BAB 2
LANDASAN TEORI

2.1 Tinjauan Teori

2.1.1 Matriks Kerusakan Jalan

Matriks kerusakan jalan merupakan kerangka kerja terstandarisasi yang digunakan untuk mengklasifikasikan jenis, tingkat keparahan (*severity level*), dan kuantitas kerusakan pada perkerasan jalan. Matriks ini berfungsi sebagai label *ground truth* untuk melatih model deteksi objek dan segmentasi.

Klasifikasi Tingkat Keparahan (*Severity Level*)

Tabel 2.1. Matriks Tingkat Kerusakan Jalan

No	Jenis Kerusakan	Rendah	Sedang	Tinggi
1	*Longitudinal Cracking*	Lebar retak $\leq$ 6 mm.	Lebar retak $>$ 6 mm s/d $\leq$ 19 mm.	Lebar retak $>$ 19 mm.
2	*Lateral Cracking*	Lebar retak $\leq$ 6 mm.	Lebar retak $>$ 6 mm s/d $\leq$ 19 mm.	Lebar retak $>$ 19 mm.
3	*Alligator Cracking*	Lebar retak $\leq$ 6 mm.	Lebar retak $>$ 6 mm s/d $\leq$ 19 mm.	Lebar retak $>$ 19 mm.
4	*Potholes* (Lubang)	Kedalaman $<$ 25 mm.	Kedalaman 25 mm s/d 50 mm.	Kedalaman $>$ 50 mm.

Sumber: [17]

Berdasarkan standar penilaian jalan yang umum digunakan (seperti *Pavement Condition Index* atau kriteria teknis terkait), tingkat keparahan biasanya dibagi menjadi tiga kategori utama: *Low* (Rendah), *Medium* (Sedang), dan *High* (Tinggi) [17].

- Parameter Berbasis Geometri: Klasifikasi ini umumnya didasarkan pada dimensi fisik kerusakan. Misalnya, untuk retak (*cracking*), parameter utamanya adalah lebar celah (*crack width*). Untuk lubang (*potholes*), parameternya adalah kedalaman dan diameter [18].
- Kriteria Matriks: Ambang batas tingkat keparahan seringkali menggunakan nilai numerik yang spesifik (contoh: lebar retak $\leq$ 6 mm untuk kategori Rendah, dan $>$ 19 mm untuk kategori Tinggi). Penggunaan MiDaS dalam penelitian ini sangat relevan untuk mengestimasi dimensi kedalaman untuk menyesesuaikan dengan kriteria matriks tersebut [18].

Jenis-Jenis Kerusakan pada Matriks

Matriks kerusakan jalan mencakup berbagai tipe kegagalan struktural dan fungsional, di antaranya:

Gambar 2.1. Gambar Kerusakan Retak

- Retak (*Cracking*): Meliputi *Longitudinal*, *Alligator* dan *Lateral Cracking*. Deteksi tipe retak ini sangat bergantung pada fitur spasial dan orientasi [17].

Gambar 2.2. Gambar Kerusakan Lubang

- Lubang (*Potholes*): Kerusakan berbentuk mangkuk dengan tingkat keparahan yang diukur berdasarkan kedalaman. Estimasi kedalaman (*depth estimation*) menjadi krusial karena semakin dalam maka semakin tinggi severity [17].

Hanya empat tipe kerusakan yang dianalisis dalam penelitian ini. Pertimbangannya adalah beberapa jenis kerusakan seperti *block*, *edge*, *reflection*, dan *transverse cracking* memiliki kesamaan klasifikasi pada tabel matriks yang digunakan.

Integrasi *Deep Learning* dalam Penilaian Matriks

Integrasi dilakukan secara sekuensial dengan memanfaatkan luaran *bounding box* dari model *Detection Transformer* (DETR) yang mendeteksi lokasi dan tipe kerusakan jalan. Area

bounding box tersebut kemudian dipetakan ke dalam *depth map* milik MiDaS untuk diekstrak nilai kedalaman relatifnya. Hasil kalkulasi nilai kedalaman ini menjadi dasar penentuan klasifikasi tingkat keparahan (Rendah, Sedang, atau Tinggi) sesuai dengan standar matriks penilaian yang ditetapkan.

2.1.2 DETR

Detection Transformer (DETR) merupakan kerangka kerja deteksi objek yang memandang proses deteksi sebagai masalah *direct set prediction problem* [19]. Model ini merampingkan alur kerja deteksi dengan menghilangkan komponen desain manual seperti pembuatan *anchors box* dan prosedur *Non-Maximum Suppression* (NMS) yang secara eksplisit mengkodekan pengetahuan awal tentang tugas deteksi. Melalui arsitektur transformer *encoder-decoder* dan fungsi kerugian berbasis *bipartite matching*, DETR mampu menghasilkan prediksi unik secara paralel [13].

Fungsi *Set Prediction Loss*

DETR menghasilkan set prediksi berukuran tetap sebanyak N dalam satu kali lintasan melalui *decoder*. Inti dari proses pelatihan adalah fungsi kerugian yang menghasilkan *bipartite matching* optimal antara objek prediksi dan objek *ground truth* menggunakan algoritma *Hungarian* [20]. Proses ini dilakukan dalam dua tahap:

1. *Optimal Bipartite Matching*: Mencari permutasi elemen N dengan biaya terendah ($\hat{\sigma}$) menggunakan rumus:

$$\hat{\sigma} = \arg\min_{\sigma \in \mathfrak{S}_N} \sum_{i}^{N} \mathscr{L}_{match}(y_i, \hat{y}_{\sigma(i)}) \tag{2.1}$$

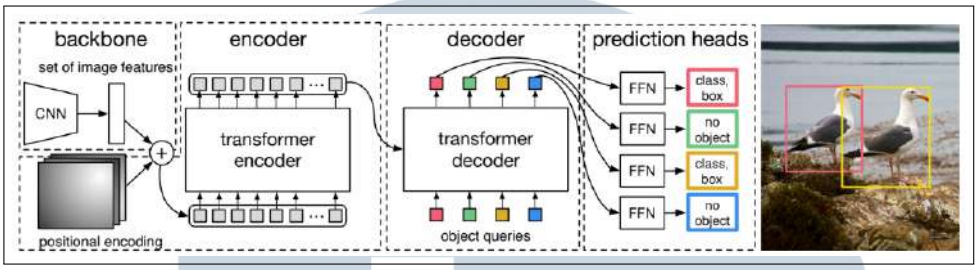
   Di mana $\mathscr{L}_{match}(y_i, \hat{y}_{\sigma(i)})$ adalah biaya kecocokan pasangan yang mempertimbangkan prediksi kelas dan kesesuaian kotak pembatas.

2. Hungarian Loss: Setelah kecocokan ditemukan, model dioptimalkan menggunakan *Hungarian loss*:

$$\mathscr{L}_{Hungarian}(y, \hat{y}) = \sum_{i=1}^{N} \left[-\log \hat{p}_{\hat{\sigma}(i)}(c_i) + \mathbb{I}_{\{c_i \neq \emptyset\}} \mathscr{L}_{box}(b_i, \hat{b}_{\hat{\sigma}(i)}) \right] \tag{2.2}$$

   Di mana $\mathscr{L}_{box}$ merupakan kombinasi linear dari *loss* l_1 dan *generalized IoU loss* yang bersifat *scale-invariant*.

Alur Pemrosesan Gambar pada DETR



Gambar 2.3. Alur Pemrosesan DETR

Berdasarkan arsitektur yang dikembangkan oleh Carion et al., alur pemrosesan gambar dalam DETR dibagi menjadi beberapa tahap sistematis [14]:

- Ekstraksi Fitur: gambar masukan diproses oleh *backbone* CNN (seperti ResNet-50) untuk menghasilkan peta aktivasi beresolusi rendah [14].
- *Transformer Encoder*: Representasi fitur 2D diratakan (*flattened*) dan digabungkan dengan *positional encoding* untuk mempertahankan informasi spasial sebelum dimasukkan ke dalam unit *encoder* [14].
- *Transformer Decoder*: *Decoder* menerima sejumlah *object queries* (parameter *positional embeddings* yang dipelajari) sebagai masukan. Setiap *query* secara paralel memperhatikan *output* dari *encoder* untuk melakukan penalaran global terhadap seluruh objek dalam gambar [14].
- Prediksi Akhir: Setiap *output* dari *decoder* diteruskan ke *Feed-Forward Network* (FFN) bersama yang memprediksi koordinat kotak pembatas dan label kelas, termasuk kelas khusus "tanpa objek" (∅) jika tidak ada objek yang terdeteksi pada slot tersebut [14].

Pemilihan DETR dalam Deteksi Kerusakan Jalan

Dalam pelatihan model *deep learning* pada arsitektur *Detection Transformer* (DETR), kualitas dan kuantitas data menentukan performa akhir model. Konsep ini dikenal sebagai *Scaling Laws* (Hukum Penskalaan). Penelitian empiris oleh [21] menunjukkan bahwa peningkatan kapasitas arsitektur model memerlukan penambahan jumlah *dataset* pelatihan secara proporsional untuk mencapai performa optimal.

Tingkat kesalahan prediksi model menurun seiring dengan bertambahnya volume data [22]. Oleh karena itu, ketersediaan *dataset* gambar berskala besar dan bervariasi menjadi syarat mutlak untuk mencapai tingkat presisi yang tinggi dalam mendeteksi dan mengklasifikasi tingkat keparahan kerusakan jalan.

Pemilihan DETR didasarkan pada efisiensi implementasi dan fleksibilitas arsitekturnya yang adaptif. Pada konteks kerusakan jalan, mekanisme perhatian (*attention*) pada *encoder* efektif memisahkan *instance* objek secara global. Kemampuan ini diperlukan untuk mengidentifikasi kerusakan yang saling bertumpang-tindih maupun kerusakan berukuran kecil di area tekstur aspal yang kompleks [23].

2.1.3 Perbandingan RF-DETR

Perkembangan metodologi deteksi objek secara umum terbagi dua, yakni jaringan berbasis konvolusi (*Convolutional Neural Networks*/CNN) dan arsitektur berbasis *Transformer*. RF-DETR (*Roboflow Detection Transformer*) mengintegrasikan mekanisme *Neural Architecture Search* (NAS) untuk mengoptimalkan efisiensi komputasi *Transformer* tanpa mengorbankan akurasi lokalisasi. Untuk memahami posisi dan keunggulan teknis dari RF-DETR, diperlukan tinjauan komparatif mendalam dengan arsitektur lainnya.

RF-DETR vs Faster R-CNN

Faster R-CNN merepresentasikan *two-stage detector* yang memisahkan proses penentuan kandidat objek melalui *Region Proposal Network* (RPN) dan tahap klasifikasi akhir melalui modul *refiner* berbasis CNN [24]. Ekstraksi fitur pada Faster R-CNN bergantung pada pencocokan matriks berbasis *Intersection over Union* (IoU) dan operasi *RoI Align* yang rentan menghasilkan peta fitur regional dengan informasi berlebih [24].

Secara konseptual, struktur *encoder-decoder* pada rumpun model DETR (termasuk RF-DETR) mengadopsi prinsip dasar desain RPN-*refiner* tersebut namun dengan perombakan mekanisme ekstraksi [24]. RF-DETR mengganti *RoI Align* dengan modul *deformable attention* sebagai komponen *proposal refiner* yang dikombinasikan dengan pencocokan *Hungarian* secara *end-to-end* [24].

Keunggulan RF-DETR atas Faster R-CNN tidak diukur semata-mata dari perbandingan metrik *Average Precision* (AP) langsung, melainkan dari efisiensi struktural berikut:

- Metode Pencocokan: Faster R-CNN menghasilkan banyak prediksi *many-to-one* yang mensyaratkan pasca-pemrosesan. Sementara itu, RF-DETR menggunakan *bipartite matching* yang memastikan prediksi objek *one-to-one*.
- Komponen Ekstraksi: Peralihan dari *RoI Align* (konvolusi spasial kaku pada Faster R-CNN) ke *deformable attention* (pada RF-DETR) memungkinkan model untuk fokus hanya pada titik-titik referensi di sekitar objek, mengeliminasi redudansi komputasi pada area latar belakang yang tidak relevan.

RF-DETR vs DETR Standar

Arsitektur DETR standar penerapan *Transformer* untuk memprediksi himpunan objek secara langsung, menghapus kebutuhan *anchor boxes* dan *Non-Maximum Suppression* (NMS) [24, 25]. DETR memiliki kendala serius berupa lambatnya waktu konvergensi dan beban komputasi akibat penggunaan mekanisme *global attention*, sehingga sulit diterapkan untuk inferensi waktu nyata (*real-time*) [25, 26]. RF-DETR mengatasi ini dengan menjadi detektor *Transformer* spesialis yang berbobot ringan (*lightweight*) [27].

Pengujian pada dataset COCO memperlihatkan perbedaan efisiensi komputasi antara varian DETR konvensional dan RF-DETR yang telah dioptimasi dengan NAS, sebagaimana dirangkum pada Tabel 2.2.

Tabel 2.2. Perbandingan Metrik Kuantitatif RF-DETR dan DETR Standar

Model Arsitektur	Dataset	Akurasi (AP)	Kecepatan / Latensi Inferensi
DINO-Deformable-DETR-R50	COCO	50,9%	5 FPS
RF-DETR (Nano)	COCO	48,0%	2,3 ms
RF-DETR (2x-Large)	COCO	60,1%	17,2 ms

Sumber: [27]

RF-DETR vs RT-DETR

RT-DETR (*Real Time Detection Transformer*) dirancang sebagai detektor *Transformer* pertama yang mampu beroperasi secara *real-time* dengan memecahkan hambatan komputasi melalui penggunaan *hybrid encoder* dan skema seleksi kueri objek dengan ketidakpastian minimum [25]. RF-DETR dibangun di atas fondasi tersebut dan diperluas menggunakan *weight-sharing Neural Architecture Search* (NAS) secara *end-to-end*, memungkinkan pencarian keseimbangan optimal antara akurasi dan latensi pada dataset spesifik tanpa perlu *retraining* [27]. Selain itu, RF-DETR menggunakan *backbone* DINOv2 yang dilatih secara *self-supervised* skala internet [28], berbeda dengan *backbone* CNN standar pada RT-DETR.

Perbandingan pada Tabel 2.3 menunjukkan bagaimana RF-DETR meningkatkan plafon akurasi dari RT-DETR pendahulunya melalui optimasi struktural tersebut.

Tabel 2.3. Perbandingan Performa Struktural RF-DETR dan RT-DETR

Model Arsitektur	Modul Pendukung	Akurasi (AP)	Performance
RT-DETR-R50	Hybrid Encoder + ResNet	53,1%	108 FPS
RF-DETR (2x-Large)	NAS + DINOv2 Backbone	60,1%	17,2 ms

Sumber: [27]

RF-DETR vs YOLO

YOLO merupakan *single-stage detector* berbasis CNN yang mengutamakan kecepatan tinggi [28]. Meskipun efisien, YOLO sangat bergantung pada NMS untuk menyring prediksi ganda, yang menuntut pencarian nilai ambang batas hiperparameter (seperti IoU) yang sering kali tidak

stabil [25]. YOLOv12 mulai mengadopsi mekanisme atensi (*area attention* dan R-ELAN), namun representasi fitur lokal pada arsitektur CNN-nya terbukti memiliki keterbatasan saat dihadapkan pada skenario spasial yang ekstrem, seperti objek yang menyamar dengan latar belakang (*camouflage*) atau tumpang tindih (*occlusion*) [28].

Pemodelan konteks global melalui *backbone* DINOv2 menjadikan RF-DETR memiliki keunggulan pengenalan spasial yang luar biasa kuat dibandingkan YOLO. Kemampuan ini sangat krusial tidak hanya untuk mendeteksi objek pertanian, namun juga dapat diterapkan secara langsung pada tugas identifikasi pola ireguler yang menuntut presisi tingkat keparahan tinggi, seperti analisis piksel pada deteksi kerusakan permukaan jalan.

Pada pengujian deteksi objek di lingkungan dengan rintangan visual ekstrem (seperti buah hijau di antara dedaunan hijau), model RF-DETR menunjukkan dominasi akurasi dan kecepatan konvergensi pelatihan dibandingkan arsitektur CNN mutakhir sebagaimana dijabarkan pada Tabel 2.4.

Tabel 2.4. Perbandingan RF-DETR dan YOLOv12 pada Kompleksitas Spasial Kebun

Model Arsitektur	Dataset	Akurasi (AP)	Waktu Konvergensi Pelatihan
RF-DETR	Kelas Tunggal	94,64% mAP@50:95	$<$ 10 epoch
YOLOv12N	Kelas Tunggal	76,20% mAP@50:95	$\sim$100 epoch
RF-DETR	Multi Kelas	82,98% mAP@50:95	$\sim$20 epoch
YOLOv12L	Multi Kelas	66,22% mAP@50:95	$\sim$100 epoch

Sumber: [28]

Meskipun mendominasi dalam hal presisi, kelemahan RF-DETR terlihat pada skalabilitas latensi ketika dipaksa beroperasi pada perangkat keras terbatas (*edge deployment*), seperti pada inspeksi termal melalui *Unmanned Aerial Vehicle* (UAV). Tabel 2.5 menunjukkan performa (*speed-accuracy trade-off*) tersebut.

Tabel 2.5. Perbandingan Efisiensi *Speed-Accuracy* RF-DETR dan Varian YOLO pada Perangkat Edge

Model Arsitektur	Dataset	Akurasi (AP)	Kecepatan Inferensi
RF-DETR	UAV Termal	82,6% mAP@0.50	12,6 FPS
YOLOv12s	UAV Termal	77,1% mAP@0.50	37,2 FPS
YOLOv10s	UAV Termal	75,4% mAP@0.50	37,1 FPS

Sumber: [26]

Relevansi DETR dalam Deteksi Kerusakan Jalan

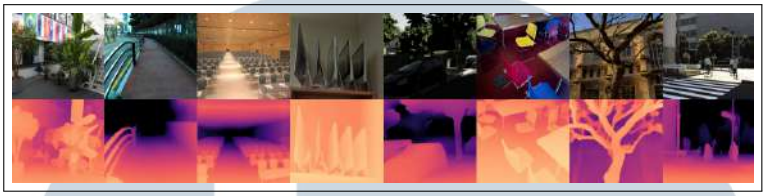
Dalam konteks deteksi kerusakan jalan melalui gambar *Unmanned Aerial Vehicle* (UAV) atau kamera kendaraan, DETR memberikan keunggulan teknis sebagai berikut [20], [12]:

1. Objek Berukuran Kecil: DETR dan variannya (seperti Drone-DETR) sangat efektif dalam menangani tantangan objek berukuran kecil (*small objects*) dan latar belakang yang rumit (*intricate backgrounds*) yang sering ditemui pada gambar permukaan jalan (seperti tekstur aspal yang tidak rata) [20].
2. Penalaran Global: Kemampuan penalaran global dari *transformer* memungkinkan deteksi yang lebih akurat pada kerusakan yang memiliki ketergantungan spasial yang kompleks, misalnya retakan yang memanjang atau pola kerusakan yang saling berhubungan [20].
3. Efisiensi Survei: Penggunaan varian *real-time* seperti RT-DETR memungkinkan pemantauan kondisi jalan yang cepat dan otomatis. Hal ini berpotensi menggantikan survei konvensional secara manual yang memakan waktu dan biaya tinggi [12].

2.1.4 MiDaS

MiDaS merupakan kerangka kerja estimasi kedalaman (*monocular depth estimation*) yang dirancang untuk menghasilkan *depth map* yang *robust* dari gambar melalui pendekatan *zero-shot cross-dataset transfer*. Model ini mengatasi keterbatasan generalisasi pada metode konvensional dengan melatih arsitektur pada gabungan berbagai dataset skala besar menggunakan fungsi kerugian (*loss function*) yang bersifat *scale-and-shift-invariant*. Dengan mengintegrasikan mekanisme *transformer-based backbone* pada varian terbarunya, MiDaS mampu menangkap hubungan spasial global dan detail tekstur permukaan jalan secara presisi, yang kemudian digunakan sebagai parameter utama dalam menentukan tingkat keparahan kerusakan berdasarkan dimensi kedalaman [29].

Konsep Dasar Monocular Depth Estimation

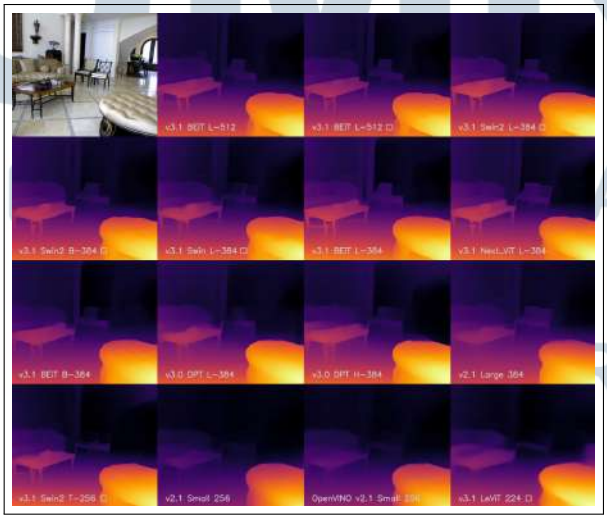


Gambar 2.4. Gambar terproses

Monocular depth estimation merujuk pada tugas meregresikan kedalaman padat (*dense depth*) secara eksklusif hanya dari sebuah gambar tunggal atau satu pandangan kamera. Kemampuan menyelesaikan masalah ini membuka peluang bagi berbagai aplikasi *downstream* esensial, seperti kecerdasan buatan generatif, rekonstruksi 3D, hingga sistem persepsi lingkungan untuk kendaraan otonom. Namun, proses mendeduksi informasi kedalaman pada piksel individu melalui satu gambar merupakan permasalahan yang sangat menantang dan bersifat *under-constrained.* Tantangan inheren utama pada masalah yang tergolong *ill-posed* ini adalah ambiguitas skala metrik (*metric scale ambiguity*) [30], [31].

Pendekatan *learning-based methods* berupaya memecahkan permasalahan ini dengan dua arah yang berbeda. Strategi pertama berupaya meregresikan kedalaman metrik secara langsung melalui pelatihan yang diawasi (*supervised training*), namun hal ini seringkali berujung pada permasalahan *overfitting* terhadap rentang kedalaman yang sempit dan mendegradasi generalisasinya di berbagai lingkungan. Sebagai alternatif yang lebih kokoh, pendekatan *relative depth estimation* (RDE) berupaya memprediksi kedalaman berbasis piksel yang akurat secara relatif antara satu piksel dengan piksel lainnya tanpa membawa makna metrik yang absolut [32].

Arsitektur dan Mekanisme MiDaS



Gambar 2.5. Gambar Tipe-tipe MiDaS

MiDaS merupakan salah satu arsitektur mutakhir dalam RDE yang memanfaatkan struktur jaringan *encoder-decoder* konvensional, di mana bagian *encoder* diadaptasi dari jaringan klasifikasi gambar. Pada versi awal perkembangannya (MiDaS v1.0 dan v2.0), model mengandalkan *encoder* konvolusional seperti keluarga arsitektur ResNet. Mengikuti perkembangan pesat dalam bidang *computer vision*, rilis MiDaS v3.0 memperkenalkan eksplorasi *backbone* berbasis *vision transformers* dasar (ViT). Pada iterasi terbaru, MiDaS v3.1 menawarkan variasi koleksi model skala besar yang mengintegrasikan *state-of-the-art transformer backbones* seperti BEiT, Swin, SwinV2, Next-ViT, dan LeViT. Beragamnya ketersediaan model ini menghadirkan keseimbangan (*trade-offs*) yang spesifik antara optimalisasi kualitas estimasi kedalaman dan kebutuhan waktu komputasi atau *framerate* [33].

Kunci keberhasilan performa *zero-shot* lintas-lingkungan pada MiDaS bersumber pada strategi pencampuran data (*dataset mixing*) saat fase pelatihan. Pelatihan dilakukan menggunakan ragam dataset heterogen yang memiliki skala kedalaman metrik dan parameter kamera yang bervariasi. Alih-alih memprediksi kedalaman ruang secara literal, regresi kedalaman pada MiDaS dilakukan pada ruang disparitas (*disparity space*), yaitu sebuah representasi inversi kedalaman yang dihitung hingga batas parameter skala dan pergeseran tertentu. Mekanisme pelatihan ini mengandalkan *scale-and-shift-invariant losses* untuk menangani ambiguitas bawaan yang ada pada label *ground truth* [34], [35].

Relevansi MiDaS dengan Estimasi Keparahan Kerusakan Jalan

Keluaran murni dari arsitektur MiDaS berbentuk *inverse relative depth map* (peta inversi kedalaman relatif). Secara visual, sistem ini merepresentasikan topografi kedalaman dalam bentuk gradasi piksel, di mana warna yang lebih terang mencerminkan nilai inversi kedalaman relatif yang lebih besar, yang secara teknis mengindikasikan bahwa objek tersebut berada pada jarak yang lebih dekat dengan proyeksi kamera [36], [37].

Pada aplikasi rekayasa infrastruktur, geometri fisis permukaan jalan normal memiliki nilai kedalaman relatif yang konsisten. Anomali topografi berbentuk kerusakan (seperti lubang atau *pothole* dan retakan dalam) secara inheren memiliki profil fisik yang menjauh ke bawah permukaan jalan. Berdasarkan logika ini, area jalan yang mengalami keausan atau berlubang akan diinterpretasikan oleh *depth map* sebagai regio yang memiliki rentang jarak proksimitas lebih jauh ke lensa kamera jika dibandingkan dengan area aspal di sekitarnya [32]. Gradien diferensial kedalaman piksel ini dapat dikuantifikasi guna memperoleh metrik volume anomali.

Meskipun model keluarga MiDaS mampu secara superior memberikan gambaran kedalaman relatif dengan ketahanan komputasi tinggi, model ini mempertahankan faktor skala dan pergeseran yang tidak diketahui identitasnya. Jika penelitian membutuhkan konversi absolut dari keparahan (*severity*) ke dimensi metrik nyata (misal: *centimeter*), informasi non-metrik luaran MiDaS tidak mencukupi untuk berdiri sendiri. Estimasi tersebut memerlukan pensejajaran (*alignment*) skala global yang digabungkan bersama teknik penyempurnaan sekunder (*fine-tuning*) komplementer, misalnya memanfaatkan modul *metric depth binning* atau integrasi sampel metrik yang jarang (*sparse metric depth reference*) [38].

Sinergi dengan Object Detection

Metode regresi berbasis piksel yang masif seringkali dibatasi oleh analisis area yang tidak terfokus (*unconstrained background processing*). Arsitektur seperti *Detection Transformer* (DETR) berperan penting untuk mendeterminasi letak kerusakan secara lokalisatoris di atas gambar RGB [38]. Dalam praktiknya, prediksi *bounding box* dari model detektor berfungsi strategis sebagai *masking operator*. Peta batas ini memproyeksikan *region* fokus murni di atas *depth map* yang diregresi oleh MiDaS. Sinergitas ini memastikan komputasi diferensial keparahan lubang/retakan hanya menargetkan piksel-piksel valid terkurung, sehingga secara komputasional mengeliminasi distraksi bias latar yang dihasilkan oleh objek tepi jalan, kendaraan eksogen, atau gangguan visual lainnya [32].

2.1.5 Pinhole Camera Model

Pinhole Camera Model merupakan model geometris dasar dalam bidang *computer vision* yang digunakan untuk merepresentasikan proses pembentukan gambar dari dunia tiga dimensi (3D) ke bidang dua dimensi (2D). Model ini mengasumsikan bahwa cahaya dari objek masuk melalui sebuah titik kecil (*pinhole*) dan diproyeksikan ke bidang gambar tanpa mempertimbangkan distorsi lensa [39, 40].

Pada model ini, setiap titik pada dunia nyata diproyeksikan ke bidang gambar melalui garis lurus yang melewati pusat proyeksi. Model *pinhole* menjadi dasar dalam berbagai aplikasi seperti rekonstruksi 3D, pengukuran jarak, dan kalibrasi kamera [41].

Model Matematis

Hubungan antara titik pada koordinat dunia dan koordinat gambar dapat dinyatakan dalam bentuk persamaan berikut:

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{K} \begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{2.3}$$

- (X, Y, Z) merupakan koordinat titik pada sistem dunia (3D)
- (u, v) merupakan koordinat piksel pada gambar (2D)
- s adalah faktor skala
- **K** adalah matriks intrinsik kamera
- **R** adalah matriks rotasi
- **t** adalah vektor translasi

Matriks intrinsik kamera **K** dinyatakan sebagai:

$$\mathbf{K} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{2.4}$$

- f_x dan f_y adalah panjang fokus kamera dalam satuan piksel
- (c_x, c_y) adalah titik pusat gambar (*principal point*)

Formulasi ini merupakan bentuk umum dari model kamera *pinhole* yang menggunakan koordinat homogen untuk merepresentasikan transformasi perspektif [42, 43].

Proyeksi Perspektif

Dalam model *pinhole* camera, proyeksi yang terjadi adalah proyeksi perspektif. Hubungan sederhana antara koordinat 3D dan koordinat gambar dapat dituliskan sebagai:

$$u = \frac{fX}{Z}, \quad v = \frac{fY}{Z} \tag{2.5}$$

Persamaan ini menunjukkan bahwa nilai kedalaman (Z) mempengaruhi posisi dan ukuran objek pada gambar. Semakin besar nilai Z, maka objek akan tampak semakin kecil pada bidang gambar [44, 45].

Karakteristik Model

Model *pinhole* camera memiliki beberapa karakteristik utama, yaitu:

- Tidak mempertimbangkan distorsi lensa
- Menggunakan satu titik pusat proyeksi
- Menghasilkan hubungan linier dalam koordinat homogen
- Bersifat ideal dan digunakan sebagai pendekatan dalam sistem nyata

Meskipun sederhana, model ini tetap menjadi dasar utama dalam pemodelan kamera pada berbagai penelitian modern di bidang *computer vision* [41].

2.2 Tools

2.2.1 Python

Python merupakan bahasa pemrograman tingkat tinggi yang telah menjadi standar dalam pengembangan sistem kecerdasan buatan dan *computer vision*. Dalam penelitian ini, Python berfungsi sebagai jembatan integrasi antara berbagai pustaka spesialisasi:

PyTorch: Merupakan kerangka kerja utama untuk membangun mekanisme *self-attention* pada arsitektur DETR. Kemampuan Python dalam menangani *dynamic computational graphs* memungkinkan proses *fine-tuning* pada model MiDaS menjadi lebih fleksibel [46].

OpenCV & NumPy: Berperan dalam pengolahan gambar digital sebelum masuk ke tahap inferensi. Python memfasilitasi transformasi gambar RGB menjadi representasi matriks yang dapat diolah untuk mengekstraksi fitur geometri kerusakan jalan [47].

2.2.2 Visual Studio Code

Visual Studio Code (VS Code) merupakan sebuah perangkat lunak penyunting kode sumber (*source code editor*) lintas platform yang dikembangkan oleh Microsoft [48]. Perangkat ini dirancang dengan arsitektur yang ringan namun memiliki performa tinggi, serta dilengkapi dengan fitur bawaan seperti *debugging*, kendali versi Git, dan penyelesaian kode otomatis (*intellisense*). Dalam implementasi sistem kecerdasan buatan, VS Code menjadi lingkungan pengembangan terpadu yang sangat populer karena mendukung integrasi ekstensi Python secara penuh untuk mengelola modul-modul *deep learning* [49].

2.2.3 Roboflow

Roboflow merupakan sebuah platform *computer vision lifecycle management* berbasis awan yang dirancang untuk menyederhanakan proses persiapan dataset sebelum fase pelatihan model [50]. Platform ini menyediakan alat untuk melakukan anotasi gambar, manajemen label, serta penerapan teknik pra-pemrosesan dan augmentasi data secara efisien. Dalam penelitian ini, *Roboflow* digunakan sebagai infrastruktur utama untuk melakukan *forking* data, pengorganisasian kelas kerusakan jalan, serta penanganan isu *missing annotation* guna meningkatkan performa model arsitektur DETR [51].

2.2.4 Streamlit

Streamlit merupakan sebuah *framework* berbasis open-source yang digunakan untuk membangun aplikasi web interaktif berbasis data menggunakan bahasa pemrograman Python secara cepat [52]. Framework ini mengonversi skrip Python standar menjadi komponen antarmuka pengguna tanpa memerlukan keahlian mendalam di bidang *front-end development* seperti HTML, CSS, atau JavaScript. *Streamlit* diterapkan pada penelitian ini untuk mengimplementasikan demo antarmuka sistem deteksi kerusakan jalan, pembuatan panel *slider* nilai *confidence threshold*, serta visualisasi tabel ringkasan kalkulasi fisik dan logistik biaya perbaikan [53].

2.2.5 Overleaf

Overleaf merupakan sebuah perangkat lunak berbasis awan (*cloud-based platform*) yang berfungsi sebagai editor LaTeX kolaboratif untuk menyusun, mengedit, dan menerbitkan dokumen akademis secara daring [54]. Platform ini mengeliminasi kebutuhan instalasi distribusi LaTeX lokal (seperti TeX Live atau MiKTeX) dengan menyediakan mesin kompilasi langsung pada

server bawaannya. Dalam ranah riset teknologi, *Overleaf* banyak diadopsi karena mempermudah standardisasi format penulisan ilmiah (seperti format IEEE), manajemen sitasi melalui integrasi file bibliografi, serta memfasilitasi kerja sama penulisan antara mahasiswa dan dosen pembimbing secara *real-time* [55].

2.3 Penelitian Terdahulu

Tabel 2.6. Ringkasan Penelitian Terdahulu

No	Peneliti & Tahun	Judul Penelitian	Metode / Model	Kontribusi Utama	Relevansi
1	Yian Zhao, dkk. (2024)	*DETRs Beat YOLOs on Real-time Object Detection*	RT-DETR	Optimasi DETR untuk kecepatan *real-time* dengan *Efficient Hybrid Encoder*.	Referensi efisiensi komputasi untuk deteksi cepat. [13]
2	Cuihua Zuo, dkk. (2025)	*Pavement-DETR: A High-Precision Real-Time Detection Transformer for Pavement Defect Detection*	Pavement-DETR	Modifikasi RT-DETR khusus untuk perkerasan jalan menggunakan *Powerful-IoU* dan *SPD-Conv*.	Validasi efektivitas DETR untuk kerusakan jalan. [15]
3	Shuo Wang, dkk. (2025)	*RT-DETRv3: Real-time End-to-End Object Detection with Hierarchical Dense Positive Supervision*	RT-DETRv3	Penggunaan *hierarchical dense positive supervision* untuk mempercepat konvergensi.	Optimasi akurasi deteksi tipe kerusakan. [19]
4	Shihua Huang, dkk. (2025)	*DEIM: DETR with Improved Matching for Fast Convergence*	DEIM-D-FINE	Fokus pada percepatan waktu pelatihan dan akurasi tinggi yang melampaui YOLO.	Referensi optimasi waktu *training* model. [20]

5	Yaning Kong, dkk. (2024)	*Drone-DETR: Efficient Small Object Detection for Remote Sensing Image Using Enhanced RT-DETR Model*	Drone-DETR	Peningkatan RT-DETR untuk objek kecil pada gambar jarak jauh (UAV).	Referensi jika data menggunakan sudut pandang *drone*. [23]

Sumber: [26]

Berdasarkan pemaparan tabel penelitian terdahulu di atas, terlihat adanya perkembangan yang signifikan dalam pemanfaatan *computer vision* untuk pemeliharaan infrastruktur jalan. Meskipun demikian, sebagian besar penelitian sebelumnya masih berfokus secara terpisah pada aspek klasifikasi tipe kerusakan atau estimasi dimensi objek saja. Oleh karena itu, penelitian ini hadir untuk mengisi celah tersebut dengan mengintegrasikan model *Detection Transformer* (DETR) dan MiDaS guna menghasilkan sistem yang mampu mendeteksi tipe sekaligus mengestimasi tingkat keparahan jalan secara sekuensial.