subbab 3.4.

3.4 Evaluasi Model

Model terlatih dievaluasi pada subset validasi dan uji (Subbab 3.2)

menggunakan metrik dan perangkat evaluasi Roboflow. Subset uji tidak digunakan dalam pelatihan dan berfungsi sebagai penilaian akhir pada citra yang belum pernah dilihat. Keluaran evaluasi yang digunakan meliputi grafik pelatihan ($mAP@50$, $mAP@50:95$, serta $loss$ kotak, kelas, dan objek per *epoch*) untuk memeriksa konvergensi selama 300 *epoch*; presisi, *recall*, dan *F1-score* pada ambang kepercayaan terpilih; kurva ambang kepercayaan untuk memilih ambang operasi (34%); AP per kelas ($mAP@50$) pada subset validasi dan uji; kinerja per kelas dan *confusion matrix* (prediksi benar, salah klasifikasi, negatif palsu, dan positif palsu); serta *vector analysis* untuk mengidentifikasi adegan sulit.

Presisi (*P*), *recall* (*R*), dan *F1-score* dihitung dari jumlah TP, FP, dan FN yang didefinisikan pada Subbab 2.2:

$$P = TP / (TP + FP)$$

$$R = TP / (TP + FN) \quad (1)$$

$$F1 = 2PR / (P + R) \quad (2)$$

Average precision (AP) adalah luas di bawah kurva presisi–*recall* suatu kelas, dan mAP adalah rata-ratanya atas *K* kelas ($K = 2$: *car* dan *free*):

$$mAP = (1/K) \sum AP_k \quad (3)$$

Presisi, *recall*, dan *F1-score* dilaporkan pada ambang kepercayaan yang memaksimalkan *F1-score* pada kurva ambang kepercayaan (34%), dan ambang yang sama digunakan untuk hasil per kelas dan estimasi okupansi. Metrik dilaporkan secara keseluruhan dan per kelas. Roboflow mencatat bahwa metriknya dilaporkan oleh penyedia model dan mungkin tidak mengikuti teknik evaluasi standar industri. Karena itu, hasil ditafsirkan sebagai indikator kinerja model pada dataset ini, dan hasil validasi dan uji dibandingkan untuk memeriksa konsistensinya.

3.5 Pengujian Video dan Akuisisi Data

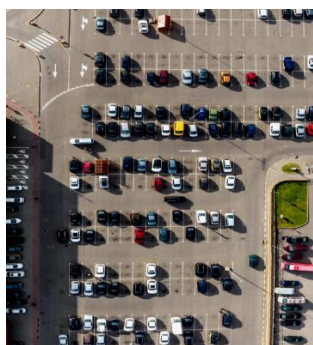
Model YOLO terlatih diuji menggunakan satu video parkir tampak

udara yang disediakan untuk penelitian ini. Video memperlihatkan area parkir dengan banyak kendaraan dan ruang parkir bertanda. Metadata berkas menunjukkan durasi 8,72 detik, resolusi 1920×1080 piksel, dan laju 25 *frame* per detik, sehingga klip memuat sekitar 218 *frame*.

Video dikirim ke *workflow* inferensi Roboflow yang telah dipublikasikan untuk diproses *frame* demi *frame*. Kelas target adalah *car* dan *free*, yang berkaitan dengan kendaraan dan ruang yang diprediksi tersedia. Prediksi model menghasilkan label objek dan koordinat *bounding box* untuk visualisasi pada setiap *frame*. Inferensi lokal melalui API juga dijajaki, tetapi *workflow* Roboflow menjadi jalur pengujian utama. Prosedur saat ini tidak menyertakan pelacakan multi-objek eksplisit; oleh karena itu, prediksi diperlakukan sebagai deteksi tingkat *frame* dan bukan identitas kendaraan yang persisten. Ringkasan pengujian disajikan pada Tabel 3 dan contoh *frame* video pada Gambar 3.

Tabel 3. Ringkasan pengaturan pengujian video

Parameter	Keterangan
Sumber video	Video parkir tampak udara (Vecteezy, berdasarkan nama berkas); format MP4
Durasi	8,72 detik
Ukuran <i>frame</i>	1920×1080 piksel
Laju <i>frame</i>	25 <i>frame</i> /detik
Perkiraan jumlah <i>frame</i>	218 <i>frame</i> (durasi $\times$ laju)
Inferensi	Per <i>frame</i> , seluruh <i>frame</i> dikirim ke <i>workflow</i> Roboflow (YOLO; <i>car</i> dan <i>free</i>)
Deret waktu okupansi	Tidak diekstraksi; deteksi tingkat <i>frame</i> dipakai untuk estimasi okupansi awal
Kapasitas parkir	Diestimasi 174 ruang (median 10 <i>frame</i> pertama)



Gambar 3. *Frame* representatif dari video parkir yang digunakan untuk pengujian (sebelum *overlay* deteksi)

3.6 Simulasi Okupansi Parkir Dinamis

Perilaku dinamis area parkir dianalisis dengan simulasi stokastik waktu diskret yang diimplementasikan dengan Python. Area parkir direpresentasikan sebagai satu kumpulan C ruang identik, dan keadaan sistem pada langkah waktu t adalah jumlah ruang terisi N_t . Model tidak merepresentasikan posisi tiap ruang, jalur kendaraan, maupun waktu pencarian; model hanya melacak perubahan jumlah total kendaraan yang parkir seiring kendaraan datang dan pergi.

Asumsi model. Waktu maju dalam langkah berdurasi sama Δt sepanjang horizon T langkah. Kendaraan datang secara independen terhadap okupansi saat ini, dan kendaraan yang datang hanya diterima bila ada ruang kosong; bila tidak, kendaraan pergi tanpa parkir. Setiap kendaraan yang parkir berangkat secara independen dengan peluang p yang sama pada tiap langkah, sehingga lama tinggal berdistribusi geometrik dengan rerata $1/p$ langkah. Kapasitas C konstan, dan keadaan awal diperoleh dari detektor (Subbab 3.3).

Variabel dan parameter. Notasi model dirangkum pada Tabel 4. Variabel keadaan adalah N_t ; jumlah kedatangan A_t dan keberangkatan D_t adalah masukan acak; dan skenario ditentukan oleh faktor beban ρ_s . Keadaan awal berasal dari detektor, sedangkan peluang keberangkatan dan faktor beban merupakan asumsi skenario karena video

yang digunakan terlalu pendek untuk mengestimasi laju kedatangan dan keberangkatan.

Tabel 4. Variabel dan parameter model simulasi

Simbol	Deskripsi	Nilai atau sumber
t	Indeks langkah waktu ($t = 0, 1, \dots, T$)	—
Δt	Durasi satu langkah waktu	1 menit
T	Jumlah langkah simulasi (horizon)	480 langkah (8 jam)
C	Kapasitas parkir (jumlah ruang)	$N_{\text{car}} + N_{\text{free}} = 174$
N_t	Jumlah ruang terisi pada langkah t (variabel keadaan)	$N_0 = N_{\text{car}} = 102$
O_t	Persentase okupansi pada langkah t	$(N_t / C) \times 100\%$
A_t	Jumlah kendaraan datang pada langkah t	Poisson dengan rerata λ_s
D_t	Jumlah kendaraan berangkat pada langkah t	Binomial, N_t percobaan, peluang p
p	Peluang keberangkatan kendaraan per langkah	0,02 (rerata tinggal 50 menit)
ρ_s	Faktor beban skenario s	0,5; 1,0; 1,5
λ_s	Rerata kedatangan per langkah pada skenario s	ρ_s, p, C
R	Jumlah pengulangan (<i>run</i>) independen per skenario	1.000

Persamaan keadaan. Pada setiap langkah, jumlah kedatangan dan keberangkatan diambil dari

$$A_t \sim \text{Poisson}(\lambda_s), \\ D_t \sim \text{Binomial}(N_t, p) \quad (4)$$

dan jumlah ruang terisi diperbarui sebagai

$$N_{t+1} = \min(C, \max(0, N_t + A_t - D_t)) \quad (5)$$

Persamaan ini membatasi okupansi pada rentang nol hingga kapasitas total, sehingga kedatangan yang tidak menemukan ruang kosong ditolak. Persentase okupansi parkir adalah

$$O_t = (N_t / C) \times 100\% \quad (6)$$

Skenario simulasi. Tiga skenario dengan kedatangan kendaraan rendah, sedang, dan tinggi dibandingkan. Skenario hanya berbeda pada laju kedatangan rerata, yang didefinisikan melalui faktor beban ρ_s , yaitu rasio antara aliran kedatangan yang diharapkan dan

aliran keberangkatan yang diharapkan pada area parkir penuh (pC):

$$\lambda_s = \rho_s \cdot p \cdot C \quad (7)$$

Tanpa batas kapasitas, okupansi yang diharapkan konvergen ke $\lambda_s/p = \rho_s C$. Bila ρ_s kurang dari satu, okupansi menetap di bawah kapasitas; bila sama dengan satu, okupansi mendekati kapasitas; dan bila lebih dari satu, permintaan melebihi kapasitas sehingga area parkir cenderung jenuh. Ketiga skenario dirangkum pada Tabel 5. Peluang keberangkatan, keadaan awal, kapasitas, langkah waktu, dan horizon identik pada semua skenario.

Tabel 5. Skenario simulasi

Skenario	ρ	λ (kendaraan / langkah)	Batas harapan λ/p	Perilaku yang diharapkan
Rendah	0,5	$0,5 pC = 1,74$	87 (50% C)	Tetap di bawah kapasitas
Sedang	1,0	$1,0 pC = 3,48$	174 (100% C)	Mendekati kapasitas
Tinggi	1,5	$1,5 pC = 5,22$	261 (150% C)	Cepat jenuh

Prosedur simulasi. Setiap pengulangan dimulai dari keadaan awal N_0 dan kapasitas C dari detektor. Untuk $t = 0, 1, \dots, T - 1$, kedatangan A_t dan keberangkatan D_t diambil, N_{t+1} dihitung dengan Persamaan (5), dan O_t dicatat (Gambar 4). Karena kedatangan dan keberangkatan bersifat acak, setiap skenario diulang sebanyak R kali secara independen dengan *seed* acak yang berbeda, dengan *seed* dasar ditetapkan 42 untuk reproduibilitas. Hasil dilaporkan sebagai rerata dan simpangan baku antar pengulangan.

```
Masukan: C, N0, p, T, R, rho(s)
untuk tiap skenario s:
  lambda <- rho(s) * p * C
  untuk r = 1..R:
    N <- N0
    untuk t = 0..T-1:
      A ~ Poisson(lambda)
      D ~ Binomial(N, p)
      N <- min(C, max(0, N+A-D))
      O[t] <- N / C * 100
    hitung Tfull, Ffull, Orata
  rata-rata dan simpangan baku
  antar pengulangan
```

Gambar 4. Algoritma simulasi okupansi parkir

Ukuran keluaran. Tiga ukuran diturunkan dari lintasan simulasi. Pertama, waktu hingga kapasitas penuh, yaitu langkah waktu pertama saat seluruh ruang terisi:

$$T_{full} = \min \{ t : N_t = C \} \quad (8)$$

Jika okupansi tidak mencapai kapasitas dalam horizon, pengulangan tersebut dicatat tidak mencapai kapasitas penuh, dan porsi pengulangan yang mencapainya dilaporkan bersama T_{full} . Kedua, porsi langkah waktu pada kapasitas penuh:

$$F_{full} = (1/T) \sum 1(N_t = C) \times 100\% \quad (9)$$

dengan $1(\cdot)$ bernilai satu bila kondisi terpenuhi dan nol bila tidak. Ketiga, okupansi rata-rata:

$$\bar{O} = (1/T) \sum O_t \quad (10)$$

Skenario dibandingkan menggunakan lintasan okupansi rerata O_t beserta simpangan bakunya antar pengulangan, yang diplot untuk ketiga skenario dalam satu grafik, serta menggunakan T_{full} , F_{full} , dan $\bar{O}$ dalam tabel ringkasan. Perbandingan difokuskan pada bagaimana laju kedatangan yang lebih tinggi menggeser waktu menuju kejenuhan dan menambah waktu pada kapasitas penuh.

3.7 Evaluasi Simulasi dan Analisis Data

Hasil simulasi dianalisis menggunakan Python untuk membandingkan dinamika okupansi pada tiga skenario kedatangan (Subbab 3.6). Setiap skenario dievaluasi selama 480 langkah waktu (delapan jam) dan diulang 1.000 kali dengan *seed* tetap 42, lalu dirangkum dengan rerata dan simpangan baku antar pengulangan. T_{full} , F_{full} , dan $\bar{O}$ dibandingkan dalam tabel ringkasan, sedangkan lintasan okupansi rerata beserta simpangan bakunya diplot untuk melihat perubahan okupansi dari waktu ke waktu. Simulasi juga diperiksa agar jumlah ruang terisi tetap berada antara nol dan kapasitas total, dan persentase okupansi dalam rentang 0%–100%.

Karena parameter kedatangan dan keberangkatan diasumsikan dan bukan diestimasi dari data lalu lintas teramati,

hasil mencerminkan perilaku sistem parkir tersimulasi pada kondisi hipotetis dan bukan prakiraan permintaan parkir yang sebenarnya.

4. HASIL DAN PEMBAHASAN

4.1 Evaluasi Model YOLO

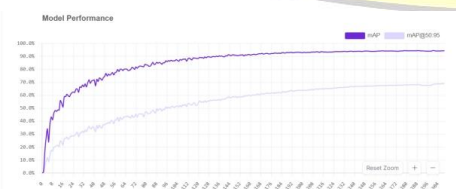
Model YOLO26s dievaluasi pada subset uji Roboflow untuk menilai kinerjanya mendeteksi kendaraan dan ruang parkir yang tersedia, mencakup perilaku pelatihan, metrik deteksi keseluruhan, dan kesalahan prediksi untuk kelas *car* dan *free*.

4.1.1 Kinerja Pelatihan

Kurva pelatihan menunjukkan tren penurunan pada *loss* kotak, *loss* klasifikasi, dan *loss* objek selama 300 *epoch* pelatihan (Gambar 5). Hal ini menunjukkan bahwa model secara bertahap mengurangi galat pelatihannya seiring optimisasi. Grafik kinerja model (Gambar 6) juga menunjukkan peningkatan metrik deteksi yang dilaporkan sebelum kurva berangsur mendatar pada *epoch-epoch* akhir. Kurva-kurva ini menggambarkan proses pelatihan; metrik pada subset uji yang dilaporkan pada subbab berikutnya digunakan secara terpisah untuk merangkum kinerja evaluasi.



Gambar 5. Kurva *loss* pelatihan untuk *loss* kotak, klasifikasi, dan objek



Gambar 6. Metrik kinerja model sepanjang *epoch* pelatihan

4.1.2 Metrik Deteksi Keseluruhan

Hasil subset uji yang dilaporkan Roboflow dirangkum pada Tabel 6. Model mencapai presisi 93,6%, *recall*

91,8%, dan *F1-score* 92,7%. *mAP@50* yang dilaporkan sebesar 95,5%, sedangkan *mAP@50:95* sebesar 71,5%. Hasil ini menunjukkan kinerja deteksi yang kuat pada ambang *IoU* 0,50. Nilai *mAP@50:95* yang lebih rendah mencerminkan evaluasi yang lebih ketat karena merata-ratakan presisi pada beberapa ambang *IoU*, sehingga lebih menekankan kualitas lokalisasi *bounding box*.

Tabel 6. Metrik deteksi keseluruhan YOLO26s pada set uji Roboflow

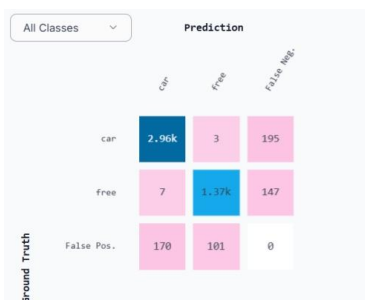
Metrik	Hasil uji	Deskripsi
Presisi	93,6%	Proporsi deteksi prediksi yang benar
<i>Recall</i>	91,8%	Proporsi objek acuan yang terdeteksi model
<i>F1-score</i>	92,7%	Rerata harmonik presisi dan <i>recall</i>
<i>mAP@50</i>	95,5%	<i>Mean average precision</i> pada <i>IoU</i> = 0,50
<i>mAP@50:95</i>	71,5%	<i>Mean average precision</i> yang dirata-ratakan pada ambang <i>IoU</i> 0,50–0,95

4.1.3 Kinerja Deteksi per Kelas

Confusion matrix diperiksa untuk menilai prediksi model pada kelas *car* dan *free* (Gambar 7). Pada kelas *car*, sekitar 2,96 ribu instans diprediksi benar sebagai *car*, tiga instans diprediksi sebagai *free*, dan 195 tercatat sebagai negatif palsu. Pada kelas *free*, sekitar 1,37 ribu instans diprediksi benar sebagai *free*, tujuh diprediksi sebagai *car*, dan 147 tercatat sebagai negatif palsu. Matriks juga melaporkan 170 prediksi positif palsu untuk *car* dan 101 untuk *free*.

Confusion matrix menunjukkan bahwa model pada umumnya dapat membedakan kedua kelas, meskipun deteksi yang terlewat dan prediksi positif palsu masih ada. Kesalahan ini relevan bagi analisis video parkir berikutnya karena deteksi yang salah atau terlewat dapat memengaruhi penafsiran apakah suatu ruang terisi atau tersedia. Nilai matriks disalin dalam format singkat yang ditampilkan antarmuka Roboflow; oleh

karena itu, nilai yang bertanda “k” bersifat aproksimasi.



		Prediction		
		car	free	False Neg.
Ground Truth	car	2,96k	3	195
	free	7	1,37k	147
	False Pos.	170	101	0

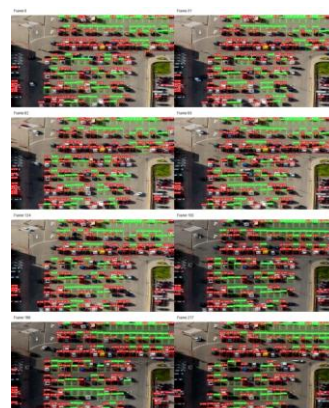
Gambar 7. *Confusion matrix* untuk kelas *car* dan *free*

Secara keseluruhan, YOLO26s mencapai metrik deteksi agregat yang tinggi pada set uji. Namun, positif palsu dan negatif palsu yang tersisa perlu dipertimbangkan dalam menafsirkan hasil pada video parkir (Subbab 4.2), karena hasil evaluasi ini tidak dengan sendirinya menetapkan akurasi deret waktu okupansi parkir.

4.2 Hasil Deteksi pada Video Parkir

4.2.1 Hasil Deteksi Kualitatif

Model YOLO26s terlatih diterapkan pada video lahan parkir untuk mengidentifikasi ruang parkir terisi dan kosong. Video diproses *frame* demi *frame*, dan hasilnya divisualisasikan dengan *bounding box* dan skor kepercayaan: kelas *car* (ruang terisi) berwarna merah dan kelas *free* (ruang kosong) berwarna hijau (Gambar 8). Model mengidentifikasi ruang parkir pada beberapa baris kendaraan dalam pandangan atas, baik ruang terisi maupun kosong dalam satu adegan, termasuk area berkepadatan tinggi.



Gambar 8. Hasil deteksi ruang parkir secara kualitatif: ruang terisi (*car*, merah) dan ruang kosong (*free*, hijau), beserta skor kepercayaan

Namun, visualisasi juga memperlihatkan *bounding box* yang saling tumpang tindih dan variasi skor kepercayaan, terutama pada baris yang padat; sebagian prediksi memiliki skor tinggi, sedangkan yang lain mendekati ambang deteksi. Variasi ini dapat memengaruhi konsistensi estimasi kapasitas dan okupansi. Hasil kualitatif menunjukkan keterterapan detektor untuk estimasi okupansi berbasis video, tetapi tidak menetapkan akurasi kuantitatifnya; penilaian tersebut memerlukan presisi, *recall*, *F1-score*, dan mAP dari set validasi dan uji (Subbab 4.1).

4.2.2 Kapasitas Parkir dan Okupansi Awal

Hasil deteksi digunakan untuk mengestimasi kapasitas parkir total dan okupansi awal. Mengikuti prosedur pada Subbab 3.3, jumlah ruang terisi (N_{car}) dan kosong (N_{free}) diperoleh dari kelas hasil deteksi, kapasitas total (C) dihitung sebagai jumlah keduanya, dan okupansi awal (O_0) sebagai rasio ruang terisi terhadap kapasitas total.

Pada *frame* pertama yang diproses, model mendeteksi 102 ruang terisi dan 75 ruang kosong, sehingga kapasitas diestimasi 177 ruang dan okupansi awal 57,6%. Untuk mengurangi pengaruh variasi deteksi tingkat *frame*, jumlah deteksi median pada 10 *frame* pertama

digunakan untuk menetapkan kondisi awal simulasi. Jumlah median adalah 102 ruang terisi dan 72 ruang kosong, yang menghasilkan estimasi kapasitas total 174 ruang dan okupansi awal 58,6% (Tabel 7). Dengan demikian, sekitar 41,4% kapasitas yang diestimasi tersedia pada awal simulasi. Nilai-nilai ini digunakan untuk menginisialisasi model okupansi parkir waktu diskret, dengan $N_0 = 102$ dan $C = 174$.

Tabel 7. Ringkasan estimasi kapasitas parkir dan okupansi awal

Parameter	Frame pertama	Median 10 frame pertama
Ruang terisi (N_{car})	102	102
Ruang kosong (N_{free})	75	72
Estimasi kapasitas (C)	177	174
Okupansi awal (O_0)	57,6%	58,6%
Proporsi ruang kosong (estimasi)	42,4%	41,4%

Catatan: nilai median dari 10 frame pertama digunakan sebagai kondisi awal simulasi okupansi parkir dinamis.

Kapasitas yang diestimasi bervariasi antara 150 dan 194 ruang pada video yang diproses, yang menunjukkan bahwa jumlah deteksi tidak sepenuhnya konsisten antar *frame*. Variasi ini menyoroti kemungkinan pengaruh derau deteksi terhadap estimasi kapasitas parkir dan mendukung penggunaan jumlah median dibandingkan hanya bergantung pada satu *frame*.

Video memuat 218 *frame* pada 25 *frame* per detik (sekitar 8,7 detik). Meskipun objek terdeteksi berubah antar *frame*, durasi ini tidak memadai untuk menetapkan laju kedatangan dan keberangkatan kendaraan yang andal. Karena itu, variasi temporal yang teramati tidak digunakan untuk menentukan laju kedatangan atau rerata durasi parkir; estimasi kapasitas dan okupansi awal hanya dipakai sebagai kondisi awal simulasi, sedangkan perilaku kedatangan

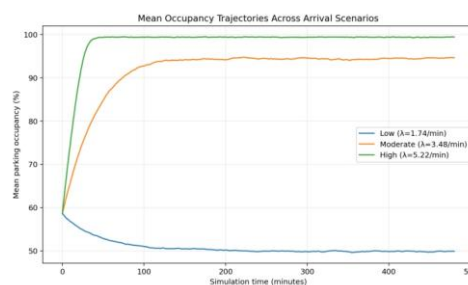
dan keberangkatan ditetapkan melalui asumsi skenario pada Subbab 3.6. Kapasitas 174 ruang dan okupansi awal 58,6% menjadi kondisi dasar untuk membandingkan skenario kedatangan rendah, sedang, dan tinggi.

4.3 Hasil Simulasi Stokastik

Simulasi stokastik diimplementasikan dengan Python untuk mengkaji okupansi parkir pada tiga skenario kedatangan kendaraan hipotetis. Kapasitas parkir ditetapkan 174 ruang dan okupansi awal 102 kendaraan. Setiap skenario disimulasikan selama 480 langkah waktu satu menit dan diulang 1.000 kali dengan *seed* acak tetap 42. Peluang keberangkatan ditetapkan 0,02 per langkah waktu, dan laju kedatangan divariasikan melalui faktor beban skenario. Laju kedatangan dihitung sebagai $\lambda = \rho p C$, yang menghasilkan 1,74; 3,48; dan 5,22 kendaraan per menit untuk skenario rendah, sedang, dan tinggi. Laju ini merupakan masukan model hipotetis dan bukan estimasi dari pergerakan kendaraan yang teramati pada video.

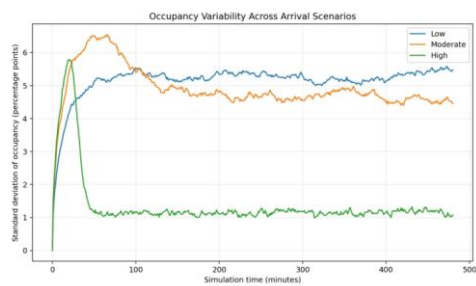
4.3.1 Lintasan Okupansi

Gambar 9 membandingkan lintasan okupansi rerata pada ketiga skenario kedatangan. Lintasan merangkum persentase okupansi rerata dari 1.000 pengulangan simulasi independen pada setiap langkah waktu. Perbedaan antar skenario mencerminkan perbedaan laju kedatangan yang diasumsikan pada kapasitas, okupansi awal, peluang keberangkatan, dan durasi simulasi yang sama.



Gambar 9. Lintasan okupansi parkir rerata pada ketiga skenario kedatangan

Kurva simpangan baku pada Gambar 10 menunjukkan variabilitas antar pengulangan simulasi dari waktu ke waktu. Sebaran ini penting karena lintasan rerata saja tidak memperlihatkan seberapa konsisten perilaku tiap pengulangan. Karena okupansi dibatasi oleh kapasitas total, lintasan dapat mendekati batas atas ketika jumlah ruang terisi mendekati 174.



Gambar 10. Simpangan baku okupansi parkir pada ketiga skenario kedatangan

4.3.2 Waktu hingga Kapasitas Penuh

Waktu hingga kapasitas penuh (T_{full}) didefinisikan sebagai menit simulasi pertama saat seluruh 174 ruang terisi. Pengulangan yang tidak mencapai kapasitas penuh dalam periode 480 menit dicatat sebagai tidak mencapai kapasitas penuh. Tabel 8 melaporkan proporsi pengulangan yang mencapai kapasitas dan rerata waktu hingga kapasitas penuh pada pengulangan tersebut.

Tabel 8. Hasil waktu hingga kapasitas penuh dari 1.000 pengulangan simulasi per skenario

Skenario	Pengulangan mencapai kapasitas (%)	Rerata T_{full} (menit; pengulangan yang mencapai)
Rendah	0,0	Tidak tercapai
Sedang	100,0	124,5
Tinggi	100,0	30,5

Ukuran waktu hingga kapasitas penuh ditafsirkan bersama proporsi pengulangan yang mencapai kapasitas. Rerata bersyarat saja dapat menyesatkan apabila banyak pengulangan tidak menjadi penuh dalam periode simulasi.

4.3.3 Durasi Kapasitas Penuh dan Okupansi Rata-rata

Proporsi waktu simulasi pada kapasitas penuh (F_{full}) dan persentase okupansi rata-rata dihitung untuk setiap pengulangan, kemudian dirata-ratakan pada 1.000 pengulangan. Tabel 9 membandingkan ukuran-ukuran ini pada skenario kedatangan rendah, sedang, dan tinggi.

Tabel 9. Durasi kapasitas penuh dan okupansi rata-rata pada ketiga skenario

Skenario	Rerata waktu pada kapasitas penuh (%)	Okupansi rata-rata (%)
Rendah	0,00	50,78
Sedang	10,01	91,76
Tinggi	62,20	98,27

Hasil ini menggambarkan perilaku simulasi yang ditetapkan pada laju kedatangan hipotetis. Hasil tersebut tidak boleh ditafsirkan sebagai prakiraan permintaan parkir yang sebenarnya, karena laju kedatangan dan peluang keberangkatan diasumsikan untuk analisis skenario dan bukan diestimasi dari deret waktu kedatangan dan keberangkatan kendaraan yang teramati.

4.4 Pembahasan

Hasil simulasi sejalan dengan perilaku yang diharapkan dari model pada Subbab 3.6. Pada skenario rendah, batas harapan λ/p sebesar 87 ruang (50% kapasitas) berada di bawah kapasitas, sehingga okupansi menurun dari 58,6% mendekati batas tersebut dan tidak ada pengulangan yang mencapai kapasitas penuh. Pada skenario sedang, batas harapan sama dengan kapasitas (174 ruang), sehingga okupansi mendekati kapasitas dan penuh pada seluruh pengulangan setelah rerata 124,5 menit, dengan 10,01% waktu simulasi berada pada kapasitas penuh. Pada skenario tinggi, batas harapan (261 ruang) melebihi kapasitas sehingga area parkir jenuh setelah rerata 30,5 menit dan menghabiskan 62,20% waktu pada kapasitas penuh. Laju kedatangan yang

lebih tinggi dengan demikian mempercepat kejenuhan dan memperpanjang waktu pada kapasitas penuh.

Dibandingkan dengan penelitian terdahulu, penelitian ini melengkapi studi deteksi okupansi parkir (Yuldashev dkk., 2023; Sobirin dkk., 2024; Saputra dkk., 2024; Pore & Nemade, 2025), yang berfokus pada kondisi parkir saat ini, dengan komponen yang memperlihatkan perubahan okupansi pada skenario kedatangan terkendali. Tujuannya berbeda dari prediksi berbasis data historis (Ye dkk., 2024): simulasi memisahkan pengaruh laju kedatangan terpilih, sedangkan prediksi menangkap pola temporal okupansi teramati. Keterbatasannya ialah laju kedatangan dan peluang keberangkatan berupa asumsi skenario; video uji hanya berdurasi 8,72 detik dengan kapasitas yang bervariasi antara 150 dan 194 ruang tanpa validasi terhadap data acuan (*ground truth*); dan tidak ada pelacakan kendaraan. Video yang lebih panjang, validasi kapasitas, serta pelacakan kendaraan akan memungkinkan laju kedatangan dan keberangkatan diestimasi dari pergerakan teramati.

5. KESIMPULAN

Penelitian ini mengembangkan kerangka kerja yang mengintegrasikan deteksi ruang parkir berbasis YOLO26s dengan simulasi okupansi stokastik. Pada set uji Roboflow, detektor mencapai presisi 93,6%, *recall* 91,8%, *F1-score* 92,7%, *mAP@50* 95,5%, dan *mAP@50:95* 71,5%. Positif palsu dan negatif palsu yang tersisa menunjukkan bahwa galat deteksi masih perlu diperhitungkan ketika keluarannya digunakan sebagai masukan simulasi.

Pada video parkir, median deteksi sepuluh *frame* pertama menghasilkan estimasi kapasitas 174 ruang, dengan 102 ruang terisi dan 72 ruang kosong, atau okupansi awal 58,6%. Kapasitas yang diestimasi bervariasi antara 150 dan 194 ruang antar *frame*, sehingga nilai-nilai ini

merupakan estimasi dan bukan hasil pengukuran terverifikasi.

Dengan kondisi awal tersebut, simulasi membandingkan tiga skenario kedatangan selama 480 langkah satu menit dan 1.000 pengulangan. Skenario rendah tetap berada di bawah kapasitas, dengan okupansi rata-rata 50,78%, dan tidak pernah mencapai kapasitas penuh. Skenario sedang dan tinggi mencapai kapasitas penuh pada seluruh pengulangan setelah rerata 124,5 dan 30,5 menit. Okupansi rata-rata keduanya 91,76% dan 98,27%, dengan 10,01% dan 62,20% waktu simulasi berada pada kapasitas penuh. Laju kedatangan yang lebih tinggi dengan demikian mempercepat kejenuhan dan memperpanjang waktu pada kapasitas penuh.

Keterbatasan utama penelitian ini adalah bahwa laju kedatangan dan peluang keberangkatan merupakan asumsi skenario dan bukan estimasi dari lalu lintas teramati. Hasil menggambarkan perilaku sistem yang dimodelkan dan bukan prakiraan permintaan parkir yang sebenarnya.

6. UCAPAN TERIMA KASIH

Penulis berterima kasih kepada Departemen Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Negeri Medan, atas dukungan yang diberikan selama penelitian ini. Penulis juga mengapresiasi komunitas Roboflow Universe yang telah menyediakan dataset parkir secara terbuka, serta Vecteezy atas video parkir yang digunakan dalam pengujian. Penelitian ini tidak menerima pendanaan khusus dari lembaga pendanaan mana pun di sektor publik, komersial, maupun nirlaba.

DAFTAR PUSTAKA

Mendoza-Silva, G. M., Gould, M., Montoliu, R., Torres-Sospedra, J., & Huerta, J. (2019). An occupancy simulator for a smart parking system: Developmental design and experimental considerations.

- ISPRS International Journal of Geo-Information*, 8(5), 212.
<https://doi.org/10.3390/ijgi8050212>
- Pore, A. A., & Nemade, P. D. (2025). A novel approach to detect parking space occupancy for efficient urban management. *Civil Engineering Journal*, 11(2), 658–669.
<https://doi.org/10.28991/CEJ-2025-011-02-015>
- Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 779–788.
<https://doi.org/10.1109/CVPR.2016.91>
- Saputra, A., Giawa, B., & Khana, R. (2024). *Vehicle and parking space detection for smart parking systems using the YOLOv5 method* (pp. 458–466). INSTICC.
<https://doi.org/10.5220/0012584700003821>
- Sharma, N., Baral, S., Paing, M. P., & Chawuthai, R. (2023). Parking time violation tracking using YOLOv8 and tracking algorithms. *Sensors*, 23(13), 5843. <https://doi.org/10.3390/s23135843>
- Sobirin, M., Tiorivaldi, & Mufit, C. (2024). *Car parking space detection using YOLOv8* (pp. 394–398). INSTICC.
<https://doi.org/10.5220/0012582600003821>
- Ultralytics. (t.t.). *Performance metrics deep dive*. Diakses dari <https://docs.ultralytics.com/guides/yolo-performance-metrics/>
- Ye, W., Kuang, H., Li, J., Lai, X., & Qu, H. (2024). A parking occupancy prediction method incorporating time series decomposition and temporal pattern attention mechanism. *IET Intelligent Transport Systems*, 18(1), 58–71.
<https://doi.org/10.1049/itr2.12433>
- Yuldashev, Y., Mukhiddinov, M., Abdusalomov, A. B., Nasimov, R., & Cho, J. (2023). Parking lot occupancy detection with improved MobileNetV3. *Sensors*, 23(17), 7642.
<https://doi.org/10.3390/s23177642>