

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian yang Relevan

Penelitian yang relevan digunakan untuk mengkaji perkembangan penelitian mengenai sistem parkir cerdas berbasis *computer vision* dan *deep learning*. Kajian dilakukan terhadap penelitian terdahulu yang membahas deteksi kendaraan, identifikasi ruang parkir, klasifikasi kondisi okupansi, serta pengembangan sistem *monitoring* parkir. Kajian penelitian terdahulu diperlukan untuk mengetahui pendekatan yang telah digunakan, hasil yang telah dicapai, serta keterbatasan yang masih dapat dikembangkan. Hasil kajian tersebut selanjutnya digunakan sebagai dasar dalam menentukan pendekatan dan posisi penelitian pada pengembangan *prototype* sistem parkir cerdas.

Penelitian terdahulu yang dikaji dapat dikelompokkan berdasarkan beberapa fokus utama. Kelompok pertama membahas penerapan *deep learning* dan *computer vision* untuk mendeteksi kendaraan pada area parkir. Kelompok kedua membahas deteksi dan klasifikasi okupansi ruang parkir, yaitu penentuan kondisi ruang parkir sebagai Kosong atau Terisi. Kelompok ketiga membahas penggunaan YOLO dan model deteksi objek lainnya untuk meningkatkan kemampuan pendeteksian kendaraan. Kelompok keempat membahas pemanfaatan *Region of Interest* (ROI), *polygon*, dan informasi posisi kendaraan untuk menghubungkan hasil deteksi dengan ruang parkir tertentu. Kelompok kelima membahas pengembangan sistem *monitoring* parkir secara terintegrasi melalui pemanfaatan kamera, pemrosesan video, dan penyajian informasi ketersediaan ruang parkir kepada pengguna maupun pengelola.

Berdasarkan kajian terhadap penelitian terdahulu, sejumlah penelitian menunjukkan bahwa *computer vision* dan *deep learning* dapat dimanfaatkan untuk mendukung proses identifikasi kendaraan dan kondisi ruang parkir. Sebagian penelitian berfokus pada peningkatan performa model deteksi, sedangkan penelitian lainnya mengembangkan metode klasifikasi okupansi atau penentuan posisi ruang parkir. Pendekatan berbasis YOLO juga digunakan dalam berbagai penelitian untuk mendukung proses deteksi objek pada lingkungan parkir. Meskipun demikian, masih terdapat ruang pengembangan pada aspek integrasi antara deteksi kendaraan, pemetaan ruang parkir menggunakan *polygon Region of Interest*, pembacaan beberapa *frame* secara berurutan,

penyimpanan hasil pembacaan, dan penyajian informasi kondisi parkir melalui suatu sistem *monitoring*.

Berdasarkan kesenjangan tersebut, penelitian ini diarahkan pada pengembangan *prototype* sistem parkir cerdas menggunakan model YOLO26n hasil *fine-tuning* untuk mendeteksi kendaraan pada video CCTV simulasi. Hasil deteksi kendaraan dipetakan ke dalam sebelas *polygon Region of Interest* (ROI) yang merepresentasikan slot A1 sampai A5 dan B1 sampai B6. Posisi kendaraan yang terdeteksi kemudian dikaitkan dengan masing-masing *polygon* untuk menentukan status slot sebagai Kosong atau Terisi. Penentuan status tidak hanya dilakukan berdasarkan satu *frame*, tetapi menggunakan pembacaan beberapa *frame* secara berurutan untuk membantu mengurangi perubahan status akibat hasil deteksi yang tidak stabil pada satu *frame*.

Hasil penentuan status selanjutnya diintegrasikan dengan *backend* FastAPI, basis data MySQL, dan *dashboard* berbasis PHP. MySQL digunakan untuk menyimpan status terbaru setiap slot serta riwayat hasil pembacaan status parkir, sedangkan *dashboard* digunakan untuk menyajikan jumlah slot Kosong dan Terisi, status setiap slot, *snapshot* CCTV, serta laporan historis status parkir. Dengan demikian, posisi penelitian ini tidak terletak pada pengembangan arsitektur *deep learning* baru, tetapi pada integrasi hasil deteksi kendaraan, pemetaan *polygon* ROI, pembacaan temporal, penyimpanan data, dan penyajian informasi dalam satu *prototype* Sistem Informasi parkir cerdas. Rincian penelitian terdahulu yang menjadi dasar dan pembanding penelitian ini ditunjukkan pada Tabel 2.1.

Tabel 2. 1 Matriks Penelitian

No	Peneliti dan Tahun	Judul	Metodologi	Hasil Penelitian	Keterkaitan Penelitian
1	Schulte et al. (2021)	<i>Parking Space Management Through Deep Learning: An Approach for Automated, Low-Cost and Scalable Real-time Detection of Parking Space Occupancy</i>	<i>Design Science Research; deep learning; YOLOv4; DeepSORT; deteksi berbasis kamera</i>	Sistem berbasis kamera dan <i>deep learning</i> dapat digunakan untuk mendeteksi okupansi parkir secara otomatis sebagai alternatif <i>monitoring</i> berbasis sensor.	Menjadi dasar bahwa <i>prototype</i> parkir berbasis video dan <i>deep learning</i> dapat digunakan untuk <i>monitoring</i> okupansi. Penelitian ini mengembangkan pendekatan tersebut dengan

					YOLO26n, <i>polygon</i> ROI, dan <i>dashboard</i> .
2	Yuldashev et al. (2023)	<i>Parking Lot Occupancy Detection with Improved MobileNetV3</i>	<i>Deep learning</i> ; klasifikasi citra; Improved MobileNetV3; dataset CNRPark-EXT dan PKLot	Improved MobileNetV3 memperoleh akurasi rata-rata 98,01% pada pengujian gabungan <i>datas et</i> , lebih tinggi dibandingkan CarNet sebesar 97,03%.	Mendukung penggunaan <i>deep learning</i> untuk menentukan kondisi parkir Kosong atau Terisi. Perbedaannya, penelitian ini menggunakan deteksi kendaraan YOLO dan pencocokan dengan <i>polygon</i> ROI, bukan klasifikasi potongan citra setiap slot.
3	Grbić dan Koch (2023)	<i>Automatic Vision-Based Parking Slot Detection and Occupancy Classification</i>	<i>Computer vision</i> ; deteksi kendaraan; <i>bird's-eye view</i> ; <i>clustering</i> ; klasifikasi okupansi	Pendekatan berbasis kamera dapat mendeteksi posisi slot parkir dan menentukan status okupansi berdasarkan informasi visual.	Mendukung konsep pemetaan lokasi parkir dan penentuan status berdasarkan posisi kendaraan. Penelitian ini menggunakan <i>polygon</i> ROI yang telah ditetapkan untuk slot A1–A5 dan B1–B6.
4	Encío et al. (2023)	<i>Visual Parking Occupancy Detection Using Extended Contextual Image Information via a Multi-Branch Output ConvNeXt Network</i>	<i>Deep learning</i> ; ConvNeXt; <i>multi-branch output</i> ; analisis konteks visual	Informasi visual dari slot dan area di sekitarnya dapat membantu sistem mengenali kondisi okupansi pada lingkungan parkir yang kompleks.	Memperkuat bahwa kondisi visual lingkungan dapat memengaruhi deteksi okupansi. Penelitian ini menggunakan YOLO untuk mendeteksi kendaraan dan <i>polygon</i> ROI untuk menentukan status slot.
5	Dung & Cho (2024)	<i>Parking Lot Occupancy Detection Using Deep Learning and Fisheye Camera for AioT System</i>	<i>Deep learning</i> ; YOLOv5; kamera fisheye; AioT; aturan keputusan	Integrasi kamera, <i>deep learning</i> , dan komputasi tepi dapat digunakan untuk <i>monitoring</i> kondisi okupansi parkir.	Relevan karena sama-sama memanfaatkan kamera dan YOLO untuk menentukan status parkir. Perbedaannya, penelitian ini menggunakan video CCTV simulasi standar dan tidak

					menggunakan kamera fisheye maupun pendekatan AioT.
6	Zhao et al. (2024)	<i>CMCA-YOLO: A Study on a Real-time Object Detection Model for Parking Lot Surveillance Imagery</i>	<i>Deep learning; YOLO; deteksi objek; real-time object detection</i>	CMCA-YOLO memperoleh mAP@0.5 sebesar 0,895 dan meningkatkan kemampuan deteksi kendaraan serta pejalan kaki berukuran kecil atau saling tumpang tindih.	Mendukung penggunaan keluarga YOLO untuk deteksi kendaraan pada area parkir. Penelitian ini tidak mengembangkan arsitektur YOLO baru, tetapi mengintegrasikan YOLO26n dengan <i>polygon</i> ROI dan <i>dashboard</i> .
7	Jiang et al. (2025)	<i>Two-Stage Efficient Parking Space Detection Method Based on Deep Learning and Computer Vision</i>	<i>Deep learning; Computer vision; metode dua tahap; deteksi titik penting dan posisi parking space</i>	Deteksi parkir membutuhkan pengenalan objek sekaligus pemetaan posisi ruang parkir agar informasi dapat digunakan secara efektif.	Mendukung penggunaan ROI atau <i>polygon</i> untuk merepresentasikan lokasi parkir. Penelitian ini menggunakan <i>polygon</i> yang ditentukan secara manual untuk menghubungkan kendaraan dengan slot tertentu.
8	Fathurrahman et al. (2025)	<i>Comparative Study of YOLO Versions for Detecting Vacant Car Parking Spaces</i>	<i>Deep learning; perbandingan beberapa versi YOLO; pengujian citra dan video</i>	YOLO dapat digunakan untuk mendukung deteksi ruang parkir secara <i>real-time</i> , tetapi performanya dipengaruhi sudut kamera, kondisi lingkungan, kepadatan kendaraan, dan keteraturan parkir.	Mendukung penggunaan YOLO untuk <i>monitoring</i> parkir berbasis video. Perbedaannya, penelitian ini menggunakan satu model YOLO26n hasil <i>fine-tuning</i> dan menentukan status melalui <i>polygon</i> ROI.
9	Popereshnyak dan Chornobryvets (2025)	<i>Real-time Parking Space Monitoring System Based on Computer Vision</i>	<i>Computer vision; YOLO; OpenCV; Region of Interest; real-time monitoring</i>	Sistem dapat mendeteksi kendaraan dan menggunakan ROI untuk menentukan serta memvisualisasik	Merupakan penelitian yang sangat relevan karena sama-sama menggabungkan deteksi kendaraan dan ROI. Penelitian ini menambahkan

				an status okupansi parkir secara <i>real-time</i> .	pembacaan beberapa <i>frame</i> , FastAPI, dan <i>dashboard</i> PHP untuk meningkatkan kestabilan dan penyajian status.
10	Wei et al. (2025)	<i>Real-time Parking Space Detection Based on Deep Learning and Panoramic Images</i>	<i>Deep learning</i> ; citra panoramik; deteksi ruang parkir; pemrosesan <i>real-time</i>	Deteksi ruang parkir secara <i>real-time</i> perlu dilanjutkan dengan pemrosesan pada modul manajemen untuk memperoleh informasi status okupansi.	Mendukung konsep pembagian area parkir menjadi lokasi tertentu sebelum informasi ketersediaan disampaikan kepada pengguna. Penelitian ini menggunakan CCTV simulasi dan <i>polygon ROI</i> pada slot tetap.
11	Rafique et al. (2023)	<i>Optimized Real-time Parking Management Framework Using Deep Learning</i>	<i>Deep learning</i> ; YOLOv5; analisis video; <i>Computer vision</i> ; <i>real-time parking management</i>	YOLOv5 dapat digunakan untuk mendeteksi kendaraan dan menghasilkan informasi ketersediaan parkir sebagai bagian dari sistem manajemen parkir <i>real-time</i> .	Sama-sama menggunakan deteksi kendaraan sebagai dasar penentuan kondisi parkir. Penelitian ini menggunakan YOLO26n hasil <i>fine-tuning</i> , <i>polygon ROI</i> pada sebelas slot, pembacaan temporal, dan <i>dashboard</i> .
12	Nugroho et al. (2023)	<i>Detection of Empty/Occupied States of Parking Slots in Multicamera System Using Mask R-CNN Classifier</i>	<i>Computer vision</i> ; Mask R-CNN; sistem multikamera; <i>mask</i> kendaraan; analisis irisan area slot	Sistem multikamera mampu menentukan status Kosong atau Terisi dengan mempertimbangkan <i>mask</i> kendaraan dan area slot serta dapat mengatasi oklusi hingga batas tertentu.	Relevan karena sama-sama mencocokkan hasil deteksi kendaraan dengan area slot. Perbedaannya, penelitian ini menggunakan YOLO26n, satu kelas <i>vehicle</i> , sebelas <i>polygon ROI</i> , dan sistem berbasis CCTV simulasi.
13	Carrera García et al. (2022)	<i>Expert System Design for Vacant Parking Space Location Using</i>	<i>Machine learning</i> ; <i>Computer vision</i> ; sistem pakar; deteksi kendaraan	Identifikasi kendaraan pada area parkir dapat digunakan untuk menentukan lokasi parkir	Mendukung hubungan antara pengenalan kendaraan dan informasi ketersediaan parkir.

		<i>Automatic Learning and Computer Vision</i>		yang masih tersedia.	Penelitian ini mengintegrasikan YOLO26n hasil <i>fine-tuning</i> , <i>polygon ROI</i> , pembacaan temporal, FastAPI, dan MySQL.
14	H. Lee et al. (2023)	<i>A Systematic Review of Computer Vision and AI in Parking Lot Management</i>	<i>Systematic review; Computer vision; machine learning; deep learning</i>	<i>Computer vision</i> dan AI dapat digunakan untuk mendeteksi kendaraan, menentukan okupansi ruang parkir, serta menyampaikan informasi parkir kepada pengguna.	Memperkuat landasan penggunaan <i>camera-based parking system</i> dan <i>deep learning</i> . Perbedaannya, penelitian ini menghasilkan <i>proto type</i> yang mengintegrasikan YOLO26n, <i>polygon ROI</i> , pembacaan temporal, FastAPI, MySQL, dan <i>dashboard PHP</i> .
15	Janowski et al. (2024)	<i>Sustainable Parking Space Management Using Machine Learning and Computer Vision</i>	<i>Machine learning; Computer vision; pengelolaan ruang parkir; pemantauan okupansi</i>	Pemanfaatan <i>machine learning</i> dan <i>Computer vision</i> dapat mendukung pemantauan okupansi serta pengelolaan ruang parkir secara lebih efisien.	Mendukung pemanfaatan hasil deteksi visual sebagai dasar pengelolaan dan penyampaian informasi ketersediaan parkir. Penelitian ini menggunakan YOLO26n hasil <i>fine-tuning</i> , <i>polygon ROI</i> , dan pembacaan temporal pada video CCTV.

Berdasarkan Tabel 2.1, penelitian terdahulu memiliki fokus kajian yang beragam. Sebagian penelitian membahas pemanfaatan *deep learning* dan *computer vision* untuk mendeteksi kendaraan dan kondisi ruang parkir, sedangkan penelitian lainnya berfokus pada klasifikasi okupansi, pengembangan model deteksi objek, penggunaan *Region of Interest* (ROI), serta pengembangan sistem *monitoring* parkir. Berbagai arah penelitian tersebut menunjukkan bahwa proses deteksi kendaraan, penentuan lokasi parkir, dan penyajian informasi merupakan komponen yang saling berkaitan dalam pengembangan

sistem parkir cerdas.

Kelompok penelitian mengenai deteksi kendaraan dan okupansi ruang parkir menunjukkan bahwa teknologi *computer vision* dan *deep learning* dapat dimanfaatkan untuk mengidentifikasi kondisi parkir secara otomatis. Schulte et al. (2021) memanfaatkan YOLOv4 dan DeepSORT untuk mendeteksi serta melacak kendaraan, sedangkan Yuldashev et al. (2023) menggunakan Improved MobileNetV3 untuk mengklasifikasikan kondisi okupansi ruang parkir. Grbić dan Koch (2023) juga mengembangkan pendekatan untuk mendeteksi ruang parkir dan mengklasifikasikan status okupansinya. Penelitian dalam kelompok ini menunjukkan kemampuan metode berbasis visual dalam mendeteksi kendaraan maupun kondisi ruang parkir. Pada penelitian ini, hasil deteksi kendaraan selanjutnya dipetakan ke area parkir tertentu untuk menghasilkan informasi status setiap slot sebagai Kosong atau Terisi.

Kelompok penelitian mengenai pengembangan model *deep learning* dan YOLO menunjukkan bahwa model deteksi objek dapat digunakan untuk mendukung identifikasi kendaraan pada lingkungan parkir. Zhao et al. (2024) mengembangkan CMCA-YOLO untuk meningkatkan kemampuan deteksi objek pada citra pengawasan parkir, sedangkan Fathurrahman et al. (2025) membandingkan beberapa versi YOLO untuk mendeteksi ruang parkir kosong. Penelitian tersebut lebih berorientasi pada pengembangan maupun evaluasi performa model. Berbeda dengan pendekatan tersebut, penelitian ini menggunakan YOLO26n hasil *fine-tuning* sebagai model deteksi kendaraan dan mengintegrasikan hasil deteksi dengan *polygon Region of Interest* pada setiap slot, sehingga informasi keberadaan kendaraan dapat diterjemahkan menjadi status lokasi parkir.

Kelompok penelitian mengenai ROI dan sistem *monitoring* parkir menunjukkan bahwa hasil deteksi kendaraan dapat dikaitkan dengan area parkir tertentu untuk menentukan kondisi okupansi. Popereshnyak & Chornobryvets (2025) menggabungkan deteksi kendaraan dan ROI untuk menentukan status lokasi parkir secara *real-time*, sedangkan Wei et al. (2025) menunjukkan bahwa hasil deteksi ruang parkir perlu dihubungkan dengan proses pengelolaan untuk menghasilkan informasi status okupansi. Temuan tersebut menunjukkan bahwa proses deteksi akan lebih bermanfaat apabila hasilnya dihubungkan dengan area parkir yang telah ditentukan dan disajikan melalui

suatu sistem *monitoring*. Pada penelitian ini, *polygon* ROI digunakan untuk merepresentasikan setiap slot parkir, sedangkan pembacaan beberapa *frame* secara berurutan digunakan untuk membantu mengurangi perubahan status akibat hasil deteksi yang tidak stabil pada satu *frame*.

Berdasarkan perbandingan tersebut, penelitian ini diarahkan pada pengembangan *prototype* sistem parkir cerdas yang mengintegrasikan YOLO26n hasil *fine-tuning* untuk mendeteksi kendaraan, *polygon* ROI untuk memetakan kendaraan ke slot parkir, serta pembacaan temporal untuk menentukan status Kosong atau Terisi. Hasil pengolahan tersebut selanjutnya diintegrasikan dengan *backend* FastAPI, basis data MySQL, dan *dashboard* berbasis PHP. MySQL digunakan untuk menyimpan status terbaru serta riwayat hasil pembacaan setiap slot, sedangkan *dashboard* digunakan untuk menyajikan informasi ketersediaan parkir, *snapshot* CCTV, dan laporan historis status parkir.

Dengan demikian, posisi penelitian ini tidak terletak pada pengembangan arsitektur *deep learning* baru, melainkan pada penerapan dan integrasi hasil deteksi kendaraan menjadi informasi operasional dalam suatu Sistem Informasi parkir cerdas. Integrasi antara deteksi kendaraan, pemetaan *polygon* ROI, pembacaan temporal, penyimpanan data, dan penyajian informasi memungkinkan kondisi setiap slot parkir dapat dipantau secara lebih terstruktur oleh pengguna maupun pengelola.

2.2 Teori Pendukung

Teori pendukung merupakan dasar konseptual yang menjelaskan hubungan antara masalah penelitian, teknologi yang digunakan, proses pengembangan, dan cara mengevaluasi hasil. Pada penelitian ini, teori disusun mulai dari konsep sistem informasi dan parkir, pengolahan data visual, pembentukan model deteksi, pemetaan hasil deteksi ke lokasi parkir, hingga penyajian informasi melalui aplikasi web.

Secara fungsional, teori tersebut membentuk empat lapisan. Lapisan masukan terdiri atas video CCTV, *frame*, *dataset*, dan anotasi. Lapisan pemrosesan mencakup *computer vision*, *deep learning*, *object detection*, YOLO26n, DETR, serta evaluasi model. Lapisan logika sistem mencakup *polygon Region of interest*, pembacaan temporal, dan basis data MySQL. Lapisan penyajian terdiri atas *backend* FastAPI, API berbasis JSON, dan *dashboard* PHP. DSRM dan pengujian sistem digunakan sebagai

landasan proses pengembangan serta evaluasi artefak yang dihasilkan.

2.2.1 Sistem Informasi Parkir Cerdas Berbasis *Computer Vision* dan *Deep Learning*

Sistem informasi parkir cerdas merupakan sistem yang memanfaatkan teknologi informasi untuk memperoleh, mengolah, menyimpan, dan menyajikan informasi mengenai kondisi area parkir. Dalam pendekatan berbasis kamera, data visual dari citra atau video dapat digunakan sebagai sumber informasi untuk mendeteksi kendaraan dan mengetahui kondisi okupansi ruang parkir. Fahim et al. (2021) menjelaskan bahwa sistem *smart parking* dapat dikembangkan menggunakan berbagai teknologi, termasuk *computer vision*, untuk mendukung proses pemantauan dan pengelolaan ruang parkir. Pendekatan berbasis kamera juga memungkinkan beberapa ruang parkir diamati melalui satu sumber visual sehingga informasi kondisi parkir dapat diperoleh tanpa memasang sensor pada setiap ruang.

Pemanfaatan *computer vision* memungkinkan sistem memperoleh informasi dari citra atau video, sedangkan *deep learning* digunakan untuk mengenali objek berdasarkan karakteristik visual yang dipelajari oleh model. Schulte et al. (2021) menunjukkan bahwa pendekatan berbasis kamera dan *deep learning* dapat digunakan untuk mendeteksi okupansi ruang parkir. Yuldashev et al. (2023) juga menunjukkan pemanfaatan *deep learning* dalam identifikasi kondisi okupansi, sedangkan Encío et al. (2023) mengembangkan pendekatan berbasis informasi visual untuk menentukan kondisi ruang parkir. Hasil penelitian tersebut menunjukkan bahwa data visual dapat dimanfaatkan sebagai dasar untuk memperoleh informasi mengenai kendaraan dan kondisi ruang parkir.

Dalam sistem parkir cerdas, hasil deteksi kendaraan perlu dikaitkan dengan lokasi parkir tertentu agar dapat menghasilkan informasi ketersediaan ruang parkir. Pendekatan *Region of Interest* (ROI) dapat digunakan untuk mendefinisikan area yang mewakili setiap slot parkir. Popereshnyak dan Chornobryvets (2025) menunjukkan relevansi penggunaan deteksi kendaraan dan ROI dalam sistem pemantauan parkir, sedangkan Wei et al. (2025) menunjukkan bahwa informasi hasil deteksi perlu dihubungkan dengan proses pengelolaan agar dapat menghasilkan informasi okupansi pada area yang telah ditentukan.

Pada penelitian ini, konsep sistem informasi parkir cerdas diwujudkan melalui integrasi beberapa komponen. Video CCTV simulasi digunakan sebagai sumber data visual, model YOLO26n hasil *fine-tuning* digunakan untuk mendeteksi kendaraan, dan

polygon Region of Interest digunakan untuk merepresentasikan setiap slot parkir. Hasil deteksi pada beberapa *frame* diproses secara temporal untuk membantu menentukan status setiap slot sebagai Kosong atau Terisi. Informasi tersebut selanjutnya dikelola melalui *backend* FastAPI, disimpan pada basis data MySQL, dan disajikan melalui *dashboard* berbasis PHP. Dengan demikian, sistem yang dikembangkan tidak hanya menghasilkan keluaran deteksi kendaraan, tetapi mengolah hasil tersebut menjadi informasi ketersediaan dan riwayat status parkir yang dapat dipantau secara terstruktur oleh pengguna maupun pengelola.

2.2.2 Parkir

Dalam penelitian ini, parkir dipahami sebagai kondisi ketika kendaraan menempati suatu area yang telah disediakan sebagai tempat parkir. Area yang digunakan untuk menempatkan satu kendaraan disebut sebagai slot parkir. Setiap slot perlu memiliki batas dan identitas yang jelas agar kondisi penggunaannya dapat dicatat, diproses, dan disajikan secara konsisten dalam suatu sistem informasi parkir.

Salah satu informasi utama dalam pengelolaan slot parkir adalah kondisi okupansi. Okupansi menunjukkan apakah suatu slot sedang digunakan oleh kendaraan atau tidak. Pada sistem deteksi okupansi, kondisi slot dapat diklasifikasikan sebagai *empty* atau *vacant* ketika tidak terdapat kendaraan dan *occupied* ketika slot ditempati kendaraan (Nugroho et al., 2023; Wei et al., 2025). Berdasarkan konsep tersebut, penelitian ini menggunakan dua status utama, yaitu Kosong dan Terisi. Status Kosong menunjukkan bahwa tidak terdapat kendaraan yang dipetakan pada area slot, sedangkan status Terisi menunjukkan bahwa terdapat kendaraan yang terdeteksi dan dipetakan pada area slot tersebut.

Pada penelitian ini, area parkir direpresentasikan menjadi sebelas slot yang diberi kode A1 sampai A5 dan B1 sampai B6. Setiap slot memiliki area yang ditentukan untuk menghubungkan keberadaan kendaraan dengan lokasi parkir tertentu. Dengan demikian, unit informasi yang digunakan dalam sistem adalah kondisi okupansi setiap slot parkir, yaitu Kosong atau Terisi.

2.2.3 Smart parking

Smart parking merupakan sistem parkir yang memanfaatkan teknologi untuk mendukung proses pemantauan, pengelolaan, dan penyampaian informasi mengenai

ketersediaan parkir. Fahim et al. (2021) menjelaskan bahwa *smart parking* dapat ditinjau berdasarkan teknologi pendeteksian, jaringan komunikasi, pemrosesan data, antarmuka pengguna, serta layanan yang dihasilkan. Pemanfaatan teknologi tersebut memungkinkan kondisi area parkir diperoleh dan diolah menjadi informasi yang dapat digunakan oleh pengguna maupun pengelola.

Secara umum, arsitektur *smart parking* dapat melibatkan lapisan penginderaan, komunikasi, pemrosesan, penyimpanan, dan aplikasi. Lapisan penginderaan digunakan untuk memperoleh data mengenai kondisi area parkir melalui sensor atau kamera. Lapisan komunikasi digunakan untuk mengirimkan data menuju komponen pemrosesan. Selanjutnya, lapisan pemrosesan mengubah data yang diperoleh menjadi informasi mengenai kondisi okupansi, sedangkan lapisan penyimpanan digunakan untuk mencatat hasil pengolahan data. Lapisan aplikasi kemudian menyajikan informasi tersebut kepada pengguna atau pengelola. Pada pendekatan berbasis kamera, beberapa slot dapat diamati melalui satu sumber visual, meskipun proses pengamatan dapat dipengaruhi oleh pencahayaan, perspektif kamera, *occlusion*, dan kestabilan posisi kamera (Grbić & Koch, 2023).

Pada penelitian ini, konsep *smart parking* diterapkan melalui integrasi video CCTV simulasi sebagai sumber data visual, YOLO26n hasil *fine-tuning* sebagai model deteksi kendaraan, *polygon Region of Interest* (ROI) sebagai representasi area setiap slot, serta pembacaan beberapa *frame* secara berurutan untuk membantu menentukan status Kosong atau Terisi. Hasil pengolahan selanjutnya dikelola melalui *backend* FastAPI, disimpan pada basis data MySQL, dan disajikan melalui *dashboard* berbasis PHP. Dengan alur tersebut, data visual dari video diubah menjadi informasi kondisi slot parkir yang dapat disimpan dan dipantau melalui sistem.

2.2.4 Computer vision

Computer vision merupakan bidang ilmu yang mempelajari bagaimana komputer memperoleh, memproses, dan menafsirkan informasi dari citra atau video. Menurut Szeliski (2021), *computer vision* mencakup berbagai tugas, seperti peningkatan kualitas citra, pengenalan pola, klasifikasi, deteksi objek, segmentasi, pelacakan, serta pemahaman hubungan spasial pada suatu *scene*. Pada data visual, citra direpresentasikan sebagai susunan piksel, sedangkan video merupakan rangkaian *frame* yang tersusun berdasarkan

urutan waktu.

Secara umum, proses *computer vision* melibatkan tahapan akuisisi data visual, praproses, pembelajaran atau ekstraksi fitur, inferensi model, dan interpretasi hasil. Akuisisi data menentukan sumber citra atau video yang digunakan, sedangkan praproses dilakukan untuk menyesuaikan data masukan agar dapat diproses oleh model. Model selanjutnya menghasilkan prediksi berdasarkan karakteristik visual yang telah dipelajari, kemudian hasil prediksi diinterpretasikan sesuai dengan kebutuhan sistem. Pada sistem parkir berbasis kamera, hasil deteksi kendaraan masih perlu dihubungkan dengan lokasi slot parkir sebelum dapat digunakan sebagai informasi mengenai kondisi okupansi.

Dalam penelitian ini, konsep *computer vision* digunakan untuk mengolah data visual yang berasal dari video CCTV simulasi. Setiap *frame* video menjadi masukan bagi model deteksi kendaraan untuk menghasilkan informasi berupa kelas objek, tingkat *confidence*, dan *bounding box*. Informasi hasil deteksi tersebut selanjutnya digunakan sebagai dasar untuk menentukan hubungan antara kendaraan dengan area slot parkir yang telah direpresentasikan menggunakan *polygon Region of Interest* (ROI). Dengan demikian, *computer vision* berfungsi sebagai dasar pemrosesan data visual sebelum hasil deteksi diolah lebih lanjut menjadi informasi status slot parkir Kosong atau Terisi.

2.2.5 Deep learning

Deep learning merupakan bagian dari *machine learning* yang menggunakan jaringan saraf dengan banyak lapisan untuk mempelajari representasi data secara bertingkat. Lapisan awal umumnya mempelajari karakteristik visual sederhana, seperti tepi, warna, dan tekstur, sedangkan lapisan yang lebih dalam membentuk representasi yang semakin kompleks. Kemampuan untuk mempelajari fitur secara otomatis membuat *deep learning* banyak digunakan pada berbagai tugas pengolahan citra, seperti klasifikasi citra, deteksi objek, segmentasi, dan analisis video Heaton (2018).

Proses pengembangan model *deep learning* pada umumnya terdiri atas tahap *training* dan inferensi. Pada tahap *training*, parameter model diperbarui berdasarkan perbedaan antara hasil prediksi dan *ground truth* agar model dapat mempelajari pola dari data pelatihan. Setelah proses pelatihan selesai, tahap inferensi dilakukan dengan menggunakan parameter yang telah dipelajari untuk menghasilkan prediksi terhadap data baru. Salah satu pendekatan yang dapat digunakan dalam pengembangan model adalah

transfer learning, yaitu memanfaatkan model yang sebelumnya telah dilatih pada dataset yang lebih besar sebagai dasar untuk pelatihan pada data yang lebih spesifik. Selanjutnya, proses *fine-tuning* dilakukan untuk menyesuaikan model tersebut terhadap karakteristik dataset penelitian. Pendekatan ini dapat mengurangi kebutuhan pelatihan dari awal serta membantu proses adaptasi model terhadap data yang digunakan (Grbić & Koch, 2023).

Dalam penelitian ini, *deep learning* diterapkan melalui model YOLO26n untuk mendeteksi objek kendaraan pada *frame* video parkir. Model YOLO26n yang digunakan merupakan model pralatih yang kemudian dilakukan *fine-tuning* menggunakan dataset internal penelitian. Proses tersebut bertujuan menyesuaikan kemampuan model dalam mendeteksi kendaraan pada karakteristik visual yang terdapat pada video CCTV simulasi. Keluaran model berupa hasil deteksi kendaraan, sedangkan penentuan status setiap slot sebagai Kosong atau Terisi tidak dilakukan secara langsung oleh model. Status slot ditentukan melalui proses pemetaan hasil deteksi ke *polygon Region of Interest* dan pembacaan beberapa *frame* secara berurutan.

2.2.6 Object detection

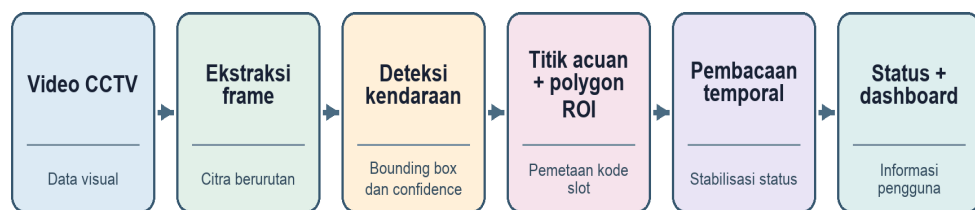
Object detection merupakan proses untuk mengenali jenis objek sekaligus menentukan lokasi objek tersebut pada gambar atau video. Berbeda dengan klasifikasi citra yang memberikan satu label pada keseluruhan gambar, *object detection* dapat mendeteksi beberapa objek dalam satu citra serta memberikan informasi lokasi untuk setiap objek yang ditemukan. Keluaran dari proses deteksi objek umumnya terdiri atas label kelas, nilai *confidence*, dan *bounding box*.

Bounding box digunakan untuk merepresentasikan lokasi objek melalui koordinat posisi dan ukuran kotak, sedangkan nilai *confidence* menunjukkan tingkat keyakinan model terhadap suatu prediksi. Nilai *confidence* dapat dibandingkan dengan ambang tertentu untuk menyaring prediksi dengan tingkat keyakinan yang rendah. Kesesuaian lokasi antara *bounding box* hasil prediksi dengan *ground truth* dapat diukur menggunakan *Intersection over Union* (IoU). Pada metode deteksi yang menghasilkan beberapa *bounding box* untuk objek yang sama, *Non-Maximum Suppression* (NMS) dapat digunakan untuk mempertahankan prediksi yang paling relevan.

Redmon et al. (2016) memperkenalkan YOLO sebagai pendekatan deteksi objek satu tahap (*one-stage detector*) yang memprediksi lokasi dan kelas objek dalam satu alur

jaringan. Pendekatan satu tahap tidak memerlukan proses usulan wilayah (*region proposal*) yang terpisah seperti pada detektor dua tahap. Dalam sistem parkir berbasis kamera, kemampuan lokalisasi objek menjadi penting karena informasi posisi kendaraan hasil deteksi selanjutnya dapat digunakan untuk menentukan keterkaitannya dengan area slot parkir.

Pada penelitian ini, *object detection* digunakan untuk mendeteksi kendaraan sebagai satu kelas objek, yaitu *vehicle*. Keluaran model berupa kelas objek, nilai *confidence*, dan *bounding box* kendaraan. Hasil deteksi tersebut belum secara langsung menentukan status suatu slot parkir. Informasi lokasi kendaraan selanjutnya dipetakan terhadap *polygon Region of Interest* (ROI) setiap slot dan dikombinasikan dengan pembacaan beberapa *frame* secara berurutan untuk menentukan status slot sebagai Kosong atau Terisi. Hubungan konseptual antara proses deteksi kendaraan dan penentuan status slot ditunjukkan pada Gambar 2.1.



Gambar 2. 1 Alur Konseptual Computer vision untuk Penentuan Status Slot Parkir

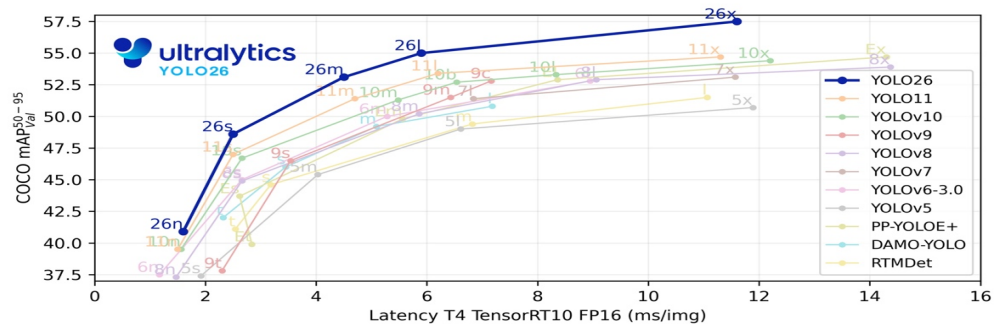
2.2.7 YOLO26n dan *Fine-tuning* Model Deteksi Kendaraan

YOLO26 merupakan keluarga model *computer vision* yang digunakan untuk melakukan *object detection* secara *end-to-end*. Pada tugas deteksi objek, model menerima citra atau *frame* sebagai masukan dan menghasilkan informasi berupa label kelas, nilai *confidence*, serta koordinat *bounding box*. Informasi tersebut memungkinkan sistem mengenali jenis objek sekaligus menentukan lokasi objek pada citra atau video Jocher et al. (2026).

Keluarga YOLO26 menggunakan kepala deteksi *one-to-one* sehingga jalur inferensi utamanya dapat menghasilkan prediksi tanpa proses *Non-Maximum Suppression* (NMS). Arsitektur YOLO26 juga menggunakan kepala regresi yang lebih ringan tanpa *Distribution Focal Loss* serta beberapa pembaruan pada proses pelatihan, seperti MuSGD, *Progressive Loss*, dan *Small-Target-Aware Label Assignment*. Karakteristik tersebut

dikembangkan untuk menjaga keseimbangan antara kemampuan deteksi dan efisiensi komputasi (Jocher et al., 2026).

YOLO26 tersedia dalam beberapa varian ukuran, yaitu YOLO26n (*nano*), YOLO26s (*small*), YOLO26m (*medium*), YOLO26l (*large*), dan YOLO26x (*extra-large*). Perbedaan ukuran model memengaruhi kapasitas representasi, kebutuhan memori, waktu inferensi, dan performa deteksi. YOLO26n merupakan varian dengan ukuran paling ringan di antara varian tersebut sehingga dapat dipertimbangkan untuk sistem yang membutuhkan proses inferensi berulang dengan sumber daya komputasi yang terbatas.



Gambar 2. 2 Perbandingan mAP50–95 dan Latensi Model YOLO26 dengan Model Deteksi Objek Lainnya

Perbandingan performa dan latensi keluarga YOLO26 dengan beberapa model deteksi objek lainnya ditunjukkan pada Gambar 2.2. Berdasarkan dokumentasi yang digunakan dalam penelitian, YOLO26n memiliki nilai mAP50–95 sebesar 40,9 pada dataset COCO dengan latensi sekitar 1,7 milidetik per citra pada konfigurasi perangkat pengujian yang tercantum pada sumber. Nilai tersebut merupakan hasil *benchmark* umum dan tidak digunakan sebagai hasil eksperimen penelitian ini karena performa aktual model dapat dipengaruhi oleh dataset, perangkat keras, serta konfigurasi inferensi yang digunakan.

Model pralatih YOLO26n menggunakan bobot awal yang telah mempelajari berbagai kategori objek umum pada dataset COCO, termasuk beberapa jenis kendaraan. Model pralatih tersebut telah memiliki representasi visual awal yang dapat dimanfaatkan untuk mendeteksi kendaraan. Namun, karakteristik data pada dataset umum tidak sepenuhnya sama dengan karakteristik citra pada area parkir yang digunakan dalam penelitian. Oleh karena itu, model pralatih dapat disesuaikan terhadap dataset yang lebih spesifik melalui proses *fine-tuning*.

Fine-tuning merupakan proses melanjutkan pelatihan model pralatih menggunakan dataset target yang lebih spesifik. Proses tersebut memanfaatkan pengetahuan yang telah diperoleh dari dataset sebelumnya, kemudian menyesuaikan parameter model terhadap karakteristik data target. Dalam proses pengembangan model, data *training* digunakan untuk memperbarui parameter, data *validation* digunakan untuk memantau performa selama pelatihan dan membantu memilih model, sedangkan data *testing* digunakan untuk mengevaluasi kemampuan model pada data yang tidak digunakan dalam proses pembaruan parameter Yosinski et al. (2014).

Secara umum, proses *fine-tuning* meliputi pemuatan bobot pralatih, penyesuaian kelas objek, pelatihan menggunakan data *training*, pemantauan performa menggunakan data *validation*, dan pemilihan bobot dengan performa validasi terbaik. Model terpilih selanjutnya dapat dievaluasi menggunakan metrik *precision*, *recall*, mAP@50, dan mAP@50–95 untuk mengetahui kemampuan model dalam mendeteksi objek pada data pengujian.

Dalam penelitian ini, YOLO26n digunakan sebagai model utama untuk mendeteksi kendaraan dalam satu kelas objek, yaitu *vehicle*. Model YOLO26n dilakukan *fine-tuning* menggunakan dataset internal penelitian agar kemampuan deteksinya lebih sesuai dengan karakteristik visual kendaraan pada video CCTV simulasi. Hasil deteksi kendaraan selanjutnya digunakan sebagai masukan bagi proses pemetaan terhadap *polygon Region of Interest* (ROI) dan pembacaan temporal untuk menentukan status setiap slot sebagai Kosong atau Terisi.

2.2.8 Faster R-CNN

Faster R-CNN merupakan salah satu metode *object detection* dua tahap (*two-stage detector*). Berbeda dengan pendekatan satu tahap seperti YOLO yang melakukan prediksi kelas dan lokasi objek secara langsung, Faster R-CNN terlebih dahulu menghasilkan kandidat wilayah objek melalui *Region Proposal Network* (RPN). Kandidat wilayah tersebut kemudian diproses pada tahap berikutnya untuk menentukan kelas objek serta melakukan penyempurnaan posisi *bounding box*. Pendekatan dua tahap memungkinkan proses deteksi dilakukan dengan memisahkan pembentukan kandidat objek dan proses klasifikasi serta regresi lokasi objek.

Pada arsitektur Faster R-CNN, citra masukan terlebih dahulu diproses oleh jaringan

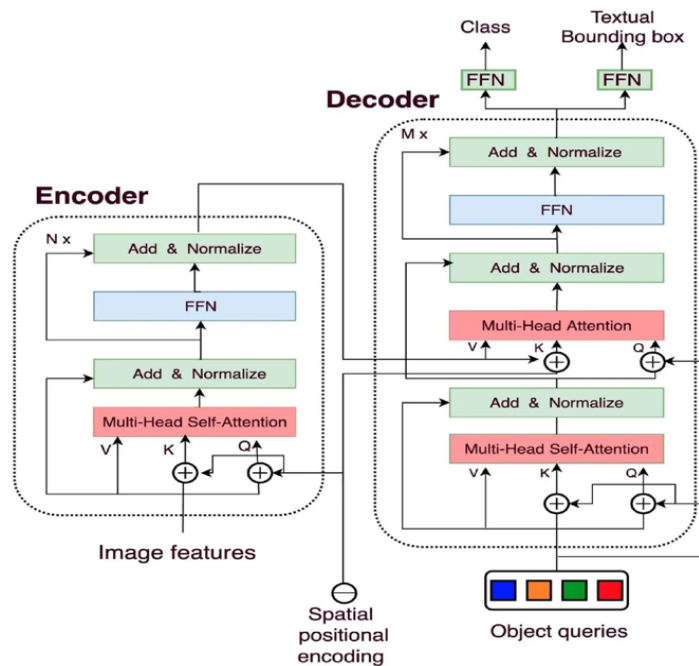
backbone untuk menghasilkan peta fitur. *Region Proposal Network* kemudian menggunakan peta fitur tersebut untuk menghasilkan sejumlah kandidat wilayah yang diperkirakan mengandung objek. Kandidat wilayah selanjutnya diproses oleh bagian deteksi untuk menghasilkan label kelas dan koordinat *bounding box*. Dengan mekanisme tersebut, Faster R-CNN termasuk pendekatan deteksi berbasis *region proposal*.

Salah satu komponen yang dapat digunakan bersama Faster R-CNN adalah *Feature Pyramid Network* (FPN). FPN memanfaatkan informasi fitur pada beberapa tingkat resolusi sehingga objek dengan ukuran yang berbeda dapat direpresentasikan melalui skala fitur yang sesuai. Penggunaan FPN dapat membantu proses deteksi ketika ukuran objek pada citra bervariasi, termasuk pada citra area parkir yang menampilkan kendaraan dengan ukuran berbeda akibat perbedaan posisi dan perspektif kamera.

Dalam penelitian ini, Faster R-CNN digunakan sebagai model pembandingan terhadap YOLO26n. Model pembandingan digunakan untuk melihat perbedaan performa deteksi kendaraan berdasarkan nilai *precision*, *recall*, mAP@50, dan mAP@50–95 pada data pengujian yang sama. Faster R-CNN digunakan karena memiliki karakteristik deteksi berbasis *region proposal*, sehingga dapat memberikan pembandingan terhadap YOLO26n yang menggunakan pendekatan deteksi satu tahap. Hasil deteksi Faster R-CNN hanya digunakan dalam evaluasi model dan tidak digunakan sebagai model utama pada alur *prototype* sistem parkir cerdas.

2.2.9 DETR (*DE*tectiOn *TR*ansformer)

DETR (*DE*tectiOn *TR*ansformer) merupakan arsitektur *object detection end-to-end* yang diperkenalkan oleh Carion et al. (2020). DETR memandang proses deteksi objek sebagai permasalahan *direct set prediction*, yaitu menghasilkan sekumpulan prediksi objek secara langsung dari citra masukan. Citra terlebih dahulu diproses menggunakan *backbone* konvolusional untuk menghasilkan peta fitur. Selanjutnya, *positional encoding* ditambahkan untuk mempertahankan informasi posisi, kemudian *Transformer encoder* menggunakan mekanisme *self-attention* untuk mempelajari hubungan global antarbagian citra. *Transformer decoder* selanjutnya menghasilkan prediksi kelas dan *bounding box* objek.



Gambar 2. 3 Arsitektur Transformer Encoder–Decoder pada DETR

Pada bagian *decoder*, DETR menggunakan sejumlah *object queries* yang dipelajari. Setiap *query* dapat menghasilkan satu prediksi kelas dan satu *bounding box*, sedangkan *query* yang tidak berkaitan dengan objek diberi kelas *no object*. Pada tahap *training*, DETR menggunakan *bipartite matching* melalui algoritma Hungarian untuk memperoleh pasangan satu-ke-satu antara hasil prediksi dan *ground truth*. Proses pencocokan mempertimbangkan kesesuaian kelas, posisi *bounding box*, dan *generalized Intersection over Union*. Mekanisme pencocokan satu-ke-satu tersebut membantu mengurangi prediksi ganda sehingga DETR tidak memerlukan *anchor* maupun *Non-Maximum Suppression* (NMS).

Dalam penelitian ini, DETR digunakan sebagai salah satu model pembanding terhadap YOLO26n dan Faster R-CNN menggunakan pembagian dataset yang sama. Penggunaan DETR sebagai pembanding memberikan representasi pendekatan deteksi yang berbeda karena DETR memanfaatkan mekanisme *Transformer* dan pemodelan konteks global, sedangkan YOLO26n digunakan sebagai model utama pada *prototype* sistem parkir cerdas. Perbandingan ketiga model digunakan untuk menganalisis perbedaan performa berdasarkan *precision*, *recall*, *mAP@50*, dan *mAP@50–95*.

2.2.10 Dataset, Anotasi, dan Pseudo-label

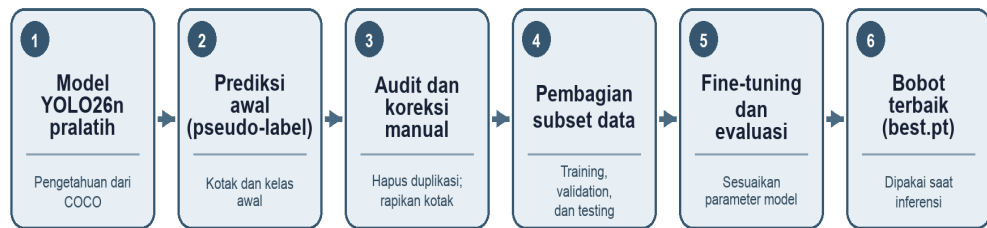
Dataset merupakan kumpulan data yang digunakan dalam proses pengembangan dan evaluasi model. Pada tugas *object detection*, setiap citra yang digunakan dalam proses pelatihan perlu memiliki anotasi yang menunjukkan jenis objek dan lokasi objek pada citra. Kualitas dataset tidak hanya ditentukan oleh jumlah data, tetapi juga oleh ketepatan anotasi, keragaman kondisi data, serta pemisahan data yang digunakan untuk proses *training*, *validation*, dan *testing*.

Anotasi pada *object detection* dapat direpresentasikan dalam bentuk *bounding box* yang menunjukkan posisi objek pada citra. Pada format YOLO, *bounding box* direpresentasikan menggunakan koordinat titik pusat, lebar, dan tinggi yang dinormalisasi terhadap ukuran citra. Anotasi tersebut digunakan sebagai *ground truth* pada proses pelatihan dan evaluasi model. Ketidaktepatan anotasi, seperti *bounding box* yang terlalu longgar, terlalu sempit, bergeser, ganda, atau memiliki kelas yang salah, dapat menghasilkan target pembelajaran yang tidak sesuai sehingga anotasi perlu diperiksa sebelum digunakan.

Dataset dalam pengembangan model umumnya dibagi menjadi data *training*, *validation*, dan *testing*. Data *training* digunakan untuk memperbarui parameter model selama proses pelatihan. Data *validation* digunakan untuk memantau performa model dan membantu menentukan model atau bobot terbaik, sedangkan data *testing* digunakan untuk melakukan evaluasi akhir terhadap kemampuan model pada data yang tidak digunakan untuk memperbarui parameter. Pemisahan ketiga subset tersebut diperlukan agar evaluasi model dapat menggambarkan kemampuan model pada data di luar proses pelatihan.

Pseudo-label merupakan label awal yang dihasilkan secara otomatis menggunakan prediksi model terhadap data yang belum memiliki anotasi manual D.-H. Lee (2013). Pendekatan ini dapat membantu mempercepat proses penyusunan dataset, tetapi hasil prediksi model belum tentu seluruhnya sesuai dengan kondisi objek sebenarnya. Oleh karena itu, *pseudo-label* perlu diperiksa dan diperbaiki melalui proses verifikasi terhadap kelas objek, posisi *bounding box*, objek yang tidak terdeteksi, prediksi ganda, serta prediksi terhadap objek yang tidak sesuai dengan kelas yang ditentukan.

Dalam penelitian ini, dataset diperoleh dari *frame* yang diekstraksi dari video CCTV simulasi dengan posisi kamera tetap. Model pralatih digunakan untuk membantu menghasilkan *pseudo-label* kendaraan dengan kelas *vehicle*. Hasil anotasi selanjutnya diperiksa dan diperbaiki secara visual sebelum digunakan dalam proses *fine-tuning*. Dataset kemudian dibagi menjadi data *training*, *validation*, dan *testing* untuk mendukung proses pengembangan serta evaluasi model deteksi kendaraan. Alur penyusunan dataset dan proses *fine-tuning* ditunjukkan pada Gambar 2.4.



Gambar 2. 4 Alur Pseudo-label dan Fine-tuning Model Deteksi

2.2.11 Evaluasi Model Deteksi

Evaluasi model deteksi dilakukan untuk mengukur kemampuan model dalam mendeteksi dan melokalisasi objek kendaraan pada data yang tidak digunakan untuk memperbarui parameter model. Hasil prediksi dibandingkan dengan *ground truth* berdasarkan kelas objek, nilai *confidence*, dan tingkat kesesuaian lokasi *bounding box*. Proses evaluasi perlu menggunakan nilai ambang *confidence* dan *Intersection over Union* (IoU) yang ditentukan secara jelas karena perubahan nilai ambang dapat memengaruhi jumlah prediksi yang dikategorikan sebagai benar maupun salah Mulyanto et al. (2024).

Dalam evaluasi *object detection*, *True Positive* (TP) merupakan prediksi objek yang berhasil dipasangkan dengan anotasi *ground truth* berdasarkan kriteria kelas dan lokasi yang ditetapkan. *False Positive* (FP) merupakan prediksi objek yang tidak memiliki pasangan anotasi yang memenuhi kriteria, sedangkan *False Negative* (FN) merupakan objek pada *ground truth* yang tidak berhasil dideteksi oleh model. Nilai TP, FP, dan FN digunakan sebagai dasar dalam menghitung *precision* dan *recall*.

Precision menunjukkan proporsi prediksi positif yang benar dibandingkan dengan seluruh prediksi positif yang dihasilkan model. Nilai *precision* dapat dihitung menggunakan Persamaan (2.1):

$$Precision = \frac{TP}{TP + FP} \quad (2.1)$$

Nilai *precision* yang tinggi menunjukkan bahwa model relatif sedikit menghasilkan prediksi positif yang salah. Dalam konteks sistem parkir, *false positive* dapat menyebabkan kendaraan terdeteksi pada kondisi yang sebenarnya tidak sesuai sehingga berpotensi memengaruhi proses penentuan status slot.

Recall menunjukkan proporsi objek pada *ground truth* yang berhasil ditemukan oleh model. Nilai *recall* dapat dihitung menggunakan Persamaan (2.2):

$$Recall = \frac{TP}{TP + FN} \quad (2.2)$$

Nilai *recall* yang tinggi menunjukkan bahwa model mampu menemukan sebagian besar objek kendaraan yang terdapat pada data pengujian. Sebaliknya, *false negative* menunjukkan kendaraan yang terdapat pada *ground truth* tetapi tidak berhasil dideteksi oleh model. Kondisi tersebut dapat memengaruhi penentuan status slot apabila kendaraan yang berada pada area parkir tidak terdeteksi.

Intersection over Union (IoU) digunakan untuk mengukur tingkat kesesuaian spasial antara *bounding box* hasil prediksi dan *bounding box ground truth*. IoU dihitung berdasarkan perbandingan luas irisan terhadap luas gabungan kedua *bounding box*, sebagaimana ditunjukkan pada Persamaan (2.3):

$$IoU = \frac{Luas(B_{prediksi} \cap B_{acuan})}{Luas(B_{prediksi} \cup B_{acuan})} \quad (2.3)$$

Nilai IoU berada pada rentang 0 sampai 1. Nilai yang semakin mendekati 1 menunjukkan tingkat tumpang tindih yang semakin tinggi antara *bounding box* prediksi dan *ground truth*. Dalam proses evaluasi, suatu prediksi dapat dikategorikan benar apabila memenuhi nilai ambang IoU yang telah ditetapkan.

Average Precision (AP) digunakan untuk merangkum hubungan antara *precision* dan *recall*, sedangkan *mean Average Precision* (mAP) digunakan sebagai metrik agregat performa deteksi. Pada penelitian ini hanya digunakan satu kelas objek, yaitu *vehicle*. Nilai mAP@50 menunjukkan performa deteksi pada ambang IoU sebesar

0,50, sedangkan $mAP@50-95$ merupakan rata-rata performa pada beberapa nilai ambang IoU dari 0,50 hingga 0,95 dengan interval 0,05. Metrik $mAP@50-95$ memberikan penilaian yang lebih ketat terhadap kualitas lokalisasi *bounding box* dibandingkan $mAP@50$ Oksuz et al. (2018).

Selain metrik tersebut, *confusion matrix* digunakan untuk memberikan gambaran mengenai jumlah prediksi yang benar, prediksi yang salah, dan objek yang tidak berhasil terdeteksi. Pada model dengan satu kelas *vehicle*, *confusion matrix* membantu menunjukkan hubungan antara prediksi kelas kendaraan dan latar belakang. Oleh karena itu, *precision*, *recall*, mAP , dan *confusion matrix* digunakan secara bersama-sama agar evaluasi tidak hanya bergantung pada satu metrik.

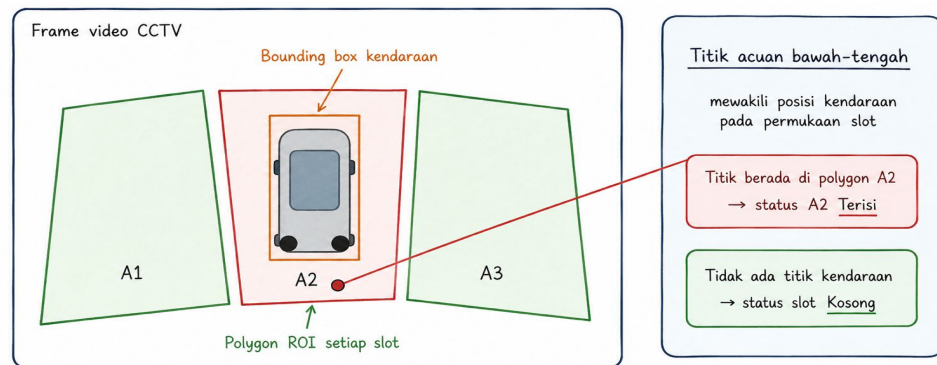
2.2.12 Video CCTV, *Polygon ROI*, dan Pembacaan Temporal

Video CCTV merupakan rangkaian *frame* yang direkam secara berurutan dari suatu area pemantauan. Kualitas informasi visual yang diperoleh dari video dapat dipengaruhi oleh posisi kamera, resolusi, pencahayaan, *frame rate*, kompresi, dan kondisi lingkungan. Penggunaan kamera dengan posisi tetap mendukung proses pemetaan area parkir karena posisi objek dan batas slot relatif konsisten antar-*frame*. Sebaliknya, perubahan posisi atau sudut kamera dapat menyebabkan koordinat area yang telah ditentukan tidak lagi sesuai dengan posisi fisik slot parkir.

Region of Interest (ROI) merupakan bagian tertentu pada citra yang ditetapkan sebagai area yang menjadi fokus proses pengolahan. Pada penelitian parkir berbasis kamera, ROI dapat digunakan untuk merepresentasikan lokasi setiap slot parkir. ROI berbentuk *polygon* dibentuk menggunakan beberapa titik koordinat sehingga lebih fleksibel dalam mengikuti bentuk slot yang terlihat miring atau mengalami perubahan perspektif pada citra. Setiap *polygon* dapat dikaitkan dengan kode slot tertentu sehingga hasil deteksi kendaraan dapat dipetakan ke lokasi parkir yang sesuai.

Pemetaan kendaraan terhadap slot parkir dapat dilakukan dengan menggunakan titik acuan dari *bounding box* hasil deteksi. Dalam penelitian ini, titik acuan ditentukan pada bagian bawah-tengah *bounding box* kendaraan karena posisi tersebut lebih dekat dengan area tempat kendaraan bersentuhan dengan permukaan parkir. Apabila titik acuan berada di dalam *polygon* suatu slot, kendaraan dipetakan ke slot tersebut. Penggunaan titik acuan membantu mengurangi kemungkinan satu kendaraan dianggap menempati

beberapa slot hanya karena ukuran *bounding box* melebar akibat perspektif kamera. Konsep pemetaan tersebut ditunjukkan pada Gambar 2.5.



Gambar 2. 5 Pencocokan Titik Acuan Bounding box dengan Polygon ROI

2.2.13 FastAPI

FastAPI merupakan *web framework* berbasis Python yang digunakan untuk membangun *Application Programming Interface* (API). API merupakan antarmuka yang memungkinkan dua atau lebih komponen perangkat lunak saling bertukar permintaan dan respons melalui *endpoint* yang telah ditentukan. FastAPI memanfaatkan deklarasi tipe pada Python dan mendukung validasi data menggunakan Pydantic sehingga struktur data yang diterima maupun dikirimkan dapat didefinisikan secara lebih terstruktur. Data respons API umumnya dapat disajikan dalam format JSON sehingga dapat digunakan oleh aplikasi atau antarmuka pengguna lainnya.

FastAPI mendukung pembuatan dokumentasi API secara otomatis berdasarkan spesifikasi OpenAPI. Dokumentasi tersebut dapat diakses melalui antarmuka interaktif seperti Swagger UI dan ReDoc sehingga memudahkan proses pemeriksaan *endpoint*, parameter, struktur permintaan, serta bentuk respons API FastAPI, (2026). Spesifikasi OpenAPI digunakan untuk mendeskripsikan antarmuka HTTP secara terstruktur, termasuk *path*, parameter, struktur data, dan respons yang disediakan oleh layanan API OpenAPI Initiative (2025).

Dalam penelitian ini, FastAPI digunakan sebagai komponen *backend* yang menghubungkan proses deteksi kendaraan dengan penyimpanan dan penyajian informasi parkir. *Backend* memproses data video CCTV simulasi, menjalankan model YOLO26n,

memetakan hasil deteksi kendaraan terhadap *polygon* ROI, menerapkan pembacaan temporal, serta menyimpan hasil status parkir ke basis data MySQL. FastAPI kemudian menyediakan informasi tersebut melalui *endpoint* agar dapat diakses oleh *dashboard* PHP.

Data yang disediakan melalui API mencakup informasi kode slot, status Kosong atau Terisi, nilai *confidence*, waktu pembaruan, serta informasi pendukung lainnya. Selain data status dalam format JSON, *backend* juga menyediakan *snapshot* CCTV yang menampilkan hasil deteksi kendaraan dan visualisasi *polygon* area parkir. Dengan demikian, FastAPI berfungsi sebagai lapisan penghubung antara proses *computer vision*, basis data, dan antarmuka pengguna.

2.2.14 PHP (*Hypertext Preprocessor*)

PHP : *Hypertext Preprocessor* merupakan bahasa pemrograman *server-side* yang banyak digunakan dalam pengembangan aplikasi web. Kode PHP dieksekusi pada sisi server untuk memproses logika aplikasi dan menghasilkan keluaran, seperti HTML atau data lainnya, yang kemudian dikirimkan ke peramban pengguna The PHP Group, (2026). Karena dijalankan pada sisi server, PHP dapat digunakan untuk mengambil, mengolah, dan menyajikan data yang diperoleh dari API maupun basis data.

Dalam sistem informasi, *dashboard* digunakan sebagai antarmuka untuk menyajikan informasi penting secara ringkas dan terstruktur. Bach et al. (2023) menjelaskan bahwa perancangan *dashboard* perlu mempertimbangkan hierarki informasi, pemilihan bentuk visual, konsistensi tampilan, dan keterbacaan. Informasi utama perlu ditampilkan secara jelas agar pengguna dapat memahami kondisi sistem tanpa harus menelusuri data secara keseluruhan.

Dalam penelitian ini, PHP digunakan sebagai lapisan penyajian pada *dashboard* sistem parkir cerdas. *Dashboard* memperoleh data status parkir melalui *endpoint* yang disediakan oleh *backend* FastAPI, kemudian menyajikannya dalam bentuk informasi yang dapat dipahami oleh pengguna dan pengelola. Informasi yang ditampilkan meliputi jumlah keseluruhan slot parkir, jumlah slot Kosong dan Terisi, status masing-masing kode slot, *snapshot* CCTV, serta laporan parkir.

Pemisahan antara *backend* FastAPI dan *dashboard* PHP memungkinkan proses deteksi kendaraan dan penyajian informasi dikelola sebagai komponen yang berbeda.

FastAPI menangani pemrosesan serta penyediaan data melalui API, sedangkan PHP berfungsi menyajikan informasi tersebut kepada pengguna melalui antarmuka web. Dengan pendekatan tersebut, perubahan pada proses deteksi tidak harus mengubah keseluruhan antarmuka selama struktur data yang disediakan melalui API tetap konsisten.

2.2.15 Basis Data MySQL

Basis data merupakan kumpulan data terstruktur yang disimpan dan dikelola agar data dapat ditambahkan, dicari, diperbarui, serta digunakan kembali secara konsisten. Dalam sistem informasi, basis data berperan dalam menyimpan hasil proses sistem sehingga informasi tidak hanya tersedia pada saat proses berlangsung, tetapi juga dapat digunakan kembali untuk kebutuhan pemantauan dan pelaporan. Pengelolaan basis data melibatkan struktur tabel, atribut, tipe data, serta hubungan antartabel agar data dapat disimpan secara terorganisasi.

MySQL merupakan sistem manajemen basis data yang menggunakan *Structured Query Language* (SQL) untuk mengelola data. MySQL Server mendukung pengelolaan data melalui tabel serta berbagai operasi untuk menyimpan dan mengambil informasi dari basis data Oracle Corporation (2026). MySQL juga mendukung penggunaan *primary key* dan *foreign key* dalam membentuk hubungan antartabel. *Foreign key* memungkinkan suatu tabel merujuk pada data yang terdapat pada tabel lain serta membantu menjaga konsistensi hubungan antara data pada tabel induk dan tabel anak Oracle Corporation (2026).

MySQL menyediakan berbagai tipe data yang dapat disesuaikan dengan karakteristik data yang disimpan, seperti tipe numerik, karakter, tanggal dan waktu, ENUM, serta JSON Oracle Corporation (2026). Tipe ENUM dapat digunakan untuk membatasi nilai suatu atribut pada sejumlah pilihan yang telah ditentukan, sedangkan tipe JSON memungkinkan penyimpanan data dengan struktur JSON dan melakukan validasi terhadap dokumen JSON yang disimpan Oracle Corporation (2026).

Dalam penelitian ini, MySQL digunakan sebagai basis data untuk menyimpan informasi status slot parkir yang dihasilkan oleh sistem. Data disusun ke dalam dua tabel utama, yaitu `parking_slots` dan `parking_status_logs`. Tabel `parking_slots` digunakan untuk menyimpan kondisi terkini setiap slot parkir yang mencakup kode slot, koordinat *polygon*, status Kosong atau Terisi, nilai *confidence*, jumlah hasil pembacaan, dan waktu

pembaruan. Sementara itu, tabel `parking_status_logs` digunakan untuk menyimpan riwayat status setiap slot parkir berdasarkan waktu.

Hubungan antara tabel `parking_slots` dan `parking_status_logs` dibentuk melalui atribut `slot_id` pada tabel `parking_status_logs` yang merujuk pada atribut `id` pada tabel `parking_slots`. Hubungan tersebut menunjukkan bahwa satu slot parkir dapat memiliki banyak catatan riwayat status, sedangkan setiap catatan riwayat berhubungan dengan satu slot parkir. Penggunaan *foreign key* pada hubungan tersebut membantu mempertahankan integritas referensial antara data slot dan data riwayat status Oracle Corporation (2026).

Dalam penelitian ini, basis data MySQL diintegrasikan dengan *backend* FastAPI. Setelah proses deteksi kendaraan, pemetaan hasil deteksi terhadap *polygon* ROI, dan pembacaan temporal menghasilkan status suatu slot, informasi kondisi terkini diperbarui pada tabel `parking_slots`, sedangkan data historis disimpan pada tabel `parking_status_logs`. Data yang tersimpan selanjutnya digunakan untuk menyajikan informasi kondisi terkini dan riwayat status parkir melalui *dashboard* PHP.

2.2.16 Pengembangan Sistem

Pengembangan sistem merupakan serangkaian aktivitas yang dilakukan untuk menerjemahkan kebutuhan pengguna dan kebutuhan sistem menjadi suatu solusi yang dapat diimplementasikan, digunakan, dan dievaluasi. ISO/IEC/IEEE 12207:2026 memberikan kerangka proses siklus hidup perangkat lunak yang mencakup aktivitas pengadaan, penyediaan, pengembangan, pengoperasian, pemeliharaan, serta penghentian perangkat lunak. Standar tersebut tidak menetapkan satu model pengembangan tertentu sehingga aktivitas pengembangan dapat disesuaikan dengan kebutuhan, ruang lingkup, dan karakteristik sistem yang dikembangkan ISO/IEC/IEEE, (2026).

Pada pengembangan sistem dalam bentuk *prototype*, aktivitas dapat mencakup identifikasi masalah, analisis kebutuhan, perancangan proses dan data, pembangunan komponen sistem, integrasi, demonstrasi, serta pengujian. Kebutuhan sistem dapat dibedakan menjadi kebutuhan fungsional dan kebutuhan nonfungsional. Kebutuhan fungsional menjelaskan layanan atau fungsi yang harus dapat dilakukan oleh sistem, sedangkan kebutuhan nonfungsional menjelaskan karakteristik kualitas maupun batasan operasional, seperti lingkungan eksekusi, waktu respons, keamanan, kemudahan

pemeliharaan, serta kondisi perangkat yang digunakan.

Dalam penelitian ini, pengembangan sistem menghasilkan *prototype* sistem parkir cerdas yang mengintegrasikan video CCTV simulasi, model YOLO26n, *polygon Region of Interest* (ROI), pembacaan temporal, *backend* FastAPI, basis data MySQL, dan *dashboard* PHP. Seluruh komponen diintegrasikan dalam satu alur mulai dari pemrosesan video, deteksi kendaraan, penentuan status slot, penyimpanan data, hingga penyajian informasi parkir kepada pengguna.

Karena sistem yang dikembangkan masih berada pada tingkat *prototype*, hasil penelitian difokuskan pada pembuktian bahwa komponen utama dapat terintegrasi dan menjalankan fungsi yang dirancang. Pengembangan tersebut belum ditujukan untuk membuktikan kesiapan sistem pada lingkungan produksi yang melibatkan banyak kamera, jumlah pengguna yang tinggi, gangguan jaringan, autentikasi tingkat produksi, maupun pengoperasian dalam jangka panjang. Hal tersebut menjadi batasan yang perlu dipertimbangkan dalam pengembangan sistem selanjutnya.

2.2.17 Unified Modeling Language (UML)

Unified Modeling Language (UML) merupakan bahasa pemodelan visual yang digunakan untuk menggambarkan, merancang, dan mendokumentasikan suatu sistem perangkat lunak. UML membantu dalam memvisualisasikan kebutuhan, struktur, perilaku, serta interaksi antarkomponen sistem sehingga rancangan sistem dapat dipahami secara lebih terstruktur. Koç et al. (2021) menjelaskan bahwa UML banyak digunakan dalam *software engineering* sebagai bahasa visual untuk mendukung proses perancangan dan dokumentasi sistem.

Dalam penelitian ini, UML digunakan untuk menggambarkan rancangan *prototype* sistem parkir cerdas yang melibatkan pengguna, pengelola, *dashboard* PHP, *backend* FastAPI, video CCTV simulasi, model YOLO26n, *polygon Region of Interest* (ROI), proses pembacaan temporal, basis data MySQL, serta laporan parkir. Diagram UML yang digunakan terdiri atas *use case diagram*, *activity diagram*, *sequence diagram*, dan *class diagram*.

Use case diagram merupakan diagram yang digunakan untuk menggambarkan hubungan antara aktor dengan fungsi atau layanan yang disediakan oleh sistem. Aktor merupakan pihak di luar sistem yang berinteraksi dengan sistem, sedangkan *use case*

menunjukkan fungsi yang dapat digunakan oleh aktor tersebut. Dalam penelitian ini, aktor utama terdiri atas pengguna dan pengelola. Pengguna dapat memperoleh informasi mengenai ketersediaan slot parkir dan melihat *snapshot* CCTV, sedangkan pengelola dapat memantau kondisi parkir serta mengakses laporan yang disediakan sistem.

Simbol yang digunakan dalam penyusunan *use case diagram* pada penelitian ini ditunjukkan pada Tabel 2.2.

Tabel 2. 2 Simbol Use Case Diagram

Simbol	Nama Simbol	Pengertian	Penggunaan pada Sistem Parkir Cerdas
	Aktor	Pihak yang berinteraksi dengan sistem untuk menggunakan fungsi yang tersedia.	Merepresentasikan Pengguna dan Pengelola sebagai pihak yang berinteraksi dengan sistem.
	<i>Use Case</i>	Fungsi atau layanan yang disediakan oleh sistem kepada aktor.	Digunakan untuk fungsi Login, Melihat Dashboard, Melihat Status Slot, Melihat <i>Snapshot</i> CCTV, Melihat Laporan, dan Logout.
	<i>Association</i>	Hubungan antara aktor dengan fungsi yang dapat digunakan.	Menunjukkan hubungan Pengguna atau Pengelola dengan fungsi sistem yang dapat diakses.
	<i>Include</i>	Hubungan yang menunjukkan bahwa suatu fungsi membutuhkan fungsi lain sebagai bagian dari prosesnya.	Menunjukkan keterkaitan proses Login dengan akses terhadap fungsi sistem.
	<i>System Boundary</i>	Batas yang menunjukkan fungsi-fungsi yang termasuk dalam suatu sistem.	Membatasi fungsi yang terdapat dalam Sistem Parkir Cerdas Berbasis Video CCTV.

Berdasarkan Tabel 2.2, simbol *use case diagram* digunakan untuk merepresentasikan aktor, fungsi sistem, serta hubungan antara aktor dengan fungsi yang tersedia pada sistem parkir cerdas.

Activity diagram digunakan untuk menggambarkan urutan aktivitas atau alur kerja suatu proses dalam sistem. Diagram ini dapat menunjukkan titik awal, aktivitas, keputusan, percabangan, penggabungan, hingga titik akhir proses. Dalam penelitian ini, *activity diagram* digunakan untuk menggambarkan alur aktivitas pengguna dan

pengelola ketika mengakses fungsi yang tersedia pada sistem parkir cerdas.

Simbol yang digunakan dalam penyusunan *activity diagram* pada penelitian ini ditunjukkan pada Tabel 2.3.

Tabel 2. 3 Simbol Activity Diagram

Simbol	Nama Simbol	Pengertian	Penggunaan pada Sistem Parkir Cerdas
	<i>Initial Node</i>	Menunjukkan titik awal suatu alur aktivitas.	Menunjukkan awal proses ketika Pengguna atau Pengelola mulai mengakses sistem.
	<i>Activity</i>	Menunjukkan aktivitas atau proses yang dilakukan dalam suatu alur.	Digunakan untuk aktivitas Login, Melihat Dashboard, Melihat Status Slot, Melihat <i>Snapshot</i> CCTV, dan Logout.
	<i>Decision Node</i>	Menunjukkan percabangan alur berdasarkan kondisi tertentu.	Digunakan dalam proses validasi login untuk menentukan apakah kredensial valid atau tidak valid.
	<i>Control Flow</i>	Menunjukkan arah perpindahan dari suatu aktivitas ke aktivitas berikutnya.	Menunjukkan urutan proses interaksi pengguna atau pengelola dengan sistem.
	<i>Final Node</i>	Membagi aktivitas berdasarkan pihak atau komponen yang bertanggung jawab.	Digunakan untuk membedakan aktivitas Pengguna/Pengelola, Sistem FastAPI, dan Basis Data MySQL.
	<i>Swimlane</i>	Menunjukkan akhir dari suatu rangkaian aktivitas.	Digunakan ketika proses penggunaan sistem telah selesai setelah Logout.

Berdasarkan Tabel 2.3, simbol *activity diagram* digunakan untuk menggambarkan urutan aktivitas, keputusan, pembagian tanggung jawab, serta awal dan akhir proses pada sistem parkir cerdas.

Sequence diagram digunakan untuk menggambarkan interaksi antarobjek berdasarkan urutan waktu. Objek atau komponen sistem ditampilkan melalui *lifeline*, sedangkan komunikasi antarkomponen direpresentasikan melalui pesan yang dikirimkan. Dalam penelitian ini, *sequence diagram* digunakan untuk menunjukkan aliran interaksi mulai dari permintaan pengguna melalui *dashboard*, pemrosesan pada

FastAPI, deteksi kendaraan menggunakan YOLO26n, pemetaan terhadap *polygon* ROI, penentuan status slot, penyimpanan data pada basis data, hingga informasi dikembalikan kepada *dashboard*.

Simbol yang digunakan dalam penyusunan *sequence diagram* pada penelitian ini ditunjukkan pada Tabel 2.4.

Tabel 2. 4 Simbol Sequence Diagram

Simbol	Nama Simbol	Pengertian	Penggunaan pada Sistem Parkir Cerdas
	<i>Actor</i>	Pihak eksternal yang memulai atau berinteraksi dalam suatu proses.	Merepresentasikan Pengguna atau Pengelola yang mengakses dashboard.
	<i>Lifeline</i>	Menunjukkan keberadaan aktor atau objek selama berlangsungnya interaksi.	Digunakan pada Dashboard PHP, FastAPI, Video CCTV, YOLO26n, proses ROI/Temporal, dan MySQL.
	<i>Activation</i>	Menunjukkan periode ketika suatu objek sedang menjalankan proses.	Menunjukkan saat FastAPI, YOLO26n, atau komponen lain sedang melakukan proses tertentu.
	<i>Message</i>	Menunjukkan permintaan atau pesan yang dikirim dari satu objek ke objek lain.	Digunakan untuk proses seperti <code>requestStatus()</code> , <code>ambilFrame()</code> , <code>deteksiKendaraan()</code> , <code>petakanKeROI()</code> , dan <code>pembacaanTemporal()</code> .
	<i>Return Message</i>	Menunjukkan respons atau hasil yang dikembalikan setelah suatu proses.	Digunakan untuk mengembalikan <i>frame</i> , hasil deteksi, status slot, serta data dari MySQL.

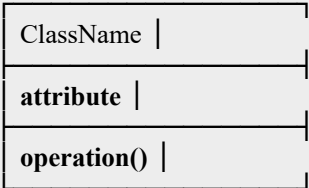
Berdasarkan Tabel 2.4, simbol *sequence diagram* digunakan untuk memperlihatkan komponen yang berinteraksi, pesan yang dikirim, respons yang diterima, serta urutan proses pembaruan status slot parkir.

Class diagram merupakan diagram UML yang digunakan untuk menggambarkan struktur statis sistem melalui kelas, atribut, operasi, serta hubungan antarkelas. Setiap kelas mewakili objek atau komponen yang memiliki karakteristik dan tanggung jawab tertentu. Hubungan antarkelas dapat dilengkapi dengan *multiplicity* atau kardinalitas untuk menunjukkan jumlah objek yang dapat berpartisipasi dalam suatu

hubungan.

Dalam penelitian ini, *class diagram* digunakan untuk memperjelas struktur data dan hubungan komponen pada sistem parkir cerdas. Salah satu hubungan utama terdapat antara data slot parkir dan data riwayat status. Satu slot parkir dapat memiliki nol atau lebih catatan riwayat status selama sistem dijalankan, sedangkan setiap catatan riwayat status berkaitan dengan satu slot parkir. Hubungan tersebut direpresentasikan menggunakan kardinalitas pada *class diagram* sehingga struktur dan keterkaitan data dapat dipahami secara lebih jelas. *Class diagram* menggunakan beberapa simbol untuk menggambarkan struktur kelas, atribut, operasi, hubungan antarkelas, serta jumlah objek yang terlibat dalam setiap hubungan. Simbol yang digunakan dalam penyusunan *class diagram* pada penelitian ini ditunjukkan pada Tabel 2.5.

Tabel 2. 5 Simbol Class Diagram

Simbol	Nama Simbol	Pengertian	Penggunaan pada Sistem Parkir Cerdas
	Class	Merepresentasikan suatu objek atau entitas dalam sistem yang memiliki nama, atribut, dan operasi.	Digunakan untuk merepresentasikan kelas CCTVVideo, VideoFrame, VehicleDetector, VehicleDetection, PolygonROI, ParkingSlot, TemporalStatus, dan ParkingStatusLog.
- slotCode : string	Attribute	Menunjukkan data atau karakteristik yang dimiliki oleh suatu kelas.	Digunakan untuk merepresentasikan data seperti slotCode, currentStatus, confidence, boundingBox, votes, dan timestamp.
+ getStatus ()	Operation / Method	Menunjukkan fungsi atau perilaku yang dapat dilakukan oleh suatu kelas.	Digunakan untuk merepresentasikan proses seperti mengambil <i>frame</i> , mendeteksi kendaraan, memetakan hasil deteksi ke ROI, menentukan status temporal, memperbarui status, dan mengambil riwayat.
	Association	Menunjukkan hubungan struktural antara dua kelas yang saling berhubungan dalam sistem.	Digunakan untuk menunjukkan hubungan seperti CCTVVideo dengan VideoFrame, PolygonROI dengan ParkingSlot, serta

			ParkingSlot dengan ParkingStatusLog.
1, 0..1, 1..*, 0..*	Multiplicity / Cardinality	Menunjukkan jumlah objek yang dapat terlibat dalam suatu hubungan antarkelas.	Digunakan untuk menjelaskan jumlah hubungan antarkelas, misalnya satu ParkingSlot dapat memiliki nol atau banyak ParkingStatusLog.
+	Public	Menunjukkan bahwa atribut atau operasi dapat diakses dari luar kelas.	Digunakan pada bagian operasi atau <i>method</i> , seperti + getStatus() dan + updateStatus().
-	Private	Menunjukkan bahwa atribut hanya dapat diakses atau dikelola di dalam kelas yang bersangkutan.	Digunakan pada atribut kelas, seperti - slotId, - currentStatus, dan - confidence.

Berdasarkan Tabel 2.5, *class diagram* pada penelitian ini menggunakan simbol kelas, atribut, operasi, *association*, dan *multiplicity* untuk menggambarkan struktur serta hubungan antarobjek pada sistem parkir cerdas. Simbol + digunakan untuk menunjukkan operasi yang bersifat *public*, sedangkan simbol - digunakan untuk menunjukkan atribut yang bersifat *private*. Kardinalitas digunakan untuk memperjelas jumlah objek yang dapat terlibat dalam setiap hubungan antarkelas.

Tabel 2. 6 Keterangan Kardinalitas Class Diagram

Notasi	Makna
1	Tepat satu objek
0..1	Tidak ada atau paling banyak satu objek
1..*	Sedikitnya satu atau dapat berjumlah banyak
0..*	Tidak ada atau dapat berjumlah banyak
*	Banyak objek

Penggunaan kardinalitas pada *class diagram* membantu memperjelas jumlah objek yang dapat berhubungan antara satu kelas dengan kelas lainnya. Pada penelitian ini, hubungan antara slot parkir dan riwayat status menggunakan kardinalitas 1 : 0..*, yang menunjukkan bahwa satu slot parkir dapat belum memiliki catatan riwayat atau dapat memiliki banyak catatan riwayat, sedangkan setiap catatan riwayat mengacu pada satu slot parkir.

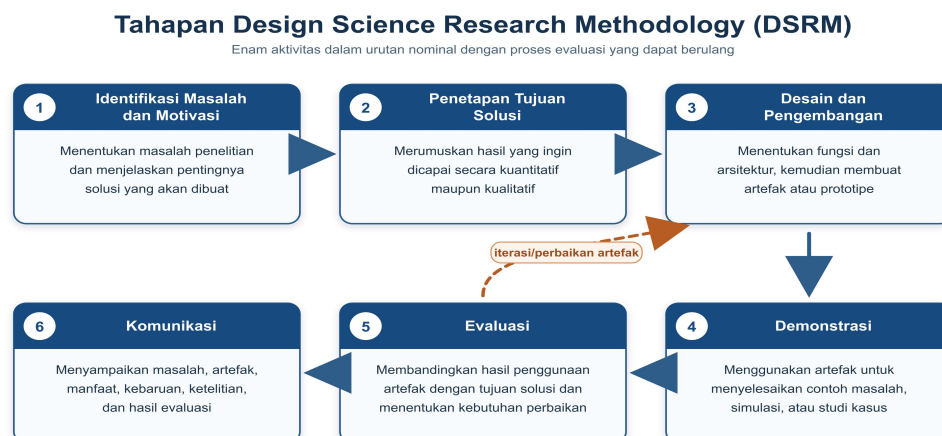
2.2.18 *Design Science Research Methodology*

Design Science Research Methodology (DSRM) merupakan pendekatan penelitian yang digunakan untuk merancang dan mengevaluasi artefak sebagai solusi terhadap suatu permasalahan. Artefak dalam penelitian sistem informasi dapat berbentuk konstruk, model, metode, maupun aplikasi atau *prototype*. Pendekatan *design science* menekankan bahwa hasil penelitian tidak hanya dinilai berdasarkan keberhasilan artefak yang dihasilkan, tetapi juga berdasarkan proses perancangan, evaluasi, serta kontribusi pengetahuan yang diperoleh dari pengembangan artefak tersebut Hevner et al. (2004).

Peppers et al. (2007) membagi DSRM menjadi enam aktivitas utama, yaitu identifikasi masalah dan motivasi, penetapan tujuan solusi, desain dan pengembangan, demonstrasi, evaluasi, serta komunikasi. Keenam aktivitas tersebut membentuk tahapan yang digunakan untuk mengarahkan proses pengembangan artefak dari identifikasi permasalahan hingga penyampaian hasil penelitian.

Tahap identifikasi masalah dan motivasi dilakukan untuk menentukan permasalahan yang akan diselesaikan serta menjelaskan pentingnya solusi yang dikembangkan. Tahap penetapan tujuan solusi digunakan untuk menentukan hasil atau kemampuan yang diharapkan dari artefak. Selanjutnya, tahap desain dan pengembangan berfokus pada perancangan dan pembangunan artefak berdasarkan kebutuhan yang telah ditentukan.

Tahap demonstrasi dilakukan untuk menunjukkan bahwa artefak dapat digunakan dalam konteks permasalahan yang diteliti. Setelah itu, tahap evaluasi digunakan untuk menilai kesesuaian artefak terhadap tujuan yang telah ditetapkan. Apabila hasil evaluasi belum memenuhi kebutuhan, proses pengembangan dapat kembali dilakukan untuk memperbaiki artefak. Tahap terakhir, yaitu komunikasi, digunakan untuk menyampaikan proses penelitian, hasil pengembangan, hasil evaluasi, manfaat, keterbatasan, dan kontribusi penelitian kepada pihak yang berkepentingan.



Gambar 2. 6 Tahapan *Design Science Research Methodology* (DSRM)

Dalam penelitian ini, DSRM digunakan karena penelitian menghasilkan artefak berupa *prototype* sistem parkir cerdas. Permasalahan penelitian berkaitan dengan kebutuhan penyediaan informasi kondisi slot parkir secara lebih terstruktur. Artefak dikembangkan dengan mengintegrasikan video CCTV simulasi, model YOLO26n hasil *fine-tuning*, *polygon Region of Interest* (ROI), pembacaan temporal, *backend* FastAPI, basis data MySQL, dan *dashboard* PHP. Artefak kemudian didemonstrasikan menggunakan video CCTV simulasi serta dievaluasi melalui pengujian model deteksi kendaraan dan pengujian fungsi sistem. Hasil pengembangan, evaluasi, serta keterbatasan penelitian selanjutnya dikomunikasikan melalui laporan penelitian.

2.2.19 Pengujian Sistem

Pengujian sistem merupakan proses yang dilakukan untuk memperoleh bukti bahwa fungsi-fungsi yang telah dikembangkan menghasilkan keluaran sesuai dengan kebutuhan dan skenario yang telah ditentukan. Salah satu metode yang dapat digunakan untuk mengevaluasi fungsi perangkat lunak adalah *black-box testing*. Pada *black-box testing*, pengujian dilakukan berdasarkan spesifikasi fungsional sistem tanpa memeriksa struktur internal maupun kode program. Setiap kasus uji dapat memuat kondisi awal, tindakan atau masukan, hasil yang diharapkan, hasil aktual, serta kesimpulan keberhasilan pengujian Black et al. (2021)

Dalam *black-box testing*, penguji menjalankan fungsi sistem dan membandingkan keluaran aktual dengan keluaran yang diharapkan. Fokus pengujian

berada pada perilaku sistem yang dapat diamati oleh pengguna, sehingga penguji tidak perlu mengetahui bagaimana kode program menghasilkan keluaran tersebut. Pengujian dapat melibatkan pihak yang memiliki keterkaitan dengan penggunaan maupun pengelolaan sistem agar fungsi yang diuji dapat dilihat dari sudut pandang kebutuhan yang berbeda.

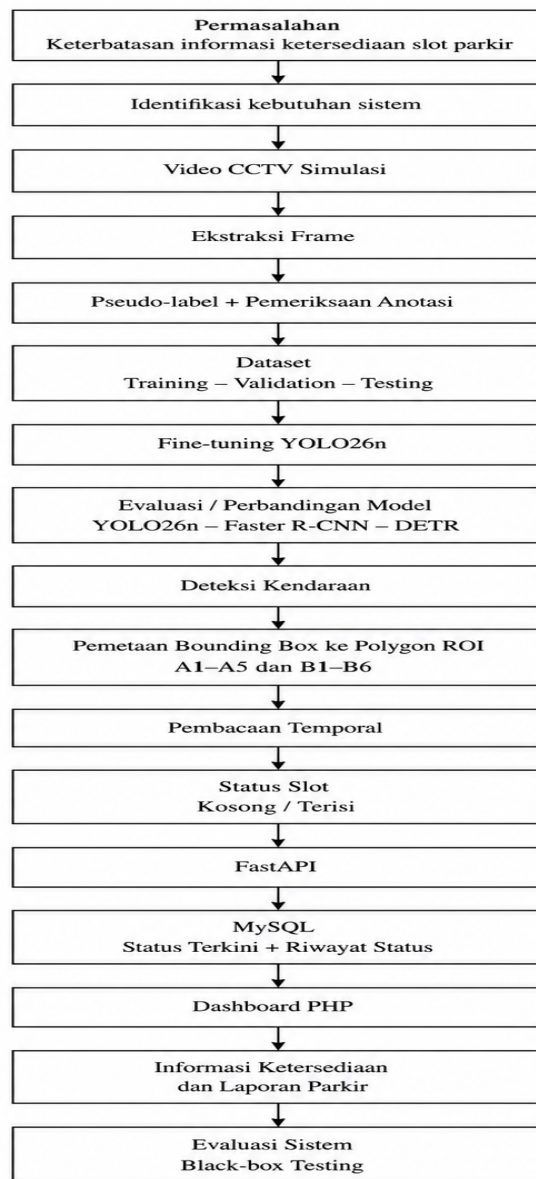
Pada penelitian ini, penguji dibedakan berdasarkan peran yang berkaitan dengan *prototype* sistem parkir cerdas. Pengelola atau petugas parkir mewakili pihak yang berkepentingan terhadap proses pemantauan kondisi parkir, mahasiswa atau staf mewakili calon pengguna informasi ketersediaan parkir, sedangkan pihak yang memahami teknologi informasi digunakan untuk membantu memeriksa fungsi antarmuka, *endpoint*, dan alur pertukaran data sistem. Setiap penguji menggunakan skenario serta kriteria keberhasilan yang sama agar hasil pengujian dapat dibandingkan secara konsisten.

Pengujian pada sistem yang mengintegrasikan model *deep learning* perlu dibedakan antara evaluasi performa model deteksi dan evaluasi fungsi aplikasi. Evaluasi model deteksi digunakan untuk mengetahui kemampuan model dalam mendeteksi kendaraan melalui metrik seperti *precision*, *recall*, mAP, dan *confusion matrix*. Sementara itu, pengujian fungsi sistem digunakan untuk memastikan bahwa komponen seperti *backend*, *endpoint* status, *snapshot* CCTV, pemetaan hasil deteksi terhadap *polygon* ROI, pembacaan temporal, penentuan status slot, penyimpanan data, *dashboard*, dan halaman laporan dapat berjalan sesuai dengan fungsi yang ditentukan. Pemisahan tersebut penting karena performa model deteksi yang tinggi tidak secara otomatis menunjukkan bahwa seluruh fungsi sistem telah bekerja dengan benar.

Dalam penelitian ini, *black-box testing* digunakan untuk menguji fungsi utama *prototype* sistem parkir cerdas. Setiap penguji menjalankan skenario yang sama, membandingkan hasil aktual dengan hasil yang diharapkan, kemudian memberikan status Berhasil apabila fungsi menghasilkan keluaran sesuai dengan skenario atau Gagal apabila keluaran tidak sesuai maupun fungsi tidak dapat dijalankan. Penguji juga dapat mencatat kendala dan saran yang ditemukan selama proses pengujian. Hasil pengujian hanya menggambarkan fungsi, data, serta kondisi yang benar-benar diuji dan tidak menunjukkan bahwa sistem telah bebas dari seluruh kemungkinan kesalahan.

2.3 Kerangka Pemikiran

Kerangka pemikiran menggambarkan alur logis penelitian dalam menghasilkan *prototype* sistem parkir cerdas berbasis video CCTV simulasi. Penelitian berangkat dari permasalahan keterbatasan informasi mengenai ketersediaan slot parkir yang dapat dipantau secara terstruktur. Kondisi tersebut menyebabkan pengguna masih perlu mengetahui ketersediaan parkir melalui pengamatan secara langsung, sedangkan pengelola memerlukan media yang dapat menyajikan kondisi setiap slot parkir secara lebih terorganisasi. Kerangka pemikiran penelitian ditunjukkan pada Gambar 2.7.



Gambar 2. 7 Kerangka Pemikiran Penelitian

Solusi yang dikembangkan berupa *prototype* sistem parkir cerdas berbasis *computer vision* dan *deep learning*. Video CCTV simulasi dengan posisi kamera tetap digunakan sebagai sumber data visual. Sejumlah *frame* kemudian diekstraksi dari video untuk membentuk dataset deteksi kendaraan. Anotasi awal dibuat melalui proses *pseudo-label*, kemudian dilakukan pemeriksaan dan perbaikan secara visual agar *bounding box* serta kelas objek sesuai dengan kendaraan yang terdapat pada citra.

Dataset yang telah disiapkan selanjutnya dibagi menjadi data *training*, *validation*, dan *testing*. Model YOLO26n dilakukan *fine-tuning* untuk mendeteksi satu kelas objek, yaitu *vehicle*. Kemampuan model dievaluasi menggunakan *precision*, *recall*, mAP@50, mAP@50–95, dan *confusion matrix*. Untuk memperoleh gambaran perbandingan performa, YOLO26n juga dibandingkan dengan Faster R-CNN dan DETR menggunakan data pengujian yang sama.

Model YOLO26n hasil *fine-tuning* digunakan sebagai komponen deteksi kendaraan pada *prototype*. Hasil deteksi berupa *bounding box* kendaraan kemudian dipetakan terhadap *polygon Region of Interest* (ROI) pada setiap slot parkir. Area parkir direpresentasikan menjadi sebelas slot, yaitu A1 sampai A5 dan B1 sampai B6. Hasil pemetaan pada beberapa *frame* dibaca secara temporal untuk membantu menghasilkan status setiap slot sebagai Kosong atau Terisi secara lebih stabil.

Informasi status slot selanjutnya diproses melalui *backend* FastAPI dan disimpan pada basis data MySQL. Basis data menyimpan kondisi terkini setiap slot parkir serta riwayat status yang digunakan untuk kebutuhan pemantauan dan pelaporan. Data tersebut kemudian disajikan melalui *dashboard* PHP dalam bentuk jumlah slot, jumlah slot Kosong dan Terisi, status masing-masing slot, *snapshot* CCTV, serta laporan riwayat parkir.

Tahap evaluasi dilakukan pada dua aspek, yaitu evaluasi model deteksi kendaraan dan pengujian fungsi sistem. Evaluasi model digunakan untuk mengetahui kemampuan model dalam mendeteksi dan melokalisasi kendaraan, sedangkan *black-box testing* digunakan untuk memeriksa kesesuaian fungsi *prototype* dengan skenario pengujian. Dengan demikian, kerangka pemikiran menghubungkan permasalahan penelitian, pengembangan model deteksi, penentuan status slot, integrasi sistem, penyimpanan data, penyajian informasi, dan pengujian dalam satu alur penelitian.