

## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Penelitian Terdahulu

Beberapa penelitian sebelumnya telah berkontribusi signifikan dalam pengembangan sistem deteksi kerusakan pada bodi mobil, terutama dengan penerapan Convolutional Neural Network (CNN) dan metode berbasis visi. Teknologi ini telah memberikan solusi inovatif untuk mengidentifikasi berbagai jenis kerusakan, seperti penyok, goresan, dan retak, dengan memanfaatkan algoritme canggih untuk menganalisis gambar secara otomatis.

Lebih jauh lagi, beberapa penelitian telah mengeksplorasi integrasi teknologi berbasis visi dengan teknik machine learning lainnya, seperti Support Vector Machines (SVM) dan Random Forest, untuk memberikan analisis yang lebih mendalam dan prediksi yang lebih andal. Kemajuan dalam bidang ini tidak hanya mendukung industri otomotif dalam mengoptimalkan proses inspeksi, tetapi juga memberikan kontribusi pada pengurangan biaya operasional dan peningkatan kualitas layanan. Berikut adalah beberapa penelitian sejenis beserta temuan dan hasilnya:

**Tabel 2. 1** Tabel Penelitian Terdahulu

No	Peneliti (Tahun)	Judul Penelitian	Isi Penelitian	Temuan	Kelebihan	Kekurangan
1	Feng et al. (2021)	Automated Vehicle Damage Detection Using Deep Learning	Sistem deteksi kerusakan kendaraan menggunakan CNN dan Faster R-CNN untuk otomatisasi evaluasi kerusakan kendaraan setelah kecelakaan.	Akurasi deteksi 92%, pengurangan waktu inspeksi manual, namun kecepatan pemrosesan masih dapat ditingkatkan dengan algoritma lebih ringan seperti YOLO.	Akurasi tinggi, mampu mendeteksi area kerusakan secara detail.	Kecepatan pemrosesan rendah dibandingkan YOLO.
2	Huang et al. (2020)	Car Damage Detection Using	Penggunaan arsitektur YOLO untuk mendeteksi	Akurasi 89%, waktu pemrosesan sangat cepat	Kecepatan tinggi, cocok untuk	Akurasi lebih rendah dibandingkan

		YOLO in Real-Time	kerusakan bodi mobil secara real-time. Dataset mencakup berbagai gambar kerusakan mobil.	(0,045 detik per gambar).	aplikasi real-time.	Faster R-CNN.
3	Lee & Kim (2022)	Vehicle Damage Recognition Using YOLO Architecture for Efficient Claim Processing	Sistem untuk mempercepat klaim kerusakan kendaraan dengan deteksi YOLO, mengklasifikasi kerusakan ke dalam kategori ringan, sedang, dan berat.	Akurasi 90%, mempercepat proses klaim asuransi, dapat diintegrasikan dengan aplikasi mobile.	Efisien untuk klaim asuransi, kompatibel dengan aplikasi mobile.	Tidak membahas deteksi kerusakan secara detail seperti model lain.
4	Wang et al. (2023)	The Role of Artificial Intelligence in Smart Manufacturing for Automotive Industry	Tinjauan literatur peran AI dalam smart manufacturing, termasuk implementasi YOLO untuk deteksi kerusakan kendaraan selama produksi dan inspeksi akhir.	YOLO dapat mengurangi kesalahan manusia, meningkatkan efisiensi produksi, keberhasilan deteksi cacat kecil sekitar 87%.	Memberikan tinjauan luas tentang implementasi AI dalam industri otomotif.	Tidak fokus secara mendalam pada satu metode atau implementasi YOLO tertentu.
5	Zhang et al. (2019)	Object Detection and Classification for Vehicle Insurance Using Deep Learning	Eksplorasi algoritma YOLO, SSD, dan Faster R-CNN untuk klaim asuransi kendaraan, menggunakan dataset dengan berbagai kondisi pencahayaan dan sudut pandang kendaraan yang berbeda.	YOLO adalah algoritma tercepat dengan akurasi 88%, efisien untuk aplikasi mobile dan proses klaim digital.	Memvalidasi kecepatan YOLO, cocok untuk implementasi mobile.	Akurasi lebih rendah dibandingkan Faster R-CNN. Tidak membahas peningkatan performa secara mendalam.

## 2.2 *Computer Vision*

*Computer Vision* merupakan salah satu dari *Artificial Intelligence* yang memiliki fokus pada perkembangan dari teknologi dan algoritma pada komputer agar dapat melihat, memproses, memahami dan menganalisis data visual berupa gambar dan video untuk dapat mengetahui dan memproses informasi yang diterima berupa tiga dimensi seperti yang dilakukan oleh manusia. *Computer Vision* adalah bidang yang dominan untuk digunakan pada penelitian dan serbaguna di era saat ini. *Computer Vision* memberikan intruksi pada komputer untuk dapat memahami dan menganalisis dengan pemahaman tingkat tinggi mengenai konten visual. Sub bidang dari *computer vision* meliputi pengenalan pada adegan atau objek, deteksi objek, pelacakan video, segmentasi objek, pose dan estimasi gerakan, pemodelan pemandangan, dan restorasi gambar (Morris dalam Diwan et al., 2023).

*Computer vision* memiliki kemampuan untuk dapat mengembangkan sistem secara otomatis dengan melihat, mengenali, dan menganalisis gambar atau video. Tujuan utama yang dimiliki oleh *computer vision* adalah agar komputer dapat melihat dan memahami dunia nyata yang berupa visual layaknya manusia, dengan menggunakan algoritma dan teknik dalam melakukan proses gambar agar dapat dianalisis dan mendapatkan informasi yang berguna (Sutisna, 2024). Mempelajari metode agar mendapatkan informasi numerik maupun simbolik merupakan bidang ilmu dari *Computer Vision*. Proses-proses yang dapat dilakukan oleh *computer vision* sangat beragam, seperti dengan melakukan penangkapan citra, peningkatan citra, segmentasi, ekstraksi pada fitur, serta pada melakukan klasifikasi pada gambar atau video. Dengan sebagian besar tugas dari *computer vision* itu sendiri adalah mendapatkan informasi mengenai peristiwa atau suatu deskripsi yang berasal dari gambar digital dan juga ekstraksi pada fitur. Dalam penggunaannya, *computer vision* dapat memecahkan masalah mengenai informasi yang akan dianalisisnya sesuai dengan domain aplikasi dan sifat dari data yang akan dianalisis (Wiley & Lucas, 2018).

## 2.3 Deep Learning

*Deep Learning* atau biasa disebut dengan istilah Pembelajaran Struktural Mendalam (*Deep Structured Learning*) atau Pembelajaran Hierarki (*Hierarchical learning*) merupakan salah satu cabang dari ilmu *Machine Learning*, yaitu algoritma pemodelan abstraksi pada tingkat tinggi yang ada pada data dengan menggunakan berbagai macam fungsi dari transformasi *non-linear* yang ditata secara berlapis-lapis dan mendalam. Teknik dan algoritma dalam *deep learning* dapat digunakan dalam memenuhi kebutuhan pembelajaran terarah (*supervised learning*), pembelajaran takterarah (*unsupervised learning*), dan semi-terarah (*semi-supervised learning*) dalam berbagai aplikasi seperti pengenalan citra, pengenalan suara, klasifikasi teks, dan sebagainya. *Deep Learning* juga disebut sebagai *Deep* karena struktur dan jumlah jaringan saraf yang ada pada algoritmanya sangat banyak hingga dapat mencapai hingga ratusan lapisan (Lorentius et al., 2019).

Pendeteksian objek dapat dilakukan salah satunya adalah dengan menggunakan metode *deep learning*. Penggunaan dari *deep learning* didasarkan oleh beberapa faktor, contohnya adalah karena penggunaan data *training* dengan jumlah yang cukup banyak untuk dapat dipelajari dan dianalisis karakteristik dari data *training* tersebut, sehingga komputer memerlukan performa yang cukup tinggi agar data *training* dapat berkembang dengan signifikan (Zhao et al., 2019). Konsep dari algoritma *deep learning* terinspirasi oleh otak manusia (Ertam & Aydın, 2017), sama seperti manusia yang berulang kali belajar dari pengalaman, *deep learning* juga akan melakukan hal yang serupa, yaitu dengan melakukan pengulangan secara terus menerus dalam melakukan tugasnya agar dapat memberikan hasil akhir dengan tingkat akurasi yang tinggi dalam melakukan pendeteksian suatu objek, melakukan pengenalan suara, menerjemahkan bahasa, dan lain sebagainya (LeCun et al., 2015).

### 2.3.1 Convolutional Neural Network (CNN)

*Convolutional Neural Network* (CNN) merupakan salah satu bagian dari metode *deep learning* yang biasanya digunakan untuk melakukan pengolahan data yang berupa gambar atau video. CNN memiliki beberapa *layer*, seperti *input layer*, *output layer*, dan *layer* tersembunyi yang terdiri dari *convolution layer* yang

berfungsi untuk mengekstraksi fitur dengan mengatur ukuran dari kernel, pergeseran setiap kernel, dan menambahkan *padding* pada *input* supaya ukuran dari *output* sama dengan ukuran *input*. *Pooling layer* yang berfungsi untuk dapat mengurangi dimensi agar *layer* berikutnya tidak memerlukan komputasi yang besar dan *fully connected layer* yang berfungsi untuk melakukan klasifikasi berdasarkan fitur yang telah diekstrak (Narayan & Muthalagu, 2021). CNN dibuat dengan menggunakan cara *Multi Layer Perceptron* dan banyak digunakan dalam pembuatan aplikasi *computer vision* seperti aplikasi untuk dapat melakukan klasifikasi pada citra, mendeteksi objek, dan melakukan pengenalan pada wajah manusia (Radikto, 2022).

Metode CNN dianggap menghasilkan data yang paling signifikan pada pengenalan citra digital, karena CNN memang diimplementasikan berdasarkan sistem pengenalan pada citra pada *visual cortex* manusia (Bili et al., 2022). Dengan memanfaatkan jaringan dari syaraf tiruan, CNN dapat mempelajari data-data yang telah tersedia dengan bantuan *deep learning*. CNN juga dapat diartikan sebagai sebuah matriks yang memiliki sebuah fungsi untuk dapat melakukan penyaringan pada suatu gambar (Lambacing, et al. 2020). Dengan beberapa lapisan yang digunakan untuk menyaring gambar dari setiap mekanismenya, hal tersebut disebut dengan *training*. Citra pada gambar yang telah disaring akan dipecah menjadi dua kategori, yaitu *sampling* yang merupakan proses pengambilan nilai pecahan atau koordinat ruang, dan kuantitasi yang merupakan mekanisme dalam melakukan *clustering* dengan nilai tahapan citra kontinu dengan dikategorikan menjadi beberapa *level* (Hermawan, 2021; Putra, 2016).

### 2.3.2 You Only Look Once (YOLO)

YOLO (*You Only Look Once*) merupakan bentuk algoritma dari *deep learning* untuk dapat melakukan deteksi pada objek secara *real time* dengan memanfaatkan algoritma dari *Convolutional Neural Network* (CNN) agar dapat memproses data yang berupa data visual. YOLO dapat mendeteksi objek dengan menerima sebuah *input* berupa gambar, kemudian akan dibagi menjadi kotak dengan ukuran  $S \times S$  untuk membuat *bounding box*, dan sistem akan memprediksi probabilitas kelas berdasarkan dengan tingkat keyakinan sistem dalam mendeteksi sebuah objek (Diwan et al., 2023). Sistem deteksi yang dilakukan oleh YOLO yaitu



dengan menggunakan *repurse classifier* atau *localizer* agar dapat melakukan pendeteksian. Dalam YOLO, citra yang diterapkan melalui berbagai model kemudian menghasilkan *score* atau nilai yang tinggi akan dianggap sebagai sebuah acuan pendeteksian (Rahma, et al. 2021). Cara kerja dari YOLO adalah membagi citra yang berasal dari gambar atau *frame* video menjadi beberapa *grid* kecil, selanjutnya YOLO akan melakukan prediksi pada kotak pembatas *grid* (*Bounding Box*) dan kelas objek bagi setiap *grid*. Setiap kotak pembatas akan memunculkan skor hasil dari deteksi YOLO yang berdasarkan keyakinan YOLO dalam membaca bahwa objek yang dideteksi benar-benar ada atau tidak, kemudian seluruh kotak pembatas *grid* akan digabungkan dan YOLO akan menyaring skor yang rendah (Gerald & Lubis, n.d.).

YOLO (*You Only Look Once*) digunakan serta dikembangkan untuk dapat meningkatkan dan mempercepat efisiensi dari deteksi objek secara *realtime*, hal ini juga yang menjadi unggulan dari penggunaan YOLO. Dengan menggunakan pendekatan deteksi pada objek berbasis *grid* menggunakan konvolusi jaringan syaraf tunggal (*single-shot*), maka akan memungkinkan YOLO untuk melakukan deteksi pada objek secara langsung pada citra dengan kecepatan yang tinggi. Dalam membagi citra, YOLO akan membaginya dengan menjadikan *grid* sel yang tumpang tindih dan setiap sel akan bertanggung jawab untuk melakukan prediksi pada setiap kotak pembatas objek serta kelas objek yang ada di dalamnya. Sehingga hal tersebut dapat memungkinkan YOLO agar dapat melakukan pendeteksian objek secara paralel serta mempercepat proses deteksi yang terjadi (Nafis Alfarizi et al., 2023).

## **2.4 Kerusakan pada Bodi Mobil**

Deteksi kerusakan pada bodi mobil merupakan aspek yang sangat penting dalam berbagai industri, termasuk asuransi, perbaikan otomotif, dan penjualan kendaraan bekas. Kerusakan pada bodi mobil bisa disebabkan oleh berbagai faktor, seperti kecelakaan, benturan, kondisi cuaca ekstrem, atau proses penggunaan jangka panjang. Kerusakan ini dapat beragam, mulai dari goresan kecil hingga penyok atau retakan besar yang mempengaruhi integritas struktural mobil.

Penentuan tingkat kerusakan bodi mobil secara akurat adalah langkah awal dalam proses evaluasi yang dapat memengaruhi keputusan perbaikan atau penggantian, klaim asuransi, dan penilaian harga jual.

Tradisionalnya, deteksi kerusakan pada bodi mobil dilakukan secara manual oleh teknisi ahli yang memeriksa kondisi fisik mobil secara visual. Proses ini, meskipun efektif pada beberapa kasus, cenderung subyektif dan rentan terhadap kesalahan manusia. Variasi pengalaman dan keterampilan teknisi dapat menyebabkan perbedaan dalam evaluasi kerusakan yang sama. Selain itu, inspeksi manual memakan waktu dan biaya yang tidak sedikit, terutama jika kendaraan harus dibawa ke bengkel atau pusat inspeksi khusus untuk pemeriksaan lebih mendalam. Menurut Feng et al. (2021), metode manual sering kali tidak efisien dan dapat menghasilkan ketidakakuratan dalam klasifikasi kerusakan kendaraan, terutama untuk kerusakan kecil yang bisa luput dari pengamatan (Feng et al., 2021). Sebagai alternatif, deteksi kerusakan berbasis teknologi *computer vision* telah diperkenalkan untuk meningkatkan efisiensi dan akurasi dalam evaluasi kendaraan. Teknologi ini memungkinkan deteksi otomatis kerusakan pada bodi mobil dengan memanfaatkan gambar atau video yang diambil dari berbagai sudut. *Computer vision*, terutama yang digabungkan dengan *deep learning*, memungkinkan model untuk "belajar" mengenali pola kerusakan dari dataset gambar yang telah dianotasi sebelumnya.

Algoritma ini mampu mengenali berbagai jenis kerusakan secara otomatis tanpa campur tangan manusia, yang mengurangi ketergantungan pada keahlian individu teknisi. Teknologi *deep learning* seperti YOLO (You Only Look Once) memainkan peran kunci dalam pengembangan deteksi kerusakan otomatis ini. YOLO adalah algoritma deteksi objek yang sangat cepat dan efisien, yang memungkinkan deteksi objek dalam gambar dengan satu kali pemrosesan. Algoritma ini secara khusus unggul dalam aplikasi *real-time*, sehingga sangat cocok untuk diterapkan dalam sistem otomatisasi deteksi kerusakan kendaraan.

Penelitian oleh Huang et al. (2020) menunjukkan bahwa penerapan YOLO dalam deteksi kerusakan kendaraan dapat mencapai akurasi yang tinggi. Penelitian ini juga menemukan bahwa YOLO mampu memproses gambar kerusakan kendaraan dengan sangat cepat, yang menjadikannya ideal untuk digunakan dalam

aplikasi berbasis mobile atau perangkat lunak inspeksi otomatis (Huang et al., 2020). YOLO mendeteksi kerusakan dengan menggunakan *bounding boxes* yang menggambarkan area kerusakan pada gambar kendaraan, serta memberikan klasifikasi mengenai tingkat keparahan kerusakan tersebut. Selain kecepatan dan akurasi, deteksi otomatis berbasis YOLO dapat memberikan manfaat tambahan dalam bentuk klasifikasi kerusakan. Klasifikasi kerusakan ini mencakup pembagian kerusakan ke dalam beberapa kategori berdasarkan tingkat keparahannya.

Klasifikasi yang umum digunakan meliputi kerusakan ringan, kerusakan sedang, dan kerusakan berat. Penggunaan YOLO untuk deteksi kerusakan juga telah diimplementasikan dalam beberapa aplikasi mobile yang dirancang untuk membantu pengguna secara langsung. Aplikasi ini memungkinkan pengguna mengambil gambar mobil mereka menggunakan kamera ponsel, kemudian menggunakan algoritma YOLO untuk mendeteksi kerusakan dan mengklasifikasikannya dalam hitungan detik. Penggunaan aplikasi ini tidak hanya meningkatkan kenyamanan pengguna, tetapi juga mengurangi biaya yang terkait dengan pemeriksaan manual di bengkel atau pusat klaim asuransi. Penelitian oleh Zhang et al. (2019) menunjukkan bahwa aplikasi mobile berbasis YOLO dapat memberikan hasil deteksi yang akurat dengan kecepatan tinggi, menjadikannya solusi praktis untuk kebutuhan sehari-hari pengguna kendaraan (Zhang et al., 2019).

Dengan perkembangan teknologi seperti *computer vision* dan *deep learning*, termasuk algoritma YOLO, deteksi kerusakan pada bodi mobil tidak lagi hanya bergantung pada pemeriksaan manual yang lambat dan subyektif. Pengembangan sistem otomatis yang dapat mendeteksi dan mengklasifikasikan kerusakan bodi mobil dengan akurasi tinggi akan membawa dampak besar dalam berbagai sektor, termasuk perbaikan kendaraan, asuransi, dan penjualan mobil bekas.



## 2.5 Dataset

Dataset memainkan peran penting dalam pengembangan model *deep learning*, terutama untuk tugas deteksi objek seperti yang dilakukan oleh YOLO. Dataset berfungsi sebagai sumber utama data pelatihan yang diperlukan untuk melatih model agar dapat mengenali pola-pola tertentu dan melakukan prediksi yang akurat. Dalam konteks penelitian ini, dataset yang digunakan akan berfokus pada gambar bodi mobil yang mengalami kerusakan, dengan berbagai kategori kerusakan yang mencakup kerusakan ringan, sedang, dan berat.

### 2.5.1 Dataset dalam Pelatihan Model

Dalam pengembangan model deteksi kerusakan bodi mobil, dataset yang digunakan harus memenuhi beberapa kriteria penting. Pertama, dataset harus mencakup gambar yang mewakili berbagai jenis kerusakan mobil dari berbagai sudut dan dalam kondisi pencahayaan yang berbeda. Hal ini penting karena YOLO menggunakan metode deteksi objek berbasis *bounding box* yang membutuhkan anotasi yang tepat untuk melatih model. Selain itu, YOLO memiliki kemampuan *real-time*, yang memungkinkannya mendeteksi objek dalam satu kali pemrosesan gambar (*single shot*), sehingga dataset yang digunakan harus mengandung keragaman yang cukup untuk memastikan model dapat menangani berbagai skenario di dunia nyata (Lin et al., 2020).

Seperti yang telah ditunjukkan oleh Lin et al. (2020) dalam pengembangan *Microsoft COCO Dataset*, dataset yang baik untuk pelatihan YOLO harus memiliki ribuan gambar dengan objek yang dianotasi secara manual menggunakan *bounding boxes*. Anotasi yang tepat dan akurat sangat penting untuk memastikan model dapat mendeteksi objek secara efektif dalam kondisi dunia nyata (Lin et al., 2020). Dalam konteks ini, dataset khusus yang berisi gambar kerusakan mobil, baik yang bersumber dari koleksi publik atau dihasilkan sendiri, sangat penting untuk memastikan bahwa YOLO dapat mengenali berbagai pola kerusakan.

### **2.5.2 Dataset Khusus untuk Deteksi Kerusakan Mobil**

Sejumlah penelitian sebelumnya telah menggunakan dataset yang relevan untuk mendeteksi kerusakan pada kendaraan, seperti yang dilakukan oleh Huang et al. (2020), yang mengembangkan dataset kerusakan mobil dengan kategori kerusakan ringan, sedang, dan berat. Dalam penelitian tersebut, dataset dikumpulkan dari berbagai sumber seperti gambar kendaraan yang rusak setelah kecelakaan, atau dari pusat inspeksi kendaraan. Dataset ini digunakan untuk melatih model YOLO agar mampu mengidentifikasi kerusakan bodi mobil dan mengklasifikasikan tingkat keparahannya. Penelitian ini menunjukkan bahwa dataset yang terdiri dari gambar-gambar kendaraan rusak dengan anotasi yang jelas memungkinkan YOLO mencapai akurasi deteksi yang tinggi, mencapai lebih dari 89% pada tugas deteksi kerusakan (Huang et al., 2020).

Penelitian lain oleh Zhang et al. (2019) menunjukkan bahwa penggunaan dataset khusus untuk deteksi kerusakan kendaraan, yang terdiri dari berbagai gambar kendaraan yang mengalami kerusakan, sangat membantu dalam meningkatkan performa model deteksi berbasis YOLO. Dataset ini mencakup gambar mobil yang rusak akibat kecelakaan dengan kondisi kerusakan yang berbeda, seperti goresan kecil, penyok, atau kerusakan parah pada bodi. Dataset ini memberikan kontribusi signifikan dalam meningkatkan kemampuan model untuk mengklasifikasikan kerusakan berdasarkan tingkat keparahan. Penelitian tersebut menggunakan dataset dengan ribuan gambar yang dikategorikan ke dalam beberapa kelas kerusakan, yang memungkinkan YOLO melakukan deteksi dengan tingkat akurasi yang memadai (Zhang et al., 2019).

### **2.5.3 Sumber Dataset yang Digunakan**

Beberapa dataset yang umum digunakan dalam penelitian deteksi objek adalah Microsoft COCO dan Pascal VOC. Kedua dataset ini menyediakan ribuan gambar yang dianotasi dengan objek yang jelas, menjadikannya dataset yang baik untuk melatih model deteksi objek seperti YOLO. Namun, karena dataset ini tidak secara khusus mencakup kerusakan kendaraan, peneliti sering kali harus mengumpulkan data tambahan atau menggunakan dataset yang lebih khusus yang

mencakup kerusakan mobil.

Dataset Microsoft COCO (Common Objects in Context) adalah salah satu dataset yang paling umum digunakan dalam pelatihan model deteksi objek. COCO menyediakan anotasi objek dengan *bounding boxes* pada berbagai objek dalam gambar, termasuk kendaraan. Meskipun tidak secara spesifik menangani kerusakan mobil, dataset ini dapat digunakan sebagai dataset dasar untuk melatih model YOLO sebelum menyesuaikan dengan dataset yang lebih spesifik mengenai kerusakan kendaraan. COCO memiliki lebih dari 200 ribu gambar yang dianotasi, dengan lebih dari 80 kategori objek, yang sangat berguna dalam memperluas kemampuan YOLO untuk mengenali berbagai objek dalam kondisi dunia nyata (Lin et al., 2020).

Pascal VOC (Visual Object Classes) adalah dataset lain yang sering digunakan dalam penelitian deteksi objek. Meskipun lebih kecil daripada COCO, dataset ini telah digunakan dalam berbagai studi yang berhubungan dengan deteksi objek dan juga menawarkan gambar yang sudah dianotasi dengan *bounding boxes*. Dataset ini juga relevan dalam penelitian ini karena mencakup anotasi untuk kendaraan, meskipun tidak secara langsung terkait dengan kerusakan bodi mobil. Pascal VOC menyediakan anotasi yang lebih sederhana dibandingkan COCO, tetapi masih berguna untuk melatih model YOLO dalam tugas deteksi objek (Everingham et al., 2015).

## 2.6 Metrik Evaluasi

Untuk mengevaluasi performa sistem deteksi jatuh, beberapa metrik kinerja yang umum digunakan meliputi:

### 1. Akurasi (*Accuracy*)

Akurasi mengukur proporsi total prediksi yang benar dalam keseluruhan dataset. Dalam sistem deteksi jatuh, akurasi mengindikasikan seberapa tepat sistem dalam mengenali insiden jatuh maupun aktivitas non-jatuh. Akurasi yang tinggi menunjukkan bahwa sistem dapat secara konsisten memberikan hasil yang benar, baik dalam mendeteksi jatuh maupun aktivitas harian lainnya.

2. Sensitivitas (*Recall*)

Sensitivitas, atau dikenal juga sebagai *true positive rate*, mengukur kemampuan sistem dalam mendeteksi jatuh dari semua insiden jatuh yang benar-benar terjadi. Sensitivitas tinggi menunjukkan bahwa model mampu mengenali sebagian besar kejadian jatuh dan meminimalkan kesalahan negatif, yaitu insiden jatuh yang tidak terdeteksi (Altman & Bland, 1994).

3. Spesifisitas (*Specificity*)

Spesifisitas mengukur kemampuan sistem dalam mengenali aktivitas normal dan menghindari kesalahan deteksi, atau false positives. Spesifisitas yang tinggi menunjukkan bahwa sistem dapat membedakan dengan baik antara aktivitas normal dan insiden jatuh, yang mengurangi risiko alarm palsu.

4. Presisi (*Precision*)

Presisi mengukur proporsi prediksi positif yang benar-benar jatuh dari total prediksi jatuh. Presisi tinggi berarti model dapat secara akurat mengidentifikasi insiden jatuh tanpa banyak kesalahan deteksi. Metrik ini penting untuk menghindari terlalu banyak alarm palsu yang dapat mengganggu pengguna.

5. F1-Score

F1-Score adalah metrik gabungan dari presisi dan sensitivitas, memberikan gambaran seimbang tentang kinerja model, terutama jika terdapat ketidakseimbangan antara kelas jatuh dan aktivitas non-jatuh.