BAB 2
TINJAUAN PUSTAKA DAN DASAR TEORI

2. 1. Tinjauan Pustaka

Penelitian mengenai klasifikasi cacat pada produk pengecoran logam, yang berbasis pada arsitektur Convolutional Neural Networks (CNN), telah banyak diterapkan, baik dalam bentuk model yang dikembangkan sendiri maupun model *pre-trained* seperti ResNet, Xception, VGG16, dan EfficientNet. Beberapa penelitian mengusulkan pendekatan ensemble *deep learning* guna menggabungkan keunggulan dari berbagai model.

Penelitian oleh Gupta dkk (Gupta dkk., 2023) menggabungkan model EfficientNet B3, ResNet18, ResNet50, dan VGG16 dengan model CNN yang dikembangkan, menggunakan teknik pembobotan rata-rata pada fitur yang diekstraksi dari masing-masing model. Hasilnya, kombinasi ResNet50 dan CNN menunjukkan performa terbaik dengan akurasi 98,18% dan presisi 99,89%. Selain itu, penelitian Bolla dkk (Bolla dkk., 2022) membandingkan model CNN yang mereka kembangkan dengan model pra-latih seperti NasNet, MobileNet, dan ResNet50. Model CNN yang diusulkan terdiri dari lapisan konvolusi, batch normalization, max pooling, dan diakhiri dengan lapisan dense. Hasil evaluasi menunjukkan bahwa model CNN yang dikembangkan sendiri memiliki performa tertinggi, dengan akurasi sebesar 99,44%, mengungguli model pra-latih yang dibandingkan.

Sementara itu, penelitian Hu dkk (Hu dkk., 2023) berfokus pada peningkatan akurasi model melalui augmentasi data, dengan membandingkan model ResNet50, CNN, InceptionV3, dan Xception. Model Xception yang dikombinasikan dengan data augmentasi memberikan performa terbaik, dengan *precision* sebesar 84,56%, *recall* 78,18%, dan F1-score 78,50%. Hal ini menunjukkan bahwa augmentasi data dapat meningkatkan ketahanan model terhadap variasi dalam dataset. Di sisi lain, penelitian Nguyen dkk (Nguyen dkk., 2021) melakukan eksperimen dengan pengambilan gambar dari empat sisi berbeda menggunakan kamera resolusi tinggi (DMK 33GP031). Model CNN yang mereka

kembangkan berhasil mengklasifikasikan cacat produk pengecoran logam dari berbagai sudut dengan akurasi berkisar antara 97,25% hingga 98,25%. Penelitian ini menyoroti pentingnya variasi sudut pengambilan gambar untuk meningkatkan akurasi deteksi cacat.

Rangkuman dari berbagai pendekatan dan hasil penelitian sebelumnya disajikan dalam Tabel 2.1 berikut.

Tabel 2. 1 Rangkuman tinjauan pustaka model klasifikasi cacat pada produk pengecoran logam

No	Peneliti	Judul Artikel, Dataset	Model dan Pencapaian
1	(Gupta dkk., 2023)	*Deep learning* model for defect analysis in industry using casting images. Penelitian ini menggunakan dataset gambar *casting* logam yang mengalami cacat produk dan produk normal. Dataset ini diperoleh melalui website www.kaggle.com	Penelitian oleh Gupta *et al.* (2023) menggunakan metode *ensemble deep learning* dalam mengklasifikasi kualitas *casting* logam. Penelitian ini menggabungkan model pra-latih EfficientNet B3, ResNet18, ResNet50, dan VGG16 dengan model CNN yang diajukan. Model CNN yang diajukan terdiri dari lapisan konvolusi, *Max Pooling*, dan dilanjutkan dengan lapisan *Dense*. Penggabungan model ini dilakukan dengan melakukan pembobotan rata-rata dari fitur yang dihasilkan oleh masing masing kandidat model. Model terbaik yang didapatkan adalah model gabungan ResNet50 dan CNN dengan skor presisi dan akurasi sebesar 99,89% dan 98,18%

2	Bolla dkk. (2023) (Bolla dkk., 2022)	Efficient *Deep learning* Methods For Identification Of Defective Casting Products. Penelitian ini menggunakan dataset gambar *casting* logam yang mengalami cacat produk dan produk normal. Dataset ini diperoleh melalui perusahaan percetakan logam.	Penelitian oleh Bolla dkk. (2023) membandingkan model CNN dari peneliti dengan model pra-latih seperti NasNet, MobileNet, dan ResNet50. Model CNN yang diajukan terdiri dari lapisan konvolusi, *Batch Normalization, Max Pooling,* diakhiri dengna lapisan *dense.* Model terbaik yang didapatkan adalah model CNN dari peneliti dengan nilai akurasi sebesar 99,44%
3	(Hu dkk., 2023)	Casting Product Image Data for Quality Inspection with Xception and Data Augmentation Penelitian ini menggunakan dataset gambar *casting* logam yang mengalami cacat produk dan produk normal. Dataset ini diperoleh melalui perusahaan percetakan logam.	Penelitian oleh Hu dkk. (2023) membandingkan pra-latih ResNet50, CNN, InceptionV3, Xception dengan model pra-latih Xception yang dilakukan proses data augmentasi. Model terbaik yang didapatkan adalah model Xception dengan data augmentasi. Model ini memiliki nilai *Precision*, *Recall*, dan *F1-Score* sebesar 84,56%, 78,18%, dan 78,50%.
4	(Nguyen dkk., 2021)	Inspecting Method for Defective Casting Products with Convolutional Neural Network (CNN). Penelitian ini menggunakan dataset gambar *casting* logam pada empat sisi yang berbeda. yang mengalami cacat produk dan produk normal. Dataset ini diperoleh melalui pengambilan kamera 5M piksel (DMK 33GP031)	Penelitian oleh Nguyen dkk. (2023) menggunakan model CNN dalam mengklasifikasi cacat produk *casting* logam pada empat sisi yang berbeda. Penelitian ini menggunakan model CNN yang terdiri dari lapisan konvolusi, *max pooling*, dan diakhir dengan lapisan *dense* Menggunakan model CNN untuk mengklasifikasi keempat sisi didapatkan nilai akurasi sebesar 98%, 97,5%, 97,25%, dan 98,25%

Penelitian yang meletakkan dasar bagi BiFPN adalah studi oleh Tan (Tan dkk., 2020) yang mengusulkan metode efisien untuk fusi fitur multi-skala dalam deteksi objek berbasis *deep learning*. Salah satu tantangan utama dalam deteksi objek adalah bagaimana mengintegrasikan informasi dari berbagai tingkat resolusi secara optimal. Feature Pyramid Network (FPN) sebelumnya telah memperkenalkan jalur top-down untuk menggabungkan fitur skala besar dengan fitur skala kecil, sementara PANet menambahkan jalur bottom-up untuk mempertajam detail. Namun, pendekatan ini masih mengalami keterbatasan karena tidak semua fitur yang dilewatkan melalui jalur tersebut memiliki bobot informasi yang sama, sehingga beberapa informasi penting dapat tereduksi atau tidak tersampaikan secara maksimal. BiFPN (Bidirectional Feature Pyramid Network) mengatasi masalah ini dengan memperkenalkan koneksi dua arah yang lebih efisien, yang memungkinkan pertukaran informasi antara fitur skala tinggi dan rendah secara lebih seimbang. Selain itu, BiFPN menerapkan mekanisme bobot belajar (weighted feature fusion), yang memungkinkan setiap koneksi diberikan bobot yang dapat disesuaikan berdasarkan relevansi informasinya, sehingga hanya fitur yang benar-benar berguna yang diperkuat dalam proses deteksi. Dengan pendekatan ini, BiFPN tidak hanya meningkatkan akurasi deteksi, tetapi juga mengoptimalkan efisiensi komputasi, menjadikannya lebih unggul dibandingkan metode fusi fitur multi-skala sebelumnya

Selanjutnya, terdapat beberapa penelitian yang mengintegrasikan metode YOLO (You Only Look Once) dengan BiFPN *(Bi-directional Feature Pyramid Network)*. Penelitian kesatu yang diacu dalam tesis ini adalah studi (Li, N. dkk., 2024) mengenai deteksi cacat pada kapasitor berukuran kecil, dengan fokus pada identifikasi cacat skala mikron dalam lingkungan visual yang kompleks. Dalam arsitektur YOLOv8 tanpa BiFPN, model mengolah fitur (yang diperoleh dari backbone) yang masih mengandung informasi global yang dominan misalnya tekstur material. Sehingga detail kecil yang berupa cacat produk tertutupi oleh elemen visual yang lebih besar atau tidak releva. Guna mengatasi keterbatasan ini, model ditingkatkan dengan BiFPN, yang menambahkan jalur koneksi lateral dua arah guna memperbaiki aliran informasi antar level fitur dan mengoptimalkan fusi

fitur dari berbagai ukuran cacat. Dengan pendekatan ini, model yang dikembangkan dalam (Li, N. dkk., 2024) menunjukkan peningkatan kinerja deteksi, dengan mean average *precision* (mAP) sebesar 95,8%, meningkat 9,5% dibandingkan YOLOv8 asli, menjadikannya lebih andal dalam mendeteksi cacat mikro pada kapasitor dengan tingkat presisi yang lebih tinggi.

Penelitian kedua yang diacu dalam tesis ini adalah penelitian (Zheng dkk., 2024), yang mengintegrasikan BiFPN pada YOLOv8 untuk deteksi objek asing pada jalur transmisi listrik. Layang-layang, balon, dan sarang burung merupakan contoh objek eksternal yang sering menjadi penyebab gangguan pada jaringan listrik, sehingga deteksi yang akurat menjadi bagian penting dalam strategi pemeliharaan. Tantangan utama dalam deteksi ini terletak pada variasi skala objek, misalnya pada objek seperti layang-layang dan balon yang memiliki ukuran kecil dengan sedikit detail tekstur, sementara objek lain seperti sarang burung bisa lebih besar dengan struktur yang kompleks. Perbedaan ini semakin diperumit oleh latar belakang lingkungan terbuka, seperti langit atau pepohonan, yang dapat mengaburkan batas objek dan meningkatkan kemungkinan false negative.

Guna mengatasi tantangan tersebut, penelitian pada (Zheng dkk., 2024) menggunakan BiFPN sebagai solusi dalam menjembatani perbedaan skala objek. BiFPN memanfaatkan koneksi horizontal tambahan serta mekanisme fusi fitur berulang, sehingga memungkinkan model mempertahankan informasi dari berbagai variasi ukuran tanpa meningkatkan kompleksitas komputasi. Dengan struktur ini, model dapat menyeimbangkan informasi dari objek kecil maupun besar, meningkatkan akurasi dalam mendeteksi berbagai jenis gangguan di jalur transmisi listrik. Salah satu langkah kunci dalam BiFPN adalah menghilangkan node dengan kontribusi informasi yang rendah, sehingga mengurangi redundansi jaringan dan meningkatkan efisiensi deteksi objek. Melalui pendekatan ini, sistem inspeksi berbasis YOLOv8-BiFPN yang diusulkan pada menunjukkan kinerja yang tinggi, dengan mean average *precision* (mAP) sebesar 95,5% dan *Recall* mencapai 93,4%, menjadikannya lebih andal dalam mendeteksi objek asing pada jalur transmisi listrik.

Penelitian ketiga yang diacu dalam tesis ini adalah penelitian (Xie & Zhao, 2025), yang mengembangkan model inspeksi berbasis YOLOv5 yang diperbaiki dengan pendekatan lightweight untuk deteksi cacat pada papan sirkuit cetak (PCB). Tantangan utama dalam deteksi cacat PCB terletak pada ukuran kecil serta kompleksitas pola sirkuit, yang sering kali menyebabkan kesulitan dalam membedakan antara cacat dan elemen desain PCB. Guna mengatasi permasalahan ini, penelitian (Xie & Zhao, 2025) mengusulkan beberapa perbaikan pada arsitektur YOLOv5, termasuk BiFPN yang memungkinkan fusi fitur yang bersifat multi-skala. Dengan pendekatan ini, model yang dikembangkan oleh Xie (2025) berhasil mencapai mean average *precision* (mAP@0.5) sebesar 94,9%, hanya 0,5% lebih rendah dibandingkan YOLOv5 asli, namun dengan pengurangan 56,2% dalam operasi floating point (GFLOPs) serta 53,7% dalam jumlah parameter. Hasil ini menunjukkan bahwa pendekatan lightweight YOLOv5-BiFPN tidak hanya meningkatkan efisiensi inspeksi PCB, tetapi juga memberikan keseimbangan optimal antara akurasi dan komputasi, sehingga lebih sesuai dengan kebutuhan industri produksi berskala besar.

Penelitian keempat yang diacu dalam tesis ini adalah penelitian (Bai & Xu, 2025), yang mengembangkan model PD-YOLOv8 untuk deteksi cacat pada papan sirkuit cetak (PCB) dengan fokus pada pengenalan objek berukuran kecil, yang menjadi tantangan umum dalam inspeksi PCB. Pendekatan PD-YOLOv8 yang dikembangkan dalam (Bai & Xu, 2025) menggunakan BiFPN untuk fusi lintas-layer, yang memastikan bahwa informasi tekstur dari lapisan bawah dan informasi semantik dari lapisan atas dapat saling melengkapi untuk meningkatkan pengenalan terhadap target kecil. Pada penelitian ini, BiFPN lebih difokuskan untuk meningkatkan deteksi target kecil, terutama dengan memperkuat interaksi fitur antar-lapisan dalam jaringan.

Rangkuman dari berbagai pendekatan dan hasil penelitian sebelumnya disajikan dalam Tabel 2.2 berikut.

Tabel 2. 2 Rangkuman tinjauan pustaka model deteksi cacat pada produk yang menggunakan pendekatan BiFPN

No	Peneliti	Judul Artikel, Dataset	Model dan Pencapaian
1	(Tan dkk., 2020)	EfficientDet: Scalable and Efficient Object Detection. Dataset: COCO	Mengembangkan BiFPN sebagai metode fusi fitur multi-skala yang lebih efisien, menggunakan koneksi dua arah dan weighted feature fusion untuk meningkatkan deteksi objek.
2	(Li, N. dkk., 2024)	Defect Detection in Miniature Capacitors using YOLOv8-BiFPN. Dataset: Micro-Capacitor Surface Defect (MCSD)	Menggunakan YOLOv8 dengan BiFPN untuk deteksi cacat skala mikron pada kapasitor. Meningkatkan mAP sebesar 9,5% dibandingkan YOLOv8 asli, mencapai mAP 95,8%.
3	(Zheng dkk., 2024)	Foreign Object Detection in Power Transmission Lines using BiFPN-enhanced YOLOv8. Dataset: Power Transmission Line Inspection Data	Mengintegrasikan BiFPN pada YOLOv8 untuk deteksi objek asing pada jalur transmisi listrik. Model mencapai mAP 95,5% dan *Recall* 93,4%.
4	(Xie & Zhao, 2025)	A Lightweight YOLOv5-based Model for Defective PCB Detection. Dataset: Industrial PCB Defect Dataset	Mengembangkan model lightweight YOLOv5 dengan dual-head detection dan GSConv, serta BiFPN untuk fusi fitur multi-skala. Model mencapai mAP 94,9%, dengan pengurangan 56,2% dalam GFLOPs dan 53,7% dalam jumlah parameter.
5	(Bai & Xu, 2025)	PD-YOLOv8: A Small-Target Optimized Model for PCB Defect Detection. Dataset: Industrial PCB Defect Dataset	Mengembangkan PD-YOLOv8 dengan ECANet dan SlimNeck serta BiFPN untuk deteksi target kecil pada PCB. Model menunjukkan peningkatan akurasi dalam mendeteksi cacat kecil dibandingkan YOLOv8 asli.

2. 2. Industri Pengecoran Logam

Industri pengecoran logam merupakan salah satu sektor kunci dalam manufaktur yang memegang peran penting dalam menghasilkan berbagai komponen logam untuk berbagai industri, seperti otomotif, konstruksi, mesin, dan energi (MSc dkk., 2024). Proses pengecoran logam melibatkan serangkaian tahapan yang dimulai dari peleburan logam, seperti besi, baja, aluminium, dan tembaga, hingga penuangan logam cair ke dalam cetakan yang telah dirancang sesuai dengan bentuk komponen yang diinginkan. Komponen yang dihasilkan dari proses ini dikenal karena daya tahan dan kekuatannya, sehingga menjadi pilihan utama dalam pembuatan produk yang memerlukan sifat mekanis khusus, seperti ketahanan terhadap tekanan, gesekan, dan deformasi.

Pengecoran logam menawarkan fleksibilitas yang tinggi dalam menghasilkan bentuk-bentuk kompleks yang sulit dicapai melalui metode manufaktur lainnya. Selain itu, proses ini juga memungkinkan produksi massal yang efisien, sehingga dapat memenuhi kebutuhan industri skala besar dengan biaya yang relatif lebih rendah (Suprapto, 2017). Dalam industri pengecoran, kualitas produk sangat dipengaruhi oleh beberapa faktor, seperti desain cetakan, komposisi logam, suhu penuangan, dan proses pendinginan. Oleh karena itu, pengendalian kualitas dan analisis cacat pengecoran menjadi aspek penting untuk memastikan produk yang dihasilkan memenuhi standar yang ditetapkan.

Pada industri kecil dan menengah tantangan yang dihadapi sering kali terkait dengan keterbatasan sumber daya, baik dari segi peralatan, teknologi, maupun tenaga kerja yang terampil. Namun, dengan penerapan metode analisis seperti Six Sigma dan penggunaan alat-alat kontrol kualitas, industri pengecoran dapat mengurangi tingkat penolakan produk dan meningkatkan kualitas secara keseluruhan. Studi kasus yang dilakukan pada industri pengecoran skala kecil menunjukkan bahwa pendekatan sistematis dalam menganalisis cacat pengecoran dan menerapkan solusi yang tepat dapat menghasilkan peningkatan signifikan dalam produktivitas dan kualitas produk (Chelladurai dkk., 2020).

Secara keseluruhan, industri pengecoran logam tidak hanya berperan penting dalam memenuhi kebutuhan komponen logam untuk berbagai sektor

industri, tetapi juga terus berkembang dengan inovasi teknologi dan metode produksi yang lebih efisien (Chelladurai dkk., 2020). Dengan demikian, industri ini tetap menjadi tulang punggung dalam manufaktur modern, terutama dalam menghasilkan komponen-komponen yang memerlukan kekuatan dan ketahanan tinggi.

2.2.1 Klasifikasi Pengecoran Logam

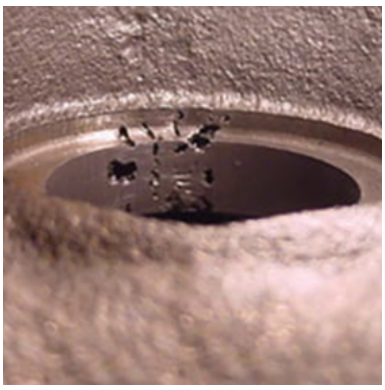
Proses pengecoran logam dibagi menjadi dua klasifikasi utama: *expendable mold casting* dan *permanent mold casting*. Pada *expendable mold casting*, seperti *sand casting*, cetakan hanya bisa digunakan sekali karena harus dihancurkan setelah produk coran diambil, sementara permanent mold casting menggunakan cetakan logam yang dapat dipakai berkali-kali untuk menghasilkan produk serupa (Bhirawa, 2013). Dalam setiap tahapan pengecoran, mulai dari persiapan cetakan, peleburan logam, penuangan, pendinginan, hingga pelepasan cetakan, risiko cacat produk seperti porositas, retak, dan deformasi bisa terjadi.

2.2.2 Cacat Produk Pengecoran Logam

Dalam proses mold casting, terdapat berbagai jenis cacat atau *defect* yang dapat terjadi pada produk coran, yang umumnya disebabkan oleh ketidaksempurnaan dalam proses penuangan, pendinginan, atau bahan cetakan (Pariri dan Buyung, 2022). Berikut merupakan kerusakan yang umum terjadi.

1) Porositas

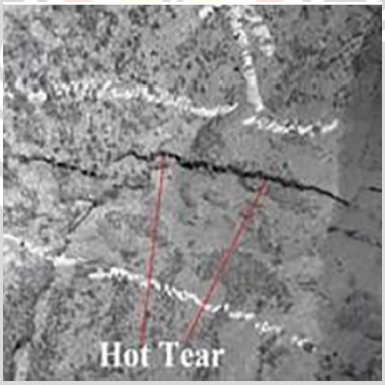
   Kerusakan yang ditandai dengan terbentuknya rongga atau lubang kecil (lihat gambar 2.1) pada produk akibat gas yang terperangkap atau pendinginan yang terlalu cepat.



Gambar 2. 1 Cacat logam porositas

2) Retak panas (*hot tear*)

Ketika logam tidak cukup menyusut selama pendinginan, sehingga menimbulkan tegangan yang menyebabkan retakan (lihat gambar 2.2).



Gambar 2. 2 Cacat logam retak panas

3) *Shrinkage* atau penyusutan

Shrinkage ditandai dengan volume logam berkurang secara tidak merata saat pembekuan, menyebabkan deformasi (lihat gambar 2.3).

Gambar 2. 3 Cacat logam *shrinkage* / penyusutan

4) Inklusi terak

Inklusi terak adalah cacat yang ditandai adanya partikel terak atau kotoran ikut terjebak dalam logam cor, sehingga mempengaruhi kualitas dan kekuatan produk.

Gambar 2. 4 Cacat logam inklusi terak

5) *Misrun*

Misrun dapat terjadi ketika logam cair gagal sepenuhnya mengisi cetakan karena suhu yang terlalu rendah atau desain saluran yang tidak optimal.

Gambar 2. 5 Cacat logam *misrun*

6) *Blowholes*

Blowholes adalah jenis cacat pengecoran yang umum ditemukan pada hasil coran. Terperangkapnya udara akibat penuangan logam cair membentuk kontur bulat atau rongga berbentuk bola.

Gambar 2. 6 Cacat logam bl*owholes*

7) *Defective Surface*

Defective surface merupakan kerusakan yang ditandai dengan Garis-garis bergaris (*streaky lines*). Garis-garis ini yang dihasilkan akibat aliran logam cair, membentuk pola garis yang terlihat seperti serangkaian saluran kecil, dikenal sebagai permukaan cacat. Cacat ini memerlukan perhatian khusus dalam kontrol kualitas untuk memastikan produk coran memenuhi spesifikasi yang diinginkan. Oleh karena itu, kontrol kualitas yang ketat

sangat diperlukan untuk memastikan produk yang dihasilkan memenuhi standar yang diinginkan.

Gambar 2. 7 Cacat logam defective surface

2. 3. Image Processing

Seiring perkembangan teknologi, inovasi dalam industri pengecoran logam semakin berkembang, terutama dalam hal kontrol kualitas menggunakan *image processing*. Teknologi ini memungkinkan otomatisasi deteksi cacat pada produk coran dengan tingkat akurasi yang tinggi, menggantikan metode inspeksi manual yang memakan waktu dan rentan terhadap kesalahan manusia. Sistem pengolahan citra memindai permukaan dan struktur produk coran secara mendetail, mengidentifikasi cacat-cacat kecil yang mungkin tidak terlihat oleh mata manusia. Penggunaan image processing dalam kontrol kualitas ini tidak hanya meningkatkan efisiensi dan kecepatan produksi, tetapi juga memastikan bahwa produk yang dihasilkan memiliki kualitas tinggi dan konsisten.

Secara umum, *image processing* adalah teknologi yang memproses citra digital untuk mengekstrak informasi atau menghasilkan analisis visual yang berguna. Proses ini melibatkan berbagai tahap seperti akuisisi gambar, *preprocessing* (peningkatan kualitas gambar), segmentasi (memisahkan objek dari latar belakang), hingga ekstraksi fitur untuk mengenali pola atau karakteristik tertentu dalam gambar. Image processing digunakan secara luas di berbagai industri, mulai dari pengenalan wajah, diagnosis medis, hingga analisis astronomi. Teknologi ini mampu menganalisis gambar dengan akurasi dan kecepatan tinggi,

memungkinkan keputusan yang lebih cepat dan lebih tepat di berbagai bidang aplikasi.

Dalam industri pengecoran logam, *image processing* digunakan secara khusus untuk otomatisasi quality control, mendeteksi cacat-cacat pada produk coran yang sulit terdeteksi oleh inspeksi manual. Dengan memindai permukaan produk secara otomatis, sistem ini dapat mengidentifikasi kerusakan seperti porositas, retak, atau deformasi dengan sangat cepat dan presisi. Tahap analisis dilakukan dengan membandingkan citra produk hasil pengecoran dengan citra standar atau model ideal yang telah ditentukan. Cacat yang ditemukan kemudian diolah dan dievaluasi dalam waktu nyata, memungkinkan tindakan korektif segera diambil dalam proses produksi. Dengan penerapan *image processing*, industri tidak hanya meningkatkan kualitas produk secara keseluruhan, tetapi juga mengurangi biaya inspeksi dan mempercepat waktu produksi.

2. 4. *Image Preprocessing*

Image Preprocessing atau praproses data gambar merupakan langkah yang dilakukan untuk memproses gambar sebelum masuk ke model CNN yang digunakan. Selain itu, Proses ini bertujuan agar model menghasilkan performa yang baik dalam memprediksi gambar (Pal dan Sudeep, 2016). Proses yang dilakukan pada penelitian ini antara lain mengubah ukuran gambar dan melakukan normalisasi gambar

2.4.1 Mengubah ukuran gambar

Tahapan mengubah ukuran gambar atau *resizing* merupakan proses penting dalam praproses data gambar untuk memastikan dimensi *input* konsisten sesuai dengan persyaratan model CNN. Hal ini diperlukan karena CNN membutuhkan *input* dengan ukuran yang tetap untuk dapat memproses gambar secara efektif. Ukuran gambar diubah menjadi dimensi tetap yang lebih kecil untuk mengurangi kebutuhan komputasi tanpa mengorbankan terlalu banyak informasi penting dari gambar tersebut (Kannojia dan Jaiswal, 2018).

2.4.2 Normalisasi gambar

Normalisasi gambar adalah proses penyesuaian nilai intensitas piksel dalam gambar yang memiliki rentang nilai piksel diantara [0, 255] agar berada dalam rentang tertentu dengan rentang [0, 1]. Proses normalisasi ini bertujuan untuk mengurangi variabilitas antar gambar dan mempercepat proses pelatihan pada model CNN. Melalui normalisasi CNN lebih mudah mengkonvergensi karena mencegah bobot di lapisan awal mengalami pembaruan ekstrem yang dapat menyebabkan *gradient vanishing* atau *exploding* (Pal dan Sudeep, 2016). Rumus yang digunakan untuk normalisasi dengan rentang [0,1] adalah:

$$X_{normalized} = \frac{X}{255} \quad , 0 \leq X \leq 255 \tag{1}$$

Dengan:

X adalah nilai intensitas piksel asli

Secara umum, selain mengubah skala piksel dari [0,1] nilai piksel juga dapat diubah menjadi skala [*min*, *max*] menggunakan persamaan (2):

$$X_{std} = X_{normalized} \cdot (\max + min) + min \tag{2}$$

Dengan:

X_{std} adalah nilai dengan skala [*min*, *max*]

max dan *min* adalah nilai maksimal dan minimal yang dikehendaki

2. 5. Augmentasi Gambar

Augmentasi gambar adalah serangkaian teknik yang digunakan untuk meningkatkan jumlah dan keragaman data pelatihan dalam model *deep learning*. Menerapkan transformasi tertentu pada gambar asli dapat membuat model menjadi lebih *robust* dan memiliki kemampuan generalisasi yang lebih baik. Teknik ini berguna dalam mengatasi masalah *overfitting* dan meningkatkan akurasi model (Shorten dan Khoshgoftaar, 2019).

2.5.1 *Flip* Vertikal dan Horizontal

Membalik gambar secara horizontal adalah menukar posisi piksel di sisi kiri dan kanan gambar. Sedangkan, membalik gambar secara vertikal adalah menukar posisi piksel di bagian atas dan bawah gambar (Shorten dan Khoshgoftaar, 2019). Secara matematis, transformasi ini dapat dinyatakan sebagai:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} W - x \\ H - y \end{pmatrix} \tag{3}$$

Dengan:

W adalah lebar gambar

H adalah tinggi gambar,

x, y adalah koordinat piksel asli.

2.5.2 *Random Rotation*

Random rotation atau rotasi acak melibatkan pemutaran gambar dengan sudut tertentu yang dipilih secara acak. Transformasi ini membantu model mengenali objek dari berbagai orientasi (Shorten dan Khoshgoftaar, 2019). Secara matematis, rotasi dapat dinyatakan dengan matriks rotasi:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \tag{4}$$

Dengan:

x dan y merupakan koordinat piksel asli dari gambar

θ adalah sudut rotasi acak yang dipilih dengan rentang tertentu

2.5.3 *Random* Zoom

Random Zoom atau perbesaran acak dengan memperbesar atau memperkecil bagian dari gambar dengan memilih faktor *zoom* secara acak. Teknik ini memungkinkan model untuk fokus pada detail yang berbeda dalam gambar dan membuat model lebih *robust* terhadap skala objek (Shorten dan Khoshgoftaar, 2019). Formula yang digunakan dalam *zoom* acak adalah dengan mengalikan koordinat piksel dengan faktor *zoom* tertentu, yang dapat ditulis sebagai:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} sx \\ sy \end{pmatrix} \quad (5)$$

Dengan:

s sebagai faktor zoom yang diambil dalam rentang tertentu.

2.5.4 *Random Brightness*

Random Brightness atau kecerahan acak adalah mengubah intensitas piksel dalam gambar untuk mensimulasikan berbagai kondisi pencahayaan (Shorten dan Khoshgoftaar, 2019). Nilai kecerahan baru I' dapat dihitung sebagai:

$$I' = I \,.\, b \quad (6)$$

Dengan:

I adalah intensitas piksel asli

b adalah faktor kecerahan yang dipilih secara acak.

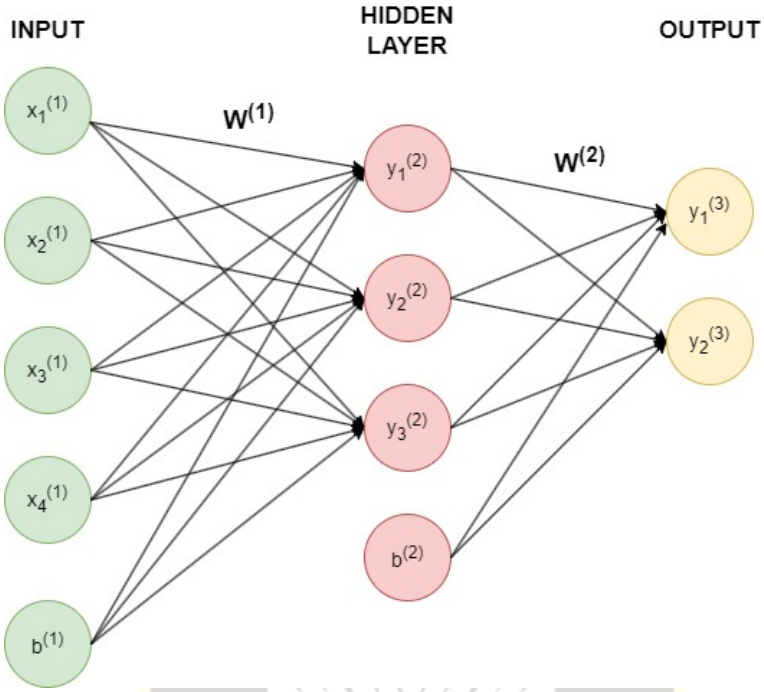
Teknik ini membantu model beradaptasi dengan perubahan pencahayaan yang mungkin terjadi di lingkungan nyata.

2. 6. Jaringan Syaraf Tiruan

Jaringan Syaraf Tiruan (JST) atau *Artificial Neural Network* (ANN) adalah model komputasi yang terinspirasi oleh cara kerja otak manusia dalam memproses informasi. JST terdiri dari kumpulan unit sederhana yang disebut *neuron* yang saling terhubung dalam lapisan-lapisan, yang akan memproses informasi melalui propagasi data dari satu *neuron* ke *neuron* lainnya. Model ini sering digunakan dalam berbagai bidang, termasuk klasifikasi, regresi, dan pengenalan pola. JST merupakan dasar dari arsitektur CNN yang lebih kompleks, dengan tujuan utama mempelajari pola dari data *input* untuk menghasilkan prediksi atau klasifikasi yang akurat (Goodfellow dkk. 2016)

2.6.1 Layer

Layer atau lapisan adalah komponen utama dalam jaringan saraf tiruan. Setiap jaringan saraf terdiri dari beberapa lapisan, yaitu lapisan *input*, lapisan tersembunyi (*hidden layer*), dan lapisan *output*. Ilustrasi dari lapisan ini dapat dilihat pada gambar 2.8 berikut.



Gambar 2. 8 Lapisan Jaringan Syaraf Tiruan (Alzubaidi dkk. 2021)

1) Lapisan *Input*: Lapisan ini menerima data mentah dari luar sistem dan mendistribusikannya ke lapisan tersembunyi berikutnya. Tidak ada komputasi yang dilakukan dalam lapisan ini.
2) Lapisan Tersembunyi: Lapisan ini melakukan sebagian besar komputasi di JST dan terdiri dari *neuron* yang mengolah data menggunakan fungsi aktivasi. Setiap *neuron* dalam lapisan ini menerima *input* dari *neuron* di lapisan sebelumnya dan menghasilkan *output* yang menjadi *input* bagi lapisan berikutnya.

3) Lapisan *Output*: Lapisan ini memberikan hasil akhir dari jaringan setelah melalui lapisan tersembunyi. *Neuron* di lapisan output menentukan prediksi atau klasifikasi berdasarkan fungsi aktivasi yang digunakan.

Secara matematis, output dari setiap *neuron* dalam lapisan lll dapat dirumuskan sebagai berikut (LeCun dkk. 2015) :

$$y^{(l)} = f(W^{(l)}.y^{(l-1)} + b^{(l)}) \quad (7)$$

Dengan:

$y^{(l)}$ adalah output lapisan ke-l

$W^{(l)}$ adalah bobot dari lapisan ke-l

$b^{(l)}$ adalah bias dari lapisan ke-l

f adalah fungsi aktivasi dari model.

2.6.2 Neuron

Neuron adalah unit dasar dalam JST yang memproses dan mentransmisikan informasi. Setiap *neuron* menerima beberapa *input*, melakukan operasi penggabungan linear, dan kemudian menerapkan fungsi aktivasi untuk menghasilkan output (LeCun dkk. 2015). Proses di setiap *neuron* dapat dirumuskan sebagai berikut:

$$y = f(\sum_{i=1}^{n} w_i x_i + b) \quad (8)$$

Dengan:

x_i adalah *input neuron*

w_i adalah parameter dari *neuron*

b adalah bias

f adalah fungsi aktivasi.

2. 7. Fungsi Aktivasi

Fungsi aktivasi merupakan komponen penting dalam jaringan saraf tiruan yang digunakan untuk memperkenalkan non-linearitas. Fungsi ini memungkinkan jaringan untuk belajar dari data yang lebih kompleks dalam mendeteksi pola. Setiap

neuron dalam jaringan biasanya dilengkapi dengan fungsi aktivasi yang membantu menentukan keluaran neuron berdasarkan bobot dan bias. Terdapat beberapa fungsi aktivasi yang umum digunakan dalam jaringan saraf, termasuk fungsi *linear*, ReLU (*Rectified Linear Unit)*, dan *sigmoid* (Goodfellow dkk. 2016).

2.7.1 Fungsi Aktivasi *Linear*

Fungsi aktivasi *linear* adalah fungsi yang sederhana dan tidak memperkenalkan non-linearitas ke dalam model. Fungsi ini mengembalikan nilai output yang sama dengan *input* yang diberikan. Fungsi aktivasi *linear* dapat dituliskan menggunakan persamaan (9):

$$f(x_i) = x_i, \quad -\infty < x_i < \infty \tag{9}$$

Dengan:

x_i merupakan nilai pada neuron ke-i

Fungsi ini sering digunakan di lapisan output dalam masalah regresi dengan hasil akhir berupa nilai kontinu. Fungsi linear dapat berguna dalam kasus-kasus tertentu, namun tidak mampu menangkap pola yang kompleks dalam data karena tidak dapat memperkenalkan non-linearitas ke dalam model (Goodfellow dkk. 2016).

2.7.2 Fungsi Aktivasi ReLU

ReLU (*Rectified Linear Unit*) adalah fungsi aktivasi yang banyak digunakan dalam jaringan saraf karena sederhana dan efektif. Fungsi ini mengembalikan *input* secara langsung jika bernilai positif dan mengembalikan nol jika negatif. Fungsi aktivasi ReLU dapat dituliskan menggunakan persamaan (10)

$$f(x_i) = \max(0, x_i), -\infty < x_i < \infty \tag{10}$$

Dengan:

x_i merupakan nilai pada neuron ke-i

ReLU memiliki keunggulan dalam mengatasi masalah *gradient vanishing* karena memberikan gradien konstan pada nilai positif. ReLU juga mudah dihitung dan lebih cepat untuk mengkonvergensi dalam pelatihan jaringan saraf. Namun, ReLU juga memiliki kelemahan seperti *dying ReLU*, yang terjadi ketika neuron tidak aktif ketika output selalu nol untuk semua *input* (Glorot dkk. 2011).

2.7.3 Fungsi Aktivasi *Sigmoid*

Fungsi *sigmoid* adalah fungsi aktivasi yang mengonversi *input* menjadi rentang nilai $[0, 1]$, sehingga sering digunakan pada lapisan *output* untuk masalah klasifikasi biner. Fungsi aktivasi *sigmoid* dapat dituliskan menggunakan persamaan (11)

$$f(x_i) = \frac{1}{1+e^{-x_i}}, \quad -\infty < x < \infty \tag{11}$$

Dengan:

x_i merupakan nilai pada *neuron* ke-i

Output dari fungsi *sigmoid* ideal untuk memodelkan probabilitas dalam klasifikasi biner. Meskipun *sigmoid* efektif dalam beberapa kasus, fungsi ini memiliki beberapa kelemahan, terutama masalah gradient vanishing yang membuat proses pembaruan bobot pada jaringan yang lebih dalam menjadi lambat (Goodfellow dkk. 2016).

2. 8. *Binary Cross Entropy Loss*

Binary Cross Entropy Loss (BCE *Loss*) adalah fungsi *loss* yang umum digunakan dalam masalah klasifikasi biner. Fungsi ini mengukur perbedaan antara prediksi probabilitas dari model dengan label yang diharapkan. BCE *Loss* menghitung kesalahan model dengan membandingkan probabilitas hasil prediksi terhadap nilai label yang benar, sehingga mengarahkan model untuk lebih akurat dalam memberikan prediksi probabilitas pada kelas yang benar. Persamaan dari BCE Loss dapat dilihat pada persamaan (12)

$$BCE\ Loss = -\frac{1}{N}\sum_{i=1}^{N}(y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)) \quad (12)$$

Dengan:

N adalah jumlah sampel

y_i adalah label sebenarnya dari sampel ke-i (0 atau 1)

$\hat{y}_i$ adalah probabilitas prediksi dari model untuk sampel ke-i (dalam rentang [0,1])

BCE *Loss* menjadi nol ketika model memprediksi probabilitas yang benar untuk semua sampel. Jika prediksi jauh dari nilai yang benar, nilai BCE Loss akan meningkat, mendorong model untuk memperbaiki kesalahannya selama proses pelatihan. Fungsi *loss* ini sangat cocok untuk kasus klasifikasi biner dan banyak digunakan dalam jaringan saraf tiruan yang menerapkan fungsi aktivasi *sigmoid* pada lapisan output (Goodfellow dkk. 2016)

2. 9. Optimasi Adam

Optimasi Adam (*Adaptive Moment Estimation*) adalah algoritma optimasi yang banyak digunakan dalam pelatihan jaringan saraf tiruan, karena mampu mempercepat proses konvergensi dan menangani berbagai kondisi data. Adam mengombinasikan keunggulan dari dua metode optimasi sebelumnya, yaitu Momentum dan RMSProp (Ruder, 2016), dengan memanfaatkan rata-rata tertimbang dari gradien (Momentum) dan kuadrat gradien (RMSProp) untuk menyesuaikan tingkat pembelajaran (*learning rate*) secara adaptif untuk setiap parameter.

Adam bekerja dengan menghitung rata-rata eksponensial dari gradien pertama (momentum) dan kedua (RMS) dari gradien. Rumus dasar dari optimasi Adam dapat dituliskan menggunakan persamaan (13)

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \quad (13)$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \quad (14)$$

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \quad (15)$$

$$\hat{v}_t = \frac{v_t}{1-\beta_2^t} \tag{16}$$

$$\theta_{t+1} = \theta_t - \alpha \frac{\hat{m}_t}{\sqrt{\hat{v}_t}+\epsilon} \tag{17}$$

Dengan:

m_t adalah estimasi Momentum yang diperbarui pada iterasi ke-t

v_t adalah estimasi RMS yang diperbarui pada iterasi ke-t

g_t adalah gradien pada iterasi ke-t

β_1 dan β_2 adalah konstanta untuk mengatur momentum dan RMS

θ_{t+1} adalah nilai parameter yang diperbarui

α adalah tingkat pembelajaran ($learning\ rate$)

ϵ adalah nilai kecil yang digunakan untuk mencegah pembagian nol

Keunggulan dari Adam adalah kemampuannya untuk menyesuaikan tingkat pembelajaran untuk setiap parameter secara individual, yang mempercepat konvergensi serta mencegah *stuck* di titik lokal minimum. Algoritma ini juga cukup robust dan cocok digunakan dalam berbagai jenis jaringan saraf, termasuk CNN, RNN, dan LSTM (Kingma dan Ba, 2014).

2. 10. *Convolutional Neural Network*

Convolutional Neural Network (CNN) adalah arsitektur jaringan saraf yang dirancang khusus untuk memproses data dengan struktur *grid* seperti gambar. CNN sangat efektif dalam mengenali pola visual seperti tepi, tekstur, dan bentuk yang lebih kompleks, sehingga banyak digunakan dalam pengenalan gambar, deteksi objek, dan pengenalan wajah. CNN terdiri dari beberapa lapisan utama, yaitu lapisan konvolusi, lapisan *padding*, dan lapisan *stride*, yang memungkinkan jaringan untuk mengekstrak fitur dari data *input* secara hierarkis dan efisien (Lecun dkk. 1998).

2.10.1 Lapisan Konvolusi

Lapisan konvolusi adalah lapisan utama dalam CNN yang bertanggung jawab untuk mengekstrak fitur dari data *input*. *Filter* (kernel) berukuran kecil

diterapkan pada data *input* untuk menghitung konvolusi, menghasilkan peta fitur yang menunjukkan fitur tertentu dari data *input*. Operasi konvolusi ini dapat dirumuskan pada persamaan (18)

$$(f * I)(x, y) = \sum_{i=-k}^{k} \sum_{j=-k}^{k} I(x + i, y + j) \cdot f(i, j) \quad (18)$$

Dengan:

I adalah data *input* (gambar)

f adalah *kernel* atau *filter*

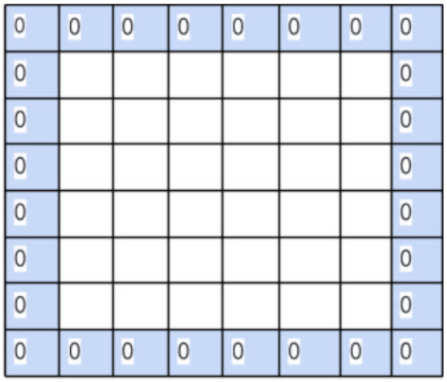
x, y adalah posisi pada gambar

k adalah setengah ukuran filte

Filter yang berbeda dapat digunakan untuk mengekstrak fitur yang berbeda, seperti tepi atau tekstur tertentu. Setiap peta fitur dihasilkan oleh filter yang berbeda, yang nantinya diproses lebih lanjut dalam lapisan-lapisan berikutnya untuk mengenali pola yang lebih kompleks (Goodfellow dkk. 2016).

2.10.2 Lapisan *Padding*

Lapisan *padding* adalah lapisan yang menambahkan piksel tambahan di sekitar tepi data *input*, yang bertujuan untuk mengendalikan ukuran output peta fitur setelah operasi konvolusi. *Padding* diperlukan agar fitur di tepi gambar tetap dipertimbangkan dalam perhitungan konvolusi (Dumoulin & Visin, 2016). Dengan padding, ukuran peta fitur dapat dipertahankan atau disesuaikan sesuai kebutuhan model.



Gambar 2. 9 Ilustrasi *Zero Padding*

Gambar 2.9 menunjukkan gambar 6x6 yang dikenai oleh padding sebanyak 1, nilai dari padding memiliki nilai 0 sehingga disebut dengan *zero padding*. Penambahan ini bertujuan untuk menjaga output dari gambar setelah dikenai oleh konvolusi. Persamaan untuk menentukan jumlah padding dapat dilihat pada persamaan (19)

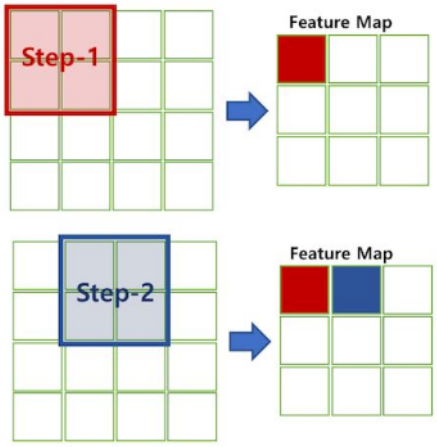
$$padding = \frac{k-1}{2} \tag{19}$$

Dengan:

k adalah ukuran kernel

2.10.3 Lapisan Stride

Lapisan stride mengontrol pergerakan filter saat melakukan konvolusi pada data *input*. Stride mengatur seberapa banyak filter bergeser di setiap langkah selama proses konvolusi. Dengan stride yang lebih besar, peta fitur akan memiliki ukuran yang lebih kecil, karena filter melewati lebih banyak piksel dalam satu kali langkah.

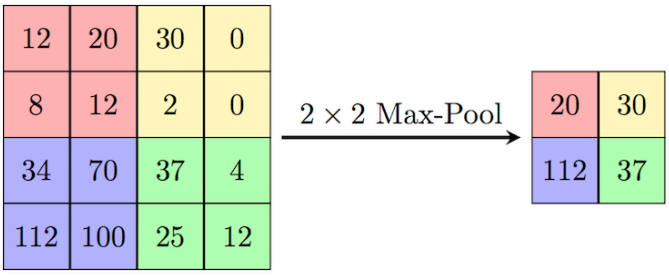


Gambar 2. 10 Ilustrasi Stride 1 pada CNN

Gambar 2.10 menunjukkan ilustasi konvolusi menggunakan Stride sebesar 1 kali. Stride yang lebih besar memungkinkan jaringan untuk mengurangi ukuran data *input* lebih cepat, namun pada saat yang sama, bisa mengakibatkan hilangnya informasi detail (Goodfellow dkk. 2016).

2. 11. *Max Pooling*

Max pooling adalah teknik yang umum digunakan dalam jaringan saraf konvolusi (CNN) untuk mengurangi dimensi peta fitur dan memperkecil jumlah parameter dalam model, sehingga mempercepat proses komputasi sekaligus membantu mengatasi masalah overfitting. Proses max pooling dilakukan dengan menerapkan filter pada peta fitur dan mengambil nilai maksimum dari setiap region yang diliputi filter tersebut (Scherer dkk. 2010).



Gambar 2. 11 Ilustasi *Max Pooling* 2x2

Gambar 2.11 menunjukkan ilustrasi *max pooling* dengan filter berukuran 2x2 dan stride 2, setiap 2x2 piksel akan dikurangi menjadi satu piksel dengan nilai maksimum dari keempat piksel tersebut, yang akan mengurangi ukuran data *input* tanpa kehilangan informasi penting.

2. 12. *Batch Normalization*

Batch normalization adalah teknik yang digunakan dalam jaringan saraf untuk mempercepat pelatihan dan meningkatkan stabilitas jaringan. Teknik ini dilakukan dengan menormalkan *input* dari setiap lapisan sehingga memiliki distribusi dengan rata-rata nol dan variansi satu. *Batch normalization* dapat membantu mengurangi masalah internal *covariate shift*, yaitu perubahan distribusi aktivasi pada setiap lapisan selama pelatihan, yang dapat memperlambat proses konvergensi dan menyebabkan ketidakstabilan (Ioffe & Szegedy, 2015). Proses batch normalization terdiri dari beberapa tahap utama:

1. Normalisasi pada setiap batch data dengan menghitung rata-rata dan variasi dari batch tersebut. Proses ini dapat dituliskan pada persamaan (20) dan (21)

$$\mu_B = \frac{1}{m}\sum_{i=1}^{m} x_i \tag{20}$$

$$\sigma_B^2 = \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_B)^2 \tag{21}$$

Dengan:

μ_B adalah rata-rata pada *batch* ke-B

σ_B^2 adalah varians pada *batch* ke-B

x_i adalah nilai *input*

m adalah jumlah data pada batch ke-B

2. Setelah normalisasi nilai-nilai yang sudah dinormalisasi $\hat{x}$ diubah menggunakan parameter skala γ dan translasi β, yang dioptimalkan selama pelatihan untuk meningkatkan fleksibilitas model. Proses ini dapat dituliskan pada persamaan (22) dan (23)

$$\hat{x}_i = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} \quad (22)$$

$$y_i = \gamma \hat{x}_i + \beta \quad (23)$$

Dengan:

$\hat{x}_i$ adalah data setelah normalisasi

y_i adalah data setelah $batch\ normalization$

γ adalah parameter skala

β adalah paramter translasi

2. 13. Koneksi Residual

Koneksi residual adalah teknik dalam arsitektur jaringan saraf, khususnya dalam *Residual Networks* (ResNet) yang memungkinkan sinyal untuk melewati satu atau beberapa lapisan melalui koneksi langsung (He et al., 2015). Koneksi residual diperkenalkan untuk mengatasi masalah degradasi dalam jaringan saraf yang sangat dalam, dimana pada kondisi ini jumlah lapisan yang lapisan justru dapat mengurangi akurasi jaringan karena terjadinya *gradient vanishing* atau *exploding*. Teknik ini membantu jaringan mempelajari identitas dari *input* aslinya, sehingga memudahkan pembelajaran pada jaringan yang lebih dalam (He et al., 2015).

Mekanisme Dalam koneksi residual adalah output dari satu atau lebih lapisan sebelumnya ditambahkan langsung ke output dari lapisan berikutnya. Secara matematis, koneksi residual dapat dinyatakan pada persamaan (24)

$$y = F(x, W_i) + x \quad (24)$$

Dengan:

x adalah *input* awal ke lapisan

$F(x, W_i)$ adalah fungsi lapisan tertentu dengan parameter W_i

y adalah output dari blok residual

2. 14. *Confusion Matrix*

Confusion matrix adalah metode evaluasi yang digunakan untuk mengukur kinerja model klasifikasi dengan menampilkan informasi detail mengenai jumlah prediksi yang benar dan salah untuk setiap kelas (Puad dkk. 2023).

		Predicted	
		Positive	***Negative***
Actual	***Positive***	***True Positive***	***False Negative (Type I Error)***
	Negative	***False Positive (Type II Error)***	***True Negative***

Gambar 2. 12 *Confusion Matrix* Klasifikasi *Biner*

Gambar 2.12 menunjukkan *confusion matrix* pada kasus klasifikasi biner. Confusion matrix ini berbentuk matriks 2 x 2 yang mencatat empat jenis hasil utama:

1. *True Positive* (TP): Jumlah kasus yang menunjukkan model memprediksi positif dan sebenarnya positif.
2. *True Negative* (TN): Jumlah kasus menunjukkan model memprediksi negatif dan sebenarnya negatif.
3. *False Positive (FP)*: Jumlah kasus menunjukkan model memprediksi positif tetapi sebenarnya negatif (dikenal sebagai kesalahan tipe I).
4. *False Negative (FN)*: Jumlah kasus menunjukkan model memprediksi negatif tetapi sebenarnya positif (dikenal sebagai kesalahan tipe II).

Melalui hasil ini dapat dihitung matriks untuk menentukan ketepatan prediksi dari model yang digunakan antara lain (Puad et al. 2023) :

1. Akurasi : Menghitung presentase prediksi yang benar

$$Akurasi = \frac{TP+TN}{TP+TN+FP+FN} \quad (25)$$

2. Presisi : Mengukur persentase jumlah kesalahan prediksi positif

$$Presisi = \frac{TP}{TP+FP} \quad (26)$$

3. *Recall* : Mengukur persentase jumlah kesalahan prediksi negative

$$Recall = \frac{TP}{TP+FN} \quad (27)$$

4. *F1-Score* : Rata-rata harmonis dari presisi dan *recall*, yang memberikan gambaran seimbang antara kedua metrik tersebut.

$$F1 - Score = 2\frac{Presisi \cdot Recall}{Presisi+Recall} \quad (28)$$

2. 15. *You Only Look Once* (YOLO)

Bagian ini menguraikan konsep dasar, arsitektur, pendekatan bertahap pada berbagai versi YOLO, serta fungsi loss dalam algoritma YOLO. Secara khusus, kami membahas empat versi YOLO beserta peningkatan yang dilakukan dalam setiap versinya dibandingkan dengan versi sebelumnya.

Pengembang YOLO [56] mengubah pendekatan deteksi objek dengan merancangnya sebagai masalah regresi, bukan klasifikasi. Dalam pendekatan ini, Convolutional Neural Network (CNN) memprediksi bounding box serta probabilitas kelas untuk semua objek dalam sebuah citra. Karena algoritma ini hanya memproses citra satu kali untuk mengenali objek dan menentukan posisinya, metode ini disebut You Only Look Once (YOLO). CNN mampu mengolah masukan visual secara efektif dengan mengekstrak fitur, dengan cara fitur tingkat rendah diteruskan dari lapisan konvolusi awal hingga lapisan konvolusi yang lebih

dalam. Tantangan utama dalam pendekatan ini adalah mengidentifikasi banyak objek secara akurat beserta posisinya dalam satu citra. Dua fitur utama dalam CNN yang berperan dalam penyelesaian masalah ini adalah parameter sharing dan penggunaan beberapa filter.

2.15.1 Arsitektur YOLO

Arsitektur YOLO terdiri dari beberapa komponen utama, yaitu backbone, neck, dan head.

1) Backbone: Berfungsi untuk mengekstrak fitur dari gambar input menggunakan convolutional neural networks (CNN). Pada YOLOv5, backbone yang digunakan adalah CSPDarknet (Cross Stage Partial Darknet) yang meningkatkan efisiensi komputasi dengan membagi dan menggabungkan fitur pada berbagai tingkat kedalaman jaringan (Jocher et al., 2020)
2) Neck: Bagian ini bertanggung jawab untuk menghubungkan fitur dari backbone ke bagian output dengan meningkatkan representasi fitur. Pada YOLOv5, digunakan PANet (Path Aggregation Network) sebagai mekanisme penguatan informasi dari berbagai skala.
3) Head: Bagian terakhir yang bertugas untuk memprediksi bounding box, confidence score, dan kelas objek dengan menerapkan mekanisme anchor-based detection serta loss function yang dioptimalkan.

2.15.2 Intersection *Over Union* (IOU)

Intersection over Union (IoU) adalah metrik yang digunakan untuk mengevaluasi kualitas deteksi objek dengan mengukur kesesuaian antara bounding box prediksi dengan bounding box ground truth. IoU didefinisikan sebagai berikut:

$$IoU = \frac{Area\ of\ Overlap}{Area\ of\ Union} \quad (29)$$

dimana :

$Area\ of\ Overlap$ = luas area perpotongan antara *bounding box* dan *ground truth*

$Area\ of\ Union$ = luas gabungan dari kedua bounding box tersebut

Nilai IoU berkisar antara 0 hingga 1, dengan nilai lebih tinggi menunjukkan tingkat kesesuaian yang lebih baik antara prediksi dan ground truth. Biasanya, deteksi dianggap valid jika IoU melebihi ambang batas tertentu, seperti 0.5 atau 0.75 tergantung pada aplikasi.

2.15.3 Prediksi *Bounding Box* YOLO

Dalam deteksi objek menggunakan YOLO, citra atau bingkai video dibagi menjadi sel grid berukuran S×S. Setiap sel grid bertanggung jawab untuk memprediksi B *bounding box*, yang mencakup informasi posisi dan ukuran, probabilitas keberadaan objek di dalam grid, serta probabilitas kelas objek yang terdeteksi. Prinsip dasar dalam pendekatan ini adalah bahwa pusat suatu objek harus berada dalam sel grid tertentu agar dapat dikenali oleh sel tersebut menggunakan salah satu bounding box yang tersedia.

Secara matematis, setiap sel grid memprediksi parameter berikut untuk satu bounding box (lihat gambar 2.13):

p_c	b_x	b_y	b_w	b_h	$p(c_1)$	$p(c_2)$	...	...	...	...	...	...	...	$p(c_n)$

Gambar 2. 13 Sel Grid S x S

Ilustrasi pada gambar 2.13 menunjukkan representasi struktur prediksi yang dilakukan oleh setiap sel grid dalam arsitektur YOLO (You Only Look Once). Setiap sel bertanggung jawab untuk mendeteksi objek yang pusatnya berada di dalam wilayah sel tersebut. Untuk satu bounding box, sel memprediksi sejumlah

parameter, yaitu probabilitas keberadaan objek dalam sel grid (pc), koordinat relatif dari pusat bounding box terhadap batas sel (bx dan by), serta lebar dan tinggi bounding box yang biasanya telah dinormalisasi terhadap dimensi keseluruhan gambar (bw dan bh). Selain itu, sel juga memprediksi probabilitas bersyarat bahwa objek yang terdeteksi termasuk dalam salah satu dari n kelas yang mungkin, yang dinyatakan dalam bentuk vektor probabilitas p(c1), p(c2), ..., p(cn), dengan asumsi bahwa objek memang ada di dalam sel tersebut. Seluruh output prediksi dari satu sel grid ini mencerminkan gabungan antara informasi spasial, keyakinan keberadaan objek, serta distribusi kelas objek. Proses ini dilakukan secara paralel oleh seluruh sel dalam grid, sehingga membentuk sistem deteksi objek end-to-end yang efisien dan cepat dalam satu kali proses inferensi terhadap citra atau bingkai video.

$$
\begin{aligned}
P_c &= \sigma(t_o) \\
b_x &= \sigma(t_x) + c_x \\
b_y &= \sigma(t_y) + c_y \\
b_w &= p_w e^{t_w} \\
b_h &= p_h e^{t_h} \\
P(c_i) &= \frac{e^{t_i}}{\sum_{j=1}^{n} e^{t_j}}
\end{aligned}
\tag{30}
$$

Dimana:

P_c = menunjukkan probabilitas keberadaan objek dalam sel grid untuk suatu bounding box

σ = Fungsi aktivasi sigmoid

t_x, t_y, t_w, t_h = hasil keluaran dari model

(b_x, b_y) = menunjukkan koordinat pusat bounding box yang diprediksi

(b_w, b_h) = mewakili ukuran bounding box yang diprediksi

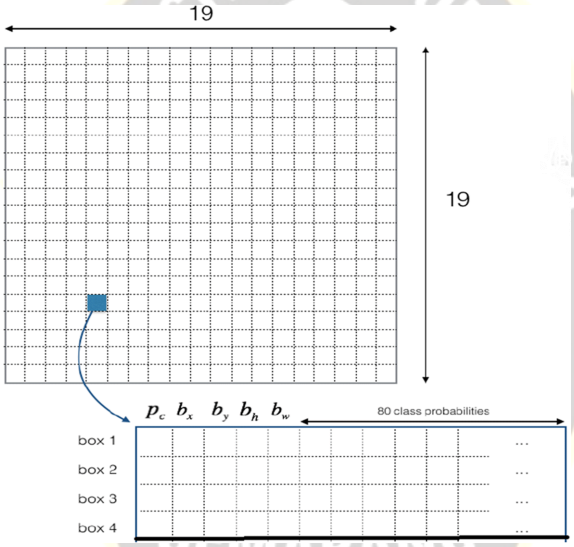
$P(_{ci})$= adalah probabilitas bersyarat bahwa objek termasuk dalam kelas ke-i jika objek berada di dalam bounding box tersebut

n = adalah jumlah kelas dalam sistem.

Input Citra dibagi menjadi sel grid berukuran S×S sehingga setiap sel grid memprediksi nilai berukuran $B \times 5 + n$. Nilai B menunjukkan jumlah bounding box per sel grid. Dengan demikian, output dari jaringan saraf berbentuk tensor berukuran:

$$S \times S \times (B \times 5 + n) \tag{31}$$

Sebagai contoh, jika sebuah citra dibagi menjadi 19 × 19 sel grid dan setiap grid memprediksi empat bounding box, maka output tensor akan sesuai dengan perhitungan di atas. Probabilitas kelas dibagikan ke semua bounding box dalam satu sel grid (lihat gambar 2.14).



Gambar 2. 14 Bounding box pada satu grid

Dalam YOLO, nilai *confidence score* (cs) dihitung untuk setiap bounding box dalam setiap sel grid dengan mengalikan P_c dengan nilai Intersection over Union (IoU) antara bounding box hasil prediksi dan ground-truth bounding box. Jika dalam suatu sel grid tidak terdapat objek, maka confidence score akan bernilai nol. Selanjutnya, class-specific score (css) dihitung untuk setiap bounding box dalam semua sel grid. Nilai ini mencerminkan probabilitas suatu kelas muncul dalam kotak pembatas serta seberapa baik kotak tersebut mencocokkan objek yang ada dalam citra.

2.15.4 *Non-Maximum Supression* (NMS)

Non-Maximum Suppression (NMS) adalah teknik pasca-pemrosesan yang digunakan dalam deteksi objek untuk menyaring bounding box yang tumpang tindih dan mempertahankan deteksi yang paling relevan (Jocher et al., 2020). Teknik ini digunakan untuk memastikan bahwa model hanya menghasilkan satu bounding box yang akurat untuk setiap objek yang terdeteksi, sehingga menghindari duplikasi atau redundansi dalam prediksi. NMS bekerja dengan memanfaatkan nilai confidence score dan Intersection over Union (IoU) untuk menentukan bounding box mana yang harus dipertahankan dan mana yang harus dihapus.
Berikut merupakan Langkah-Langkah Non-Maximum Suppression (NMS)

1) Menghapus Prediksi dengan Confidence Score di Bawah Ambang Batas
2) Mengurutkan Bounding Box Berdasarkan Confidence Score Tertinggi
3) Menghitung IoU antara Bounding Box dengan Skor Tertinggi dan Bounding Box Lainnya.
4) Menghapus Bounding Box yang Memiliki IoU Lebih Tinggi dari Threshold Tertentu
5) Mengulangi Proses untuk Bounding Box yang Tersisa

2.15.5 Fungsi Loss YOLO

Fungsi loss YOLO merupakan gabungan dari beberapa komponen, yaitu loss untuk regresi posisi bounding box, loss untuk confidence score, dan loss untuk klasifikasi. Secara umum, fungsi loss dapat diformulasikan menggunakan persamaan (32).

$$L = \lambda_{coord} \sum_{i=1}^{S^2} \sum_{j=1}^{B} 1_{ij}^{obj} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2]$$

$$+ \lambda_{coord} \sum_{i=1}^{S^2} \sum_{j=1}^{B} 1_{ij}^{obj} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right]$$

$$+ \sum_{i=1}^{S^2} \sum_{j=1}^{B} 1_{ij}^{obj} \left(C_i - \hat{C}_i \right)^2$$

$$+ \lambda_{noobj} \sum_{i=1}^{S^2} \sum_{j=1}^{B} 1_{ij}^{noobj} \left(C_i - \hat{C}_i \right)^2$$

$$+ \sum_{i=1}^{S^2} 1_i^{obj} \sum_{c \in classes} \left(p_i(c) - p_i(c) \right)^2 \quad (32)$$

dimana:

S = Jumlah *grid cell*

B = Jumlah bounding box per grid cell

1_i^{obj} = Indikator bahwa objek ada pada grid cell i dan bounding box ke-j

C_i = *Confidence score*

$p_i(c)$ = Probabilitas kelas

λ_{coord} dan λ_{noobj} = Parameter untuk menyeimbangkan komponen loss

Fungsi loss pada model YOLO dioptimalkan menggunakan algoritma backpropagation dan gradient descent. Proses ini memastikan bahwa model belajar untuk meminimalkan kesalahan prediksi bounding box, confidence score, dan kelas objek secara bersamaan.

2. 16. Bidirectional Feature Pyramid Network (Bi-FPN)

Bi-directional Feature Pyramid Network (Bi-FPN) pertama kali diperkenalkan dalam penelitian *Tan et al. (2020)* pada arsitektur EfficientDet. Bi-FPN merupakan pengembangan dari Feature Pyramid Network (FPN) dan PANet (Path Aggregation Network), yang dirancang untuk meningkatkan fusi fitur multi-skala secara efisien. Berbeda dengan FPN tradisional yang hanya menggunakan

aliran top-down, Bi-FPN menggabungkan jalur top-down dan bottom-up secara bersamaan, memungkinkan pertukaran informasi dua arah antar lapisan dengan resolusi berbeda. Hal ini meningkatkan kemampuan model dalam menangkap konteks global dan lokal, terutama untuk objek berukuran kecil atau tumpang tindih (Tan et al., 2020).

Bi-FPN menggunakan mekanisme weighted feature fusion untuk menggabungkan fitur dari berbagai skala. Setiap node pada Bi-FPN menerima input dari dua jalur:

1. Lapisan dengan resolusi lebih tinggi (misalnya, lapisan yang lebih dalam dari backbone).
2. Lapisan dengan resolusi lebih rendah (misalnya, lapisan yang dihasilkan dari operasi upsampling).

Menggunakan pendekatan ini memungkinkan model untuk menyeimbangkan kontribusi fitur dari berbagai skala secara adaptif (Tan et al., 2020). Proses fusi fitur dapat dimodelkan secara matematis sebagai:

$$P_i^{out} = Conv\left(\frac{w_1 \cdot P_i^{in} + w_2 \cdot UpSample(P_{i+1}^{out}) + w_3 \cdot DownSample(P_{i-1}^{out})}{w_1 + w_2 + w_3}\right) \quad (33)$$

dimana :

w_1, w_2, w_3 = bobot yang dipelajari selama pelatihan

P_i = fitur pada level ke-i