



Pengembangan Algoritma Berbasis *Convolutional
Neural Network* (CNN) dalam Pengukuran Kualitas
Melalui Klasifikasi pada Produk Ban

PROPOSAL PENELITIAN PADA UJIAN KUALIFIKASI

Armando Tirta Dwilaga

99223119

PROGRAM DOKTOR TEKNOLOGI INFORMASI
UNIVERSITAS GUNADARMA

2024

Pengembangan Algoritma Berbasis *Convolutional Neural Network* (CNN) dalam Pengukuran Kualitas Melalui Klasifikasi pada Produk Ban

PROPOSAL PENELITIAN DISERTASI

Untuk Memenuhi Salah Satu Syarat Meraih Gelar Doktor Teknologi Informasi di bawah
Pimpinan Rektor Universitas Gunadarma Profesor Doktor E.S. Margianti, SE, MM

Dipertahankan dalam Ujian Kualifikasi Universitas Gunadarma
Pada Hari Jum'at, 16 Agustus 2024

Armando Tirta Dwilaga

99223119

PROGRAM DOKTOR TEKNOLOGI INFORMASI
UNIVERSITAS GUNADARMA
2024

DAFTAR ISI

DAFTAR ISI	ii
DAFTAR TABEL	vi
DAFTAR GAMBAR	vii
DAFTAR RUMUS	x
1 PENDAHULUAN	
1.1 Latar Belakang Masalah.....	1
1.3 Rumusan Masalah	7
1.4 Tujuan Penelitian	7
1.5 Batasan Masalah.....	8
1.6 Kontribusi dan Manfaat Penelitian.....	8
2 TELAAH PUSTAKA	
22.1 <i>Artificial Intelligence</i>	9
2.2. <i>Machine Learning</i>	9
2.2.1 Jenis Pembelajaran Utama	10
2.3 <i>Deep Learning</i>	11
2.4. <i>Neural Network</i>	12
2.4.1 Segmentasi <i>Neural Network</i>	14
2.4.2 Prediksi <i>Neural Network</i>	14
2.4.3 Desain Eksperimen <i>Neural Network</i>	15
2.4.5 Kelebihan <i>Neural Network</i>	15
2.4.5 kekurangan <i>Neural Network</i>	16
2.4.6. Fungsi Aktivasi	17
2.4.6.1 <i>Rectified Linear Unit (ReLU)</i>	17
2.4.6.2 <i>Sigmoid Function</i>	18
2.4.6.3 <i>Tangent Function (tanh)</i>	18
2.5. <i>Convolutional Neural Network (CNN)</i>	19

2.5.1.	kelompok <i>Convolutional Neural Network</i> (CNN)	23
2.5.1.1	<i>Overfitting</i>	24
2.5.1.2	<i>Epoch</i>	25
2.5.1.3	<i>Learning Rate</i>	25
2.5.2	Kelebihan <i>Convolutional Neural Network</i> (CNN).....	27
2.5.3	Kekurangan <i>Convolutional Neural Network</i> (CNN).....	28
2.5.4.	<i>Backpropagation</i>	28
2.5.4.1	<i>Forward Pass</i>	29
2.5.5.2	<i>Backward Pass</i>	29
2.6.	<i>Library Python</i>	29
2.6.1	<i>Convolution Layer</i>	30
2.6.2	<i>Conv2D Layer</i>	31
2.6.3	<i>Maxpooling2D Layer</i>	31
2.6.4	<i>Flatten Layer</i>	32
2.6.5	<i>Dense Layer</i>	32
2.6.6	<i>Dropout</i>	32
2.7	Klasifikasi dan Prediksi.....	33
2.8	<i>Dataset</i>	33
2.9	<i>Data Preparation</i>	34
2.10	<i>Data Augmentation</i>	34
2.10.1	<i>Translation</i>	34
2.10.2	<i>Rotation</i>	35
2.10.3	<i>Flip</i>	35
2.10.4	<i>Resizing</i>	35
2.10.5	<i>Shift, Shear, & Zoom</i>	35
2.11	<i>Data Splitting</i>	35
2.12	<i>Training Data, Validation Data, dan Test Data</i>	36
2.13	<i>Confusion Matrix</i>	37
2.14.	Citra.....	38
2.14.1	Citra Analog	39
2.14.2	Citra Digital.....	39

2.15. Komponen Sistem Pengolahan Citra	40
2.15.1 Perangkat Keras Pengolahan Citra.....	41
2.15.2 Perangkat Lunak Pengolahan Citra.....	41
2.16. Jenis-Jenis Citra	41
2.16.1 Citra Berwarna	42
2.16.2 Citra <i>Grey Level</i>	43
2.16.3 Citra Biner	44
2.17. Ruang Warna dan Dasar-Dasar Pengolahan Citra	45
2.17.1 Sifat Dasar Cahaya.....	45
2.17.2 Ruang Warna.....	46
2.18. Segmentasi Citra	47
2.18.1 Segmentasi Warna Citra.....	47
2.18.2 Segmentasi Tekstur Citra	48
2.19. Pemrosesan Citra.....	48
2.19.1 Operasi Pemrosesan Citra	48
2.20 Roda dan Ban	49
2.21 Jenis Cacat Ban	52
2.22 Penelitian Terkait	55

METODOLOGI PENELITIAN

3.1 Gambaran Umum Penelitian	70
3.2. Tahapan Penelitian	72
3.2.1 Studi Literatur	73
3.2.2 Data <i>Aquisition</i>	74
3.2.3 Data <i>Preprocessing</i>	75
3.2.4 Data <i>Augmentation</i>	75
3.2.5 Data <i>Splitting</i>	77
3.2.6 Model <i>Building</i>	79
3.2.7 Model <i>Evaluation & Testing</i>	81

3.3	Arsitektur <i>Convolutional Neural Network</i> (CNN).....	81
-----	---	----

4 HASIL DAN PEMBAHASAN

4.1	Implementasi Keras.....	90
4.2	Persiapan <i>Dataset</i>	90
4.3	Augmentasi Data.....	91
4.4	Model <i>Sequential</i>	92
4.5	<i>Confusion Matrix</i>	94
4.5	Hasil Perbandingan	103

DAFTAR PUSTAKA

DAFTAR TABEL

2.1	Contoh <i>Confusion Matrix</i>	37
2.2	<i>State of The Art</i>	55
4.1	Model <i>Sequential</i>	92
4.2	<i>Confusion Matrix</i> untuk <i>Train Data</i>	94
4.3	<i>Confusion Matrix</i> untuk <i>Validation Data</i>	95
4.4	<i>Confusion Matrix</i> untuk <i>Test Data</i>	95
4.5	Hasil Klasifikasi <i>Train Data</i>	98
4.6	Hasil Klasifikasi <i>Validation Data</i>	99
4.7	Hasil Klasifikasi <i>Test Data</i>	101
4.8	Hasil Akurasi Model Klasifikasi Cacat Ban untuk Beberapa Nilai <i>Layer</i> dan <i>Epoch</i>	104
4.9	Hasil Akurasi Model Klasifikasi Cacat Ban untuk Rasio <i>Dataset</i>	104
4.10	Hasil Akurasi Model Klasifikasi Cacat Ban untuk Optimasi Algoritma	105
4.11	Hasil Akurasi Model Klasifikasi Cacat Ban untuk <i>Learning</i> <i>Rate</i>	106

DAFTAR GAMBAR

2.1	Bagian dari <i>Artificial Intelligence</i>	9
2.2	Arsitektur <i>Neural Network</i>	13
2.3	Arsitektur <i>Deep Neural Network Multi-Level</i>	13
2.4	Arsitektur Umum Neural Network dengan Fungsi Aktivasi	14
2.5	Arsitektur <i>Convolutional Neural Network</i>	20
2.6	Operasi Konvolusi Antara Gambar dan Filter	21
2.7	Konvolusi Bagian Pertama Gambar	21
2.8	Operasi <i>Pooling</i> Maksimal	22
2.9	Operasi <i>Pooling</i> Rata-Rata	22
2.10	Diagram Arsitektur <i>Convolutional Neural Network</i> (CNN)	23
2.11	Kondisi <i>Early Stop</i> pada Saat Terjadi <i>Overfitting</i>	24
2.12	Data <i>Splitting</i>	36
2.13	Komponen-Komponen Sistem Akuisisi Pengolahan Citra	40
2.14	(a) Citra Berwarna Berukuran 300 x 500 Piksel, (b) ((a) Representasi 3 Warna Setiap Pixel dan (b) Representasi Masing-Masing Warna (RGB)	43
2.15	(a) Citra <i>Grey Level</i> Berukuran 300 x 500 Piksel, (b) Nilai <i>Luminance</i> Citra <i>Grey Level</i> dengan Ukuran 5 x 5 Pixel	44
2.16	(a) Citra Biner Berukuran 300 x 500 Piksel, (b) Nilai Biner dari Citra Biner dengan Ukuran 5 x 5 Pixel	45
2.17	(a) Spektrum Gelombang Cahaya, (b) Absorpsi dan Refleksi Gelombang	46
2.18	Penajaman Citra (a) Citra Awal dan (b) Citra Penajaman	49
2.19	Pengurangan Derau (a) Citra Awal dan (b) Citra Pengurangan Derau	49

2.20	Anti Pengaburan (a) Citra Kabur dan (b) Citra	
	Anti Pengaburan.....	49
2.21	Ekstraksi Tepi (a) Citra Awal dan (b) Citra Tepi-Tepi	
	Diekstrasi	50
2.22	Binerisasi (a) Citra Awal dan (b) Citra	
	Konversi Hitam Putih.....	50
2.23	Pengaburan (a) Citra Awal dan (b) Citra	
	Setelah Pengaburan	50
2.24	Peningkatan Kontras (a) Citra Awal dan (b) Citra Memperbaiki	
	Kontras	51
2.25	Segmentasi (a) Citra Awal dan (b) Citra Setelah Segmentasi.....	51
2.26	Kecacatan dengan Material yang Sama dari Material	
	Ban (CH01)	52
2.27	Kecacatan dengan Material yang Berbeda dari Material	
	Ban (CH02)	53
2.28	Kerusakan oleh Suhu dan Tekanan (CH03).....	53
2.29	Retak (CH04)	54
2.30	Kerusakan Mekanisme Terhadap Integritas (CH05).....	54
2.31	Material <i>Etched</i> (CH06).....	54
3.1	Blok Diagram Gambaran Umum Penelitian	70
3.2	Tahapan Penelitian	73
3.3	Tahapan <i>Study Literature</i>	74
3.4	Tahapan Data <i>Aquisition</i>	74
3.5	Tahapan Data <i>Preprocessing</i>	75
3.6	Tahapan Data <i>Augmentation</i>	77
3.7	Tahapan <i>Splitting</i> Data	78
3.8	Tahapan <i>Building Model</i>	80
3.9	Tahapan <i>Model Evaluation & Testing</i>	81
3.10	Tahapan <i>Convolutional Neural Network</i> (CNN) dengan	
	Model Keras	82
4.1	Contoh Data Citra Ban Pelatihan Tanpa Augmentasi Data	91

4.2	Contoh Data Citra Ban Pelatihan dengan Augmentasi Data.....	92
4.3	<i>Training Evaluation</i>	106

DAFTAR RUMUS

2.1	Rumus Fungsi Aktivasi <i>Rectified Linear Unit</i> (ReLU)	18
2.2	Rumus Bentuk Kedua Fungsi Aktivasi <i>Rectified Linear Unit</i> (ReLU)	18
2.3	Rumus Aktivasi <i>Sigmoid Function</i>	18
2.4	Rumus Aktivasi <i>Tangent Function</i>	19
2.5	Rumus Akurasi.....	37
2.6	Rumus Presisi.....	38
2.7	Rumus <i>Recall</i>	38
2.8	Rumus <i>F1-Score</i>	38
2.9	Rumus <i>Error</i>	38

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pertumbuhan industri menurut kementerian perindustrian republik Indonesia pada triwulan III tahun 2023, kinerja sektor industri pengolahan terus membaik. Berdasarkan data yang dikeluarkan Badan Pusat Statistik (BPS), sektor industri pengolahan tumbuh sebesar 5,20 persen pada periode ini, melampaui pertumbuhan PDB sebesar 4,94 persen pada periode yang sama. Sehingga industri manufaktur utamanya sudah memberikan kontribusi berdasarkan nilai perekonomian nasional. Ada banyak harapan positif terkait implikasi pembangunan berkelanjutan dari teknologi Industri 4.0 selama beberapa tahun terakhir. Kurangnya pemahaman yang akurat tentang proses di mana teknologi Industri 4.0 memungkinkan manufaktur berkelanjutan adalah penghalang utama bagi bisnis yang mengejar digitalisasi dan pemikiran berkelanjutan (Ching, Ghobakhloo, Iranmanesh, Maroufkhani, & Asadi, 2022).

Semakin meningkatnya persaingan di dunia usaha, harus sejalan dengan meningkatnya teknologi pada proses operasi setiap perusahaan. Suatu produk dianggap cacat atau tidak berhasil jika tidak memenuhi standar yang telah ditetapkan perusahaan. Memilih suatu produk, pembeli mempertimbangkan beberapa aspek, salah satunya adalah kualitas produk. Teknologi telah menjadi faktor utama dalam mendukung produktivitas dan keberlanjutan usaha kecil dan menengah. Penggunaan teknologi oleh usaha kecil dan menengah masih terbatas, yang sering kali disebabkan oleh keterbatasan kemampuan dan pengetahuan sumber daya manusia (SDM). Konteks ini tidak hanya dilihat sebagai alat untuk meningkatkan efisiensi operasional tetapi juga sebagai sarana untuk meningkatkan daya saing dan memperluas pasar (Hernita, Surya, Perwira, Abubakar, & Idris, 2021).

Landasan pengendalian kualitas dalam produksi massal adalah penggunaan metrik yang berbeda untuk mengukur seberapa dekat produk mematuhi spesifikasi

yang ditetapkan perusahaan. Mengenai produksi massal, fitur yang paling umum diukur dalam pengendalian kualitas tradisional adalah dimensi geometris, parameter material, dan parameter visual. Selain harus memverifikasi kondisi permukaan produk tertentu untuk memastikan kelengkapan produk, parameter visual merupakan subjek yang sangat penting dan menantang untuk diotomatisasi. Pengujian ini dapat dilakukan tanpa menghentikan pergerakan benda yang diangkut. Pemeriksaan terhadap data yang dikumpulkan dan asumsi yang sering diabaikan atau salah mengenai keakuratannya adalah sumber permasalahannya (Oborski & Wysocki, 2022). Produksi massal dalam permulaan produk yang cacat adalah masalah besar karena hal ini berarti membutuhkan biaya pengerjaan ulang atau pembongkaran yang besar. Jika cacat tidak segera ditemukan, cacat tersebut mungkin akan terus berlanjut selama proses berlangsung dan menyebabkan masalah yang lebih serius. Mengidentifikasi penyebab mendasar dari kelemahan ini, apakah berasal dari produksi massal atau masalah terkait proses lainnya, sangat penting. Cacat yang terkait dengan produk muncul dari beberapa alasan yang bergantung pada sifat cacat, sedangkan cacat yang terkait dengan proses timbul dari optimalisasi parameter proses identifikasi cacat yang di bawah standar (Hassan, Hamdan, Shahin, Adelmaksoud, & Bitar, 2023). Ketika ban dibuat dalam jumlah besar yang dikenal sebagai produksi massal, melakukan pemeriksaan kualitas akhir yang mempertimbangkan karakteristik kualitatif dan visual ban sebelum diluncurkan ke pasar menjadi sangat menantang. Karakteristik visual ban sebagian besar terdiri dari ciri-ciri formal (tetapi perlu) seperti anotasi, kode batang, atau ciri-ciri lain yang diperlukan untuk mengidentifikasi produk, sedangkan aspek kualitatif berkonsentrasi pada bahan, geometri, tampilan ban, dan fungsi akhir. Konten visual yang bebas cacat merupakan tanda produk berkualitas tinggi. Menjamin produksi massal ban berkualitas tinggi, data dari proses manufaktur harus dikumpulkan, dan pemeriksaan kualitas akhir harus didigitalkan sebelum produk keluar dari pabrik (Kuric, Klarak, Saga, Cisar, Hajducik, & Wiecek, 2021).

Kemajuan teknologi yang pesat ini juga berdampak pada industri. Industri 4.0 adalah perpaduan antara aset fisik dan teknologi canggih seperti kecerdasan buatan, IoT, robot, pencetakan 3D, komputasi awan, dan lain-lain. Organisasi yang

mengadopsi 4.0 bersifat fleksibel dan siap mengambil keputusan berdasarkan data. Kemajuan teknologi kecerdasan buatan dan komputasi kognitif membawa dunia manufaktur ke kecepatan tinggi dan meningkatkan efisiensi bisnis (Adel, 2022).

Jika menyangkut masalah kualitas, salah satu teknologi yang digunakan seperti kecerdasan buatan, harus mampu memberikan jawabannya. Karena pembelajaran mesin dan kecerdasan buatan khususnya semakin banyak digunakan dalam operasi manufaktur. Salah satu pendekatan pengukuran dan pengendalian kualitas adalah penerapan pembelajaran mesin (Fahle, Prinz, & Kuhlenkotter, 2020). Menggunakan teknologi ini, di pabrik-pabrik cerdas menggunakan kecerdasan buatan untuk melakukan tugas-tugas tingkat tinggi. Kini juga dapat memutuskan dan belajar dari pengalaman dalam berbagai situasi yang sedang berlangsung. Teknologi AI memungkinkan untuk melakukan tugas-tugas tingkat tinggi, belajar dari pengalaman, dan meningkatkan sistem otomatisasi dalam pengukuran kualitas. Membantu produsen mencapai volume produksi yang besar dan meningkatkan jaminan kualitas sebelum produk dipasarkan (Javaid, Heleem, Singh, & Suman, 2021). Menurut penelitian Tulbure, Adrian, & Dulf (2021), mengatakan salah satu metode yang digunakan dalam hal ini adalah klasifikasi dan pengenalan pola foto. Beberapa tahun terakhir, visi komputer telah diubah oleh pengembangan algoritme pembelajaran mendalam dan banyaknya GPU yang memasuki pasar, sehingga memungkinkan komputasi yang kuat dan berskala besar. Sebelum pembelajaran mendalam, seseorang harus memiliki banyak pengetahuan dalam fitur pemetaan dan deskripsi fitur, agar dapat membuat fitur dan mendeteksi objek dari gambar. Pentingnya metode *deep learning* seperti CNN dalam klasifikasi gambar dan deteksi objek, yang sejalan dengan penggunaan algoritma pengenalan pola untuk mengatasi masalah kualitas produk. Deteksi otomatis diharapkan dapat mengidentifikasi cacat pada setiap produk yang diproduksi. Menurut penelitian Tamborski, Rojek, & Mikolajewski (2023), penggabungan kecerdasan buatan telah mengubah sejumlah industri baru-baru ini, termasuk sektor ban. Konteks pengukuran dan pengendalian kualitas produk ban di industri, identifikasi cacat ban otomatis masih menjadi masalah ilmiah dan teknis yang signifikan dan sulit. Kemampuan kecerdasan buatan untuk meningkatkan kontrol kualitas ban dapat

merevolusi bisnis ban dan menghasilkan produksi ban yang lebih aman, lebih dapat diandalkan, dan tidak terlalu berbahaya bagi lingkungan.

Metode inspeksi visual sebagai salah satu metode pengujian non-destruktif yang digunakan dalam industri manufaktur dan Inspeksi visual disebutkan sebagai metode yang cepat dan berbiaya rendah dalam pengujian non-destruktif (Yan, Ren, Sun., & Williams, 2024). Kemampuan kecerdasan buatan yang dipadukan dengan pembelajaran mendalam menjadikan pengendalian kualitas sangat bermanfaat, terutama di bidang inspeksi visual produk. Karena kemudahan penggunaannya, inspeksi visual sering digunakan sebagai metode pengujian kualitas tanpa menimbulkan kerusakan pada benda pertama yang diperiksa. Salah satu cara paling mendasar dan cepat untuk mengevaluasi dan menguji kualitas suatu produk tanpa menyebabkan kerusakan pada material (Kuric, Klarak, Saga *et al.*, 2021).

Pemeriksaan kualitas berbasis teknologi adalah salah satu prosedur utama yang digunakan untuk menilai barang dan menentukan apakah akan menerima atau menolaknya. Kebanyakan kasus operator manusia memeriksa produk selama tahap inspeksi visual atau inspeksi akhir untuk memverifikasi kesesuaian, meskipun presisi dan keandalan inspeksi sering kali tidak memadai. Beberapa faktor yang mempengaruhi proses inspeksi visual sehingga menghasilkan akurasi inspeksi secara keseluruhan sekitar 80% di industri yang pada tujuan inspeksi 100% dalam sistem manufaktur canggih, inspeksi visual manual memakan waktu dan biaya. Hal tersebut menciptakan sumber teknologi informasi terkait teknologi baru dalam klasifikasi dengan menyajikan pendekatan berbasis kecerdasan buatan untuk inspeksi visual proses dengan menggunakan pembelajaran mendalam. Pendekatan ini mencakup *Convolutional Neural Network* (CNN) untuk inspeksi dan aplikasi komputer yang dapat diterapkan di lantai pabrik untuk membuat proses inspeksi mudah digunakan. Penggunaan teknologi algoritma berbasis *Computer Vision* telah membantu mengotomatisasi bagian-bagian proses inspeksi visual (Sundaram & Zaid, 2023). Menurut penelitian Benbarrad, Salhaoui, Kenitar, & Arioua (2021), sumber teknologi informasi terkait teknologi baru dalam klasifikasi sebagai teknologi inovatif, *machine vision* memungkinkan inspeksi 24/7 yang andal dan cepat serta membantu produsen meningkatkan efisiensi operasi manufaktur. Data

yang dapat diakses oleh peralatan visi akan digunakan untuk mengidentifikasi dan melaporkan produk cacat. Menurut penelitian Nguyen, Yu, Shin, Kwon, Kwak, & Kim (2021), sumber teknologi informasi terkait teknologi baru dalam klasifikasi dengan menerapkan pra-pemrosesan pada data yang dikumpulkan untuk mengekstrak *Region of Interest* (ROI) dari gambar. Karena keterbatasan ketersediaan data berlabel yang sesuai, maka menggunakan metode pembelajaran transfer untuk melatih kembali model klasifikasi dengan model yang telah dilatih sebelumnya. Setelah melakukan tugas klasifikasi, pada kumpulan data kabel listrik dan kumpulan data transmisi yang diterbitkan sebelumnya, menggunakan berbagai jaringan saraf dalam termasuk *VGGNet*, *ResNet*, *DenseNet*, dan *GoogLeNet*, peneliti menganalisis hasil yang dicapai oleh sistemnya. Hasil eksperimen menunjukkan bahwa sistem mampu mengklasifikasikan produk cacat dengan cepat dan akurat di lingkungan manufaktur dunia nyata. Menurut penelitian Kiruthikaa, Pon, Samyuktha, Swathi, & Vinush (2023), sumber teknologi informasi terkait teknologi baru dalam klasifikasi di mana model *Convolutional Neural Network* (CNN) berfungsi sebagai dasar untuk *database* gambar, yang kemudian digunakan untuk memvalidasi kebenaran model menggunakan metrik yang sesuai. Eksperimen menunjukkan keakuratan, kinerja luar biasa, dan biaya rendah dari pendekatan estimasi umur ban yang disarankan.

Saat ini penelitian sudah banyak membahas mengenai deteksi otomatis yang memanfaatkan teknologi komputer untuk menangani permasalahan terkait pengukuran kualitas produk. Salah satu teknik pembelajaran mesin yang dapat mengatasi masalah identifikasi pola dan kategorisasi objek dalam gambar adalah *Convolutional Neural Network* (CNN) dengan aspek kecerdasan buatan maka prosedur ekstraksi dan klasifikasi fitur dapat dilakukan dengan pembelajaran mesin pada arsitektur yang sama. Menurut penelitian Wang, Guo, Lu, & Zhang (2019), penggunaan metode *Convolutional Neural Network* (CNN) dengan model arsitektur *Fully Convolutional Network* dan VGG16 dibangun sebagai arsitektur dasar untuk direpresentasikan gambar ban secara akurat menemukan dan mengelompokkan cacat pada gambar ban. Hasil perbandingan VGG16 dengan tiga lapisan menghasilkan akurasi 78.9% yang menyatu tidak hanya bisa akurat

mendeteksi cacat skala yang berbeda tetapi juga menghasilkan lebih tepat prediksi menghasilkan jenis cacat ban yang berbeda. Menurut penelitian Lin (2023), menggunakan metode *deep learning* yaitu *Convolutional Neural Network* (CNN) khususnya metode *ShuffleNet* yang ditingkatkan dengan fokus pentingnya kualitas ban dan risiko yang terkait dengan cacat seperti lubang kecil dan retakan pada permukaan ban. Penelitian ini membandingkan kinerja lima metode berbeda, termasuk akurasi klasifikasi *GoogLeNet* adalah 82,7%, *ShuffleNet* tradisional adalah 85,3%, *VGGNet* 87,3%, *ResNet* adalah 90%, dan *ShuffleNet* yang ditingkatkan memiliki akurasi tertinggi adalah 94,7% menjadikannya efektif dalam mendeteksi kerusakan ban. Menurut penelitian. Menurut penelitian Li, Fan, Zhang, & Jiang (2021), menggunakan metode algoritma *Convolutional Neural Networks* (CNN) dan arsitektur *TireNet* pengembangan sebuah kerangka kerja *end-to-end* untuk deteksi cacat pada gambar sinar-X ban dengan melibatkan 11 jenis cacat ban. Algoritma yang diusulkan, *TireNet* berhasil mencapai tingkat *recall* tertimbang sebesar 94,7% pada *dataset* yang ditawarkan. Hal ini menunjukkan kemampuan algoritma untuk mendeteksi cacat ban dengan tingkat keberhasilan yang tinggi. Menurut penelitian Wu, Jiao, Sun, & Chen (2020), Mengusulkan metode deteksi cacat ban berdasarkan R-CNN digunakan untuk deteksi objek dengan menggabungkan *Convolutional Neural Network* (CNN) dengan algoritma pemilihan wilayah. Metode deteksi cacat ban berfokus pada deteksi dinding samping, benda asing pada tapak, dan gelembung dinding samping yang menggabungkan fitur konvolusi lapisan ketiga dan kelima dalam arsitektur Zeiler & Fergus Network untuk mengekstraksi karakteristik mendalam sebagai masukan untuk Faster R-CNN. Metode deteksi cacat ban yang diusulkan mencapai klasifikasi dan lokalisasi cacat ban yang akurat, dengan tingkat pengenalan pengujian rata-rata hingga 95,4%.

Berdasarkan penjelasan penelitian terdahulu salah satunya yang mewakili menurut Wang, Guo, Lu *et al* (2019) permasalahan penggunaan algoritma *Convolutional Neural Network* (CNN) dengan arsitektur *Fully Convolutional Network* mungkin tidak sensitif terhadap cacat kecil pada gambar ban, seperti gelembung, karena hilangnya detail yang disebabkan oleh operasi pengumpulan

data. Berdasarkan permasalahan yang terjadi masalah sensitivitas terhadap cacat kecil pada gambar ban dapat diatasi dengan menggunakan algoritma *Convolutional Neural Network* (CNN) dengan membangun model atau kerangka kerja menggunakan Keras. Keras menyediakan berbagai model *pre-trained* yang dapat digunakan. Menambahkan lapisan-lapisan seperti *convolutional*, *MaxPooling2D*, *Flatten*, dan *Dense* yang akan membantu model untuk mempelajari fitur-fitur yang lebih spesifik terkait dengan gambar ban dan meningkatkan sensitivitas terhadap cacat. Sehingga dapat memanfaatkan model-model ini untuk meningkatkan sensitivitas terhadap cacat pada gambar ban.

1.2 Rumusan Masalah

Berikut adalah rumusan masalah yang melandasi dalam penelitian ini:

1. Bagaimana membangun model algoritma *Convolutional Neural Networks* (CNN) dalam melakukan klasifikasi cacat pada produk ban?
2. Bagaimana mengimplementasikan dan menghasilkan usulan model algoritma *Convolutional Neural Networks* (CNN) dalam membangun sistem deteksi cacat produk ban?
3. Bagaimana mengukur kinerja dan performa model algoritma *Convolutional Neural Networks* (CNN) dalam melakukan klasifikasi cacat pada produk ban?

1.3 Tujuan Penelitian

Berikut adalah tujuan penelitian yang melandasi dalam penelitian ini:

1. Membangun model algoritma *Convolutional Neural Networks* (CNN) dalam melakukan klasifikasi cacat pada produk ban.
2. Mengimplementasikan dan menghasilkan usulan model algoritma *Convolutional Neural Networks* (CNN) dalam membangun sistem deteksi cacat produk ban.
3. Mengukur kinerja dan performa model algoritma *Convolutional Neural Networks* (CNN) dalam melakukan klasifikasi cacat pada produk ban.

1.4 Batasan Masalah

Penelitian ini memiliki batasan masalah yaitu fokus pada pengembangan algoritma *Convolutional Neural Networks* (CNN) dan implementasi dengan membangun menggunakan Keras pada lapisan-lapisan seperti *convolutional*, *MaxPooling2D*, *Flatten*, dan *Dense*.

1.5 Kontribusi dan Manfaat Penelitian

Penelitian ini memiliki kontribusi dalam bidang teknologi, ilmu pengetahuan, dan akademik untuk menyesuaikan kebijakan kepada pembaca, *stakeholders* dan konsumen sebagai referensi pembelajaran dan penyampaian informasi lebih cepat sebagai berikut:

1. Kontribusi bidang teknologi kendali mutu pada sektor manufaktur, khususnya pada bisnis ban, sangat terbantu dengan adanya penelitian ini. Proses pengukuran kualitas produk ban dapat diotomatisasi mengetahui akurasi dan konsistensi dengan menggunakan metode *Convolutional Neural Network* (CNN).
2. Kontribusi bidang ilmu pengetahuan pada pemahaman yang lebih baik tentang bagaimana metode pembelajaran mesin, khususnya algoritma *Convolutional Neural Network* (CNN), diterapkan pada pengukuran kualitas produk manufaktur. Para peneliti dapat mengetahui apa yang mempengaruhi kinerja algoritma *Convolutional Neural Network* (CNN) dalam mengklasifikasikan kualitas ban melalui uji coba dan analisis yang dilakukan dalam pekerjaan ini.
3. Kontribusi bidang akademik menyumbangkan pengetahuan dan wawasan baru pada literatur ilmiah terkait pengukuran kualitas dan penggunaan algoritma *Convolutional Neural Network* (CNN) dalam industri manufaktur.
4. Kontribusi pada masyarakat, mempengaruhi tingkat keselamatan masyarakat (pengemudi dan penumpang) karena ketika ban dalam kondisi baik dan bebas cacat sangat penting mencegah kecelakaan.

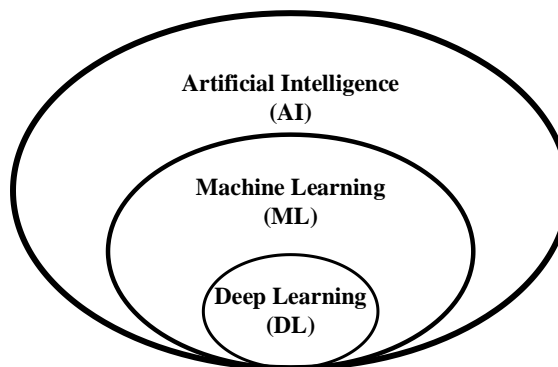
BAB II

TELAAH PUSTAKA

2.1 *Artificial Intelligence*

Artificial Intelligence (AI) adalah ilmu yang mempelajari bagaimana membuat komputer melakukan tugas-tugas cerdas yang sebelumnya hanya bisa dilakukan oleh manusia. *Artificial Intelligence* (AI) menggunakan komputer untuk mensimulasikan perilaku cerdas manusia dan melatih komputer untuk belajar perilaku manusia seperti pembelajaran, penilaian, dan pengambilan keputusan. AI adalah proyek pengetahuan yang berfokus pada akuisisi pengetahuan, analisis, dan metode ekspresi pengetahuan untuk mencapai simulasi aktivitas intelektual manusia (Zhang & Lu, 2021).

Sebagai gambaran tingkat tinggi ada bagian-bagian di dalam konsep *artificial intelligence*, sederhananya *deep learning* adalah bagian dari *machine learning*, dan *machine learning* adalah bagian dari *artificial intelligence*, seperti pada Gambar 2.1



Gambar 2.1 Bagian dari *Artificial Intelligence*

2.2. *Machine Learning*

Machine learning adalah cabang dari kecerdasan buatan yang mempelajari cara membuat algoritma yang dapat meningkatkan kinerja mereka melalui pengalaman (data). Empat jenis masalah terpenting yang perlu diselesaikan melalui pembelajaran mesin adalah prediksi, pengelompokan, klasifikasi, dan reduksi dimensi. Inti dari *machine learning* adalah penggunaan algoritma yang dapat

belajar dari data untuk membuat keputusan atau prediksi tentang data baru. Metode *machine learning* mencakup berbagai jenis pembelajaran, seperti pembelajaran terawasi, tidak terawasi, semi-terawasi, dan pembelajaran penguatan (Zhang & Lu, 2021).

2.2.1 Jenis Pembelajaran Utama

Meskipun bidang *machine learning* sangat luas, algoritma *machine learning* biasanya diklasifikasi menjadi dua jenis "*supervised*" dan "*unsupervised*". Algoritma *supervised, training* data berisi fitur dan label. Tujuannya adalah mempelajari fungsi yang mengambil vektor fitur sebagai masukan dan mengembalikan label prediksi. *Supervised* selanjutnya dapat dikategorikan ke dalam klasifikasi dan regresi. Jika label bersifat kategoris, tugas pembelajaran biasanya disebut "klasifikasi". Tugasnya adalah memprediksi suatu kondisi nilai numerik yang kontinu yang disebut regresi. Algoritma *unsupervised, training* data tidak memiliki label. Tujuan dari algoritma *unsupervised* adalah untuk menganalisis pola dalam data tanpa memerlukan anotasi. Contoh umum dari algoritma tanpa pengawasan mencakup pengelompokan dan pengurangan dimensi (Swaminathan & Parente, 2023).

Upaya dalam membantunya menyelesaikan tugasnya, *machine learning* dapat dikategorikan menurut cara pembelajarannya. Berdasarkan cara pembelajarannya, pembelajaran mesin dipisahkan menjadi empat kategori (Zhang & Lu, 2021):

1. *Supervised Learning*

Menggunakan data berlabel untuk melatih model agar dapat memprediksi tipe atau nilai data baru. Berdasarkan hasil prediksi yang berbeda, *supervised learning* dapat dibagi menjadi dua kategori klasifikasi dan regresi. Metode yang umum digunakan adalah *Support Vector Machine* (SVM) dan diskriminasi linear. Masalah regresi mengacu pada prediksi *output* nilai kontinu, sedangkan masalah klasifikasi mengacu pada prediksi *output* nilai *diskret*.

2. *Unsupervised Learning*

Menemukan pola dalam data tidak berlabel. Algoritma mencoba mengidentifikasi struktur atau pola yang mendasari dalam data tanpa panduan

dari label. Metode yang umum digunakan dalam *unsupervised learning* adalah *k-clustering* dan *principal component analysis* (PCA).

3. *Semi Supervised Learning*

kombinasi dari data berlabel dan tidak berlabel. Algoritma menggunakan sejumlah kecil data berlabel bersama dengan sejumlah besar data tidak berlabel untuk meningkatkan akurasi pembelajaran. Algoritma yang umum digunakan termasuk *self-training*, *graph-based semi-supervised learning*, dan *semi-supervised support vector machines* (S3VM).

4. *Reinforcement Learning*

Metode memperoleh penghargaan dengan berinteraksi dengan lingkungan, menilai kualitas tindakan berdasarkan tingkat penghargaan, dan kemudian melatih model. Dasar dari *reinforcement learning* berasal dari psikologi perilaku. Algoritma ini melatih model melalui mekanisme *trial and error* untuk menemukan tindakan dan perilaku terbaik yang memberikan hasil terbesar.

2.3 *Deep Learning*

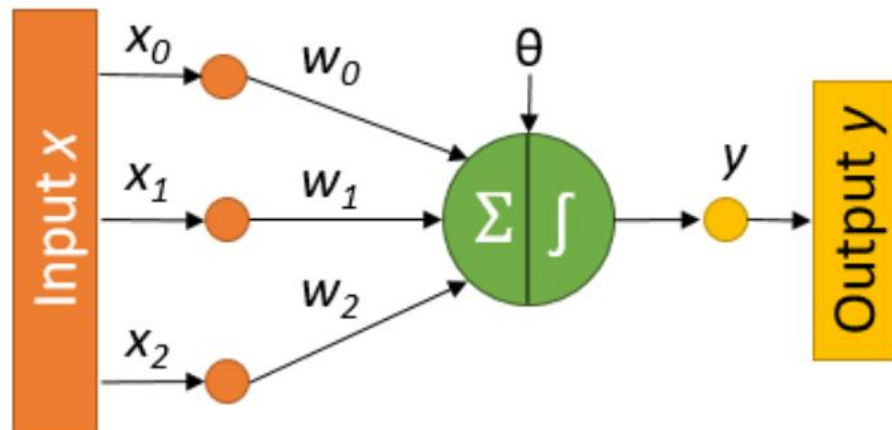
Deep learning adalah cabang dari *machine learning* yang menggunakan jaringan saraf buatan yang terdiri dari beberapa lapisan (*multilayer*). *Deep learning* memungkinkan komputer untuk belajar dari *big data* dan melakukan prediksi atau keputusan berdasarkan data tersebut. Algoritma *deep learning* yang umum mencakup *Restricted Boltzmann Machine* (RBM), *deep belief network* (DBN), *convolutional neural network* (CNN), dan *stacked auto-encoder* (SAE). Teknik ini menggali karakteristik distribusi data dengan menggabungkan fitur tingkat rendah untuk membentuk kategori atribut atau fitur tingkat tinggi (Zhang & Lu, 2021).

2.4. *Neural Network*

Artificial intelligence generasi pertama didasarkan pada simbol, yang meyakini bahwa informasi sensorik diekspresikan dengan cara pengkodean tertentu. *Artificial intelligence* generasi kedua membangun koneksi stimulus-respon dalam jaringan saraf. Berbagai arsitektur jaringan saraf mendukung beragam aplikasi pemahaman persepsi. Saat ini, penelitian tentang jaringan saraf dalam

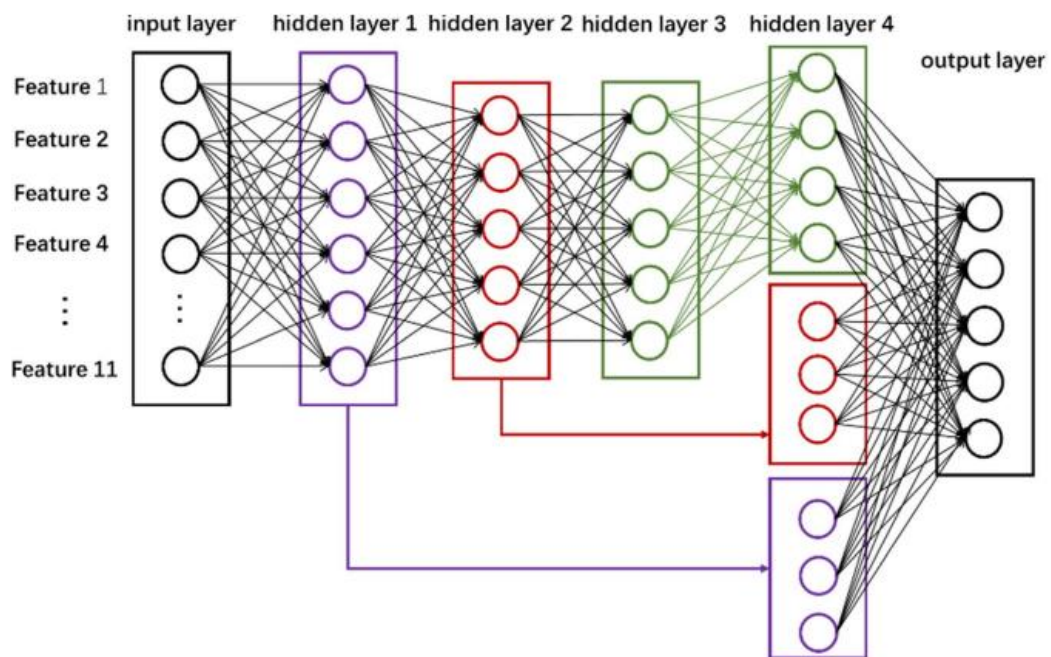
pemrosesan bahasa alami, pemrosesan data visual, pemrosesan sinyal ucapan, dan lain-lain mengalami kemajuan pesat. Telah banyak digunakan di bidang industri seperti keamanan cerdas, kesehatan medis, dan inspeksi industri. Tugas pengenalan berbasis jaringan saraf melibatkan ekstraksi fitur dari konten masukan model dan pembentukan hubungan pemetaan antara fitur yang diekstraksi dan atribut sampel yang dapat diidentifikasi (kategori, lokasi, dan lain-lain) (Li & Wang, 2021).

Neural network adalah kelompok unit yang saling berhubungan yang disebut neuron yang disusun dalam lapisan. Lapisan pertama disebut lapisan masukan, dan biasanya mempunyai dimensi yang sama dengan masukan. Lapisan terakhir disebut lapisan keluaran. Keluaran *Neural network* dapat berupa prediksi label kelas, gambar, teks, dan lain-lain. Setiap neuron di lapisan perantara diberikan sejumlah masukan. Menghitung fungsi non-linier pada jumlah *input* tertimbang. Keluaran yang dihasilkan dapat dimasukkan ke sejumlah neuron pada lapisan berikutnya. Fungsi non-linier yang terkait dengan neuron disebut fungsi aktivasi. Contoh umum fungsi aktivasi antara lain tan hiperbolik (\tanh), sigmoid, *Rectified Linear Unit* (ReLU), dan Leaky ReLU. Ada dua langkah penting dalam merancang *Neural network* untuk tugas tertentu. Langkah pertama berhubungan dengan desain arsitektur jaringan, ini menentukan jumlah lapisan, konektivitas, dan jenis neuron. Langkah kedua memparametrikkan bobot pada tepi jaringan saraf menggunakan prosedur pengoptimalan yang sesuai untuk mencocokkan distribusi keluaran dengan distribusi target. Istilah pembelajaran mendalam digunakan untuk menggambarkan rangkaian *mod* pembelajaran mesin, model dan metode yang arsitekturnya menggunakan *Neural network* sebagai komponen inti. Kata "dalam" sesuai dengan fakta bahwa algoritma pembelajaran biasanya menggunakan model *Neural network* dengan banyak lapisan, berbeda dengan jaringan dangkal yang biasanya memiliki satu atau dua lapisan perantara (atau tersembunyi) (Swaminathan & Parente, 2023).



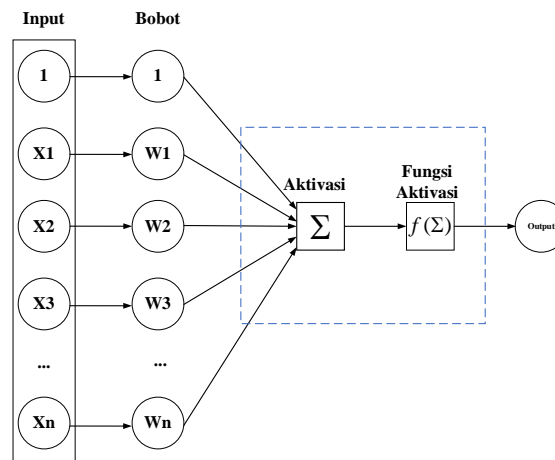
Gambar 2.2 *Arsitektur Neural Network*
(Sumber: Burgueni, Cabot, Li, & Gerard, 2022)

Selanjutnya adalah kelanjutan dari arsitektur *neural network* di mana terdapat banyak lapisan yang tersusun sehingga terdapat arsitektur *Deep Neural Network Multi-Level* yang terdapat lebih dari 2 lapisan yang tersusun untuk mereduksi dimensi.



Gambar 2.3 *Arsitektur Deep Neural Network Multi-Level*
(Sumber: Yang & Jiang, 2021)

Selanjutnya adalah pemodelan dari arsitektur *neural network* di mana terdapat fungsi aktivasi di dalam nya untuk memperkenalkan non-linearitas ke dalam model, yang memungkinkan jaringan untuk belajar dan mewakili hubungan yang lebih kompleks dalam data.



Gambar 2.4 Arsitektur Umum Neural Network dengan Fungsi Aktivasi
(Sumber: Junaidi, Beno, Farkhan, Sapartha, Pasaribu, Kmurawak, Supiyanto, Sroyer, Reba, Fitriyanto, Syafiqoh, & Rizal, 2024)

Neural network dapat dianggap lebih luas dan umum sebagai grafik komputer yang terdiri dari proses matematika. Ada dua properti utama yang membedakan jaringan saraf (Junaidi, Beno, Farkhan, Sapartha, Pasaribu, Kmurawak, Supiyanto, Sroyer, Reba, Fitriyanto, Syafiqoh, & Rizal, 2024):

1. Arsitektur *neural network*: menjelaskan kelompok koneksi, yang mungkin berupa *single* sampai *multi layer*, *feedforward*, *recurrent* atau keduanya.
2. *Training*: menjelaskan definisi khas pelatihan. *gradient descent* dan *backpropagation* adalah dua metode yang paling sering digunakan untuk melatih jaringan saraf.

2.4.1 Segmentasi Neural Network

Sementasi berbasis jaringan saraf paling sering ditemukan dalam tugas visi. Segmentasi citra berbeda dengan klasifikasi dan pemantauan citra. Tugas klasifikasi citra adalah mengidentifikasi isi suatu gambar, sedangkan tugas pemantauan citra adalah mengidentifikasi isi suatu gambar dan juga memantau lokasinya. Segmentasi gambar adalah tugas klasifikasi gambar tingkat piksel berdasarkan pengklasifikasiannya setiap piksel suatu gambar (Li & Wang, 2021).

2.4.2 Prediksi Neural Network

Memperkenalkan penerapan jaringan saraf untuk masalah prediksi. Bagian regresi di mana keterkaitan antara variabel independen dan dependen diprediksi

menggunakan regresi. Pembelajaran masalah regresi mirip dengan pembelajaran penyesuaian fungsi dengan memilih kurva fungsi yang sesuai dengan data yang diketahui dan memprediksi secara akurat data yang tidak diketahui. Selanjutnya prediksi deret waktu adalah barisan angka-angka yang disusun dalam urutan tertentu, dan urutan ini biasanya ditentukan oleh waktu ini merupakan sarana penting bagi manusia untuk memahami dunia objektif dan fenomena alam. Selanjutnya prediksi sinyal satu dimensi jaringan saraf juga banyak digunakan dalam beberapa tahun terakhir, beberapa peneliti telah menemukan bahwa sinyal osilasi frekuensi tinggi memiliki korelasi tertentu dengan penyakit tertentu, yang dapat membantu meningkatkan keakuratan lokasi lesi dan meningkatkan tingkat keberhasilan operasi klinis. Selanjutnya prediksi video dua dimensi di mana teknologi prediksi video memprediksi *frame* video berikutnya ketika beberapa panjang *frame* video kontinu disediakan dan ini adalah topik penting dalam bidang visi komputer dan memiliki prospek penerapan yang signifikan (Li & Wang, 2021).

2.4.3 Desain Eksperimen *Neural Network*

Bidang *artificial intelligence*, variabel-variabel ini disebut sebagai fitur masukan. Seseorang tentu saja dapat melatih *neural network* untuk data *input-output* yang tidak terkait, namun hal ini sering kali mengakibatkan *neural network overfitted*. Model *neural network* diperkirakan dapat berhasil melakukan ekstrapolasi ke data yang tidak diketahui asalkan terdapat hubungan *input-output* yang jelas yang dapat dipelajari oleh *neural network*. Lebih jauh lagi, dalam praktik optimasi, memilih kombinasi fitur terbaik mungkin terkadang bersifat paradoks. Akibatnya, fitur sering kali dipilih secara sembarangan dalam proses yang disebut rekayasa fitur. Ketersediaan data historis juga mempengaruhi pemilihan fitur (Li & Wang, 2021).

2.4.4 Kelebihan *Neural Network*

Selama empat tahun terakhir, *neural network* telah mendapatkan popularitas karena banyaknya pengaplikasian nya, yang mencakup data, masalah data yang rumit, kekuatan komputasi, dan penelitian akademis. Pendorong utama tren ini

adalah (Saleh, 2020):

1. Data

Database yang sangat besar sekarang dapat dikumpulkan dan disimpan karena kemajuan dalam perangkat keras dan perangkat lunak. *Neural network* terkenal karena kapasitasnya dalam menggunakan data dalam jumlah besar. Semakin banyak data yang dimasukkan ke dalam jaringan saraf, hal ini memungkinkan jaringan tersebut menunjukkan potensi penuhnya.

2. Masalah data yang kompleks

Menangani tantangan data rumit yang tidak dapat ditangani oleh Algoritma pembelajaran mesin lainnya, jaringan saraf sangat unggul. Hal ini sebagian besar disebabkan oleh kapasitas untuk menangani volume besar kumpulan data dan mengungkap pola yang kompleks.

3. Kekuatan komputasi

Kekuatan komputasi yang tersedia saat ini juga meningkat karena terobosan teknologi, dan ini penting untuk melatih model jaringan saraf yang membutuhkan jutaan titik data.

4. Penelitian akademis

Ketiga faktor yang disebutkan di atas telah membuat semakin banyak studi ilmiah mengenai subjek ini dapat diakses secara *online*, yang tidak hanya memungkinkan penelitian baru setiap hari tetapi juga membantu menjaga algoritma dan persyaratan perangkat keras/perangkat lunak tetap terkini.

2.4.5 Kekurangan *Neural Network*

Meskipun ada banyak manfaat menggunakan *neural network*, tidak semua masalah data harus diselesaikan dengan cara ini. Tidak ada satu algoritma pun yang dapat menyelesaikan setiap masalah data, pilihan metode yang digunakan harus mempertimbangkan masalah data dan sumber daya yang tersedia. Adapun kekurangannya yang akan dijelaskan sebagai berikut (Saleh, 2020):

1. Persyaratan data

Beberapa tantangan data, faktor utama yang membedakan teknik ini dari Algoritma alternatif mungkin adalah banyaknya jumlah data yang diperlukan

untuk mencapai hal ini. Ketika pekerjaan yang dilakukan memerlukan pengawasan dan pelabelan data, hal ini menjadi masalah yang lebih signifikan. Hasil yang optimal mungkin mempunyai kelebihan dan kekurangan. Lebih banyak data diperlukan untuk *neural network* dibandingkan algoritma pembelajaran mesin konvensional.

2. Waktu pelatihan

Selain kelemahan yang disebutkan di atas, pelatihan kumpulan data berukuran besar membutuhkan waktu lebih lama dibandingkan dengan algoritma pembelajaran mesin pada umumnya di mana sebuah prosedur yang mungkin tidak dapat dilakukan dalam keadaan tertentu. Penggunaan GPU dapat mempersingkat waktu pelatihan dengan meningkatkan kinerja komputasi.

3. Mahal secara komputasi

Sekali lagi, proses pelatihan *neural network* memerlukan anggaran komputasi yang besar. Metode pembelajaran mesin tertentu dilatih dalam hitungan jam atau menit, namun satu jaringan saraf mungkin memerlukan waktu berminggu-minggu untuk terbentuk sepenuhnya. *Neural network* yang lebih dalam memerlukan lebih banyak waktu untuk dilatih dibandingkan jaringan neural yang lebih dangkal, dan kedua faktor tersebut memengaruhi seberapa besar daya komputasi yang diperlukan.

2.4.6. Fungsi Aktivasi

Fungsi aktivasi dalam *Convolutional Neural Network* (CNN) adalah komponen penting yang menambahkan non-linearitas pada keluaran neuron. Ini memungkinkan jaringan untuk mempelajari dan mewakili pola data yang kompleks. Tanpa fungsi aktivasi, jaringan saraf akan menjadi model linear sederhana yang tidak mampu memecahkan masalah non-linear. Suatu neuron harus diaktifkan atau tidak yang berlandaskan input yang diterimanya merupakan proses dari fungsi aktivasi (wang, Li, Song, & Rong, 2020).

2.4.6.1 Rectified Linear Unit (ReLU)

Rectified Linear Unit (ReLU) telah menjadi salah satu aktivasi *default*

fungsi untuk banyak jaringan saraf, terutama *Convolutional Neural Network* (CNN) karena model dengan *Rectified Linear Unit* (ReLU) dilatih lebih cepat dan dapat memberikan kinerja yang lebih tinggi. *Rectified Linear Unit* (ReLU) berfungsi untuk menentukan apakah neuron harus diaktifkan atau tidak, berdasarkan *input* yang diterimanya. Berikut Fungsi Aktivasi Rectified Linear Unit (ReLU) dirumuskan (Job, Bhateja, Gupta, Bingi, & Prusty, 2022):

$$R(z) = \max(0, z) \quad (2.1)$$

atau

$$R(z) = \begin{cases} 0 & \text{untuk } z \leq 0 \\ z & \text{untuk } z > 0 \end{cases} \quad (2.2)$$

2.4.6.2 Sigmoid Function

Sigmoid Function, kadang-kadang disebut sebagai fungsi logistik, memiliki beberapa ciri yang patut diperhatikan, antara lain perilaku yang berkesinambungan (dihaluskan), berkembang secara monoton, dan dibatasi. Hal ini membantu dalam menciptakan rangkaian pengujian yang andal untuk menilai kinerja model, menyelidiki penyempurnaan model, dan membuat prediksi menggunakan data baru. Ketika fungsi hasil mengambil nilai antara 0 dan 1, fungsi sigmoid masuk akal. Jaringan saraf tiruan menggunakan fungsi ini untuk masalah kategorisasi. *Machine Learning Foundation* memanfaatkan fungsi ini secara ekstensif, sebagian besar untuk regresi logistik dan beberapa aplikasi lainnya. Berikut merupakan rumus yang digunakan dalam *sigmoid function* (Mercioni & Holban, 2020).

$$f(x) = \text{sigmoid}(x) = \frac{1}{1 + e^{-x}} \quad (2.3)$$

2.4.6.3 Tangent Function (tanh)

Muncul sebagai solusi untuk sigmoid negatif ketika hasilnya negatif dan sigmoid menghasilkan nilai yang hampir sama dengan 0. Hal ini menunjukkan bahwa akan sulit untuk membuat lapisan terkubur menjadi simpul "mati" dengan mengubah bobot yang melekat padanya. Mirip dengan sigmoid, fungsi tanh memiliki rentang -1 hingga 1. Nilai Tanh akan mendekati -1 jika x adalah bilangan bulat negatif yang sangat besar dan mendekati 1 jika x adalah bilangan positif yang

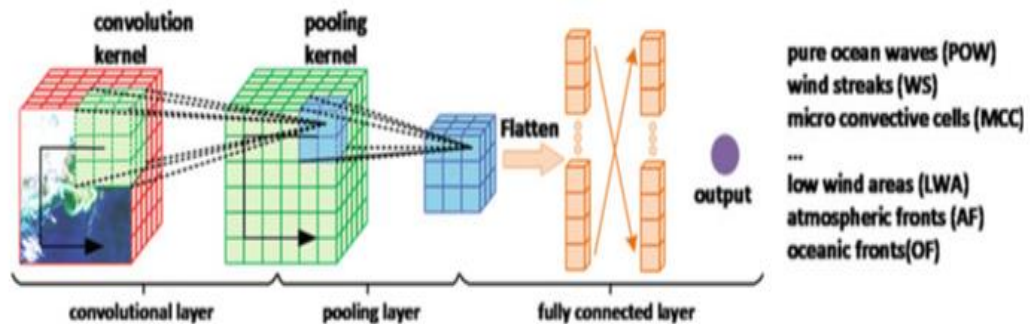
sangat besar. Berikut merupakan rumus yang digunakan dalam *tangent function* (Mercioni & Holban, 2020).

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (2.4)$$

2.5. *Convolutional Neural Network* (CNN)

Convolutional Neural Network (CNN) adalah jaringan neural yang menggunakan konvolusi untuk mengukur kecocokan pola lokal. *Convolutional Neural Network* (CNN) adalah jaringan *feed-forward* dengan satu atau lebih lapisan konvolusi. *Convolutional Neural Network* (CNN) digunakan secara luas dalam analisis gambar, dan baru-baru ini, grafik yang memodelkan struktur yang terhubung seperti molekul. *Convolutional Neural Network* (CNN) memiliki lapisan masukan, lapisan tersembunyi, dan lapisan keluaran. Masukan ke CNN adalah tensor dalam bentuk *input x input height x input width x input channels*. Tinggi dan parameter lebar sesuai dengan ukuran gambar asli. Jumlah saluran masukan biasanya tiga (merah, hijau, dan biru) untuk gambar. Masing-masing lapisan tersembunyi dapat menjadi salah satu dari (i) *convolutional layer*, (ii) *pooling layer* (iii) *fully connected layer*. *Convolutional layer* mengambil masukan berupa gambar, atau keluaran dari lapisan lain, dan mengeluarkan peta fitur (Swaminathan & Parente, 2023).

Keuntungan terbesar *Convolutional Neural Network* (CNN) adalah pada pengurangan parameter. Misalnya, gambar dalam format jpg dengan resolusi 480 x 480 direpresentasikan di komputer sebagai tensor 480 x 480 x 3, dan tiga dimensi tersebut sesuai dengan tinggi, lebar, dan jumlah saluran tensor 3D. *Convolutional Neural Network* (CNN) mendapatkan namanya dari operasi linier matematis antar matriks yang disebut konvolusi. *Convolutional Neural Network* (CNN) merupakan metode pembelajaran representasi dengan struktur multilayer, yang sebagian besar terdiri dari *convolutional layer*, *pooling layer*, dan *fully connected layer*. *Convolutional layer* dan *fully connected layer* berisi parameter sedangkan *pooling layer* tidak seperti pada gambar 2.5 (Li & Wang, 2021).



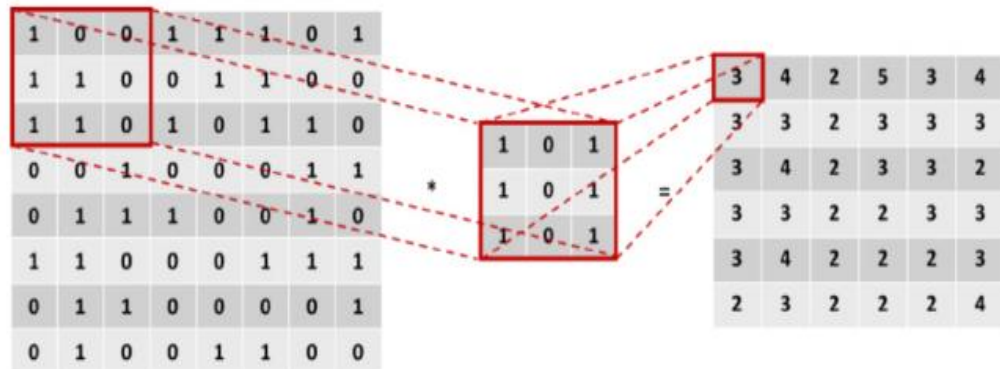
Gambar 2.5 Arsitektur *Convolutional Neural Network*
(Sumber: Li & Wang, 2021)

Seperti yang ditunjukkan pada Gambar 2.2, gambar dimasukkan ke *Convolutional Neural Network* (CNN) dan kemudian melewati *convolutional layer* dan *pooling layer* secara bergantian, yang meratakan fitur gambar menjadi vektor fitur berdimensi satu. *Convolutional Neural Network* (CNN) akhirnya mengeluarkan hasilnya melalui lapisan yang terhubung sepenuhnya. *Convolutional layer* dan penyatuan setara dengan struktur ekstraksi fitur, yang digunakan untuk mengekstraksi fitur dari tensor *input*. *Fully connected layer* setara dengan pengklasifikasi, yang digunakan untuk mengklasifikasikan vektor fitur yang diratakan.

Melanjutkan berdasarkan Gambar 2.5 maka dapat dijelaskan lebih lanjut mengenai *convolutional layer*, *pooling layer*, dan *fully connected layer* sebagai berikut (Saleh, 2020):

1. *Convolutional Layers*

Antara gambar (diwakili sebagai matriks piksel) dan filter, komputasi *convolutional* dilakukan pada lapisan ini. Hasil komputasi ini berupa peta fitur yang kemudian digunakan sebagai masukan untuk lapisan berikutnya. Perhitungannya mengalikan nilai subbagian matriks gambar yang memiliki bentuk filter yang sama. *Output* untuk area gambar tersebut kemudian diatur ke jumlah produk, contoh lebih lanjutnya akan dicontohkan seperti yang diilustrasikan oleh Gambar 2.6 Operasi Konvolusi Antara Gambar dan Filter di bawah ini:



Gambar 2.6 Operasi Konvolusi Antara Gambar dan Filter
(Sumber: Saleh, 2020)

Berdasarkan Gambar 2.6 Operasi Konvolusi Antara Gambar dan Filter, data masukan diwakili oleh matriks di sebelah kiri, filter oleh matriks tengah, dan hasil komputasi oleh matriks kanan. Berikut merupakan perhitungan yang dilakukan menggunakan nilai pada kotak yang disorot:

$$1 * 1 + 1 * 1 + 1 * 1 + 0 * 0 + 1 * 0 + 1 * 0 + 0 * 1 + 0 * 1 + 0 * 1 = 3$$

Gambar 2.7 Konvolusi Bagian Pertama Gambar
(Sumber: Saleh, 2020)

2. Pooling Layers

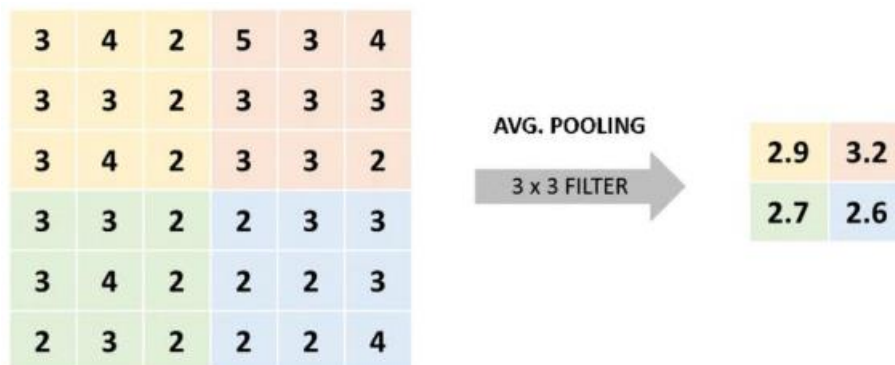
Convolutional layer dapat mengekstraksi informasi penting dari gambar, namun saat mempelajari bentuk geometris yang rumit, hasilnya mungkin luar biasa. Hal ini membuat proses pelatihan menjadi sulit secara komputasi, itulah sebabnya *pooling layer* dibuat. Lapisan ini tidak hanya mencapai tujuan *convolutional layer* untuk mengurangi keluaran, namun juga menghilangkan gangguan apa pun dari fitur yang diambil, sehingga menambah akurasi model dalam prosesnya. Tujuan di balik dua jenis lapisan penggabungan utama yang dapat diterapkan adalah untuk mengidentifikasi wilayah yang memiliki dampak lebih besar pada gambar sehingga daerah lain bisa terabaikan. Penggabungan maksimal adalah operasi yang melibatkan pemilihan subbagian matriks dengan ukuran tertentu dan menggunakan jumlah terbesar subbagian tersebut sebagai hasil operasi.



Gambar 2.8 Operasi *Pooling* Maksimal
(Sumber: Saleh, 2020)

Berdasarkan Gambar 2.8, filter *pooling* maksimum 3 x 3 digunakan untuk hasil ditunjukkan di sebelah kanan. Batasannya ada empat di bagian kuning (pojok kiri atas) dan maksimal lima di bagian oranye (pojok kanan atas).

Penggabungan rata-rata adalah *output* yang memenuhi aturan diperoleh dengan operasi penggabungan rata-rata dari subbagian matriks. *Output* nya adalah rata-rata semua angka di subbagian yang relevan:



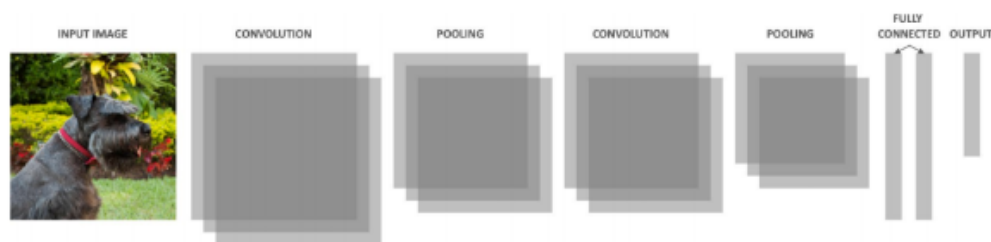
Gambar 2.9 Operasi *Pooling* Rata-Rata
(Sumber: Saleh, 2020)

Rata-rata seluruh angka yang ada di bagian kuning (pojok kiri atas) adalah 2,9 dalam hal ini diperoleh dengan menggunakan filter 3 x 3. Rata-rata angka pada sektor oranye (pojok kanan atas) adalah 3,2.

3. *Fully Connected Layers*

Terakhir, *fully Connected layer* digunakan di akhir *Convolutional Neural Network* (CNN) untuk mengambil fitur-fitur yang terdeteksi oleh lapisan sebelumnya (dikenal sebagai peta fitur) dan menampilkan probabilitas pengelompokan fitur-fitur tersebut ke dalam label kelas, yang digunakan untuk menghasilkan keputusan akhir. Hal ini karena jaringan tidak akan berguna jika

hanya dapat mendeteksi sekumpulan fitur tanpa memiliki kemampuan untuk mengklasifikasikannya ke dalam label kelas. *Fully Connected layer*, seperti ANN dengan menghitung keluaran dari masukan menggunakan perceptron. Penting juga untuk dicatat bahwa *Convolutional Neural Network* (CNN) biasanya menyertakan beberapa lapisan yang sepenuhnya terhubung di bagian atas arsitekturnya. Semua ide ini dapat digabungkan untuk menciptakan arsitektur *Convolutional Neural Network* (CNN) standar. Jumlah setiap jenis lapisan bisa jadi tidak terbatas dan jumlah filter yang tidak terbatas (masing-masing untuk tugas tertentu) di setiap *convolutional layer*. Lebih jauh lagi, seperti yang diilustrasikan gambar berikut, lapisan penggabungan harus memiliki jumlah filter yang sama dengan lapisan konvolusional sebelumnya.

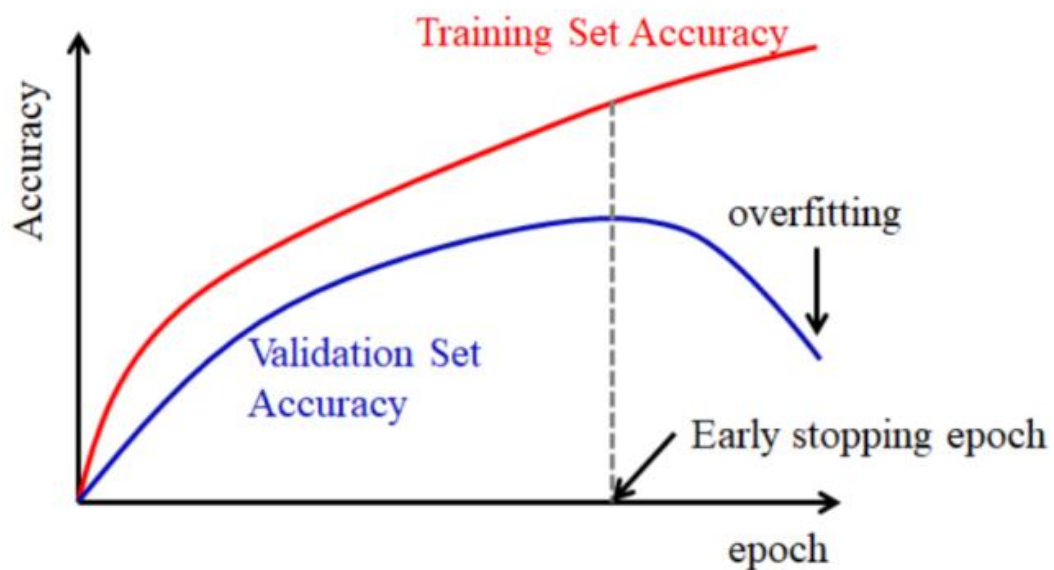


Gambar 2.10 Diagram Arsitektur *Convolutional Neural Network* (CNN)
(Sumber: Saleh, 2020)

2.5.1. Kelompok *Convolutional Neural Network* (CNN)

Lapisan *Convolutional Neural Network* (CNN) dibagi menjadi dua kelompok yaitu lapisan ekstraksi fitur dan lapisan klasifikasi, bergantung pada fungsinya masing-masing. Lapisan ekstraksi fitur adalah lapisan yang bertugas mengambil fitur dari gambar masukan. Karakteristik yang dihasilkan pada lapisan ekstraksi fitur digunakan oleh lapisan klasifikasi untuk mengkategorikan citra. Teori dan aplikasi *Convolutional Neural Network* (CNN) digunakan dalam pembelajaran mendalam untuk kategorisasi gambar, memanfaatkan fitur yang dihasilkan pada lapisan ekstraksi fitur. Lapisan ekstraksi fitur terdiri dari *convolutional layer*, *pooling layer*, dan *fully Connected layer* dengan fungsi aktivasi. Lapisan klasifikasi terdiri dari lapisan tertaut penuh, *dropout*, dan *softmax*. Pengoptimalan *Convolutional Neural Network* (CNN) menggunakan aliran propagasi mundur, yang mencakup lintasan maju dan mundur. Selama *forward pass*, *Convolutional Neural Network* (CNN)

menghasilkan fungsi kerugian atau kesalahan. Fungsi kerugian adalah perbedaan antara target dan prediksi, untuk meminimalkan fungsi kerugian, diterapkan teknik optimasi yang disebut penurunan gradien. Umpan balik *backward* digunakan untuk meningkatkan fungsi kerugian. Proses ini dilanjutkan hingga periode maksimum tercapai atau *validation* data terlalu sesuai. Berikut merupakan Gambar 2.8 Kondisi *early stop* pada saat terjadi *overfitting* (Setiawan, 2020).



Gambar 2.11 Kondisi *Early Stop* pada Saat Terjadi *Overfitting*
(Sumber: Setiawan, 2020)

2.5.1.1 *Overfitting*

Model *Convolutional Neural Network* (CNN) memiliki kemungkinan yang sangat tinggi untuk menghasilkan data *overfitting* selama fase pelatihan karena banyaknya parameter yang terhubung secara rumit. Jika hal ini terjadi, kapasitas model untuk berperforma baik pada data yang diuji akan berkurang. Kurangnya *training* data adalah salah satu penyebab masalah *overfitting*, yang mengakibatkan distribusi yang dipelajari tidak mencerminkan distribusi yang mendasarinya secara akurat. Model pembelajaran mesin terlalu menghafal pola dari *training* data yang tersedia, sehingga kinerjanya menurun secara signifikan saat diuji dengan data baru yang tidak dilihat sebelumnya (Alzubaidi, Zhang, Humaidi, Dujaili, Duan, Shamma, Santamaria, Fadhel, Amidie, & Farhan, 2021).

2.5.1.2 *Epoch*

Sebuah epoch merujuk pada satu putaran lengkap dari seluruh dataset pelatihan melalui algoritma pembelajaran. Hal ini berarti model akan melihat setiap sampel dalam dataset sekali dalam satu epoch. *Epoch* sering digunakan untuk mengukur berapa kali model telah dilatih pada keseluruhan dataset. Proses pembelajaran mesin khususnya menggunakan jaringan saraf tiruan seperti CNN, sebuah epoch adalah periode di mana seluruh dataset pelatihan telah diproses oleh jaringan sekali penuh. Berarti bahwa selama satu *epoch*, model melihat dan memproses setiap sampel dalam dataset pelatihan satu kali (Seo, Back, Lee, Park, Kim, & Lee, 2020).

Epoch dan iterasi memiliki peran penting dalam proses pelatihan CNN. Epoch adalah siklus penuh dari dataset melalui algoritma pelatihan, sedangkan iterasi adalah proses pembaruan bobot berdasarkan *batch* data yang lebih kecil atau *update* dalam proses pelatihan model. Iterasi terjadi setiap kali model memproses satu *batch* dari data pelatihan. Model CNN biasanya memerlukan banyak *epoch* untuk konvergen ke solusi yang optimal, dan dalam setiap epoch terdapat beberapa iterasi tergantung pada ukuran batch yang digunakan (Habib, & Qureshi, 2020).

2.5.1.3 *Learning Rate (Optimizer)*

Learning rate adalah hyperparameter yang mengatur seberapa besar bobot jaringan harus disesuaikan sehubungan dengan gradien kerugian. Mengingat sekarang sedang mengoptimalkan jaringan saraf yang baru dibangun dengan gradien yang menurun. Langkah yang besar dapat memaksa model untuk melompat ke titik minimum atau optimal yang diinginkan jika *learning rate* terlalu tinggi. Pelatihan akan berjalan lambat atau model tidak dapat menyatu sama sekali jika *learning rate* terlalu rendah. Pembaruan yang dilakukan oleh *optimizer* ini adalah bagian *learning rate* dan juga parameter (bobot dan bias) dari setiap neuron di lapisan *Convolutional Neural Network* (CNN). Beberapa algoritma optimasi dalam *learning rate* sebagai berikut (Yaqub, Feng, Zia, Arshid, Jia, Rehman, & Mehmood, 2020):

1. *Adaptive Momentum* (Adam)

Setiap parameter dalam pelatihan gradien, kecepatan pembelajaran adaptif diperkirakan menggunakan teknik *Adaptive Momentum* (Adam). Merupakan metode yang sangat mendasar dan efisien secara komputasi untuk optimasi stokastik yang menangani gradien orde pertama dengan kebutuhan memori rendah. Metode yang disarankan diterapkan pada masalah pembelajaran mesin yang melibatkan ruang parameter berdimensi tinggi dan kumpulan data yang cukup besar di mana kecepatan pembelajaran dihitung secara individual untuk parameter berbeda menggunakan perkiraan yang menggabungkan momen orde pertama dan kedua.

2. *Stochastic Gradient Descent* (SGD)

Keterkaitan *deep learning*, *Stochastic Gradient Descent* (SGD) sering digunakan. Algoritma ini terdiri dari proses yang terdefinisi dengan baik yang mengambil masukan dan menghasilkan hasil yang tepat. *Relaps strategies*, *linear regression*, dan *neural network* adalah algoritma terbaik berdasarkan biaya pekerjaan dan target kerja. Tujuan utama *Stochastic Gradient Descent* (SGD) adalah menekan biaya kerja. Penurunan gradien adalah sistem komputer yang sangat canggih jika terdapat set persiapan yang besar.

3. *Adaptive Gradien* (Adagrad)

Adagrad memodifikasi kecepatan pembelajaran berdasarkan parameter, mengeksekusi pembaruan yang lebih besar untuk parameter yang tidak konsisten dan lebih sedikit pembaruan untuk parameter berikutnya.

4. *Adaptive Delta* (AdaDelta)

AdaDelta adalah tambahan AdaGrad yang memecahkan masalah penurunan kecepatan pembelajaran, di mana tidak mengumpulkan gradien kuadrat masa lalu seperti yang dilakukan AdaGrad. Menutup jendela gradien yang dikumpulkan sebelumnya. Dibandingkan dengan metode sebelumnya, teknik AdaDelta hanya menggunakan waktu orde pertama untuk mengubah bobot secara signifikan dan memerlukan sumber daya pemrosesan yang paling sedikit.

5. *Adaptive Max Pooling* (Adamax)

Adamax mengambil inspirasi dari Adam dan memodifikasi cara penerapan

norma tak terhingga (ut). Lapisan pengelompokan *Convolutional Neural Network* (CNN) untuk ekstraksi fitur dari peta fitur.

6. *Root Mean Square Propagation (RMSProp)*

Menggunakan rata-rata pergerakan gradien kuadrat, yang memanfaatkan ukuran penurunan gradien terbaru untuk normalisasi gradien, untuk mengatasi kecepatan pembelajaran Adagrad yang menurun tajam. Algoritma akan berkembang dalam arah horizontal dengan langkah-langkah yang lebih besar yang menyatu lebih cepat seiring dengan peningkatan kecepatan pembelajaran.

7. *Nestrov Accelerated Gradient (NAG)*

Optimalisasi akselerasi Nesterov ibarat bola yang menggelinding menuruni puncak yang tahu kapan harus melambat sebelum kemiringan bukit naik lagi. Dapat menghitung gradien terhadap langkah berikutnya, bukan langkah yang sedang dijalankannya. menghitung gradien yang bertambah dan memperbarui bobotnya sesuai dengan signifikansinya. Performanya sedikit lebih baik dibandingkan momentum biasa karena dapat mengoptimalkan penurunan lebih cepat sambil menuruni puncak sehingga dapat melihat ke masa depan.

2.5.2 Kelebihan *Convolutional Neural Network* (CNN)

Convolutional Neural Networks (CNN) dapat secara otomatis mempelajari hierarki fitur dari gambar input tanpa perlu ekstraksi fitur manual oleh manusia. Sangat membantu dalam pengenalan pola yang kompleks seperti wajah dan objek. *Convolutional Neural Networks* (CNN) sangat akurat dalam mendeteksi dan mengklasifikasikan objek, serta dapat mengenali pola dan fitur yang rumit dari data *input*. Membuatnya sangat efektif dalam tugas pengenalan objek dan visi komputer. *Convolutional Neural Networks* (CNN) memiliki kapasitas yang lebih besar untuk mengekstraksi fitur dari gambar yang dapat tahan terhadap distorsi dan translasi. Ini berarti *Convolutional Neural Networks* (CNN) dapat mengenali objek bahkan jika posisinya berubah atau terdapat distorsi dalam gambar. *Convolutional Neural Networks* (CNN) banyak digunakan dalam berbagai aplikasi seperti pengenalan wajah, analisis citra medis, pemrosesan gambar, dan keamanan siber. Dalam analisis citra medis, *Convolutional Neural Networks* (CNN) dapat mendeteksi

tumor dan kelainan lainnya dengan akurasi tinggi pada gambar X-ray dan MRI. Salah satu keuntungan utama *Convolutional Neural Networks* (CNN) adalah pengurangan jumlah parameter yang signifikan dibandingkan dengan jaringan syaraf tiruan tradisional. Sehingga membantu dalam mengurangi waktu pelatihan dan kebutuhan komputasi, membuatnya lebih efisien (Taye, 2023).

2.5.3 Kekurangan *Convolutional Neural Network* (CNN)

Kekurangan *Convolutional Neural Networks* (CNN) memerlukan daya komputasi yang besar, terutama untuk jaringan yang dalam dan *dataset* besar. Ini karena jumlah besar parameter yang perlu dipelajari dan dihitung selama proses pelatihan dan inferensi. Semakin dalam jaringan, semakin lama waktu yang diperlukan untuk melatih model tersebut. Penambahan lapisan konvolusi dan lapisan koneksi penuh meningkatkan waktu yang dibutuhkan untuk pelatihan. CNN memerlukan sejumlah besar data berlabel untuk pelatihan agar dapat mencapai kinerja yang tinggi. Kurangnya data berlabel dapat mengurangi kemampuan generalisasi model dan menyebabkan *overfitting*. Karena jumlah parameter yang besar, *Convolutional Neural Networks* (CNN) rentan terhadap *overfitting*, terutama ketika *dataset* pelatihan tidak cukup besar untuk mendukung kompleksitas model. Teknik seperti regularisasi dan augmentasi data sering digunakan untuk mengatasi masalah ini. (Taye, 2023).

2.5.4. *Backpropagation*

Backpropagation adalah algoritma yang paling sering digunakan untuk melatih jaringan saraf tiruan yang dalam. Algoritma ini sangat sukses dalam berbagai aplikasi pembelajaran mesin, termasuk pengenalan gambar, pengenalan suara, dan terjemahan bahasa. *Backpropagation* menghitung bagaimana perubahan kecil pada setiap kekuatan sinaps akan mengubah kesalahan jaringan menggunakan aturan rantai kalkulus. Algoritma ini melakukan perhitungan ini untuk semua kekuatan sinaps secara bersamaan dan hanya memerlukan jumlah komputasi yang sama dengan yang diperlukan untuk satu pass propagasi maju melalui jaringan. Teknik pelatihan ini dikenal dengan sebutan *backpropagation* karena apabila

jaringan *output* berbeda dengan *output* yang diprediksi, maka *output* tersebut akan *backward* pada *hidden layer* ke unit-unit pada lapisan *input* (Lillicrap, Santoro, Marris, Akerman, & Hinton, 2020).

2.5.4.1 Forward Pass

Forward pass adalah metode dalam pelatihan *neural network* yang mentransfer data input dari setiap neuron pada lapisan tersembunyi hingga ke lapisan keluaran. Selama *forward pass*, aktivasi dari setiap lapisan dihitung hingga mencapai lapisan *output* di mana kesalahan dihitung. *Forward pass* bertanggung jawab untuk menghitung aktivasi dari lapisan-lapisan jaringan dan menghasilkan prediksi berdasarkan *input* yang diberikan (Akintoye, Han, Lloyd, Zhang, Dancey, Chen, & Zhang, 2023).

2.5.4.2 Backward Pass

Backward pass adalah proses di mana kesalahan yang dihitung pada lapisan *output* selama *forward pass* digunakan untuk memperbarui bobot pada setiap neuron secara mundur dari lapisan keluaran kembali ke lapisan *input*. Selama *backward pass*, gradien dari fungsi *loss* dihitung untuk setiap parameter jaringan, dan parameter tersebut diperbarui menggunakan teknik optimisasi seperti *gradient descent*. *Backward pass* bertujuan untuk meminimalkan kesalahan yang dihitung dengan memperbarui bobot jaringan berdasarkan gradien kesalahan yang dihitung selama *forward pass* (Akintoye, Han, Lloyd, Zhang, Dancey, Chen, & Zhang, 2023).

2.6. Library Python

Library Python memainkan peran penting dalam pengembangan *artificial intelligence* dan *machine learning*. *Python* dikenal sebagai bahasa pemrograman yang paling disukai untuk komputasi ilmiah, *data science*, dan pembelajaran mesin karena kemampuannya dalam meningkatkan kinerja dan produktivitas dengan memungkinkan penggunaan pustaka tingkat rendah dan API tingkat tinggi yang bersih. *Python* menawarkan berbagai pustaka seperti NumPy untuk operasi aljabar

linier pada *array* multidimensional, *SciPy* untuk komputasi ilmiah, *Pandas* untuk analisis data, dan pustaka pembelajaran mesin seperti *Scikit-learn*, *TensorFlow*, dan *PyTorch*. *Python* dikenal karena sintaksisnya yang sederhana dan mudah dipelajari, membuatnya ideal untuk pemula sekaligus cukup kuat untuk pengguna yang berpengalaman. Pustaka seperti *Scikit-learn* digunakan untuk pembelajaran mesin klasik, sementara *TensorFlow* dan *PyTorch* banyak digunakan dalam pembelajaran mendalam (Raschka, Patterson, & Nolet, 2020).

2.6.1 Convolution Layer

Merupakan bagian awal dari arsitektur *Convolutional Neural Networks* (CNN) setelah input layer, yang terdiri dari kombinasi filter konvolusi (neuron). Setiap filter terkait dengan bagian kecil yang disebut *receptive field*. Lapisan konvolusi bekerja dengan membagi gambar input menjadi potongan-potongan lebih kecil (*receptive fields*) dan mengkonvolusinya dengan seperangkat bobot tertentu. Proses ini penting untuk ekstraksi fitur dalam *Convolutional Neural Networks* (CNN). *Convolution layer* memiliki beberapa jenis layer sebagai berikut (Saleem, Senan, Wahid, Aamir, Samad, & Khan, 2022):

1. Konvolusi Standar

Fungsinya menerapkan serangkaian filter ke *input* untuk menghasilkan peta fitur. Operasinya setiap filter meluncur ke seluruh masukan secara spasial, menghitung perkalian titik antara filter dan *patch* masukan.

2. Konvolusi yang Dapat Dipisahkan Secara Mendalam

Penggunaannya ditemukan pada model seperti *Xception* dan *MobileNet*. Operasinya metode ini membagi konvolusi menjadi dua lapisan, pertama konvolusi mendalam (menerapkan filter konvolusional tunggal per saluran masukan) dan kedua konvolusi titik (menerapkan konvolusi 1x1 untuk menggabungkan keluaran konvolusi mendalam). Hal ini mengurangi biaya komputasi secara signifikan dibandingkan dengan konvolusi standar.

3. Konvolusi Atrous (*Dilated*)

Penggunaannya dalam model seperti *DeepLab*. Operasinya memperkenalkan lubang (atau nol) di antara bobot filter, yang secara efektif memperbesar ukuran

filter tanpa menambah jumlah parameter. Berguna untuk tugas prediksi padat seperti segmentasi semantik.

4. Konvolusi yang Di kelompokkan

Penggunaannya dalam model seperti *ResNeXt*. Operasinya membagi saluran masukan menjadi beberapa kelompok dan melakukan operasi konvolusi dalam setiap kelompok secara terpisah, memungkinkan komputasi yang lebih efisien dan peningkatan kapasitas.

5. Konvolusi yang Ditransposisikan (*Deconvolutions*)

Penggunaannya dalam model generatif seperti GAN atau *autoencoder*. Operasinya merupakan kebalikan dari konvolusi normal, menyebarkan informasi masukan ke seluruh area spasial yang lebih besar untuk meningkatkan pengambilan sampel masukan.

2.6.2 Conv2D Layer

Conv2D adalah jenis *layer* konvolusi yang digunakan dalam *Convolutional Neural Networks* (CNN). *Layer* ini berfungsi untuk mengekstraksi fitur dari gambar *input* dengan menggunakan filter konvolusi yang melintasi seluruh gambar. Setiap filter menghasilkan peta fitur baru berdasarkan operasi konvolusi antara filter dan bagian-bagian gambar yang disebut bidang reseptif. *Input Image* nya adalah gambar dengan dimensi tinggi x lebar x filter. Hasil dari operasi konvolusi adalah peta fitur baru yang merepresentasikan fitur-fitur spesifik yang terdeteksi oleh filter (Alem, & Kumar, 2022).

2.6.3 Maxpooling2D Layer

Max-pooling layer digunakan untuk menyaring fitur-fitur hasil konvolusi dengan mengekstrak bagian yang paling aktif dari fitur-fitur lokal. Layer ini membantu dalam mengurangi redundansi fitur dan *overfitting* dengan hanya mempertahankan hasil konvolusi yang paling bernilai dan menghilangkan fitur-fitur yang tidak diperlukan dan hanya mempertahankan hasil konvolusi yang paling bernilai. Proses ini menghasilkan vektor fitur dengan dimensi tetap (Ma, tang, zhang, Cui, ji, Chen, & Abraham, 2023).

2.6.4 Flatten Layer

Tensor masukan multidimensi yang dapat direduksi menjadi satu dimensi dengan menggunakan *flatten layer*. Merestrukturisasi tensor sehingga bentuknya sesuai dengan jumlah komponen yang dikandungnya. Memasukkan data ke dalam lapisan berikutnya, ratakan menjadi *array* satu dimensi. Jaringan *neural* yang terhubung sepenuhnya dengan sejumlah lapisan tersembunyi yang bervariasi menggunakan keluaran *flatten* sebagai masukan untuk menentukan kompleksitas non-linier dari representasi fitur. Menyediakan semua parameter yang dapat dilatih langsung ke jaringan saraf yang terhubung sepenuhnya, lapisan datar kemudian dihubungkan ke lapisan yang terhubung sepenuhnya (Rane, Kanade, Parkhi, Anasune, suresh, Bande, baranjalekar, & Adhav, 2023).

2.6.5 Dense Layer

Lapisan di *Convolutional Neural Network* (CNN) yang paling umum dan sering digunakan. Setiap neuron pada *solid layer* menerima semua keluaran dari lapisan sebelumnya, dan setiap neuron kemudian mengirimkan satu keluaran ke lapisan di bawahnya. Jaringan saraf, ini adalah salah satu lapisan paling mendasar. Karena setiap neuron pada lapisan ini terhubung ke setiap neuron lain pada lapisan di atasnya, lapisan ini juga dikenal sebagai lapisan terhubung penuh. Setiap neuron masukan terhubung sepenuhnya ke setiap unit aktivasi dan berfungsi sebagai lapisan keluaran secara signifikan. Saat menyetelnya ke tidak ada, fungsi aktivasi *default* lapisan adalah aktivasi linier (Rane, Kanade, Parkhi, *et al.*, 2023).

2.6.6 Dropout

Teknik yang digunakan untuk mengurangi *overfitting* dan meningkatkan performa model selama proses pelatihan *neural network*. *Dropout* bekerja dengan menonaktifkan secara acak neuron-neuron tertentu dalam jaringan selama setiap langkah pelatihan. Ini mencegah neuron-neuron tersebut menjadi terlalu spesifik terhadap data pelatihan dan membantu jaringan untuk lebih umum terhadap data baru yang belum dilihat (Norhikmah, Lutfhi, & Rumini, 2022).

2.7 Klasifikasi dan Prediksi

Menemukan model yang menggambarkan sekelompok fakta atau konsep adalah proses klasifikasi. Tujuannya adalah untuk dapat memprediksi kelas objek yang label kelasnya tidak diketahui dengan menggunakan model ini. Kumpulan *training* data dianalisis untuk membuat model yang dihasilkan. Format berikut dapat digunakan untuk menampilkan model turunan mulai dari Aturan Klasifikasi (*if-then*), Pohon keputusan, Rumus matematika, dan Jaringan saraf tiruan. Daftar fungsi yang terlibat dalam proses ini adalah sebagai berikut (Yahya, 2022):

1. Klasifikasi

Asumsikan bahwa suatu objek memiliki label kelas yang tidak diketahui.. Tujuannya adalah untuk menemukan model turunan yang mendeskripsikan dan membedakan kelas atau konsep data. Model turunan didasarkan pada kumpulan analisis *training* data, yaitu objek data yang label kelasnya sudah dikenal.

2. Prediksi

Penggunaan sebagai pengganti label kelas untuk mengantisipasi nilai data numerik yang tidak ada atau tidak tersedia. Prediksi adalah penggunaan analisis regresi yang biasa. Berdasarkan fakta yang ada, prediksi juga dapat digunakan untuk menentukan tren distribusi.

2.8 Dataset

Mirip dengan struktur data yang ditemukan dalam *database*, *dataset* adalah objek yang menyimpan data dan hubungannya dalam memori. *Dataset* menampilkan data tabel dan data, dua kategori *dataset* menjadi sebuah pembeda, khususnya (Yahya, 2022):

1. Kumpulan data yang tidak dapat diakses publik

Data yang diperoleh atau diambil dari organisasi atau Perusahaan yang berfungsi sebagai lokasi atau subjek studi disebut sebagai kumpulan data pribadi.

2. Kumpulan data yang dapat diakses publik

Kumpulan data yang dapat diambil dari repositori publik, atau penyimpanan publik, dan yang telah diputuskan oleh para peneliti untuk tersedia untuk umum dikenal sebagai kumpulan data publik kepada masyarakat umum.

2.9 Data Preparation

Persiapan data diutamakan saat membuat model pembelajaran mendalam, tentu saja, setelah pengumpulan data. Hal ini diperlukan jika ingin menentukan cakupan proyek dengan benar menggunakan data yang tersedia saat ini. Hal ini sering kali diabaikan oleh banyak data *scientist*, sehingga menghasilkan model yang berperforma buruk atau bahkan tidak berguna karena tidak menangani masalah data sejak awal. Tiga tugas utama terdiri dari proses persiapan data (Saleh, 2020):

1. Memahami informasi dan menyelesaikan masalah yang mungkin terjadi.
2. Mengubah skala fitur untuk memastikan tidak terjadi bias yang tidak disengaja.
3. Membagi data agar kinerja dapat diukur secara tepat.

2.10 Data Augmentation

Merupakan metode yang digunakan untuk meningkatkan ukuran dan keberagaman data pelatihan tanpa harus mengumpulkan data baru. Teknik ini bertujuan untuk mengatasi masalah *overfitting* dan meningkatkan kemampuan generalisasi model pembelajaran mesin. Augmentasi data dapat dilakukan dengan berbagai cara, termasuk transformasi gambar (seperti *rotation*, *flipping*, dan *scaling*), penyuntikan derau, operasi intensitas (seperti penyesuaian kontras dan kecerahan), serta metode deformasi yang lebih kompleks seperti registrasi gambar *deformable* dan model bentuk statistik (Chlap, Min, Vandenberg, Dowling, Holloway, & Haworth, 2021).

Persyaratan jaringan *deep learning* terhadap data dalam jumlah besar untuk augmentasi data telah memicu terciptanya berbagai teknik, mulai dari modifikasi sederhana seperti transformasi geometri hingga gambar mosaik yang rumit. Beberapa teknik augmentasi data yang paling banyak digunakan mulai dari *translation*, *rotation*, *flip*, *shift*, *resizing*, *zoom* (Isazala & Jimenez, 2022).

2.10.1 Translation

Pertama adalah *translation*, metode augmentasi data di mana suatu gambar digeser dengan tetap mempertahankan posisi relatif pikselnya dalam gambar

aslinya. Tanpa mengubah orientasi atau kemiringannya, gambar akan digulir atau digeser ke arah yang berbeda (Isazala & Jimenez, 2022).

2.10.2 Rotation

Memutar gambar di sekitar posisi aslinya adalah cara melakukan augmentasi data berbasis *rotation*. *Rotation* sama seperti *translation* yang menggunakan sistem sumbu koordinat baru dengan tetap menjaga posisi piksel relatif tetap sama (Isazala & Jimenez, 2022).

2.10.3 Flip

Transformasi geometris lainnya dalam augmentasi data adalah flip, yang melibatkan pembalikan posisi piksel terhadap salah satu dari dua sumbu secara horizontal atau vertikal (dalam kasus data dua dimensi) (Isazala & Jimenez, 2022).

2.10.4 Resizing

Proses *resizing* atau *rescale* melibatkan pengalokasian posisi baru dalam kaitannya dengan faktor skala, yang dapat memiliki proporsi berbeda atau sama untuk semua sumbu. Memperbesar (faktor skala >1) atau memperkecil (faktor skala <1) adalah definisi tepat dari *rescale* (Isaza & Jimenez, 2022).

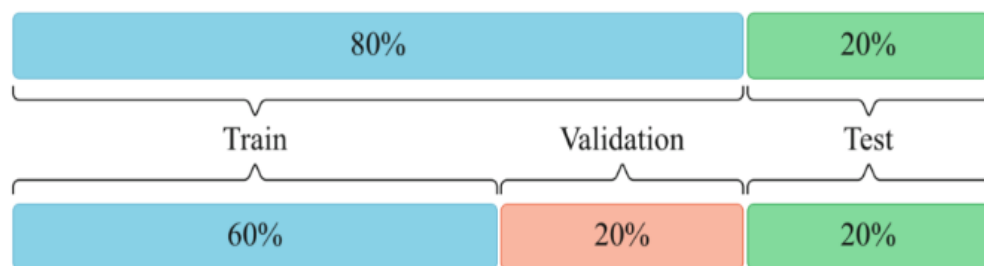
2.10.5 Shift, Shear, & Zoom

Merupakan bagian transformasi geometris di mana augmentasi dasar yang paling umum digunakan. *Shift* adalah transformasi geometris yang memindahkan setiap titik gambar dengan jarak tertentu dalam arah horizontal atau vertikal. *Shear* adalah transformasi geometris yang memindahkan satu bagian dari gambar ke arah horizontal atau vertikal, sementara bagian lainnya tetap pada posisinya. *Zoom* adalah transformasi geometris yang memperbesar atau memperkecil gambar (Chlap, Min, Vandenberg, *et al.*, 2021).

2.11 Data Splitting

Pemisahan data mencoba melatih model pembelajaran mesin dengan benar.

Memisahkan data dengan tepat akan menjamin penilaian yang baik, menjaga model agar tidak *overfitting*, dan meningkatkan performa. Data dibagi menjadi dua sub kumpulan di mana kumpulan *training* data dan kumpulan *testing* data. Metode pemisahan data yang paling banyak digunakan, merupakan kebiasaan untuk membagi menjadi tiga himpunan bagian. Pengumpulan data dibagi menjadi *subset training* data, *validation* data, dan *testing* data. Berikut merupakan contoh Gambar 2.9 Data *Splitting* (Hossain, 2023).



Gambar 2.12 Data *Splitting*
(Sumber: Hossain, 2023)

2.12 Training Data, Validation Data, dan Test Data

Model tidak akan siap untuk memulai pelatihan sampai model tersebut memiliki sekumpulan vektor fitur dan kumpulan data. Terutama karena tidak ingin berlatih dengan seluruh kumpulan data. Harus membagi data menjadi setidaknya dua *subset* tetapi idealnya tiga *subset*, karena sebagian data harus digunakan untuk tujuan lain. *Subset* ini disebut sebagai *testing* data, *validation* data, dan *training* data seperti berikut (Kneusel, 2021):

1. Training Data

subset yang digunakan untuk melatih model. Hal yang penting di sini adalah memilih vektor fitur yang mewakili distribusi induk data dengan baik

2. Test Data

Bagian yang menilai kinerja model pelatihan. Karena akan menguji model pada data yang dilihat sebelumnya, tidak pernah menggunakan *testing* data saat melatih model tidak etis. Setelah model siap, sisihkan kumpulan *testing* data dan gunakan untuk menilai model, tanpa perlu menyentuhnya.

3. Validation Data

Meskipun model pembelajaran mendalam mendapat manfaat dari memiliki

kumpulan *validation* data, tidak semua model memerlukannya. Karena untuk mengukur seberapa efektif pelatihan dilakukan, menggunakan kumpulan *validation* data sebagai *testing* data selama proses pelatihan. Membantu dalam mengambil keputusan tentang penghentian pelatihan dan pemilihan model.

2.13 Confusion Matrix

Salah satu teknik yang sering digunakan dalam penambangan data untuk mengukur akurasi adalah *confusion matrix*. Prediksi akurat sistem klasifikasi untuk klasifikasi dirinci dalam *confusion matrix*. Empat kombinasi berbeda dari nilai yang diharapkan dan nilai aktual membentuk tabel *confusion matrix*. *True Positive*, *True Negative*, *False Positive*, dan *False Negative* adalah empat istilah dari proses kategorisasi dalam *confusion matrix*. Berikut Tabel 2.1 Contoh *Confusion Matrix* (Supida, Udayana, Kharisma, Efitri, Ariana, & Rachman, 2023).

Tabel 2.1 Contoh Confusion Matrix

Aktual	Prediksi	
	Positif	Negatif
Positif	TP	FN
Negatif	FP	TN

Berdasarkan Tabel 2.1 Contoh *Confusion Matrix* dapat dijelaskan bahwa memiliki keterangan masing-masing untuk setiap nilai *confusion matrix* seperti TP adalah aktual bernilai positif (1) yang diberi klasifikasi prediksi positif (1), FN adalah aktual bernilai positif (1) yang diberi klasifikasi prediksi negatif (0). TN adalah aktual bernilai negatif (0) yang diberi klasifikasi prediksi negatif (0). FP adalah aktual bernilai negatif (0) yang diberi klasifikasi prediksi positif (1).

Selanjutnya melanjutkan dari *confusion matrix* tinggal mencari nilai akurasi, presisi, *recall*, dan *error*. Metrik yang digunakan untuk mengidentifikasi persentase prediksi yang diklasifikasikan dengan benar. Menggunakan hasil dari *confusion matrix*, akurasi dapat didefinisikan sebagai persamaan berikut (Churcher, Ullah, Ahmad, Rehman, Masood, Gogate, Alqahtani, Nour, & Buchanan, 2021):

$$\text{Akurasi} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (2.5)$$

Selanjutnya Presisi digunakan untuk menentukan rasio hasil positif yang benar dari

semua hasil positif yang diprediksi, Presisi dapat didefinisikan sebagai persamaan berikut (Churcher, Ullah, Ahmad, *et al.*, 2021):

$$\text{Presisi} = \frac{TP}{TP + FP} \quad (2.6)$$

Selanjutnya *Recall* digunakan untuk menentukan rasio hasil positif yang benar dari semua hasil dalam kelas yang diberikan. *Recall* dapat didefinisikan sebagai persamaan berikut (Churcher, Ullah, Ahmad, *et al.*, 2021):

$$\text{Recall} = \frac{TP}{TP + FN} \quad (2.7)$$

Selanjutnya *F1-Score* adalah rata-rata harmonis dari presisi dan *recall*, yang menghasilkan nilai antara 0 dan 1. *F1-Score* dianggap sebagai metrik kinerja yang lebih baik daripada akurasi, terutama dalam kasus distribusi kelas yang sangat tidak seimbang. *F1-Score* dapat didefinisikan sebagai persamaan berikut (Churcher, Ullah, Ahmad, *et al.*, 2021):

$$F1 - Score = 2 \times \frac{\text{Presisi} \times \text{Recall}}{\text{Presisi} + \text{Recall}} \quad (2.8)$$

Nilai *error* didefinisikan sebagai kasus yang salah diidentifikasi dalam sekumpulan data, sehingga memungkinkan tingkat kesalahan pada sistem yang digunakan ditampilkan secara visual. Persamaan rumus di bawah ini dapat digunakan untuk menghitung persentase *error* (Churcher, Ullah, Ahmad, *et al.*, 2021):

$$\text{Error} = \frac{FP}{TP} \quad (2.9)$$

2.14. Citra

Representasi visual suatu benda, orang, atau pemandangan yang dihasilkan oleh alat optik seperti lensa, kamera, atau kaca disebut Citra. Penggambarannya bersifat dua dimensi meskipun menampilkan suatu objek atau lokasi peristiwa dalam tiga dimensi (3D) (Siahaan & sianipar, 2020).

Citra adalah representasi fisik atau visual dari informasi yang dikandungnya sesuai dengan tujuan penggunaannya. Kandungan informasi suatu gambar memungkinkan mata manusia untuk mempelajari dan memahaminya. Informasi dasar dan informasi abstrak membentuk dua kategori informasi yang terkandung

dalam citra. Informasi yang dapat diolah secara langsung tanpa memerlukan keahlian khusus lebih lanjut disebut informasi dasar. Warna, bentuk, dan tekstur merupakan bentuk yang menyimpan informasi mendasar tersebut. Analisis citra tingkat rendah adalah studi tentang data citra fundamental. Informasi yang tidak dapat langsung diproses tanpa menggunakan pengetahuan khusus disebut sebagai informasi abstrak. Ekspresi wajah dalam citra yang dapat menyampaikan keadaan emosi seseorang seperti bahagia, sedih, marah, dan lain-lainnya adalah contoh informasi abstrak. Penjelasan tentang apa yang terjadi ketika citra diambil adalah jenis informasi abstrak lainnya. Komputer tidak dapat menilai dan mengenali salah satu dari hal ini kecuali menggabungkan pengetahuan yang lebih khusus dengan informasi dasar (Madenda, 2015).

2.14.1 Citra Analog

Definisi matematis suatu gambar adalah fungsi dua dimensi (x,y) , di mana f adalah nilai intensitas warna pada koordinat x dan y dan x dan y adalah koordinat spasial (bidang). Terdapat nilai terbatas untuk f , x , dan y . Gambar analog adalah gambar yang ditampilkan pada monitor TV, komputer, atau foto cetak ketika nilai-nilai ini kontinu (Madenda, 2015).

2.14.2 Citra Digital

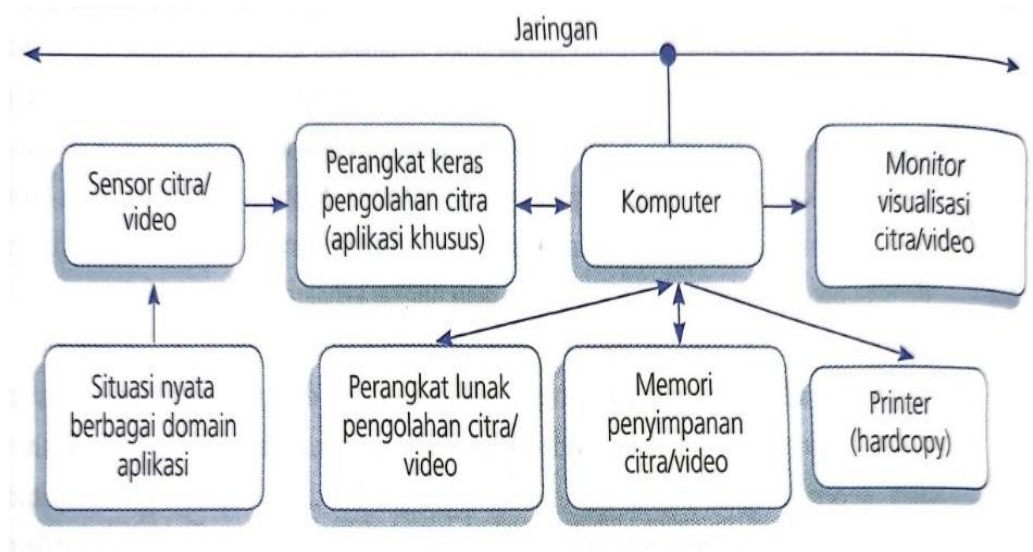
Citra digital adalah representasi dua dimensi yang terdiri dari sejumlah titik yang terbatas, juga dikenal sebagai piksel, pel, atau elemen gambar. Satu atau lebih nilai bilangan bulat mewakili setiap piksel dalam foto berwarna, diperlukan tiga nilai untuk mencerminkan intensitas komponen merah (R), hijau (G), dan biru (B). Intensitas piksel dalam gambar monokrom (abu-abu) diwakili oleh nilai tunggal, sering kali berada di antara $[0, 255]$ (Siahaan & sianipar, 2020).

Suatu citra dikatakan digital jika nilainya diskrit; contoh gambar jenis ini termasuk yang terlihat pada CD-ROM dan memori komputer. Citra digital biasanya direpresentasikan sebagai matriks dengan jumlah komponen yang terbatas, dan biasanya berbentuk dua dimensi (2D). Setiap elemen pada matriks gambar mempunyai nilai selain posisi tertentu pada koordinat x dan y . Citra digital biasanya

didefinisikan sebagai representasi piksel 2D yang diekspresikan dalam matriks dengan N baris dan M kolom. "Elemen gambar, elemen gambar, atau pel" mengacu pada setiap elemen dalam matriks gambar sebagai piksel. Intensitas warna diwakili oleh nilai setiap piksel (f) pada koordinat x dan y . Gambar berwarna (yang memiliki tiga komponen warna RGB: R = merah, G = hijau, dan B = biru) dapat memiliki pengkodean 24 bit, gambar tingkat abu-abu dapat memiliki 8 bit, dan gambar biner dapat memiliki 1 bit (Madenda, 2015).

2.15. Komponen Sistem Pengolahan Citra

Sistem pengolahan gambar dan video digital hingga saat ini telah dikembangkan dalam berbagai model yang telah disesuaikan dengan kebutuhan sektor aplikasinya masing-masing. Semuanya didasarkan pada gagasan dasar yang sama. Sistem ini terdiri dari sensor untuk akuisisi dan digitalisasi gambar, komputer, memori untuk menyimpan, perangkat untuk mencetak dan memvisualisasikan gambar, perangkat keras untuk pemrosesan gambar untuk penggunaan tertentu, dan perangkat lunak pengolah gambar. Setiap teori pemrosesan citra digital adalah teori pemrosesan video digital karena citra adalah komponen video. Berikut merupakan Gambar 2.10 Komponen-Komponen Sistem Akuisisi Pengolahan Citra (Madenda, 2015).



Gambar 2.13 Komponen-Komponen Sistem Akuisisi Pengolahan Citra
(Sumber: Madenda, 2015)

2.15.1 Perangkat Keras Pengolahan Citra

Digitalizer adalah jenis teknologi pemrosesan citra yang mengubah citra analog menjadi citra digital, yang kemudian ditransfer ke komputer untuk diproses lebih lanjut sesuai kebutuhan. Perangkat aplikasi khusus, seperti filter elektronik, transformasi geometri citra, transformasi ruang warna citra, dan lain-lain, juga dapat dianggap sebagai perangkat keras pengolah citra. Salah satu jenis perangkat keras yang digunakan untuk menyaring konten informasi pada frekuensi tertentu adalah filter elektronik, yang misalnya dapat digunakan untuk menyaring kebisingan pada frekuensi tinggi. Memperhitungkan perubahan geometri pada citra, seperti variabel rotasi, translasi, dan penskalaan (zoom), transformasi geometri citra diterapkan. Beberapa aplikasi dan juga untuk presentasi di layar monitor, diperlukan *transcoding* ruang warna citra secara *real-time*. Sebagai ilustrasi, pertimbangkan untuk mengonversi data perolehan gelombang frekuensi warna merah, hijau, dan biru (RGB) menjadi gelombang *luminance* Y dan *chrominance* C (juga dikenal sebagai gelombang Y/C atau gelombang S-Video) (Madenda, 2015).

2.15.2 Perangkat Lunak Pengolahan Citra

Perangkat lunak untuk pengolahan citra dapat digunakan untuk menguji dan menerapkan teori dan algoritma yang berkaitan dengan pengolahan citra baik untuk aplikasi luas maupun khusus. Penggunaan perangkat lunak, pengguna dapat membuat modul khusus dengan operasi pemrosesan gambar dasar dan menulis aplikasi kode sumber sendiri. Modul yang dapat menangani masalah pemrosesan gambar yang umum atau rumit disertakan dengan paket perangkat lunak tertentu. Memenuhi kebutuhan aplikasi yang lebih spesifik, pengguna dapat menggunakan modul ini dan melengkapinya dengan modul dari algoritma yang di kembangkan sendiri (Madenda, 2015).

2.16. Jenis-Jenis Citra

Gambar diklasifikasikan menjadi digital (dalam bentuk gelombang diskrit) atau analog (gelombang kontinu), seperti yang telah dibahas pada sub-bab

sebelumnya. Selanjutnya gambar dapat dikategorikan sebagai gambar biner, skala abu-abu, dan berwarna berdasarkan konten visualnya (Madenda, 2015).

2.16.1 Citra Berwarna

Citra berwarna sebenarnya adalah secara visual yang secara nyata mewakili informasi warna menggunakan nilai piksel yang mencakup komponen *luminance*, *hue* dan *chrominance/saturation*. *luminance* adalah metrik untuk kecerahan warna. Warna dapat dibuat lebih cerah atau lebih gelap dengan mengubah nilai kecerahan. Selanjutnya *hue* adalah salah satu karakteristik utama warna, dengan nilai derajat berkisar antara 0° hingga 360°. Nilai *hue* untuk rona dasar, seperti merah, hijau, dan biru, masing-masing nilai adalah 360°, 120°, dan 240°. Nilai rona warna lainnya seperti kuning yang merupakan gabungan merah dan hijau mempunyai nilai *hue* 60°, oranye yang merupakan kombinasi kuning dan merah memiliki nilai *hue* 30°; ungu yang merupakan kombinasi merah dan biru memiliki nilai *hue* 300°, cyan atau juga dikenal sebagai biru langit yang merupakan kombinasi hijau dan biru memiliki nilai rona 180°, dan warna lainnya memiliki nilai *hue* berbeda-beda. Selanjutnya *chrominance* atau *saturation* melambangkan konsentrasi cahaya putih tinggi dan rendah suatu warna. Warnanya menjadi pucat (memutih) hingga berubah menjadi putih atau abu-abu sebanding dengan penurunan nilai *chrominance* (mendekati nol). Sebaliknya warna semakin mendekati saturasi (warna murni) semakin besar nilai *chrominance*-nya. Salah satu contoh warna biru jenuh adalah biru 100%. Kombinasi perubahan nilai intensitas tiga warna primer mulai dari merah R (merah), hijau G (hijau), dan biru B (biru)—dapat menciptakan variasi warna ini. Representasi komputer dari gambar digital berwarna didasarkan pada ini. Tiga komponen warna, R, G, dan B, membentuk setiap titik atau piksel dalam gambar berwarna. Komponen-komponen ini biasanya direkam dengan masing-masing delapan bit, dengan total $3 \times 8 = 24$ bit (tiga *byte*) per titik atau piksel. Hasilnya, variasi warna maksimum untuk suatu gambar berwarna adalah 224 (16.777.216 variasi warna) Berikut adalah Gambar 2.14 (Madenda, 2015).



(a)

88	73	73	56	40
134	129	127	114	115
97	90	88	75	80
79	75	73	62	55
110	122	126	120	97
73	84	88	84	66
81	77	84	83	78
107	112	127	121	123
77	81	94	90	91
79	77	78	79	76
89	93	119	119	107
63	69	89	89	79
81	71	91	70	69
71	84	106	110	100
41	55	78	81	75

(a)

		97	90	88	75	80
		73	84	88	84	66
	134	129	127	114	115	91
	110	122	126	120	97	79
88	73	73	56	40	123	75
79	75	73	62	55	107	
81	77	84	83	78	100	
79	77	78	79	76		
81	71	91	70	69		

(b)

(b)

Gambar 2.14 (a) Citra Berwarna Berukuran 300 x 500 Piksel
, (b) ((a) Representasi 3 Warna Setiap Pixel dan (b) Representasi Masing-Masing Warna
(RGB)
(Sumber: Madenda, 2015)

2.16.2 Citra Grey Level

Citra *gray-level* (skala ke abuan) adalah citra yang memiliki skala abu-abu yang berkisar antara 0 hingga 255 ($2^8 - 1$) dan nilai pikselnya hanya diwakili oleh nilai *luminance* yang biasanya disimpan dalam 8 bit. Angka antara 0 dan 255 masing-masing menandakan hitam dan putih, sedangkan warna keabu-abuan yang berada di antara keduanya adalah hitam dan cerah dan akhirnya putih. Citra tingkat ke abuan yang dimaksud diwakili oleh komponen Y, V, atau L dari ruang warna. Contoh citra tingkat ke abuan dengan skala ke abuan 256 yang dihasilkan dengan mengkonversi citra berwarna RGB pada Gambar 2.14 menggunakan ruang warna YCbCr ditampilkan pada Gambar 2.15. Nilai kecerahan setiap piksel pada wilayah 5×5 piksel yang terletak di pojok kiri atas gambar ditampilkan pada Gambar 2.16 (Madenda, 2015).



(a)

116	110	109	101	103
97	105	108	104	90
96	99	111	106	110
81	86	106	105	96
66	78	97	99	91

(b)

**Gambar 2.15 (a) Citra Grey Level Berukuran 300 x 500 Piksel
, (b) Nilai Luminance Citra Grey Level dengan Ukuran 5 x 5 Piksel
(Sumber: Madenda, 2015)**

2.16.3 Citra Biner

Citra biner adalah *subset* citra skala abu-abu yang hanya mempunyai dua tingkat ke abuan (0 untuk hitam, 1 untuk putih). Hasilnya, setiap piksel dalam gambar biner disimpan hanya dengan satu bit. Citra biner ditentukan dengan menerapkan nilai ambang batas: nilai diubah menjadi 1 putih jika lebih besar atau sama dengan nilai ambang batas, dan menjadi 0 hitam jika kurang dari ambang batas. Nilai ambang batas sebesar 75, maka citra biner turunan citra tingkat ke abuan pada Gambar 2.15 ditampilkan pada Gambar 2.16. Meskipun tidak ada standar yang ditetapkan untuk pemilihan nilai ambang batas, namun sering kali ditentukan berdasarkan permintaan aplikasi. Selanjutnya nilai biner piksel 5×5 piksel di sudut kiri atas gambar biner ditunjukkan pada Gambar 2.16 (Madenda, 2015).



(a)

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
0	1	1	1	1

(b)

**Gambar 2.16 (a) Citra Biner Berukuran 300 x 500 Piksel
, (b) Nilai Biner dari Citra Biner dengan ukuran 5 x 5 Piksel
(Sumber: Madenda, 2015)**

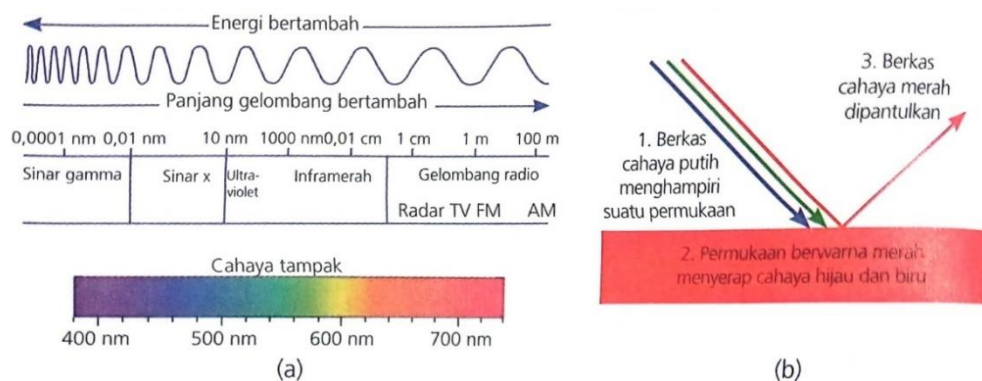
2.17. Ruang Warna dan Dasar-Dasar Pengolahan Citra

Informasi warna, seperti warna apa yang terdapat dalam ruang batas pandangan, pada posisi apa, dan merupakan bagian dari objek apa, mudah dianalisis dan diproses oleh sistem saraf penglihatan manusia. Jelas bahwa sistem seperti sistem visual manusia diperlukan jika komputer atau perangkat lain ingin digunakan untuk tugas-tugas tersebut (Madenda, 2015).

2.17.1 Sifat Dasar Cahaya

Cahaya adalah jenis radiasi elektromagnetik, yaitu variasi medan listrik dan magnet yang merambat sebagai gelombang elektromagnetik ke seluruh ruang. Seperti terlihat pada Gambar 2.16, spektrum gelombang tersebut terdiri dari

gelombang radio, infra merah, cahaya tampak, ultraviolet, sinar-x, dan sinar gamma. Dua dari banyak sifat gelombang elektromagnetik adalah panjang gelombang dan frekuensinya. Sedangkan frekuensi dinyatakan dalam Hertz (Hz), sedangkan panjang gelombang, yaitu jarak antara dua puncak gelombang yang berurutan, diukur dalam nanometer (nm). Panjang gelombang atau spektrum cahaya yang terlihat dengan mata manusia tanpa bantuan hanya berkisar antara 390 nm hingga 750 nm. Spektrum cahaya tampak ini setara dengan pita frekuensi sekitar 400790 THz dalam satuan frekuensi. Spektrum cahaya tampak adalah istilah yang digunakan untuk menggambarkan rentang cahaya ini. Berikut Gambar 2.17 (a) Spektrum Gelombang Cahaya, (b) Absorpsi dan Refleksi Gelombang (Madenda, 2015).



Gambar 2.17 (a) Spektrum Gelombang Cahaya, (b) Absorpsi dan Refleksi Gelombang
(Sumber: Madenda, 2015)

2.17.2 Ruang Warna

Sel kerucut dan batang di retina mata mendeteksi gelombang cahaya tampak, yang kemudian ditransmisikan ke neuron visual otak untuk menghasilkan berbagai persepsi warna. Menurut tulisan Hunt dalam bukunya *Measurim Color*, CIE (*Commission Internationale de l'Eclairage*, atau *International Commission on Illumination*) secara praktis menggambarkan pengertian warna ini sebagai berikut (Madenda, 2015):

1. *Brightness* (kecerahan) Cara mata memandang cahaya ketika mengidentifikasi area berbeda dengan tingkat intensitas kecerahan berbeda.
2. *Hue* (warna) yaitu persepsi visual yang dimiliki mata ketika melihat suatu area yang warnanya sama dengan area lain, sangat mirip, atau berbeda warna..

3. *Colorfulness* adalah kemampuan mata untuk mendeteksi apakah suatu area berwarna (chromatic) atau tidak (achromatic atau *grey level*).
4. *Saturation* (saturasi) adalah persepsi visual yang terjadi ketika kecerahan suatu warna dikontraskan dengan warna murninya, atau rona murni
5. *Chroma* (kemurnian) yaitu persepsi visual mata terhadap kecerahan suatu warna dalam kaitannya dengan warna referensi, biasanya putih (juga identik dengan saturasi).
6. *Lightness/luminance* yang merupakan persepsi visual mata terhadap kecerahan suatu wilayah dalam kaitannya dengan acuan warna putih dalam jangkauan penglihatannya.

2.18. Segmentasi Citra

Agar suatu citra lebih mudah dipahami dan dipelajari, segmentasi citra mencoba merepresentasikan wilayah suatu citra menjadi beberapa bagian. Segmen dapat menggambarkan fitur suatu objek atau sebagian dari suatu objek dalam citra. Karakteristik berikut ini terkait dengan pembuatan segmen (Madenda, 2015):

1. Segmentasi yang sama akan memiliki piksel yang saling terhubung, memiliki nilai intensitas warna atau tekstur yang sangat sebanding, dan terhubung.
2. Border berbentuk garis pemisah memisahkan piksel-piksel yang saling berkaitan satu sama lain namun mempunyai nilai intensitas warna atau tekstur berbeda.

2.18.1 Segmentasi Warna Citra

Setiap warna piksel yang menyusun suatu objek mungkin menunjukkan perbedaan atau perbandingan satu objek dalam citra dengan objek lain dalam hal warna luas permukaannya. Suatu benda bisa homogen (hanya mempunyai satu warna) atau heterogen (memiliki banyak corak). Segmentasi warna citra adalah proses membagi suatu gambar menjadi beberapa bagian berdasarkan kemiripan warnanya. Segmen akan dibuat menggunakan prosedur segmentasi berdasarkan atribut warna. Konstruksi teori segmentasi warna didasarkan pada dua kata kunci yaitu (1) analisis kemiripan atau kesamaan intensitas/warna, dan (2) dua atau sekelompok piksel di sekitarnya. Teori ruang warna dan alat untuk mengukur

ketidaksamaan warna dan/atau kesamaan jarak dapat ditemukan dengan menggunakan kata kunci pertama. Kata kunci kedua memberikan informasi tentang cara mengukur dan berapa banyak piksel di sekitarnya yang akan digunakan (Madenda, 2015).

2.18.2 Segmentasi Tekstur Citra

Citra yang diambil dari benda asli di alam tidak selalu menampilkan area citra dengan warna atau intensitas yang sama secara keseluruhan. Contohnya gambaran lantai kayu parket tidak seragam, namun memiliki variasi warna dan intensitas yang berulang dalam suatu pola. Ilustrasi kulit jeruk mungkin bisa menjadi contoh lain. Permukaan yang berbintik-bintik di berbagai tempat memberikan warna tampak homogen, meskipun cahaya yang dipantulkan memiliki intensitas yang bervariasi. Terlepas dari perbedaan antara kedua contoh ini, keduanya diklasifikasikan sebagai tekstur. Analisis kekasaran permukaan serta analisis struktur dan orientasi pola selalu disebutkan dalam kaitannya dengan analisis tekstur dan segmentasi gambar berbasis tekstur. Teori analisis tekstur gambar telah dikembangkan berdasarkan dua poin ini.

2.19. Pemrosesan Citra

Teknik elektro, ilmu komputer, fisika, matematika, dan disiplin ilmu lainnya semuanya berkontribusi pada topik multidisiplin pemrosesan gambar. Tiga tahapan proses pengolahan citra adalah sebagai berikut (Siahaan & sianipar, 2020):

1. Tingkat rendah: Pengoperasian sederhana dengan gambar sebagai masukan dan keluaran, seperti peningkatan kontras dan pengurangan derau.
2. Tingkat Menengah: Menghilangkan karakteristik atau elemen tertentu dari gambar, seperti area, tepi, dan kontur.
3. Tingkat tinggi: Pemeriksaan dan interpretasi isi gambar.

2.19.1 Operasi Pemrosesan Citra

Bagian ini akan dijelaskan secara singkat beberapa prosedur pengolahan citra dan akan mencakup berbagai teknik dan algoritma yang termasuk dalam

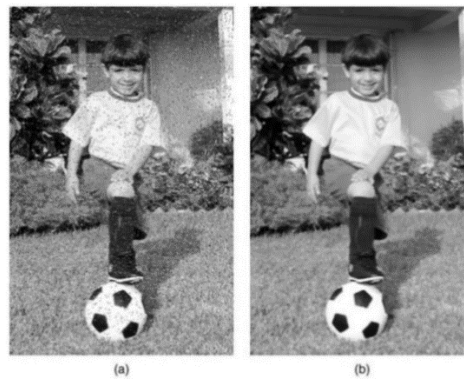
pengolahan citra (Siahaan & sianipar, 2020):

1. Penajaman: Sebuah metode yang meningkatkan batas gambar dan detail kecil untuk inspeksi visual.



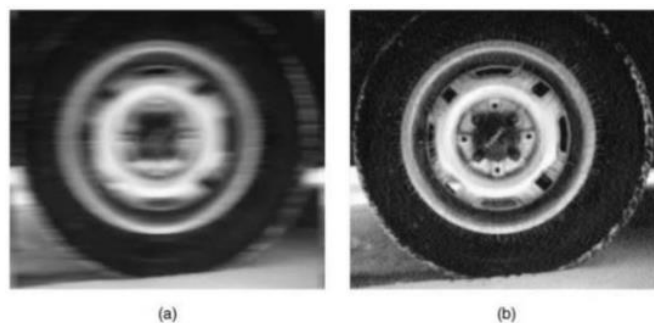
Gambar 2.18 Penajaman Citra (a) Citra Awal dan (b) Citra Penajaman
(Sumber: Siahaan & sianipar, 2020)

2. Pengurangan Derau: Tingkat kebisingan pada gambar dapat dikurangi dengan menerapkan filter pemrosesan gambar.



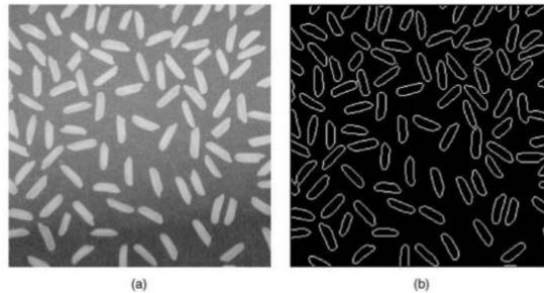
Gambar 2.19 Pengurangan Derau (a) Citra Awal dan (b) Citra Pengurangan Derau
(Sumber: Siahaan & sianipar, 2020)

3. Anti Pengaburan: Karena berbagai alasan, termasuk fokus lensa yang tidak memadai atau kamera yang terlalu cepat untuk merekam objek bergerak, gambar menjadi buram.



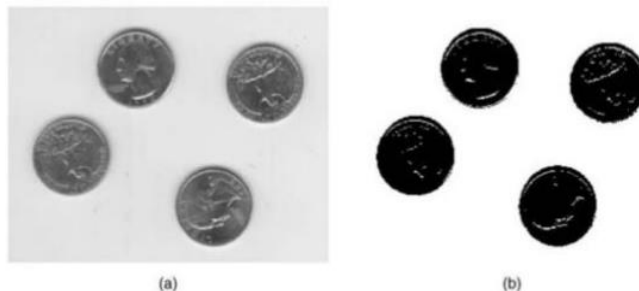
Gambar 2.20 Anti Pengaburan (a) Citra Kabur dan (b) Citra Anti Pengaburan
(Sumber: Siahaan & sianipar, 2020)

4. Ekstraksi Tepi: Salah satu langkah pra-pemrosesan yang jarang namun mendasar yang digunakan untuk membedakan objek satu sama lain sebelum pengenalan gambar adalah ekstraksi tepi.



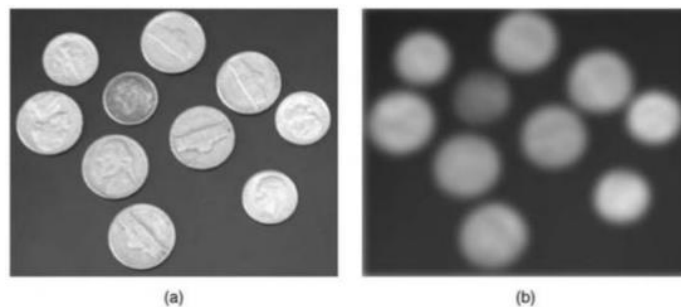
Gambar 2.21 Ekstraksi Tepi (a) Citra Awal dan (b) Citra Tepi-Tepi Diekstraksi
(Sumber: Siahaan & sianipar, 2020)

5. Binerisasi: Mengurangi jumlah tingkat *ke abuan* pada gambar monokrom sering kali diperlukan dalam aplikasi analisis gambar untuk menyederhanakan dan mempercepat proses interpretasi. Binerisasi adalah proses mereduksi gambar skala abu-abu menjadi hanya dua tingkat abu-abu: hitam dan putih.



Gambar 2.22 Binerisasi (a) Citra Awal dan (b) Citra Konversi Hitam Putih
(Sumber: Siahaan & sianipar, 2020)

6. Pengaburan: Untuk mengurangi kekasaran dan detail halus, terkadang gambar perlu diburamkan. Dalam skenario ini, benda-benda dapat diidentifikasi dengan lebih jelas berdasarkan bentuknya.



Gambar 2.23 Pengaburan (a) Citra Awal dan (b) Citra Setelah Pengaburan
(Sumber: Siahaan & sianipar, 2020)

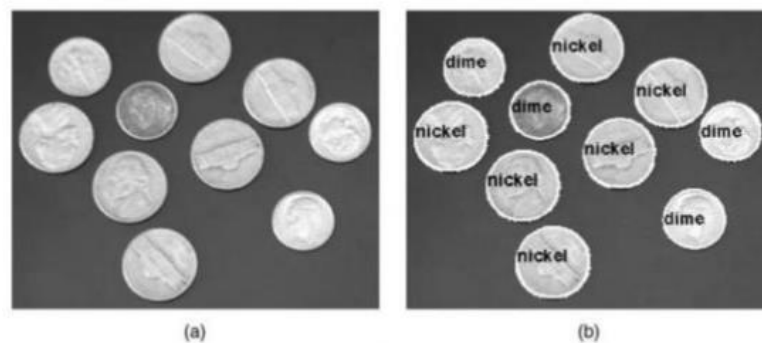
7. Peningkatan Kontras: Seringkali perlu menaikkan kontras gambar untuk

mempermudah pengamatan visual atau untuk melakukan aktivitas pemrosesan gambar tambahan (seperti deteksi tepi).



Gambar 2.24 Peningkatan Kontras (a) Citra Awal dan (b) Citra Memperbaiki Kontras (Sumber: Siahaan & sianipar, 2020)

8. Segmentasi: Hampir semua sistem klasifikasi dan pengenalan objek memerlukan tugas melakukan segmentasi dan mengidentifikasi item dalam suatu gambar. Fitur item terkait dapat diambil dan digunakan untuk mengkategorikan, membedakan, mengelompokkan, dan mengidentifikasi objek setelah objek tersebut disegmentasi dan diberi label.



Gambar 2.25 Segmentasi (a) Citra Awal dan (b) Citra Setelah Segmentasi (Sumber: Siahaan & sianipar, 2020)

2.20 Roda dan Ban

Karena roda biasanya digunakan sebagai penopang saat kendaraan melaju, maka roda merupakan bagian integral dari kendaraan. Agar penggunaan kendaraan lebih stabil dan aman, ban berfungsi sebagai lapisan luar pada roda. Kontruksi ban kendaraan terdiri dari *carcass*, *tread*, *sidewall*, *breaker* dan *beada*. Kode ban ditetapkan berdasarkan ukuran. Contoh metrik (kode ban 120/70-17 67H), nilai 120 menunjukkan lebar ban dalam milimeter. Kode rasio tinggi ban terhadap lebar ban adalah 70. Semakin sempit rasio aspeknya, semakin baik performa ban dalam hal stabilitas dan pengendalian kendaraan ($70 \times 120 \text{ mm} = 85,2 \text{ mm}$). Angka 17

mewakili diameter roda dalam inci. Kode 67 mewakili beban tertinggi yang mampu ditanggung oleh sebuah ban, yaitu 307 kg. Informasi ini dapat ditemukan di tabel indeks beban/data LI. Batas kecepatan penggunaan dilambangkan dengan H (Prayogi, 2018).

Menyambung kembali dari penelitian menurut Prayogi (2018), bahwa Produk ban mempunyai tanggal kadaluwarsa dan biasanya, ini adalah tiga tahun sejak tanggal pembuatan atau sekitar 60.000 kilometer. Jumlah digit dan kodenya unik untuk setiap pabrik ban. Bagian sisi ban sebelah velg tercetak kode pabrikan. Setiap produsen ban memiliki serangkaian angka unik, berkisar antara lima hingga delapan, untuk mengidentifikasi ban yang diproduksi. Minggu dan tahun produksi ban ditunjukkan dengan empat digit nomor belakang, yang mengikuti standar internasional.

2.21 Jenis Cacat Ban

Beberapa klaim mengklasifikasikan cacat ban berdasarkan dari mana asalnya dan apa yang terjadi selama produksi. Karyawan di lokasi inspeksi manufaktur biasanya memiliki akses ke katalog yang merinci potensi masalah pada ban. Menyederhanakan klasifikasi cacat yang paling umum, enam cacat mendasar telah diidentifikasi sebagai kategori cacat (Kuric, Klarak, Saga *et al.*, 2021).

1. Kecacatan dengan Material yang Sama dari Material Ban (CH01).

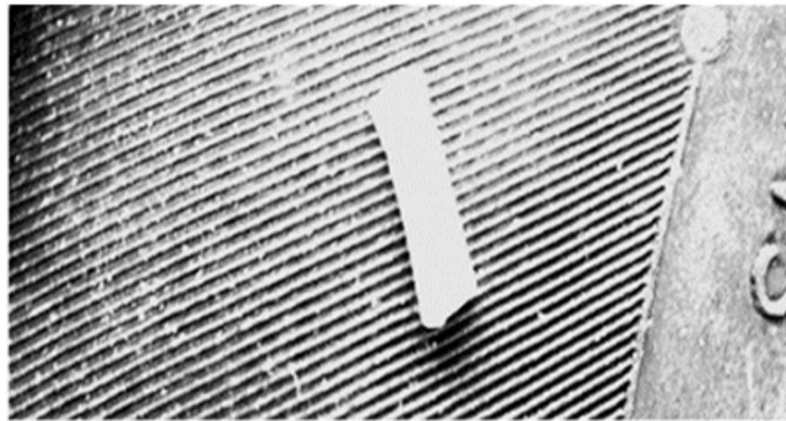
Ketidaksempurnaan atau kecacatan yang terdiri dari bahan yang sama dengan ban itu sendiri. Misalnya, cacat ini juga terjadi pada karet jika ban dari karet.



Gambar 2.26 Kecacatan dengan Material yang sama dari Material Ban CH01
(Sumber: Kuric, Klarak, Saga *et al.*, 2021)

2. Kecacatan dengan Material yang Berbeda dari Material Ban (CH02)

Mengacu pada cacat yang terbuat dari bahan selain yang digunakan dalam produksi ban. Contohnya adalah cacat berbahan plastik atau logam yang melekat atau bukan bagian pada ban.



Gambar 2.27 Kecacatan dengan Material yang Berbeda dari Material Ban (CH02)
(Sumber: Kuric, Klarak, Saga *et al.*, 2021)

3. Kerusakan oleh Suhu dan Tekanan (CH03)

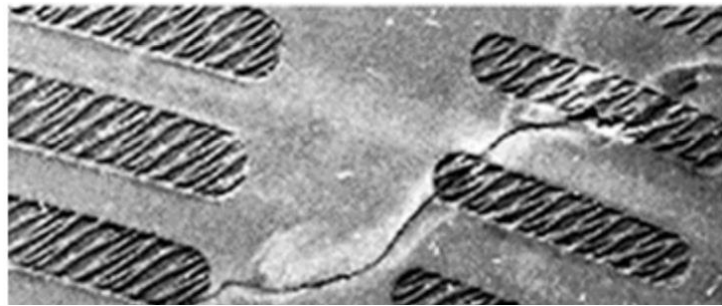
Menunjukkan kerusakan pada ban akibat dampak tekanan dan suhu tinggi. Hilangnya kekuatan struktural akibat tekanan tinggi atau panas adalah contohnya.



Gambar 2.28 Kerusakan oleh Suhu dan Tekanan (CH03)
(Sumber: Kuric, Klarak, Saga *et al.*, 2021)

4. Retak (CH04)

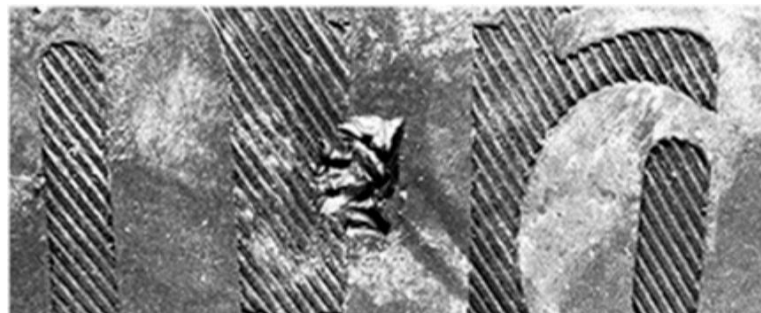
Mengacu pada adanya retakan pada struktur atau permukaan ban. Keausan, tekanan mekanis, dan cacat pada proses pembuatan ban hanyalah beberapa penyebab keretakan.



Gambar 2.29 Retak (CH04)
(Sumber: Kuric, Klarak, Saga *et al.*, 2021)

5. Kerusakan Mekanisme terhadap Integritas (CH05):

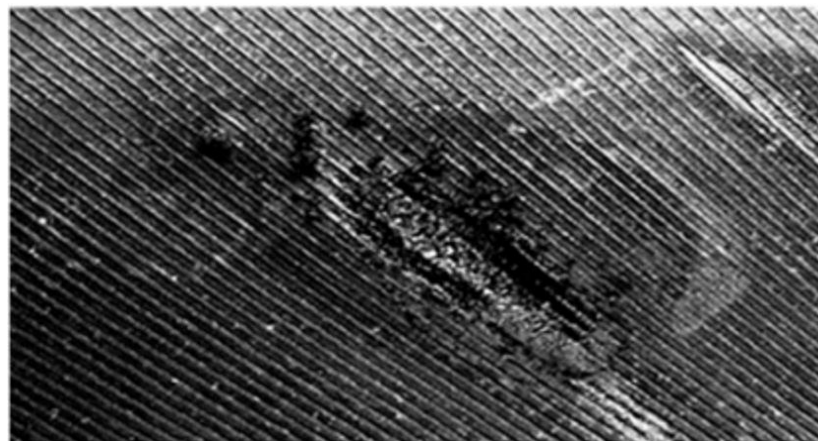
Menjelaskan kerugian yang ditimbulkan oleh elemen mekanis yang membahayakan integritas larangan tersebut. Patah, sobek, atau menyusut akibat benturan atau tekanan fisik adalah beberapa contoh jenis cacat ini.



Gambar 2.30 Kerusakan Mekanisme Terhadap Integritas (CH05)
(Sumber: Kuric, Klarak, Saga *et al.*, 2021)

6. Material *Etched* (CH06):

Menunjukkan bahan terlarang yang terkorosi atau terekspos akibat reaksi kimia dengan bahan tertentu. Reaksi kimia dapat mengubah warna, tekstur, atau karakteristik fisik bahan yang tergores.



Gambar 2.31 Material *Etched* (CH06)
(Sumber: Kuric, Klarak, Saga *et al.*, 2021)

2.22 Penelitian Terkait

Sebelum memulai proses pembuatan model *Convolutional Neural Networks* (CNN) untuk pengukuran kualitas produk ban, penting untuk memahami studi terkait atau studi sebelumnya di lapangan. Beberapa bidang penelitian sebelumnya yang dapat memberikan dasar untuk penelitian masa depan antara lain sebagai berikut pada Tabel 2.1 *State of The Art*.

Tabel 2.2 *State of The Art*

N O	Nama Penulis/ Judul Penelitian/ Tahun Penelitian	Metode Penelitian	Kelebihan/ Kekurangan	Hasil Penelitian
Ban				
1.	Shih-Lin Lin/ <i>Research on Tire Crack Detection Using Image Deep Learning Method/</i> 2023	<p>Menggunakan metode <i>deep learning</i> yaitu <i>Convolutional Neural Networks</i> (CNN) khususnya metode GoogLeNet, ShufeNet tradisional dan ShufeNet yang ditingkatkan yang merupakan metode yang dikembangkan untuk mempercepat dan meningkatkan efisiensi. Kategori kelas <i>cracked</i> dan normal ban</p> <p>Fokus pentingnya kualitas ban dan risiko yang terkait dengan cacat seperti lubang kecil dan retakan pada permukaan ban.</p> <p>Permasalahan: dilatarbelakangi oleh cacat ban seperti retak dan serpihan yang dapat menimbulkan risiko keselamatan dan mempengaruhi keselamatan berkendara.</p> <p>Tujuan:</p>	<p>Kelebihan: Algoritma pembelajaran mendalam, seperti ShuffleNet yang ditingkatkan, memiliki ekstraksi fitur yang kuat dan kemampuan belajar mandiri parameter, menjadikannya efektif dalam mendeteksi kerusakan ban</p> <p>Kekurangan: Model pembelajaran mendalam memiliki keterbatasan dalam menangani pola kerusakan ban yang kompleks dan beragam, sehingga memerlukan optimalisasi berkelanjutan dan perluasan cakupan penerapannya.</p>	<p>Menggunakan <i>dataset</i> sebanyak 150 gambar ban dengan data <i>splitting</i> yang dilakukan yaitu 70% untuk <i>training</i> data, dan 30% untuk <i>validation</i> data dengan 10 sampai 130 <i>epoch</i>, <i>batch size</i> (4), <i>dropout</i> (0,1) dan menggunakan 22 <i>layer cov</i> serta 5 <i>layer pooling</i>. Penelitian ini membandingkan kinerja lima metode berbeda, termasuk akurasi klasifikasi <i>GoogLeNet</i> adalah 82,7%, <i>ShufeNet</i> tradisional adalah 85,3%, <i>VGGNetis</i> 87,3%, <i>ResNet</i> adalah 90%, dan <i>ShufeNet</i> yang ditingkatkan adalah 94,7%.</p> <p>Memberikan keunggulan</p>

N O	Nama Penulis/ Judul Penelitian/ Tahun Penelitian	Metode Penelitian	Kelebihan/ Kekurangan	Hasil Penelitian
		Mengembangkan sistem deteksi cacat ban dengan menggunakan <i>deep learning</i> . Penelitian ini bertujuan untuk meningkatkan ShuffleNet tradisional dan mengusulkan metode ShuffleNet yang lebih baik untuk mendeteksi.		seperti objektivitas, konsistensi, otomatisasi, dan kemampuan melakukan pemeliharaan preventif, menjadikannya alat yang berharga dalam aplikasi praktis untuk inspeksi ban.
F2	Ren wang, Qiang Guo, Shanmei Lu, & Caiming Zhang/ <i>Tire Defect Detection Using Fully Convolutional Network</i> / 2019.	<p>Menggunakan metode <i>Convolutional Neural Networks</i> (CNN) dengan model arsitektur <i>Fully Convolutional Network</i> dan VGG16, Alex, VGG11, VGG13.</p> <p>Metode ini terdiri dari tiga fase: jaringan dalam tradisional untuk ekstraksi fitur, lapisan pengambilan sampel untuk menghasilkan keluaran dengan ukuran yang sama dengan gambar asli, dan fusi peta fitur multi-skala untuk menyempurnakan hasil segmentasi.</p> <p>Permasalahan: Berfokus pada penerapan jaringan konvolusional penuh (FCN) untuk mendeteksi cacat pada gambar sinar-X ban di bidang industri.</p> <p>Tujuan:</p>	<p>Kelebihan: Metode yang diusulkan berdasarkan jaringan konvolusional penuh (FCN) secara akurat menemukan dan mengelompokkan cacat pada gambar ban. Metode ini mengatasi keterbatasan metode proyeksi linier dan cocok untuk mendeteksi cacat tidak beraturan pada gambar ban.</p> <p>Kekurangan: FCN standar mungkin tidak sensitif terhadap cacat kecil pada gambar ban, seperti gelembung, karena hilangnya detail yang disebabkan oleh operasi pengumpulan.</p>	<p>Metode ini memiliki tingkat deteksi yang tinggi di bandingkan dengan metode lainnya, metode yang diusulkan dapat memperoleh lebih banyak hasil yang akurat.</p> <p>Menggunakan <i>dataset</i> sebanyak 914 gambar ban dengan data <i>splitting</i> yang dilakukan yaitu 80% untuk <i>training</i> data, dan 20% untuk <i>testing</i> data dengan <i>input size</i> 256 x 256, batch size 256 dan menggunakan 3 dan 5 <i>cov layer</i>.</p> <p>VGG16 dengan tiga lapisan yang menyatu tidak hanya akurat mendeteksi cacat skala yang berbeda tetapi juga</p>

N O	Nama Penulis/ Judul Penelitian/ Tahun Penelitian	Metode Penelitian	Kelebihan/ Kekurangan	Hasil Penelitian
		Secara akurat menemukan dan mengelompokkan cacat pada gambar ban menggunakan metode yang diusulkan berdasarkan FCN.		menghasilkan lebih tepat prediksi menghasilkan jenis cacat yang berbeda dengan akurasi 78.91%.
3	Yan Zhang, Xuehong Cui, Yun Lin, & Bin Yu/ <i>Tire Defects Classification Using Convolution Architecture for Fast Feature Embedding</i> / 2018.	<p>Menggunakan metode <i>deep learning</i> dengan algoritma <i>Convolutional Neural Networks</i> (CNN) dan model AlexNet yang telah dilatih sebelumnya untuk klasifikasi kerusakan ban.</p> <p>Permasalahan: Skema klasifikasi tradisional tidak cukup untuk mengatasi kesulitan yang dihadapi dalam 6 kategori cacat yang umum, yaitu <i>Belt Foreign-Matter</i> (BFM), <i>Sidewall-Foreign-Matter</i> (SFM), <i>Belt-Joint-Open</i> (BJO), <i>Cords-Distance</i> (CD), <i>Bulk Sidewall</i> (BS) dan <i>Normal-Cords</i> (NC),</p> <p>Tujuan: Mengusulkan metode pembelajaran mendalam yang tertanam fitur terawasi untuk klasifikasi cacat ban menggunakan <i>Convolutional Neural Networks</i> (CNN)</p>	<p>Kelebihan: Algoritma ini juga mampu mengungguli beberapa metode <i>state-of-the-art</i> lainnya dalam klasifikasi seperti PCA+BP, ScSPM09, LLC10, dan KSPM, menunjukkan kinerja yang baik dalam aplikasi dunia nyata.</p> <p>Kekurangan: Penggunaan sampel pelatihan yang kurang memadai, yang dapat mempengaruhi kinerja algoritma. Selain itu, kompleksitas struktur material ban yang beragam juga menjadi tantangan dalam pengembangan algoritma klasifikasi cacat ban.</p>	<p>Menggunakan <i>dataset</i> sebanyak 1582 gambar dengan data <i>splitting</i> yang dilakukan yaitu 60% untuk <i>training</i> data, 20% untuk <i>testing</i> data dan 20% untuk <i>validation</i> data dengan <i>dropout</i> (0,5), dengan 200 <i>epochs</i>, <i>input size</i> 127x127, <i>learning rate</i> 0,001 dan menggunakan 5 <i>layer</i> Cov2D.</p> <p>Metode klasifikasi cacat ban berbasis pembelajaran mendalam, dengan penggunaan jaringan saraf konvolusional dan teknik regularisasi gabungan, efektif dalam mengatasi tantangan klasifikasi cacat ban dalam tugas industri dunia nyata. Metode <i>deep learning</i> yang diusulkan mampu mencapai akurasi</p>

N O	Nama Penulis/ Judul Penelitian/ Tahun Penelitian	Metode Penelitian	Kelebihan/ Kekurangan	Hasil Penelitian
				klasifikasi keseluruhan sebesar 96,51%, dengan akurasi validasi sebesar 98,37%.
4	Ying Li, Binbin Fan, Weiping zhang, & Zhiqiang Jiang/ <i>TireNet: A high recall rate method for practical application of tire defect type classification</i> / 2021.	<p>Menggunakan metode algoritma <i>Convolutional Neural Networks</i> (CNN) dan arsitektur F R-CNN, YOLOv3, SSD512, <i>TireNet</i> pengembangan sebuah kerangka kerja <i>end-to-end</i> untuk deteksi cacat pada gambar sinar-X ban. Melibatkan 11 jenis cacat ban.</p> <p>Permasalahan: Deteksi cacat pada gambar sinar-X ban merupakan tugas yang sulit dan membutuhkan waktu yang lama jika dilakukan secara manual. Selain itu, ada kesulitan dalam membedakan antara bagian ban yang cacat dan bagian yang layak.</p> <p>Tujuan: Mengembangkan sebuah metode <i>end-to-end</i> yang dapat mendeteksi cacat pada gambar sinar-X ban secara otomatis dan memenuhi persyaratan praktis industri.</p>	<p>Kelebihan: Algoritma yang diusulkan, <i>TireNet</i>, berhasil mencapai tingkat <i>recall</i> tertimbang sebesar 94,7% pada <i>dataset</i> yang ditawarkan. Hal ini menunjukkan kemampuan algoritma untuk mendeteksi cacat dengan tingkat keberhasilan yang tinggi.</p> <p>Kekurangan: Meskipun algoritma ini berhasil dalam mendeteksi berbagai jenis cacat, masih terdapat kebutuhan untuk meningkatkan deteksi pada jenis cacat tertentu, seperti gelembung yang memiliki perbedaan yang halus dari bagian sekitarnya.</p>	<p>Menggunakan <i>dataset</i> sebanyak 120.000 gambar (100.000 gambar ban rusak, dan 20.000 gambar ban normal) dengan data <i>splitting</i> yang dilakukan yaitu 70% untuk <i>training</i> data, dan 30% untuk <i>validation</i> data dengan <i>dropout</i> (0,2), dengan <i>input size</i> 900 x 900, dan menggunakan 3 <i>layer</i>.</p> <p>Pengembangan metode baru yang disebut <i>TireNet</i>, yang berhasil mencapai tingkat <i>recall</i> tertimbang sebesar 94.7% pada <i>dataset</i> yang ditawarkan. Metode ini telah diimplementasikan di perusahaan manufaktur ban dan berhasil mencapai kinerja yang lebih baik daripada manusia dalam aplikasi praktis. Selain itu, metode ini juga berhasil memenuhi persyaratan</p>

N O	Nama Penulis/ Judul Penelitian/ Tahun Penelitian	Metode Penelitian	Kelebihan/ Kekurangan	Hasil Penelitian
				praktis industri, termasuk tingkat <i>recall</i> tertimbang dan tingkat presisi tertimbang yang tinggi.
5	Sang-Kwon Lee a, Hwajin Lee a , Jiseon Back a , Kanghyun An a , Youngsam Yoon b , Kiho Yum b , Sungdae kim b , & Sung-Uk Hwang/ <i>Prediction of tire pattern noise in early design stage based on convolutional neural network/</i> 2021	Menggunakan model <i>Convolutional Neural Network</i> (CNN) untuk memprediksi tingkat kebisingan pola ban dalam tahap desain awal. model <i>Artificial Neural Network</i> (ANN) juga dikembangkan untuk memprediksi level kebisingan pola ban. Permasalahan: Memprediksi tingkat kebisingan pola ban pada tahap desain awal tanpa melakukan eksperimen pada ban sungguhan. Tujuan: Mengembangkan model <i>Convolutional Neural Networks</i> (CNN) dan <i>Artificial Neural Network</i> (ANN) yang dapat memprediksi tingkat kebisingan pola ban dengan korelasi yang tinggi, bahkan tanpa memiliki ban sungguhan.	Kelebihan: model <i>Convolutional Neural Networks</i> (CNN) yang dapat memprediksi tingkat kebisingan pola ban pada tahap desain awal tanpa melakukan eksperimen pada ban sungguhan. Kekurangan: Model <i>Artificial Neural Network</i> (ANN) tidak dapat digunakan untuk memprediksi tingkat kebisingan pola ban pada tahap desain awal karena membutuhkan ban sungguhan untuk dilakukan pemindaian laser.	Model <i>Convolutional Neural Networks</i> (CNN) dan <i>Artificial Neural Network</i> (ANN) kemudian diuji dengan <i>training</i> data dan data uji yang berbeda. Menggunakan <i>dataset</i> sebanyak 300 gambar dari 28 ban dengan data <i>splitting</i> yang dilakukan yaitu 80% untuk <i>training</i> data, 10% untuk data <i>test</i> , dan 10% untuk <i>validation</i> data, menggunakan 2 layer (cov dan <i>pooling</i>), <i>batch size</i> 64, dan <i>epoch</i> (120). Hasilnya menunjukkan bahwa model mampu memprediksi tingkat kebisingan pola ban dengan korelasi di atas 0.87.
6	Zezu Wu, Cuijuan Jiao, Jianyuan Sun, & Liang Chen/ <i>Tire Defect Detection Based on Faster R-</i>	Mengusulkan metode deteksi cacat ban berdasarkan <i>Faster R-CNN</i> , yang	Kelebihan: Metode ini menggabungkan teknik pemrosesan gambar dan	Menggunakan <i>dataset</i> sebanyak 1022 gambar ban dengan data <i>splitting</i> yang

N O	Nama Penulis/ Judul Penelitian/ Tahun Penelitian	Metode Penelitian	Kelebihan/ Kekurangan	Hasil Penelitian
	CNN/ 2021	<p>menggabungkan fitur konvolusi lapisan ketiga dan kelima dalam arsitektur Zeiler & Fergus Network untuk mengekstraksi karakteristik mendalam sebagai masukan untuk <i>Faster R-CNN</i>.</p> <p>Metode deteksi cacat ban berfokus pada deteksi dinding samping, benda asing pada tapak, dan gelembung dinding samping.</p> <p>Permasalahan: Cacat ban seperti dinding samping benda asing, tapak benda asing, dan gelembung dinding samping dapat muncul selama produksi ban dan secara langsung atau tidak langsung mempengaruhi masa pakai ban. Metode tradisional untuk mendeteksi cacat bergantung pada inspeksi visual manusia, yang bersifat subyektif dan rentan terhadap kesalahan.</p> <p>Tujuan: Mengembangkan metode deteksi cacat ban yang baru dan efisien berdasarkan <i>Faster R-CNN</i>.</p>	<p>Algoritma pembelajaran mendalam, memanfaatkan jaringan saraf konvolusional untuk mengekstrak fitur dan menganalisis semantik, sehingga meningkatkan otomatisasi dan efisiensi proses deteksi.</p> <p>Kekurangan: Penelitian ini mengakui bahwa hasil deteksi cacat gelembung sedikit buruk dan menyarankan penelitian lebih lanjut dengan menggunakan kombinasi metode deteksi langsung dan tidak langsung.</p>	<p>dilakukan yaitu 80% untuk <i>training data</i>, dan 20% untuk <i>validation data</i> dengan 10 <i>max epoch</i>, <i>batch size</i> (256), <i>dropout</i> (0,2).</p> <p>Metode deteksi cacat R-CNN Lebih Cepat yang diusulkan mencapai klasifikasi akurat dan lokalisasi cacat ban pada gambar sinar-X, dengan tingkat pengenalan pengujian rata-rata hingga 95,4%. Metode ini menunjukkan peningkatan akurasi dalam mendeteksi cacat dinding samping benda asing dibandingkan dengan algoritma sebelumnya.</p>
7	Vinod Vasan, Naveen Venkatesh Sridharan, Anoop Prabhakaranpillai	Makalah ini menggunakan pendekatan jaringan saraf dalam dengan	Kelebihan: Algoritma memberikan pendekatan sistematis	Menggunakan <i>dataset</i> sebanyak 1200 gambar ban dengan data

N O	Nama Penulis/ Judul Penelitian/ Tahun Penelitian	Metode Penelitian	Kelebihan/ Kekurangan	Hasil Penelitian
	Sreelatha, & Sugumaran Vaithiyanathan/ <i>Tire Condition Monitoring Using Transfer Learning-Based Deep Neural Network Approach</i> /2023	<p>membandingkan VGG-16, GoogLeNet, AlexNet, dan ResNet-50 pada <i>Convolutional Neural Networks</i> (CNN) berbasis pembelajaran transfer untuk memantau kondisi ban.</p> <p>Permasalahan: Memantau kondisi ban sangat penting untuk keselamatan dan keekonomian mobil.</p> <p>Tujuan: Mengembangkan sistem pemantauan kondisi ban dengan menggunakan pendekatan <i>deep neural network</i> berbasis transfer <i>learning</i>.</p>	<p>dan terstruktur untuk pemecahan masalah, memastikan efisiensi dan akurasi dalam analisis data dan pengambilan keputusan.</p> <p>Kekurangan: Mengembangkan dan menerapkan Algoritma dapat memakan waktu dan sumber daya yang intensif, sehingga memerlukan keahlian dalam pemrograman dan analisis data.</p>	<p><i>splitting</i> yang dilakukan yaitu 80% untuk <i>training</i> data, 20% untuk data <i>test</i>, dan 20% untuk <i>validation</i> data dengan 246 <i>epoch</i>, <i>batch size</i> (10), <i>learning rate</i> 0,0001, <i>optimizer algorithm</i> RMSPROP dengan <i>input size</i> 224 x 224.</p> <p>Metode yang diusulkan mencapai akurasi klasifikasi maksimum sebesar 93,80% .Studi tersebut menunjukkan efektivitas pendekatan yang diusulkan dalam memantau kondisi ban.</p>
8	ZHOUSHOU ZHENG, SEN ZHANG, BIN YU, QINGDANG LI, & YAN ZHANG/ <i>Defect Inspection in Tire Radiographic Image Using Concise Semantic Segmentation</i> / 2020	<p>Menggunakan <i>Concise Semantic Segmentation Network</i> (Concise-SSN) untuk inspeksi visual ban otomatis. Model ini menggabungkan kekuatan jaringan segmentasi semantik yang dioptimalkan dan <i>Convolutional Neural Networks</i> (CNN) yang ringkas untuk klasifikasi.</p> <p>Permasalahan: Bagaimana menciptakan inspeksi visual ban otomatis sangat</p>	<p>Kelebihan: Model ini dapat mencapai deteksi dan klasifikasi cacat ban secara piksel ujung ke ujung, memastikan inspeksi visual ban otomatis yang akurat dan efisien.</p> <p>Kekurangan: Diperlukan untuk mengevaluasi kinerja model pada kumpulan data yang lebih besar dan beragam, serta untuk menilai ketahanannya dalam skenario pemeriksaan ban di dunia nyata.</p>	<p>Hasil eksperimen menunjukkan kinerja yang unggul pada tugas segmentasi dan klasifikasi cacat dibandingkan dengan model canggih, dengan ukuran model yang lebih kecil dan komputasi yang lebih cepat.</p> <p>Penggunaan <i>dataset</i> sebanyak 3234 data gambar dengan data <i>splitting</i> yang dilakukan yaitu 70% untuk</p>

N O	Nama Penulis/ Judul Penelitian/ Tahun Penelitian	Metode Penelitian	Kelebihan/ Kekurangan	Hasil Penelitian
		<p>penting untuk memastikan kualitas ban dan keselamatan berkendara.</p> <p>Tujuan: mengembangkan model deteksi cacat ban baru menggunakan <i>Concise Semantic Segmentation Network</i> (Concise-SSN) untuk inspeksi visual ban otomatis.</p>		<p><i>training</i> data, 20% untuk data <i>test</i>, dan 10% untuk <i>validation</i> data dengan 3000 <i>epoch</i>, <i>batch size</i> (64 dan 32), <i>drpooout</i> (0.5), <i>learning rate</i> 0,001 dan menggunakan 7 layer Cov serta 3 <i>layer pooling</i>. Concise-SSN mencapai skor mPA sebesar 85,13%, skor mIoU sebesar 77,34%, dan akurasi klasifikasi cacat rata-rata sebesar 96,5% pada set pengujian.</p>
9	<p>Ivan Kuric, Jaromír Klarák, Vladimír Bulej, Milan Sága, Matej Kandra, Adrián Hajdučík, & Karol Tucki/ <i>Approach to Automated Visual Inspection of Objects Based on Artificial Intelligence/</i> 2022</p>	<p>Menggunakan prinsip <i>Transfer Learning</i>, di mana <i>Convolutional Neural Networks</i> (CNN) terlatih (AlexNet) dimodifikasi dan dilatih pada sampel yang lebih kecil dari data nya sendiri untuk mempercepat pengembangan sistem deteksi cacat optik. Mendeteksi cacat yang bervariasi dalam bentuk, dimensi, dan lokasi.</p> <p>Permasalahan: Bagaimana bidang inspeksi visual otomatis terhadap objek, khususnya cacat yang terjadi pada permukaan ban mobil bisa teratasi.</p>	<p>Kelebihan: Algoritma seperti jaringan saraf konvolusional dalam, dapat digunakan untuk mengotomatiskan proses inspeksi visual, meningkatkan efisiensi dan akurasi dalam deteksi cacat.</p> <p>Kekurangan: Data masukan yang tidak memadai dapat menimbulkan tantangan dalam pengembangan dan pelatihan Algoritma, yang berpotensi menyebabkan rendahnya akurasi deteksi</p>	<p>Implementasi dan verifikasi pendekatan yang diusulkan, yang melibatkan penggunaan jaringan saraf konvolusional terlatih (<i>AlexNet</i>) dan pembelajaran transfer, menghasilkan keberhasilan deteksi cacat dengan nilai kepastian berkisar antara akurasi 85,15% hingga 99,34%. Menggunakan <i>dataset</i> sebanyak 60 gambar cacat ban, <i>epoch</i> (10), <i>learning rate</i> 0,000001, menggunakan 5 <i>layer</i> dan <i>batch</i></p>

N O	Nama Penulis/ Judul Penelitian/ Tahun Penelitian	Metode Penelitian	Kelebihan/ Kekurangan	Hasil Penelitian
		Tujuan: Merancang dan mengimplementasikan sistem inspeksi visual otomatis pada ban mobil menggunakan kecerdasan buatan.		<i>size</i> (32). Penelitian ini juga menyoroti potensi untuk melatih jenis jaringan saraf dan detektor lainnya menggunakan metodologi yang diusulkan.
10	C. Sivamani, M. Rajeswari, E. Golden Julie, Y. Harold Robinson, Vimal Shanmuganathan, Seifedine Kadry, & Yunyoung Nam7/ <i>Tyre Inspection through Multi-State Convolutional Neural Networks/</i> 2021	Mengusulkan penggunaan <i>Multi-state Convolutional Neural Networks</i> (CNN) terinspirasi oleh AlexNet dan terdiri dari lapisan Konvolusional yang bertumpuk. Permasalahan: Kecelakaan di jalan raya berpotensi menimbulkan risiko bagi kehidupan pengemudi dan orang yang lewat, dan banyak kecelakaan terjadi karena kondisi ban kendaraan yang tidak tepat setelah penggunaan jangka panjang. Tujuan: Mengembangkan model <i>Multi-state Convolutional Neural Network</i> (CNN) untuk menganalisis pola tapak ban, memprediksi keausan, dan menilai ketahanan ban.	Kelebihan: Algoritma dan penelitian dalam pembelajaran mesin, seperti <i>Convolutional Neural Networks</i> (CNN), telah menunjukkan hasil yang menjanjikan di berbagai bidang, termasuk prediksi keausan ban. Kekurangan: Performa Algoritma sangat bergantung pada kualitas dan keterwakilan kumpulan data yang digunakan untuk pelatihan, sehingga hal ini mungkin sulit diperoleh.	Menggunakan <i>dataset</i> sebanyak 50 gambar ban dengan <i>dropout</i> (0,2), <i>batch size</i> (20) <i>learning rate</i> 0,001, dengan 100 <i>epoch</i> , <i>input size</i> 50x50 dan 200x200. Model <i>Convolutional Neural Networks</i> (CNN) Multi-negara yang diusulkan mencapai persentase akurasi yang tinggi, dengan 9 set mencapai akurasi tertinggi sebesar 99,31%
11	Dongbeom Ko, Sungjoo Kang, Hyunsuk Kim, Wongok Lee,	Menggunakan model <i>Convolutional Neural Network</i>	Kelebihan: Metode pelatihan yang efisien untuk deteksi area anomali	Metodologi yang diusulkan meningkatkan rata-rata

N O	Nama Penulis/ Judul Penelitian/ Tahun Penelitian	Metode Penelitian	Kelebihan/ Kekurangan	Hasil Penelitian
	Yousuk Bae, & Jeongmin Park/ <i>Anomaly Segmentation Based on Depth Image for Quality Inspection Processes in Tire Manufacturing</i> /2021	<p>(CNN) dengan DeeplabV3+ dan mengusulkan proses empat langkah untuk deteksi area anomali berbasis pembelajaran mendalam pada gambar kedalaman ban: masukan gambar, pembuatan gambar sorotan, penumpukan gambar, dan pelatihan gambar.</p> <p>Permasalahan: Ban sering kali memiliki ukuran dan bentuk yang berbeda, sehingga sulit untuk melatih model pembelajaran mendalam.</p> <p>Tujuan: Mengembangkan metode pelatihan yang efisien untuk deteksi area anomali berbasis pembelajaran mendalam pada gambar kedalaman ban.</p>	<p>berbasis pembelajaran mendalam pada gambar kedalaman ban.</p> <p>Kekurangan: Kurang data dan masuknya kesalahan manusia mempengaruhi nilai presisi dan perolehan.</p>	<p><i>Intersection over Union</i> lebih dari 7% dan IoU untuk kesalahan vent spew lebih dari 10% dibandingkan dengan metode umum.</p> <p>Data <i>splitting</i> yang dilakukan yaitu 80% untuk <i>training</i> data, 10% untuk data <i>test</i>, dan 10% untuk <i>validation</i> data dengan menggunakan 100-1000 <i>epoch</i>, <i>batch size</i> (150), dan <i>learning rate</i> 0,001 dan 0,0001.</p> <p>Waktu yang diperlukan agar IoU rata-rata tetap stabil pada 60% berkurang sebesar dengan akurasi 80%.</p>
12	Ivan Kuric, Jaromír Klarák, Milan Sága, Miroslav Císar, Adrián Hajdučík, & Dariusz Wiecek/ <i>Analysis of the Possibilities of Tire-Defect Inspection Based on Unsupervised Learning and Deep Learning</i> / 2021.	<p>Menggunakan metode pembelajaran yang diawasi dan tidak diawasi untuk pemeriksaan kerusakan ban. Basis <i>Convolutional Neural Network</i> (CNN) dengan VGG-16 digunakan untuk klasifikasi cacat.</p> <p>Permasalahan: Terletak pada kebutuhan akan</p>	<p>Kelebihan: Algoritma, seperti <i>Convolutional Neural Network</i> (CNN), dapat digunakan untuk mengotomatiskan proses inspeksi visual, meningkatkan efisiensi dan akurasi dalam deteksi cacat.</p> <p>Kekurangan: Kurang data masukan yang unik dapat membatasi efektivitas pembelajaran</p>	<p>Menggunakan <i>dataset</i> sebanyak 5000 gambar ban dengan data <i>splitting</i> yang dilakukan yaitu 90% untuk <i>training</i> data, dan 10% untuk <i>validation</i> data dengan <i>optimizer</i> (ADAM), 20 <i>epochs</i>, <i>dropout</i> (0,5), <i>batch size</i> 96, <i>learning rate</i> 0,01, dengan <i>input size</i> 64x64,</p>

N O	Nama Penulis/ Judul Penelitian/ Tahun Penelitian	Metode Penelitian	Kelebihan/ Kekurangan	Hasil Penelitian
		<p>sistem pemeriksaan kerusakan ban yang akurat dan efisien dalam penerapan di kehidupan nyata.</p> <p>Tujuan: Pengembangan metode pengolahan data dinding samping ban, otomatisasi proses transformasi kutub menggunakan regresi polinomial, serta deteksi dan klasifikasi cacat menggunakan <i>unsupervised clustering</i> dan jaringan saraf VGG-16.</p>	<p>transfer, sehingga memerlukan konstruksi dan pelatihan jaringan saraf dalam tertentu pada kumpulan data besar.</p>	<p>VGG-16 menggunakan (13 <i>cov layer</i>, 5 <i>pooling layer</i>, 3 FCL), AlexNet menggunakan 8 <i>layer</i>, GoogLeNet menggunakan (4 <i>cov layer</i>, 3 <i>pooling layer</i>, 5 FCL).</p> <p>Keberhasilan pengembangan sistem pemeriksaan ban hibrida yang menggabungkan metode pembelajaran yang diawasi dan tidak diawasi. Sistem mencapai deteksi dan klasifikasi cacat mencapai akurasi pengenalan cacat 94% atau lebih tinggi.</p>
Roda				
13	Zheming Tong, Jie gao, & Sshuiguang Tong/ <i>A convolutional neural network approach for visual recognition in wheel production lines/</i> 2020	<p>Menggunakan metode <i>v</i> menggunakan model jaringan sisa ResNet-18 ditingkatkan dan dievaluasi berdasarkan <i>testing</i> data roda.</p> <p>Langkah-langkah penting untuk identifikasi roda meliputi akuisisi gambar, pra-pemrosesan gambar, dan klasifikasi.</p> <p>Permasalahan: Metode pengenalan tradisional terutama</p>	<p>Kelebihan: Pendekatan <i>Convolutional Neural Networks</i> (CNN) meningkatkan akurasi, ketahanan, dan kecepatan pemrosesan dalam klasifikasi jenis roda di lini produksi sebenarnya. Metode CNN memungkinkan pengenalan gambar yang sangat halus tanpa memerlukan fitur buatan yang sulit diekstraksi.</p> <p>Kekurangan: Metode CNN adalah</p>	<p>Menggunakan <i>dataset</i> sebanyak 70.000 gambar velg dengan data <i>splitting</i> yang dilakukan yaitu 80% untuk <i>training</i> data, dan 20% untuk <i>testing</i> data dengan <i>dropout</i> (0,5), <i>batch size</i> 128, <i>learning rate</i> 0,01, dengan <i>input size</i> 32x32, dan menggunakan 2 <i>layer</i>.</p> <p>Pendekatan</p>

N O	Nama Penulis/ Judul Penelitian/ Tahun Penelitian	Metode Penelitian	Kelebihan/ Kekurangan	Hasil Penelitian
		<p>didasarkan pada pencocokan fitur yang diekstraksi. Akurasi, ketahanan, dan kecepatan pemrosesan sering kali sangat terganggu produksi sebenarnya.</p> <p>Tujuan: Mengembangkan sistem pengenalan visual yang dapat secara otomatis mengklasifikasikan jenis roda pada lini produksi roda.</p>	<p>membutuhkan banyak data latih, proses pelatihan (<i>training</i>) yang memakan waktu, dan <i>overfitting</i>.</p>	<p><i>Convolutional Neural Networks</i> (CNN) yang diusulkan mencapai akurasi lebih dari 98% pada hampir 70.000 gambar roda dan menunjukkan waktu pemrosesan gambar tunggal dalam tingkat milidetik. Eksperimen menunjukkan bahwa metode ini meningkatkan model ResNet-18 dan secara efektif mengklasifikasikan jenis roda di lini produksi.</p>
Sekrup				
14	<p>Johannes Breitenbach, Isabelle Eckert, Vanessa Mahal, Hermann Baumgartl, & Ricardo Buettner/ <i>Automated Defect Detection of Screws in the Manufacturing Industry Using Convolutional Neural Networks/</i> 2022.</p>	<p>Menggunakan model <i>Convolutional Neural Network</i> (CNN) dengan VGG16 digunakan untuk mengklasifikasikan sekrup yang tidak rusak dan sekrup yang cacat dengan jenis cacat yang berbeda.</p> <p>Permasalahan: Metode kendali mutu manual bisa jadi bias, dan diperlukan metode yang andal, akurat, hemat biaya, dan cepat untuk mendeteksi cacat pada sekrup.</p> <p>Tujuan:</p>	<p>Kelebihan: <i>Convolutional Neural Networks</i> (CNNs), telah menunjukkan hasil yang menjanjikan. CNN telah terbukti akurat, hemat biaya, dan cepat dalam mengklasifikasikan sekrup yang tidak rusak dan sekrup yang cacat dengan berbagai jenis cacat.</p> <p>Kekurangan: Pelatihan model CNN memerlukan kapasitas dan waktu pemrosesan yang besar, dan pengumpulan <i>training</i> data dalam jumlah besar mungkin tidak selalu dapat dilakukan.</p>	<p>Menggunakan <i>dataset</i> sebanyak 3.354 gambar sekrup dengan data <i>splitting</i> yang dilakukan yaitu 60% untuk <i>training</i> data, 20% untuk <i>validation</i> data, dan 20% untuk <i>testing</i> data dengan 200 <i>epochs</i>, <i>dropout</i> (0,5), <i>batch size</i> 32, <i>learning rate</i> 0,001, dengan <i>input size</i> 224x224, dan menggunakan (13 <i>cov layer</i> dan 2 <i>FC layer</i>).</p> <p>Pendekatan <i>Convolutional Neural Networks</i></p>

N O	Nama Penulis/ Judul Penelitian/ Tahun Penelitian	Metode Penelitian	Kelebihan/ Kekurangan	Hasil Penelitian
		Mengembangkan metode deteksi cacat otomatis menggunakan <i>Convolutional Neural Networks</i> (CNNs) untuk sekrup di industri manufaktur.		(CNN) dengan VGG16 mencapai akurasi lebih dari 97% dalam mengklasifikasi sekrup yang tidak rusak dan sekrup yang cacat dengan berbagai jenis cacat. Pendekatan CNN terbukti akurat, hemat biaya, dan cepat dibandingkan dengan metode pengendalian kualitas manual.

Berdasarkan Tabel 2.2 *State of The Art* maka dapat disimpulkan mengenai kebaruan yang dapat diambil dari penelitian dengan topik pengukuran kualitas melalui klasifikasi. Ditemukan dari referensi-referensi terkait yang dibagi menjadi kelompok objek data pertama adalah ban dan pendukung seperti roda dan sekrup bahwa *Convolutional Neural Networks* (CNN) menjadi model yang paling banyak sekali digunakan. Adapun pengelompokan dalam ketertarikan pengambilan topik penelitian ini berdasarkan referensi penelitian terdahulu di mana diketahui pertama berdasarkan penerapan *Convolutional Neural Networks* (CNN) dalam industri, untuk menemukan cacat pada barang-barang seperti ban, roda, dan sekrup dengan teknik-teknik yang digunakan dalam penelitian jenis ini dapat dipelajari dan diterapkan pada situasi khusus produk ban karena dilatarbelakangi beberapa penelitian bahwa cacat ban seperti retak dan serpihan yang lolos dapat menimbulkan risiko keselamatan dan mempengaruhi keselamatan berkendara kendaraan. Kedua adalah penggunaan *Convolutional Neural Networks* (CNN) dalam pemeriksaan visual, yang telah banyak digunakan dalam sistem inspeksi visual untuk mendeteksi, mengklasifikasikan, dan mengidentifikasi objek dan fitur dalam gambar. Penelitian sebelumnya di bidang ini dapat memberikan wawasan tentang bagaimana *Convolutional Neural Networks* (CNN) dapat diterapkan untuk

menganalisis gambar produk ban dan melakukan pengukuran kualitas. Ketiga segmentasi citra, metode penting dalam analisis gambar untuk pengukuran kualitas adalah segmentasi gambar. Penelitian sebelumnya mengenai segmentasi gambar dapat menawarkan strategi dan teknik yang dapat digunakan untuk mengisolasi daerah yang signifikan atau bermasalah dalam foto produk ban. Keempat analisis dan klasifikasi fitur, penelitian terkait analisis dan klasifikasi ciri-ciri pada barang manufaktur dapat menjadi sumber inspirasi untuk memahami ciri-ciri yang relevan dengan produk dan bagaimana menerapkan teknik klasifikasi untuk menentukan kualitas.

Penggunaan *dataset* pada *literature* ada beberapa *splitting* data yang dilakukan, yaitu pertama (80% *training* data, 20% *validation* data), kedua (80% *training* data, 10% *validation* data, 10% *testing* data), ketiga (70% *training* data, 0% *validation* data, 10% *testing* data), keempat (60% *training* data, 20% *validation* data, 20% *testing* data). Penggunaan *batch size* (4, 10, 20, 16, 32, 64, 150, 256). Penggunaan *layer convolutional* (2, 3, 5, 7, 22), *layer pooling* (2, 3, 5). Penggunaan *learning rate* (0.01, 0.001, 0.0001, 0.000001). Penggunaan *epoch* (10, 20, 30, 100, 120, 200). Penggunaan nilai *dropout* (0.1, 0.2, 0.5). Penggunaan *input size* (256x256, 127x127, 900x900, 50x50, 64x64). Penelitian yang dilakukan oleh Vasan, Sridharan, Sreelatha, & Vaithiyanatha (2023) melakukan perbandingan dilakukan untuk *pretrained model* VGG-16, GoogLeNet, AlexNet, ResNet-50.

Permasalahan yang ditimbulkan oleh masing-masing jurnal referensi dominan seperti dalam beberapa penelitian yang menggambarkan dominansi permasalahan seperti Wang, Guo, Lu *et al* (2019) permasalahan penggunaan algoritma *Convolutional Neural Networks* (CNN) dengan arsitektur *convolutional*, *pooling*, *Fully Convolutional Network* mungkin tidak sensitif terhadap cacat kecil pada gambar ban, seperti gelembung, karena hilangnya detail yang disebabkan oleh operasi pengumpulan data. Dilanjutkan oleh penelitian Wu, Jiao, Sun *et al* (2021) penelitian ini mengakui bahwa hasil deteksi cacat gelembung sedikit buruk dan menyarankan penelitian lebih lanjut dengan menggunakan kombinasi metode deteksi langsung dan tidak langsung. Li, Fan, Zhang *et al* (2021) menyatakan meskipun algoritma ini berhasil dalam mendeteksi berbagai jenis cacat, masih

terdapat kebutuhan untuk meningkatkan deteksi pada jenis cacat tertentu, seperti gelembung yang memiliki perbedaan yang halus dari bagian sekitarnya.

Permasalahan yang terjadi masalah sensitivitas terhadap cacat kecil pada gambar ban dapat diatasi dengan menggunakan algoritma *Convolutional Neural Networks* (CNN) dengan membangun model atau kerangka kerja menggunakan Keras (sebuah *library deep learning* yang dirancang untuk mempermudah proses pengembangan dan eksperimen model jaringan saraf). Membangun model menggunakan Keras, sebenarnya mendefinisikan arsitektur model dengan menambahkan lapisan-lapisan yang sesuai. Lapisan-lapisan tersebut akan membentuk struktur atau arsitektur dari model yang kemudian dapat dilatih dan dievaluasi. Keras menyediakan berbagai model *pre-trained* yang dapat digunakan, dengan menambahkan lapisan-lapisan *convolutional*, *MaxPooling2D*, *Flatten*, dan *Dense* (*fully connected layer*) yang akan membantu model untuk mempelajari fitur-fitur yang lebih spesifik terkait dengan gambar ban dan meningkatkan sensitivitas terhadap cacat.

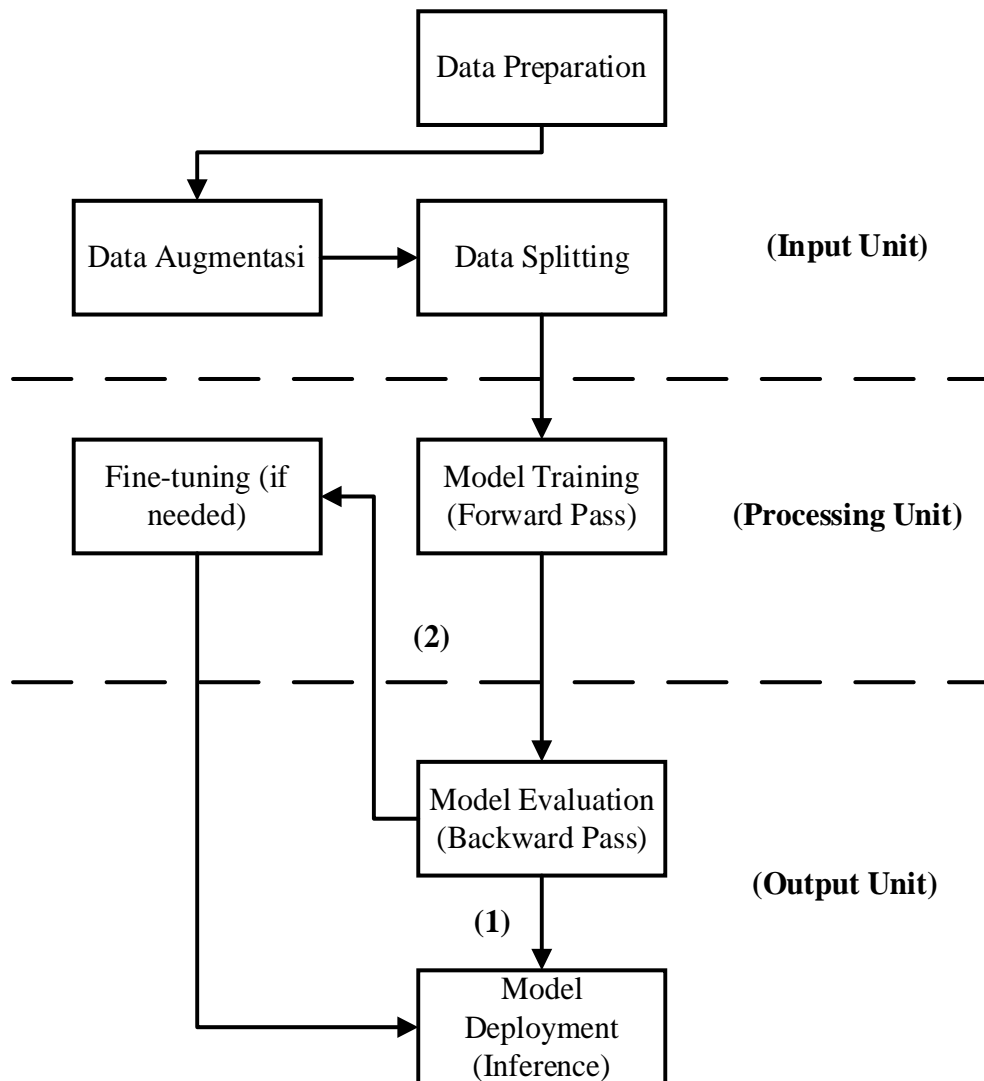
Pengembangan algoritma *Convolutional Neural Networks* (CNN) salah satunya dengan melakukan perbandingan pada komponen-komponen di dalamnya berdasarkan referensi sumber penelitian terdahulu sehingga membentuk perbandingan versi baru yang dibuat seperti melakukan perbandingan penggunaan *layer* yang sama (konvolusi dan *pooling*) mulai dari (3, 4, 5, 6) *layer*, perbandingan penggunaan seberapa banyak jumlah *epoch* (20, 50, 100), perbandingan format *splitting data* (80:20, 60:20:20, 70:20:10, 80:10:10), perbandingan penggunaan algoritma optimasi (*Stochastic Gradient Descent* (SGD), *Root Mean Square Propagation* (RMSProp) dan *Adaptive Momentum* (Adam)), dan perbandingan penggunaan *learning rate* (0.001, 0.0001, dan 0,00001). Melakukan Evaluasi dan penyesuaian di mana setelah melatih model dengan kumpulan data, maka harus menilai performanya menggunakan ukuran terkait seperti skor F1, akurasi, presisi, dan perolehan. Sehingga meningkatkan sensitivitas terhadap cacat gambar ban dapat dicapai dengan menggabungkan teknik deteksi langsung dan tidak langsung, menambahkan lebih banyak lapisan, dan menggunakan model terlatih.

BAB III

METODE PENELITIAN

3.1 Gambaran Umum Penelitian

Penelitian ini digunakan untuk mengatasi sensitivitas terhadap cacat pada gambar ban dengan melibatkan penggunaan jaringan syaraf menggunakan algoritma *Convolutional Neural Network* (CNN) dan membangun model atau kerangka kerja menggunakan Keras. Berikut adalah Gambar 3.1 Blok Diagram Gambaran Umum Penelitian.



Gambar 3.1 Blok Diagram Gambaran Umum Penelitian

Berdasarkan Gambar 3.1 Blok Diagram Gambaran Umum Penelitian maka dapat dijelaskan di blok tersebut terbagi menjadi 3 bagian yaitu bagian pertama adalah unit masukan berisikan data *preparation* di mana gambar ban dimuat, diubah menjadi format yang sesuai dipersiapkan untuk pelatihan model *Convolutional Neural Network* (CNN) seperti pemrosesan gambar ban, selanjutnya data *augmentation* di mana data dibuat lebih bervariasi dari *training* data yang ada sehingga dapat meningkatkan keberagaman *training* data tanpa harus mengambil data baru, mencakup (rotasi, pergeseran horizontal/vertikal, perbesar gambar, perubahan kecerahan gambar, sampai mengubah nilai pixel), selanjutnya data di mana *dataset* yang telah di augmentasi dan disiapkan dibagi menjadi *subset* yang berbeda untuk *training* untuk melatih model, *validation* untuk menyempurnakan model serta memvalidasi performanya selama pelatihan, dan *testing* untuk mengevaluasi kinerja model akhir. *Dataset* dibagi menjadi *training* data, *validation* data, dan *testing* data dalam proporsi tertentu.

Bagian kedua adalah unit pemrosesan yang bertindak adalah model *training* (*forward Pass*, tahap di mana *input* diproses melalui model untuk menghasilkan prediksi), tujuannya melatih model *Convolutional Neural Network* (CNN) menggunakan *dataset* pelatihan di mana data dari unit masukan diteruskan melalui jaringan neural di lakukan transformasi linier (konvolusi) dan non-linier (fungsi aktivasi) dilakukan pada data di setiap lapisan untuk menghasilkan *output* prediksi yang melibatkan komputasi di setiap neuron dan lapisan jaringan, yang merupakan inti dari proses pembelajaran dalam jaringan saraf.

Selanjutnya unit pemrosesan *Fine-tuning* tujuannya dilakukan untuk menyempurnakan model lebih lanjut setelah pelatihan awal dengan *dataset* yang lebih kecil atau lebih spesifik nantinya. Proses di dalam *Fine-tuning* menyesuaikan bobot (menggunakan kumpulan data yang lebih kecil untuk menyesuaikan bobot model untuk performa yang lebih baik), pelatihan khusus (fokus pada fitur data yang lebih relevan dengan objek).

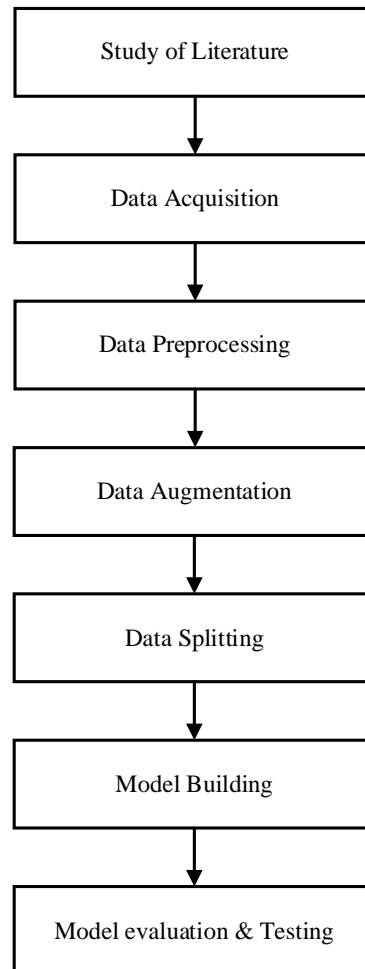
Bagian ketiga adalah unit keluaran yang bertindak ada proses model *evaluation* (*backward pass*, tahap di mana gradien (memperbarui parameter model

dalam arah yang akan mengurangi fungsi *loss*) dari fungsi *loss* (metrik yang mengukur seberapa baik atau buruk model melakukan prediksi dibandingkan nilai aktualnya) dihitung dan digunakan untuk memperbarui parameter model selama pelatihan) tujuannya mengevaluasi performa model yang dilatih dan model dievaluasi menggunakan metrik yang relevan (*accuracy*, *precision*, *recall*, dan *F1-score*) berdasarkan prediksi yang dihasilkan dari model terhadap validasi atau uji data. *Output* dari proses ini adalah tentang hasil evaluasi model, yang memberikan informasi kinerja model. Selanjutnya ada dua alur pilihan yang bisa dilakukan, alur pertama jika hasil prediksi sudah sesuai dengan keinginan maka bisa langsung masuk ke model *deployment* (*inference*), dan alur kedua jika hasil prediksi masih perlu diperbaiki pada bagian unit pemrosesan terlebih dahulu *fine tuning* untuk penggunaan data set lebih kecil (jika menunjukkan model belum mencapai performa yang diharapkan) baru masuk ke model *deployment* (*inference*) tujuannya menerapkan model terlatih untuk membuat prediksi pada data baru yang belum terlihat.

Model *deployment* (*inference*) yang telah dilatih digunakan untuk membuat prediksi pada data baru atau dalam situasi dunia nyata, tahap di mana model menerima *input* baru dan menghasilkan *output* berdasarkan pada pembelajaran yang dilakukan selama proses pelatihan dan merupakan *output* akhir dari keseluruhan proses, di mana model "mengambil keputusan" atau "membuat prediksi" berdasarkan pada pengalaman yang telah diperoleh selama pelatihan.

3.2. Tahapan Penelitian

Penelitian ini di dalamnya terdapat tahapan-tahapan yang dilakukan untuk membentuk satu kesatuan yang utuh dari awal sampai akhir dan membentuk kerangka penelitian mengenai klasifikasi pada produk ban menggunakan algoritma *Convolutional Neural Network* (CNN). Berikut Gambar 3.2 Tahapan penelitian.



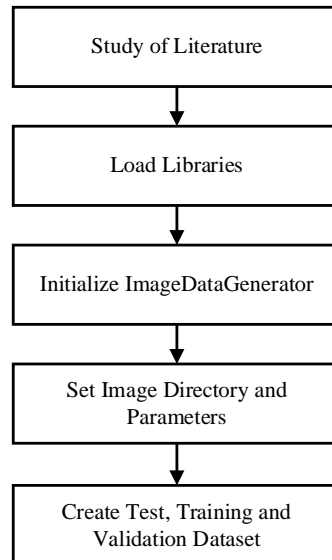
Gambar 3.2 Tahapan Penelitian

Berdasarkan Gambar 3.2 Tahapan Penelitian maka dapat dijelaskan proses yang terlibat di dalamnya ada 8 yaitu studi literatur, data *aquisition*, data *preprocessing*, data *augmentation*, *texture feature* extraction, data *splitting*, model *building*, dan model *evaluation & testing* di mana tahap ke dua sampai lima merupakan tahap proses menyiapkan sebuah data sebelum dilakukan pemodelan.

3.2.1 Studi Literatur

Tahap pertama adalah studi literatur di mana studi yang dilakukan berasal dari artikel ilmiah dan buku yang menunjang dalam menganalisis terkait dengan metode pengukuran kualitas, mengenai klasifikasi produk ban, meninjau penggunaan pembelajaran mesin algoritma *Convolutional Neural Network* (CNN) dari beberapa tahun ke belakang dalam konteks pengukuran kualitas untuk

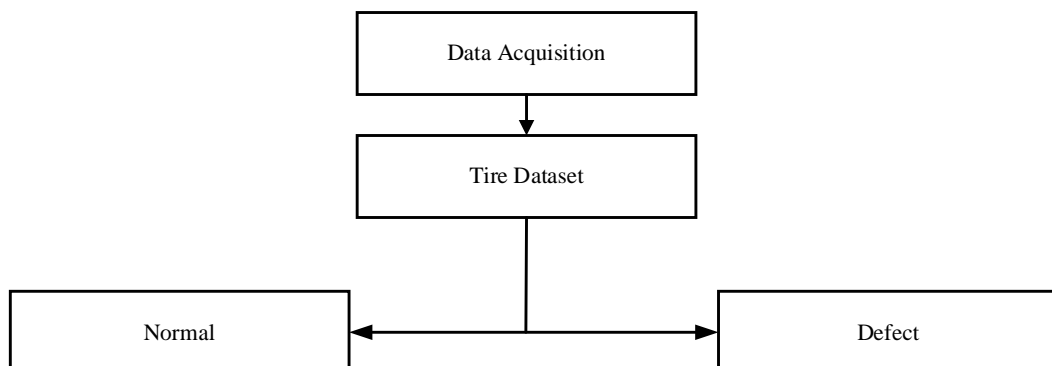
klasifikasi terhadap kondisi-kondisi produk ban. Sehingga dapat menemukan teknik terbaik yang dapat diaplikasikan pada masalah yang ada. Berikut merupakan Gambar 3.3 Tahapan *Study Literature*.



Gambar 3.3 Tahapan *Study Literature*

3.2.2 Data Aquisition

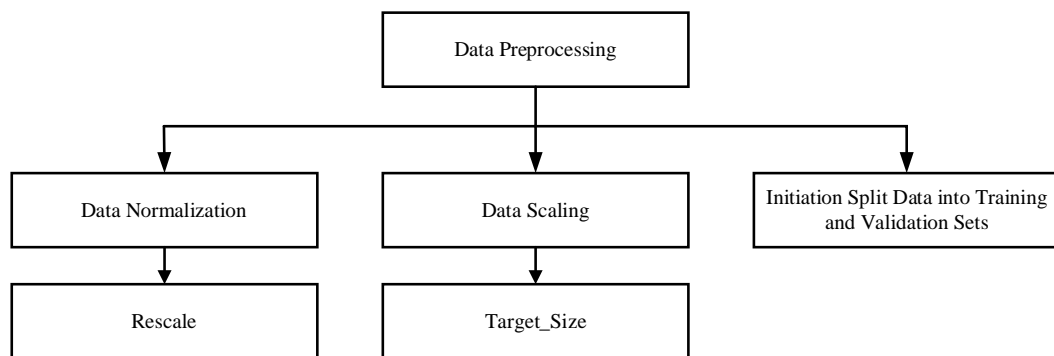
Tahap kedua adalah data *aquisition* dengan mengumpulkan kumpulan data sesuai tujuan penelitian dengan target untuk kumpulan data gambar ban untuk *training* data, *validation* data, dan *testing* data, memastikan bahwa kumpulan data tersebut memiliki varian yang secara akurat memang mewakili kondisi produk ban dan diperoleh dari sumber-sumber terpercaya. Berikut merupakan Gambar 3.4 Tahapan Data *Aquisition*.



Gambar 3.4 Tahapan Data *Aquisition*

3.2.3 Data Preprocessing

Tahap ketiga adalah data *preprocessing* melakukan pra-pemrosesan data untuk menyiapkan gambar untuk model pelatihan dan pengujian proses ini meliputi normalisasi dan penskalaan dengan fitur dalam program (*ImageDataGenerator*). Bermaksud merapikan, menata, dan menyiapkan data untuk pemeriksaan tambahan. Normalisasi data, pengkodean variabel, mengatasi nilai yang hilang, menghapus data yang tidak relevan atau hilang, dan modifikasi data lainnya untuk memenuhi persyaratan analisis adalah persiapan data. Berikut merupakan Gambar 3.5 Tahapan Data Preprocessing.



Gambar 3.5 Tahapan Data Preprocessing

3.2.4 Data Augmentation

Tahap keempat adalah data *augmentation* meningkatkan variasi dalam *dataset* dengan teknik augmentasi data, menggunakan operasi seperti rotasi, pergeseran horizontal/vertikal, perbesar gambar, perubahan kecerahan gambar, sampai mengubah nilai pixel untuk memperkaya *dataset* dan mengurangi *overfitting* (Saat disajikan dengan data baru yang belum pernah dilihat sebelumnya, performa model akan menurun drastis karena model tersebut dapat menyesuaikan diri dengan kumpulan *training* data dengan sangat efektif). Augmentasi data dilakukan dengan dua cara secara statis dan dinamis yang artinya secara statis yaitu menambah data secara fisiknya dan dinamis tidak menambah secara fisik tetapi secara kuantitas *dataset* yang dapat diakses secara fisik di komputer tidak bertambah ketika *ImageDataGenerator* digunakan pada *dataset*. Sebaliknya, pada saat *runtime* hanya menghasilkan variasi dari gambar yang sudah ada dibuat secara dinamis dan cukup bagi model untuk berlatih dari berbagai kondisi gambar ban

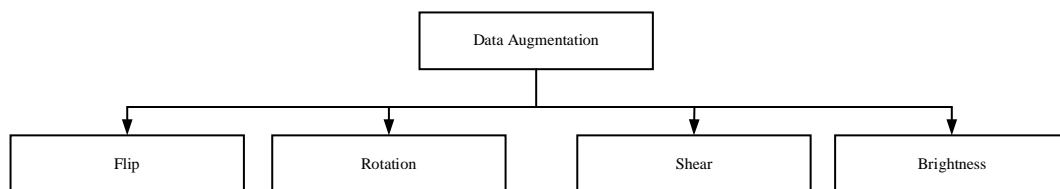
yang ada pada kenyataannya.

Secara lebih jelas nilai teknik augmentasi pertama dilakukan dengan manual menggunakan bantuan dari *website roboflow* dengan *resize* gambar menjadi 640 x 640, pada augmentasinya menggunakan model *flip* (horizontal dan vertikal), 90° pemutaran (searah jarum jam, berlawanan arah jarum jam, dan terbalik), rotasi (-45° dan 45°), *shear* ($\pm 5^\circ$ horizontal dan $\pm 5^\circ$ vertikal), *brightness* (-20% sampai 20%). Data asli pada *dataset* berjumlah 1.028 data gambar dan setelah dilakukan augmentasi secara fisik menggunakan *website roboflow* ada data yang tidak dapat diidentifikasi ada 3 gambar sehingga total gambar asli yang berhasil di *upload* dan dijadikan data asli yang tetap berjumlah 1.025 data gambar dan setelah di augmentasi bertambah menjadi 2.050 data gambar dan. Rinciannya pada data asli *training* adalah 560 gambar dan setelah dilakukan augmentasi bertambah menjadi sebanyak 1.121 gambar. Rincian data asli pada *Validation* data berjumlah 140 gambar dan setelah dilakukan augmentasi bertambah menjadi sebanyak 279 gambar. Rincian data asli pada *testing data* berjumlah 328 gambar dan setelah dilakukan augmentasi bertambah menjadi sebanyak 650 gambar. *Testing* data pada prosesnya sebenarnya tidak mengalami augmentasi karena pada proses pengujian atau evaluasi model, ingin menggunakan data asli yang sebenarnya untuk melihat kinerja model pada kasus-kasus yang belum pernah dilihat sebelumnya.

Augmentasi kedua yaitu dilakukan rotasi melakukan pemutaran gambar secara penuh dan secara acak dengan nilai 360 atau rentang nilai 0-360 derajat, kedua *width shift range*, yang menggeser gambar secara acak ke kiri atau kanan dengan nilai 0.05 atau gambar dapat digeser sampai 5% dari lebar aslinya. Ketiga *height shift range* gambar dapat digeser secara vertikal dengan nilai 0.05 atau gambar dapat digeser sampai 5% dari tinggi aslinya. Keempat *shear range* untuk menggeser gambar dengan sudut geser berlawanan arah jarum jam dengan nilai 0.05. kelima *zoom range* memperbesar gambar sebanyak 0.05 atau gambar dapat diperbesar sampai 5%. Keenam *horizontal flip* adalah memberikan variasi tambahan dengan mengubah orientasi gambar secara horizontal acak dengan keterangan nilai *true*. ketujuh *vertical flip* adalah memberikan variasi tambahan dengan mengubah orientasi gambar secara vertikal acak dengan keterangan nilai

true. Kedelapan *brightness range* mengubah atau menentukan kecerahan pada gambar secara acak dengan nilai rentan [0.75, 1.25] atau kecerahan dapat diubah mulai dari rentnag 75% sampai 125% dari kecerahan asli gambarnya. Kesembilan *rescale* mengubah nilai skala piksel 0.1 dengan membaginya setiap nilai piksel pada nilai 255, sehingga dapat membantu untuk normalisasi data. Kesepuluh *validation split* mengatur pembagian data untuk validasi dengan nilai 0.2 atau 20% data dari keseluruhan data untuk alokasi *validation* data dan 80% untuk alokasi *training* data.

Merupakan pendekatan augmentasi awal di mana sebelum data masuk ke model untuk proses pelatihan dan akan diperbesar sebelum pembagian *dataset* menjadi *batch* untuk setiap *epoch* nya, sehingga model akan dilatih menggunakan *dataset* yang telah diaugmentsi sejak awal dan seluruh augmentasi akan diterapkan pada setiap *epoch* nya dengan penggunaan ukuran *batch* 64 dengan jumlah *batch training* 36 dan validasi 10. Rinciannya data asli pada *training data* berjumlah 1.121 gambar dan setelah dilakukan augmentasi bertambah sebanyak 1.152 gambar, sehingga data pada *training data* berjumlah total menjadi 2.273 gambar. Rincian data asli pada *Validation data* berjumlah 279 gambar dan setelah dilakukan augmentasi bertambah sebanyak 320 gambar, sehingga data pada *Validation data* berjumlah total menjadi 599 gambar. *Testing* data tidak mengalami augmentasi karena pada proses pengujian atau evaluasi model, ingin menggunakan data asli yang sebenarnya untuk melihat kinerja model pada kasus-kasus yang belum pernah dilihat sebelumnya. Berikut merupakan Gambar 3.6 Tahapan Data Augmentation.



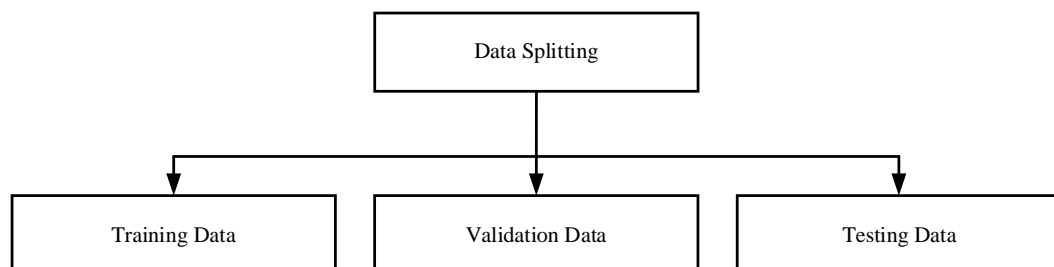
Gambar 3.6 Tahapan Data Augmentation

3.2.5 Data Splitting

Tahap kelima data *splitting* dengan membagi *file dataset* menjadi *subset training* data, *validation* data, dan *testing* data berisikan gambar ban normal dan gambar ban tidak normal sehingga *subset training* data digunakan untuk melatih

model, sedangkan *subset validation* data digunakan untuk menguji kinerja model. Sebenarnya langkah-langkah dalam proses pra-pemrosesan data yang mempersiapkan data mentah untuk digunakan dalam pelatihan model adalah tahapan yang sudah disebutkan sebelumnya data *aquisition*, data *prerocessing*, data *augmentation*, dan *splitting* data. Prosedur yang disebutkan di atas berkonsentrasi pada pengumpulan, sanitasi, pengorganisasian, dan penambahan jumlah data yang diperlukan untuk pelatihan model.

Rinciannya yaitu *file* yang tersimpan di dalam komputer total data gambar sebanyak 2.050 gambar yang dibagi menjadi dua pertama adalah *file testing* data dengan jumlah data tersimpan sebanyak 650 gambar yang dibagi menjadi *sub file* "crack" berjumlah 420 dan *sub file* "normal" berjumlah 230 data. kedua adalah *file training* data dengan jumlah data tersimpan sebanyak 1400 gambar yang dibagi menjadi sub file "crack" berjumlah 654 dan sub file "normal" berjumlah 746 data. maka ketika dilakukan data *splitting* pada program secara otomatis yang pada data augmentasi diatur menjadi pembagian 80% untuk *training* data dan 20% untuk *validation* data yaitu untuk *Train Data* sebanyak 1.121 gambar dengan 2 kelas, *validation data* sebanyak 279 gambar dengan 2 kelas, dan *Test Data* sebanyak 650 gambar dengan 2 kelas. *Testing* data bernilai tetap hal ini bertujuan agar kuantitas data awal yang telah ditentukan sebelumnya tetap terjaga dan *testing* data tidak terpengaruh oleh prosedur pemisahan. Setelah model dilatih dan divalidasi, *testing* data digunakan untuk mengevaluasi performa akhir model. Akibatnya, *testing* data tidak terbagi, dan rincian asli 648 foto masih berlaku. Berikut merupakan Gambar 3.7 Tahapan *Splitting* Data.



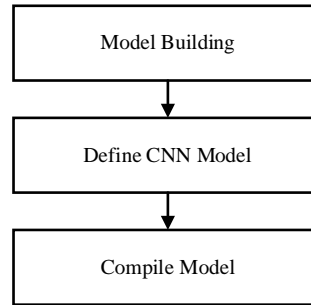
Gambar 3.7 Tahapan *Splitting* Data

3.2.6 Model Building

Tahap keenam adalah model *building* (membangun model *Convolutional Neural Network* (CNN) dengan Keras) membangun arsitektur model *Convolutional Neural Network* (CNN) menggunakan Keras, mengatur lapisan-lapisan seperti *convolutional*, *MaxPooling2D*, *Flatten*, dan *Dense* untuk membangun model. *learning rate* dalam penggunaan algoritma optimasi menggunakan *Adaptive Momentum* (Adam) untuk menghasilkan pembelajaran yang adaptif, pemilihan penggunaan *Adaptive Momentum* (Adam) jika dibandingkan dengan *learning rate* lain seperti *Stochastic Gradient Descent* (SGD) karena kecepatan pembelajaran adaptif untuk *Adaptive Momentum* (Adam) bisa secara otomatis menyesuaikan *learning rate* untuk setiap parameter dalam model klasifikasi dan sedangkan *Stochastic Gradient Descent* (SGD) memiliki *learning rate* tetap selama pelatihan model klasifikasi dan yang penentuannya dari *user* dan tidak bisa menyesuaikan *learning rate* secara otomatis berdasarkan kondisi aktual dari setiap parameter. Selanjutnya secara kestabilan dan konvergensi *Adaptive Momentum* (Adam) menyambung dari awal dapat mengubah kecepatan pembelajaran secara adaptif, sehingga membuatnya lebih stabil dan kecil kemungkinannya terjebak pada tingkat minimum lokal (nilai yang dianggap sebagai titik terendah dari *loss function* dalam model) sehingga *Adaptive Momentum* (Adam) cenderung mencapai konvergensi (tingkat kinerja yang diharapkan) lebih cepat dan andal dalam berbagai keadaan, sedangkan *Stochastic Gradient Descent* (SGD) mungkin lebih *stuck* pada nilai minimum atau terjebak pada nilai minimum lokal yang disebabkan oleh kemungkinan bergantung pada seberapa tepat kecepatan pembelajaran dipilih, kecepatan pembelajaran yang tetap dapat membuat model mencapai konvergensi terlalu cepat atau terlalu lambat.

Adapun melakukan pendekatan kedua penambahan data ketika masuk ke model *building* dan terjadi proses pemodelan setelah menggunakan *epoch*. Augmentasi data diterapkan setelah data melewati beberapa *epoch* selama proses pelatihan, sehingga variasi data yang dihasilkan akan berbeda-beda pada setiap *epoch* dan model dapat terus-menerus terlatih dengan variasi data yang lebih besar. Menggunakan 100 *epoch*, sehingga total *training* data yang diproses menjadi

230.400 gambar dan *validation* data menjadi 6.400 gambar. Sehingga jumlah data yang diproses selama pelatihan menjadi sangat besar dan pada akhirnya nanti akan menyiapkan model kompilasi dalam mengatur pengoptimal (Adam), fungsi kerugian (*biner crossentropy*), dan metrik evaluasi (akurasi). Berikut merupakan Gambar 3.8 Tahapan *Building Model*.



Gambar 3.8 Tahapan *Building Model*

Berdasarkan hasil analisis sebelumnya maka dapat diketahui untuk jumlah data asli (*training data*) adalah 1.121, jumlah data asli (*validation data*) adalah 279, jumlah data asli (*training data*) setelah augmentasi adalah 1152, jumlah data asli (*validation data*) setelah augmentasi adalah 320, jumlah *epoch* yang digunakan sebanyak 100, dan ukuran *batch* adalah 64. Berikut merupakan perhitungan manualnya ketika masuk ke model *building* dan terjadi proses pemodelan setelah menggunakan *epoch*.

1. Jumlah *batch* per *epoch* untuk *training data*.

$$\begin{aligned}
 \text{Step per epoch} &= \left(\frac{\text{Jumlah Data Train} + \text{Jumlah Data Augmentasi Train}}{\text{Batch Size}} \right) \\
 &= \left(\frac{1121 + 1152}{64} \right) \\
 &= \left(\frac{2273}{64} \right) \\
 &= 35,5156 \approx 36 \text{ batch}
 \end{aligned}$$

2. Jumlah *batch* per *epoch* untuk *validation data*.

$$\begin{aligned}
 \text{Step per epoch} &= \left(\frac{\text{Jumlah Data Valid} + \text{Jumlah Data Augmentasi Valid}}{\text{Batch Size}} \right) \\
 &= \left(\frac{279 + 320}{64} \right)
 \end{aligned}$$

$$= \left(\frac{599}{64} \right)$$

$$= 9,3593 \approx 10 \text{ batch}$$

3. Total jumlah data setelah augmentasi untuk semua *epoch*.

a. *Training Data*

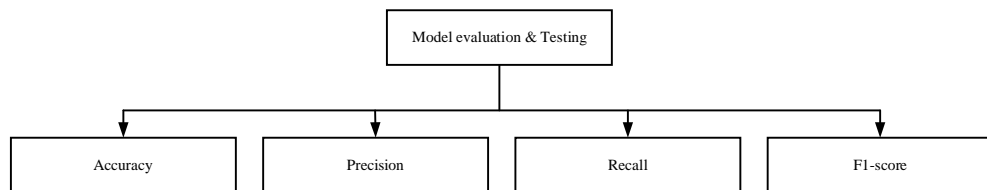
$$\begin{aligned} \text{Total Training data} &= \text{Jumlah Batch} \times \text{Batch Size} \times \text{Jumlah Epochs} \\ &= 36 \times 64 \times 100 \\ &= 230.400 \end{aligned}$$

b. *Validation Data*

$$\begin{aligned} \text{Total Validation data} &= \text{Jumlah Batch} \times \text{Batch Size} \times \text{Jumlah Epochs} \\ &= 10 \times 64 \times 100 \\ &= 64.000 \end{aligned}$$

3.2.7 Model Evaluation & Testing

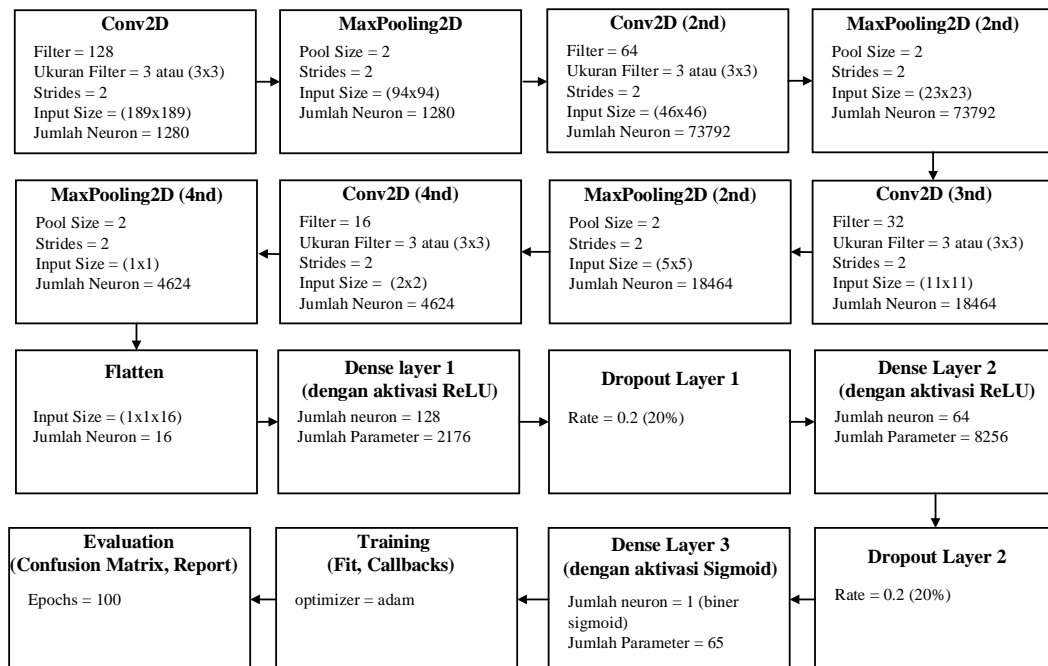
Tahap kedelapan adalah model *evaluation & testing* digunakan sebagai bahan terusan pada model *building* yang dibuat untuk melakukan evaluasi performanya dengan menggunakan bagian pengujian, dan parameter yang digunakan pada metrik evaluasi seperti akurasi, presisi, *recall*, dan *F1-score*. Berikut merupakan Gambar 3.9 Tahapan Model *Evaluation & Testing*.



Gambar 3.9 Tahapan Model *Evaluation & Testing*

3.3 Arsitektur Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) yang dibangun menggunakan model atau kerangka kerja yang pada dasarnya menggunakan Keras dan juga *tensorflow* dengan menambahkan beberapa model lapisan-lapisan seperti lapisan *convolutional* (*Conv2D*), lapisan *pooling* (*MaxPooling2D*), *Flatten*, dan lapisan *fully connected* (*Dense*). Berikut merupakan Gambar 3.10 Tahapan *Convolutional Neural Network* (CNN) dengan Model Keras.



Gambar 3.10 Tahapan *Convolutional Neural Network* (CNN) dengan Model Keras

Berdasarkan Gambar 3.10 Tahapan *Convolutional Neural Network* (CNN) dengan Model Keras maka dapat dijelaskan mulai dari yang mencakup lapisan-lapisan konvolusi yang telah dilatih pada *dataset* besar seperti *ImageNet* untuk mengekstrak fitur dari gambar-gambar, penggunaan *image size* diatur dengan (379, 379) *batch size* 64, *kernel size* 3, *strides* (2 untuk *cov2d* dan 2 untuk *maxpooling2d*), dan *pool size* 2. Selanjutnya *Conv2D* yang merupakan *convolutional layer* pertama yang berfungsi untuk mengekstrak fitur-fitur visual dari gambar. Filter *convolutional layer* pertama yang berfungsi untuk mengekstrak fitur-fitur visual diterapkan pada gambar untuk menghasilkan fitur-fitur yang lebih abstrak, formula untuk mengetahui jumlah *training* datanya dengan. Selanjutnya *MaxPooling2D* di mana tahap *pooling* digunakan untuk mengurangi dimensi spasial dari setiap *feature map* yang dihasilkan oleh layer sebelumnya. *Max pooling* memilih nilai maksimum di dalam jendela *pooling* untuk mengurangi ukuran fitur dan mempertahankan informasi penting. Selanjutnya *Conv2D* dan *MaxPooling2D* diulang sampai 4 layer karena untuk terus mengekstrak fitur-fitur yang semakin kompleks dari gambar. Selanjutnya *flatten* (digunakan untuk mengubah tensor multi-dimensi menjadi tensor satu dimensi) di mana setelah serangkaian *layer* konvolusi dan *pooling*, masukan dari *layer* terakhir perlu diubah menjadi vektor

tunggal sebelum dimasukkan ke dalam *layer dense*. *Flatten layer* melakukan hal ini dengan mengubah matriks *output* menjadi *array* satu dimensi.

Selanjutnya *dense layers* (lapisan *dense* digunakan sebagai lapisan *output* dalam model klasifikasi, di mana jumlah neuron dalam lapisan *output* sesuai dengan jumlah kelas yang harus diprediksi) di mana ada tiga lapisan *dense* ditambahkan, dengan fungsi pertama dan kedua menggunakan ReLu sebagai $f(x) = \max(0, x)$ yang artinya menunjukkan bahwa keluarannya nol jika masukannya negatif atau nol dan *output* x jika masukannya positif dengan 128 unit neuron dan pada *dense* kedua 64 unit neuron karena tugasnya mengurangi dimensi representasi pada lapisan *dense* pertama maka model dapat mempelajari pola yang lebih rumit dan mendalam dari data dengan menambahkan lapisan yang lebih padat, yang dapat meningkatkan performa model dalam tugas klasifikasi gambar. Lapisan *dense* ketiga dengan fungsi aktivasi sigmoid untuk *output biner*, dengan menunjukan kelas prediksi dari gambar yaitu normal atau *crack*. Di antara tiga lapisan *dense* di ikuti dengan lapisan *dropout* (untuk mencegah *overfitting* di mana model pembelajaran mesin terlalu menghafal pola dari *training* data yang tersedia, sehingga kinerjanya menurun secara signifikan saat diuji dengan data baru yang tidak dilihat sebelumnya) juga dimasukkan setelah setiap lapisan *dense* untuk mencegah *overfitting* dengan secara acak menonaktifkan sebagian unit sebanyak 0.2 atau 20% dari neuron selama pelatihan.

Selanjutnya *training* di mana model diterapkan pada *training* data dengan menggunakan metode *fit* dan *callback*. Model *fit* digunakan untuk melatih model dengan *training* data dan model *callback* menggunakan "*modelCheckpoint*" untuk menyimpan model terbaik selama pelatihan berkaitan dengan performa pada *validation* data mengontrol proses pelatihan. Terakhir *evaluation* di mana performa model pada *testing* data dinilai menggunakan hasil klasifikasi dan *confusion matrix* untuk memahami kinerjanya *testing* data.

Banyaknya parameter atau bobot dan jumlah data yang harus dipelajari selama pelatihan bergantung pada jumlah neuron pada lapisan. Jumlah data yang harus dipelajari model selama pelatihan tercermin dalam jumlah parameter ini. Berikut merupakan perhitungan dalam mengetahui total neuron yang dikerjakan

oleh setiap lapisan.

1. *First Conv2D*

$$\begin{aligned}
 \text{Total Neuron} &= (\text{Ukuran Filter} \times \text{Jumlah Channel Input} + 1) \times \text{Filter} \\
 &= (3 \times 3 \times 1 + 1) \times 128 \\
 &= (10) \times 128 \\
 &= 1280
 \end{aligned}$$

Kedalaman gambar yang diproses lapisan konvolusi sebenarnya ditunjukkan oleh jumlah saluran masukan. Tiga saluran merah, hijau, dan biru membentuk sebuah gambar jika diwarnai, artinya ada tiga saluran masukan. Karena kata *grayscale* digunakan untuk mendeskripsikan gambar ini, hanya ada satu saluran warna dan bernilai 1. Sehingga jumlah neuronnya 1280, yang berarti ada 1280 parameter yang harus dipelajari selama pelatihan dan akan mencerminkan jumlah data yang harus dipelajari ketika pelatihan. Selanjutnya adalah dalam penentuan ukuran spasialnya setiap filter diubah menjadi setengah dari ukuran *input* nya (379x379) menjadi (189x189) sebagai berikut.

$$\begin{aligned}
 &= \frac{\text{Input Size} - \text{Kernel Size}}{\text{Stride}} + 1 \\
 &= \frac{379 - 3}{1} + 1 \\
 &= \frac{376}{2} + 1 \\
 &= 189
 \end{aligned}$$

2. *First Maxpooling2D*

Tidak ada parameter baru yang ditambahkan, dan jumlah neuron (dalam contoh ini, lapisan konvolusi pertama) tetap sama. Setiap filter diubah menjadi setengah dari ukuran inputnya (189x189) menjadi (94x94) dan jumlah neuronnya 1280 mengikuti lapisan konvolusi pertama. Bisa juga menggunakan rumus seperti berikut untuk mengetahui dimensi atau ukuran filter.

$$\begin{aligned}
 &= \frac{\text{Input Size} - \text{Pool Size}}{\text{Stride}} + 1 \\
 &= \frac{189 - 2}{2} + 1
 \end{aligned}$$

$$\begin{aligned}
&= \frac{187}{2} + 1 \\
&= 94
\end{aligned}$$

3. *Second Cov2D*

$$\begin{aligned}
\text{Total Neuron} &= (\text{Ukuran Filter} \times \text{Jumlah Channel Input} + 1) \times \text{Filter} \\
&= (3 \times 3 \times 128 + 1) \times 64 \\
&= (1153) \times 128 \\
&= 73792
\end{aligned}$$

Jumlah *channel input* menggunakan jumlah filter dari *convolutional* layer sebelumnya bukan lagi *channel input* di awal. Sehingga jumlah neuronnya 73792, yang berarti ada 73792 parameter yang harus dipelajari selama pelatihan dan akan mencerminkan jumlah data yang harus dipelajari ketika pelatihan. Selanjutnya adalah dalam penentuan ukuran spasialnya setiap filter diubah menjadi setengah dari ukuran *input* nya (94x94) menjadi (46x46) sebagai berikut.

$$\begin{aligned}
&= \frac{\text{Input Size} - \text{Kernel Size}}{\text{Stride}} + 1 \\
&= \frac{94 - 3}{2} + 1 \\
&= \frac{91}{2} + 1 \\
&= 46
\end{aligned}$$

4. *Second Maxpooling2D*

Tidak ada parameter baru yang ditambahkan, dan jumlah neuron (dalam contoh ini, lapisan konvolusi kedua) tetap sama. Setiap filter diubah menjadi setengah dari ukuran inputnya (46x46) menjadi (23x23) dan jumlah neuronnya 73792 mengikuti lapisan konvolusi kedua. Bisa juga menggunakan rumus seperti berikut untuk mengetahui dimensi atau ukuran filter.

$$\begin{aligned}
&= \frac{\text{Input Size} - \text{Pool Size}}{\text{Stride}} + 1 \\
&= \frac{46 - 2}{2} + 1
\end{aligned}$$

$$\begin{aligned}
&= \frac{44}{2} + 1 \\
&= 23
\end{aligned}$$

5. *Third Cov2D*

$$\begin{aligned}
\text{Total Neuron} &= (\text{Ukuran Filter} \times \text{Jumlah Channel Input} + 1) \times \text{Filter} \\
&= (3 \times 3 \times 64 + 1) \times 32 \\
&= (577) \times 32 \\
&= 18464
\end{aligned}$$

Jumlah *channel input* menggunakan jumlah filter dari *convolutional* layer sebelumnya bukan lagi *channel input* di awal. Sehingga jumlah neuronnya 18464, yang berarti ada 18464 parameter yang harus dipelajari selama pelatihan dan akan mencerminkan jumlah data yang harus dipelajari ketika pelatihan. Selanjutnya adalah dalam penentuan ukuran spasialnya setiap filter diubah menjadi setengah dari ukuran *input* nya (23x23) menjadi (11x11) sebagai berikut.

$$\begin{aligned}
&= \frac{\text{Input Size} - \text{Kernel Size}}{\text{Stride}} + 1 \\
&= \frac{23 - 3}{2} + 1 \\
&= \frac{20}{2} + 1 \\
&= 11
\end{aligned}$$

6. *Third Maxpooling2D*

Tidak ada parameter baru yang ditambahkan, dan jumlah neuron (dalam contoh ini, lapisan konvolusi ketiga) tetap sama. Setiap filter diubah menjadi setengah dari ukuran inputnya (11x11) menjadi (5x5) dan jumlah neuronnya 18464 mengikuti lapisan konvolusi ketiga. Bisa juga menggunakan rumus seperti berikut untuk mengetahui dimensi atau ukuran filter.

$$\begin{aligned}
&= \frac{\text{Input Size} - \text{Pool Size}}{\text{Stride}} + 1 \\
&= \frac{11 - 2}{2} + 1
\end{aligned}$$

$$\begin{aligned}
&= \frac{9}{2} + 1 \\
&= 5
\end{aligned}$$

7. *Fourth Cov2D*

$$\begin{aligned}
\text{Total Neuron} &= (\text{Ukuran Filter} \times \text{Jumlah Channel Input} + 1) \times \text{Filter} \\
&= (3 \times 3 \times 32 + 1) \times 16 \\
&= (289) \times 16 \\
&= 4624
\end{aligned}$$

Jumlah *channel input* menggunakan jumlah filter dari *convolutional* layer sebelumnya bukan lagi *channel input* di awal. Sehingga jumlah neuronnya 4624, yang berarti ada 4624 parameter yang harus dipelajari selama pelatihan dan akan mencerminkan jumlah data yang harus dipelajari ketika pelatihan. Selanjutnya adalah dalam penentuan ukuran spasialnya setiap filter diubah menjadi setengah dari ukuran *input* nya (5x5) menjadi (2x2) sebagai berikut.

$$\begin{aligned}
&= \frac{\text{Input Size} - \text{Kernel Size}}{\text{Stride}} + 1 \\
&= \frac{5 - 3}{2} + 1 \\
&= \frac{1}{2} + 1 \\
&= 1.5 \approx 2
\end{aligned}$$

8. *Fourth Maxpooling2D*

Tidak ada parameter baru yang ditambahkan, dan jumlah neuron (dalam contoh ini, lapisan konvolusi keempat) tetap sama. Setiap filter diubah menjadi setengah dari ukuran inputnya (2x2) menjadi (1x1) dan jumlah neuronnya 4624 mengikuti lapisan konvolusi keempat. Bisa juga menggunakan rumus seperti berikut untuk mengetahui dimensi atau ukuran filter.

$$\begin{aligned}
&= \frac{\text{Input Size} - \text{Pool Size}}{\text{Stride}} + 1 \\
&= \frac{1 - 2}{2} + 1 \\
&= \frac{1}{2} + 1
\end{aligned}$$

$$= 0.5 \approx 1$$

9. *Flatten*

Tidak mengubah parameter yang ada karena fungsi flatten hanya mengubah matriks multidimensi menjadi vektor tunggal berdasarkan hasil dari jumlah filter pada lapisan lima conv2d yaitu 8 dan maxpooling2D dengan ukuran inputnya (1x1) sehingga menjadi matriks multidimensi (1, 1, 16) diubah menjadi nilai vektor tunggal dengan panjang 16 atau menjadi jumlah neuron sebanyak 16.

10. *Dense Layer 1*

$$\begin{aligned}\text{Total Neuron} &= (\text{Jumlah Neuron Input} + 1) \times \text{Jumlah Neuron Output} \\ &= (16 + 1) \times 128 \\ &= (17) \times 128 \\ &= 2176\end{aligned}$$

11. *Dropout Layer 1*

Menggunakan 0.2 yang artinya sebanyak 20% dari neuron dalam *dense layer 1* akan dinonaktifkan secara acak.

12. *Dense Layer 2*

$$\begin{aligned}\text{Total Neuron} &= (\text{Jumlah Neuron Input} + 1) \times \text{Jumlah Neuron Output} \\ &= (128 + 1) \times 64 \\ &= (129) \times 64 \\ &= 8256\end{aligned}$$

13. *Dropout Layer 2*

Menggunakan 0.2 yang artinya sebanyak 20% dari neuron dalam *dense layer 2* akan dinonaktifkan secara acak.

14. *Dense Layer 2*

$$\begin{aligned}\text{Total Neuron} &= (\text{Jumlah Neuron Input} + 1) \times \text{Jumlah Neuron Output} \\ &= (64 + 1) \times 1 \\ &= (65) \times 1 \\ &= 65\end{aligned}$$

Ketika dimensi spasial (tinggi dan lebar) dikurangi menggunakan operasi lapisan *pooling* seperti *maxpooling*, jumlah neuron di setiap lapisan *pooling* akan menurun. Misalnya, dimensi spasial setiap filter (tinggi dan lebar) di lapisan

maxpooling disesuaikan menjadi setengah dari dimensi masukannya. Karena hanya separuh dari masukan yang diproses lebih lanjut, hal ini juga menyebabkan berkurangnya jumlah neuron pada lapisan tersebut. Sedangkan penurunan pada *dense* terjadi karena penentuan jumlah neuron.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Implementasi Keras

Dalam implementasinya *Convolutional Neural Network* (CNN) dengan Keras menggunakan perpustakaan yang diperlukan untuk bekerja dengan data sebelum melakukan analisis dan pemodelan, di mana terbagi menjadi 4 bagian di dalamnya. Pertama bagian analisis menggunakan *pandas* yang digunakan untuk membaca, mengolah, dan menganalisis data dalam bentuk *DataFrame* dan *numpy* menyediakan dukungan untuk operasi matematika dan *array* numerik. Kedua bagian visualisasi menggunakan *matplotlib.pyplot* dasar untuk *plotting* grafik (x, y) di Python dan *seaborn* menyediakan antarmuka yang lebih tinggi untuk membuat plot yang lebih cantik dan informatif. Ketiga bagian membangun model *neural network* dan *convolutional neural network* menggunakan model Keras mulai dari augmentasi data, menentukan kelas *sequential* membuat model *neural network* secara lebih beraturan, memasukan semua kelas lapisan yang tersedia di Keras seperti *convolutional* (*Conv2D*), *MaxPooling2D*, *Flatten*, dan *Dense*. Kelima bagian evaluasi untuk tugas-tugas pembelajaran mesin menggunakan fungsi *confusion_matrix*, dan *classification_report* digunakan untuk mengevaluasi kinerja model klasifikasi.

4.2 Persiapan Dataset

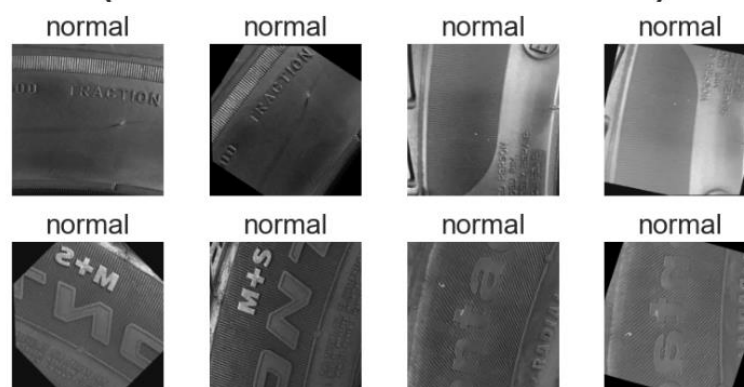
Dataset ini diperlukan karena sebagai inti untuk bahan dalam mengklasifikasi keadaan dengan objek data ini adalah ban yang dibentuk ke dalam *dataset folder* dan dibagi menjadi dua *sub folder* yaitu *testing* data dan *training* data dan masing-masing memiliki *sub foldernya* sendiri berupa data *crack* dan normal. Penetapan nilai lain di dalamnya yaitu ukuran gambar adalah 379 x 379 *pixel*, ukuran *batch* atau penentuan jumlah gambar yang akan diproses sebanyak 64, kecepatan (1,2,3) memastikan bahwa augmentasi data yang diterapkan pada gambar akan konsisten dari satu iterasi pelatihan ke iterasi berikutnya, warna *grayscale* dengan hitam dan

putih atau *ke abuan* bertujuan untuk fokus menyoroti struktur dan konter gambar tanpa distraksi dari warna serta efisiensi ruang penyimpanan data gambar, menentukan kelas target yang terdiri dari kelas 0 sebagai normal dan kelas 1 sebagai *Cracked*, dan nilai *shuffle* adalah benar untuk memastikan model mempelajari gambar yang normal dan cacat secara bergantian.

4.3 Augmentasi Data

Proses di dalamnya melibatkan augmentasi data yang digunakan untuk memberikan variasi lebih pada *dataset* pelatihan dengan mengubah secara acak gambar-gambar di dalam *dataset* seperti rotasi untuk secara acak memutar gambar 0-360 derajat, pergeseran horizontal/vertikal diatur rentang pergeseran gambar dapat digeser sebanyak 5% dari lebar dan tinggi, perbesar gambar dengan rentang sebanyak 5%, perubahan kecerahan gambar diatur rentang perubahan kecerahan secara acak pada gambar 75%-125%, sampai mengubah nilai pixel rentang antara 0 dan 1 untuk memperkaya *dataset* dan mengurangi *overfitting*. Augmentasi secara fisik dengan manual menggunakan bantuan dari *website roboflow* dengan *resize* gambar menjadi 640 x 640, pada augmentasinya menggunakan model *flip* (horizontal dan vertikal), 90° pemutaran (searah jarum jam, berlawanan arah jarum jam, dan terbalik), rotasi (-45° dan 45°), *shear* ($\pm 5^\circ$ horizontal dan $\pm 5^\circ$ vertikal), *brightness* (-20% sampai 20%). Berikut merupakan Gambar 4.1 Pelatihan Tanpa Augmentasi Data.

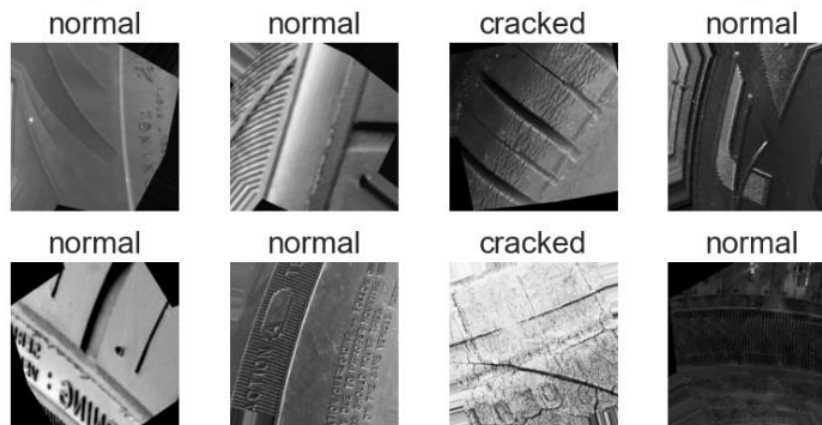
GAMBAR UJI BATCH PERTAMA (TANPA AUGMENTASI DATA)



Gambar 4.1 Contoh Data Citra Ban Pelatihan Tanpa Augmentasi Data

Berdasarkan Gambar 4.1 Pelatihan Tanpa Augmentasi Data bahwa gambar memiliki kurang variasi dan mengidentifikasi bahwa gambar tersebut adalah dalam keadaan normal semua karena *dataset* uji dalam kondisi tidak dimodifikasi untuk menciptakan variasi baru dalam gambar, berbeda dengan ketika menggunakan augmentasi data yang dapat mengidentifikasi data tidak normal sehingga terdapat variasi dari 2 kelas yaitu normal dan *crack* berasal dari berbagai variasi gambar ketika label diekstraksi dari *dataset* pelatihan yang telah diubah. Berikut Gambar 4.2 Pelatihan Tanpa Augmentasi Data.

GAMBAR PELATIHAN BATCH PERTAMA (DENGAN AUGMENTASI DATA)



Gambar 4.2 Contoh Data Citra Ban dengan Augmentasi Data

4.4 Model *Sequential*

Salah satu tipe model di perpustakaan Keras yang memungkinkan pemodelan *sequential* yaitu, penambahan lapisan secara berurutan satu per satu adalah *sequential* hasil dari *Convolutional Neural Network* (CNN). Berikut adalah hasil Tabel 4.1 Model *Sequential*.

Tabel 4.1 Model *Sequential*

<i>Layer (type)</i>	<i>Output Shape</i>	Param
(Conv2D) 1	(189, 189, 128)	1280
(Max_pooling2d) 1	(94, 94, 128)	0
(Conv2D) 2	(46, 46, 64)	73792
(Max_pooling2d) 2	(23, 23, 64)	0
(Conv2D) 3	(11, 11, 32)	18464
(Max_pooling2d) 3	(5, 5, 32)	0
(Conv2D) 4	(2, 2, 16)	4624

<i>Layer (type)</i>	<i>Output Shape</i>	Param
(Max_pooling2d) 4	(1, 1, 16)	0
(Flatten)	(16)	0
(Dense) 1	(128)	2176
(Dropout) 1	(128)	0
(Dense) 2	(64)	8256
(Dropout) 2	(64)	0
(Dense) 3	(1)	65

Berdasarkan Tabel 4.1 Model *Sequential* maka dapat dijelaskan bahwa terbagi menjadi jenis pertama *convolutional layer* (Conv2D) menghasilkan *output shape* berupa tensor tiga dimensi dengan komposisi dimensi tinggi 189 piksel, lebar 189 piksel, dan jumlah filter sebanyak 128 filter pada *ouput* tensor, selanjutnya param atau jumlah parameter yang dipelajari oleh setiap lapisan berbeda dan untuk Conv2D pertama yaitu 1280 yang artinya lapisan tersebut dapat mengekstrak 128 pola atau fitur berbeda dari gambar *input*. Kedua *pooling layer* (MaxPooling2D) menghasilkan *output shape* untuk mengurangi dimensi tinggi dan lebar dari Conv2D dengan komposisi dimensi menjadi lebih kecil tinggi 94 piksel, lebar 94 piksel, dan jumlah filter sebanyak 128 filter, selanjutnya param atau jumlah parameter yang dipelajari 0 karena tidak ada operasi perhitungan yang terjadi di dalamnya. *Convolutional layer* (Conv2D) dan *pooling layer* (MaxPooling2D) diulangi sampai 4 kali sampai dimensi yang dihasilkan lebih kecil sambil meningkatkan representasi fitur yang lebih tinggi.

Ketiga lapisan *flatten* menghasilkan *output shape* mengubah tensor multi dimensi menjadi vektor satu dimensi sebelum diberikan ke lapisan *dense* dengan jumlah unit neuron sebanyak 16, selanjutnya param atau jumlah parameter yang dipelajari 0 karena hanya melakukan pembentukan ulang tensor. Keempat *fully connected layer* (*dense*) di mana *output shape* berupa vektor satu dimensi dengan jumlah neuron *dense* 1 adalah 128, *dense* 2 adalah 64, dan *dense* 3 yang ditentukan dalam bentuk nilai biner yaitu bernilai 1, selanjutnya param atau jumlah parameter yang dipelajari adalah 65. Kelima lapisan *dropout* ditampilkan karena hanya digunakan untuk mengurangi *overfitting* dan memiliki nilai *output shape* yang sama dengan lapisan *dense*.

Param ini secara lebih jelas merupakan jumlah parameter yang digunakan pada setiap lapisan, dalam konteks ini, bobot dan bias yang harus dipelajari selama proses pelatihan model disebut sebagai parameter. Jumlah parameter mempengaruhi kecepatan pelatihan, penggunaan memori, serta kompleksitas dan kapasitas model, menjadikannya faktor yang penting. Kemungkinan *overfitting* meningkat seiring dengan banyak jumlah parameter yang digunakan dalam model, yang juga membuatnya lebih kompleks. Ketika jumlah parameter yang sedikit berpotensi membatasi kemampuan model untuk mengidentifikasi pola rumit dalam data.

4.5 *Confusion Matrix*

Merupakan Instrumen untuk mengevaluasi kinerja model klasifikasi yang memungkinkan mengetahui seberapa baik model bekerja dengan membandingkan hasil yang diharapkan dari model dengan label *testing* data yang sebenarnya menggunakan *Convolutional Neural Network* (CNN). Berikut adalah Tabel 4.2 *Confusion Matrix* untuk *Train data*.

Tabel 4.2 *Confusion Matrix* untuk *Train Data*

		Prediksi	
		normal	<i>Cracked</i>
Aktual	normal	354	243
	<i>Cracked</i>	296	228

Berdasarkan Tabel 4.2 *Confusion Matrix* untuk *Train Data* maka dapat dijelaskan terdapat empat sel yang mewakili empat kemungkinan kombinasi prediksi dan label sebenarnya dengan pembagian menjadi nilai *True Negative* (TN), *False Positive* (FP), *False Negative* (FN), dan *True Positive* (TP). Melihat nilai *confusion matrix*, maka nilai *True Negative* (TN) dengan nilai 228 yaitu berarti jumlah gambar yang benar-benar diklasifikasikan sebagai "normal" oleh model dan memang benar-benar "normal" berdasarkan label sebenarnya. Nilai *False Positive* (FP) dengan nilai 296 yaitu berarti jumlah gambar yang salah diklasifikasikan sebagai "normal" oleh model, padahal sebenarnya "*Cracked*". Nilai *False Negative* (FN) dengan nilai 243 yaitu berarti jumlah gambar yang salah diklasifikasikan sebagai "*Cracked*" oleh model, padahal sebenarnya "normal". Nilai *True Positive*

(TP) dengan nilai 354 yaitu berarti jumlah gambar yang benar-benar diklasifikasikan sebagai "*Cracked*" oleh model dan memang benar-benar "*Cracked*" berdasarkan label sebenarnya. Berikutnya adalah Tabel 4.3 *Confusion Matrix* untuk *Validation Data*.

Tabel 4.3 *Confusion Matrix* untuk *Validation Data*

		Prediksi	
		normal	<i>Cracked</i>
Aktual	normal	71	78
	<i>Cracked</i>	53	77

Berdasarkan Tabel 4.3 *Confusion Matrix* untuk *Validation data* maka dapat dijelaskan terdapat empat sel yang mewakili empat kemungkinan kombinasi prediksi dan label sebenarnya dengan pembagian menjadi nilai *True Negative* (TN), *False Positive* (FP), *False Negative* (FN), dan *True Positive* (TP). Melihat nilai *confusion matrix*, maka nilai *True Negative* (TN) dengan nilai 77 yaitu berarti jumlah gambar yang benar-benar diklasifikasikan sebagai "normal" oleh model dan memang benar-benar "normal" berdasarkan label sebenarnya. Nilai *False Positive* (FP) dengan nilai 53 yaitu berarti jumlah gambar yang salah diklasifikasikan sebagai "normal" oleh model, padahal sebenarnya "*Cracked*". Nilai *False Negative* (FN) dengan nilai 78 yaitu berarti jumlah gambar yang salah diklasifikasikan sebagai "*Cracked*" oleh model, padahal sebenarnya "normal". Nilai *True Positive* (TP) dengan nilai 71 yaitu berarti jumlah gambar yang benar-benar diklasifikasikan sebagai "*Cracked*" oleh model dan memang benar-benar "*Cracked*" berdasarkan label sebenarnya. Berikutnya adalah Tabel 4.4 *Confusion Matrix* untuk *Test Data*.

Tabel 4.4 *Confusion Matrix* untuk *Test Data*

		Prediksi	
		normal	<i>Cracked</i>
aktual	normal	209	21
	<i>Cracked</i>	157	263

Berdasarkan Tabel 4.4 *Confusion Matrix* untuk *Test Data* maka dapat dijelaskan terdapat empat sel yang mewakili empat kemungkinan kombinasi prediksi dan label sebenarnya dengan pembagian menjadi nilai *True Negative* (TN), *False Positive* (FP), *False Negative* (FN), dan *True Positive* (TP). Melihat nilai *confusion matrix*, maka nilai *True Negative* (TN) dengan nilai 263 yaitu berarti

jumlah gambar yang benar-benar diklasifikasikan sebagai "normal" oleh model dan memang benar-benar "normal" berdasarkan label sebenarnya. Nilai *False Positive* (FP) dengan nilai 157 yaitu berarti jumlah gambar yang salah diklasifikasikan sebagai "normal" oleh model, padahal sebenarnya "Cracked". Nilai *False Negative* (FN) dengan nilai 21 yaitu berarti jumlah gambar yang salah diklasifikasikan sebagai "Cracked" oleh model, padahal sebenarnya "normal". Nilai *True Positive* (TP) dengan nilai 209 yaitu berarti jumlah gambar yang benar-benar diklasifikasikan sebagai "Cracked" oleh model dan memang benar-benar "Cracked" berdasarkan label sebenarnya.

Berdasarkan hasil nilai *Confusion Matrix*, maka dapat dijelaskan secara perhitungan manualnya terkait nilai akurasi, presisi, *recall*, dan F1-score nya sebagai berikut:

1. Akurasi (Seberapa sering model membuat prediksi akurat berdasarkan total data diukur berdasarkan keakuratannya)

$$\begin{aligned}\text{Akurasi} &= \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \\ &= \frac{209 + 263}{209 + 263 + 157 + 21} \\ &= \frac{472}{650} \\ &= 0,7262\end{aligned}$$

2. Presisi (Semua prediksi positif yang dibuat model, presisi menghitung berapa banyak prediksi positif yang ternyata akurat).

Presisi untuk kelas 0 (Normal) adalah 0,5710

Presisi untuk kelas 1 (*Cracked*) adalah 0,9261

Presisi dihitung dengan menggunakan rumus:

$$\begin{aligned}\text{Presisi Kelas 0 (Normal)} &= \frac{\text{True Positive (TP)}}{\text{True Positive (TP)} + \text{False Positive (FP)}} \\ &= \frac{209}{209 + 157} \\ &= \frac{209}{366} \\ &= 0,5710\end{aligned}$$

$$\begin{aligned}
 \text{Presisi Kelas 1 (Crack)} &= \frac{\text{True Negative (TN)}}{\text{True Negative (TN)} + \text{False Negative (FN)}} \\
 &= \frac{263}{263 + 21} \\
 &= \frac{263}{284} \\
 &= 0,9261
 \end{aligned}$$

3. *Recall* (Mengukur proporsi klasifikasi kelas positif nyata yang diprediksi secara akurat oleh model).

Recall untuk kelas 0 (Normal) adalah 0,9087

Recall untuk kelas 1 (Cracked) adalah 0,6262

Recall dihitung dengan menggunakan rumus:

$$\begin{aligned}
 \text{Recall Kelas 0 (Normal)} &= \frac{\text{True Positive (TP)}}{\text{True Positive (TP)} + \text{False Negative (FN)}} \\
 &= \frac{209}{209 + 21} \\
 &= \frac{209}{230} \\
 &= 0,9087
 \end{aligned}$$

$$\begin{aligned}
 \text{Recall Kelas 1 (Crack)} &= \frac{\text{True Negative (TN)}}{\text{True Negative (TN)} + \text{False Positive (FP)}} \\
 &= \frac{263}{263 + 157} \\
 &= \frac{263}{420} \\
 &= 0,6262
 \end{aligned}$$

4. *F1-Score* (mean harmonik presisi dan *recall*. Hal ini berguna ketika kelas tidak seimbang karena menjaga keseimbangan antara kedua ukuran tersebut).

F1-Score untuk kelas 0 (Normal) adalah 0,7013

F1-Score untuk kelas 1 (Cracked) adalah 0,7472

F1-Score dihitung dengan menggunakan rumus:

$$\text{F1 Score Kelas 0 (Normal)} = 2 \times \frac{\text{Presisi} \times \text{Recall}}{\text{Presisi} + \text{Recall}}$$

$$\begin{aligned}
&= 2 \times \frac{0,5710 \times 0,9087}{0,5710 + 0,9087} \\
&= 2 \times \frac{0,5189}{1,4797} \\
&= 2 \times 0,3507 \\
&= 0,7013 \\
F1 \text{ Score Kelas 1 (Crack)} &= 2 \times \frac{Presisi \times Recall}{Presisi + Recall} \\
&= 2 \times \frac{0,9261 \times 0,6262}{0,9261 + 0,6262} \\
&= 2 \times \frac{0,5799}{1,5522} \\
&= 2 \times 0,3736 \\
&= 0,7472
\end{aligned}$$

Berdasarkan hasil perhitungan manual hasil terkait nilai akurasi, presisi, *recall*, dan F1-score nya yang diperoleh cocok dan sama sesuai dengan perolehan hasil dari sistem menggunakan *Convolutional Neural Network* (CNN). Berikut adalah Tabel 4.5 Hasil klasifikasi *Train Data*:

Tabel 4.5 Hasil klasifikasi *Train Data*

	<i>Precision</i>	<i>Recall</i>	<i>f1-score</i>	<i>Support</i>
0	0.5446	0.5930	0.5678	597
1	0.4841	0.4351	0.4583	524
<i>Accuracy</i>			0.5192	1121
<i>Macro avg</i>	0.5143	0.5140	0.5130	1121
<i>Weighted avg</i>	0.5163	0.5192	0.5166	1121

Berdasarkan Tabel 4.5 Hasil klasifikasi *Train Data* maka dapat dijelaskan *Precision* mengukur seberapa banyak dari prediksi positif yang sebenarnya benar-benar positif. Kasus ini, untuk kategori 0 (kelas normal) dengan *precision* adalah 0.5446 yang berarti dari semua gambar yang diprediksi sebagai normal, sekitar 54.46% benar-benar normal. Selanjutnya kategori 1 (kelas *Cracked*), *precision* adalah 0.4841, yang berarti dari semua gambar yang diprediksi sebagai retak, sekitar 48.41% benar-benar retak.

Recall (atau *sensitivity*) mengukur seberapa banyak dari semua kelas positif yang diprediksi dengan benar oleh model. Kategori 0, *recall* adalah 0.5930, yang

berarti model berhasil mendeteksi sekitar 59.30% dari semua gambar yang sebenarnya normal. Kategori 1, *recall* adalah 0.4351, yang berarti model hanya berhasil mendeteksi sekitar 43.51% dari semua gambar yang sebenarnya retak.

F1-score adalah ukuran gabungan dari *precision* dan *recall*. Ini adalah *harmonic mean* dari kedua metrik tersebut, memberikan nilai yang seimbang antara keduanya. *Support* adalah jumlah kemunculan sampel dengan masing-masing kelas dalam data uji, untuk kelas 0 berjumlah 597 data, untuk kelas 1 berjumlah 524 sehingga total datanya berjumlah 1121 data. *Accuracy* adalah ukuran keseluruhan kinerja model, yaitu persentase prediksi yang benar secara keseluruhan, akurasi adalah 0.5192 atau 51.92% yang berarti model memprediksi label kelas dengan benar untuk sekitar 51.92% dari total sampel dalam *dataset* uji.

Macro average adalah rata-rata dari metrik evaluasi untuk setiap kelas tanpa memperhitungkan ketidakseimbangan kelas, ini adalah rata-rata dari metrik evaluasi (*precision*, *recall*, f1-score, dan lain-lain) untuk setiap kelas secara terpisah, tanpa memperhatikan distribusi kelas. *Macro avg precision* adalah 0.5143, *recall* adalah 0.5140, dan *f1-score* adalah 0.5130. *Weighted average* adalah rata-rata yang diberi bobot berdasarkan dukungan (jumlah sampel) untuk setiap kelas ini juga merupakan rata-rata dari metrik evaluasi untuk setiap kelas, namun mempertimbangkan distribusi kelas dalam *dataset*. Dalam *weighted avg precision* adalah 0.5163, *recall* adalah 0.5192, dan *f1-score* adalah 0.5192. *Weighted avg* memberikan lebih banyak bobot pada kelas yang lebih besar, sehingga memberikan gambaran tentang performa model secara keseluruhan dengan mempertimbangkan distribusi kelas. Selanjutnya adalah Tabel 4.6 Hasil klasifikasi *Validation Data*.

Tabel 4.6 Hasil klasifikasi *Validation data*

	<i>Precision</i>	<i>Recall</i>	<i>f1-score</i>	<i>Support</i>
0	0.5726	0.4765	0.5201	149
1	0.4968	0.5923	0.5404	130
<i>Accuracy</i>			0.5305	279
<i>Macro avg</i>	0.5347	0.5344	0.5302	279
<i>Weighted avg</i>	0.5373	0.5305	0.5292	279

Berdasarkan Tabel 4.6 Hasil klasifikasi *validation data* maka dapat dijelaskan *Precision* mengukur seberapa banyak dari prediksi positif yang

sebenarnya benar-benar positif. Kasus ini, untuk kategori 0 (kelas normal) dengan *precision* adalah 0.5726 yang berarti dari semua gambar yang diprediksi sebagai normal, sekitar 57.26% benar-benar normal. Selanjutnya kategori 1 (kelas *Cracked*), *precision* adalah 0.4968, yang berarti dari semua gambar yang diprediksi sebagai retak, sekitar 49.68% benar-benar retak.

Recall (atau *sensitivity*) mengukur seberapa banyak dari semua kelas positif yang diprediksi dengan benar oleh model. Kategori 0, *recall* adalah 0.4765, yang berarti model berhasil mendeteksi sekitar 47.65% dari semua gambar yang sebenarnya normal. Kategori 1, *recall* adalah 0.5923, yang berarti model hanya berhasil mendeteksi sekitar 59.23% dari semua gambar yang sebenarnya retak.

F1-score adalah ukuran gabungan dari *precision* dan *recall*. Ini adalah *harmonic mean* dari kedua metrik tersebut, memberikan nilai yang seimbang antara keduanya. *Support* adalah jumlah kemunculan sampel dengan masing-masing kelas dalam data uji, untuk kelas 0 berjumlah 149 data, untuk kelas 1 berjumlah 130 sehingga total datanya berjumlah 279 data. *Accuracy* adalah ukuran keseluruhan kinerja model, yaitu persentase prediksi yang benar secara keseluruhan, akurasi adalah 0.5305 atau 53.05% yang berarti model memprediksi label kelas dengan benar untuk sekitar 53.05% dari total sampel dalam *dataset* uji.

Macro average adalah rata-rata dari metrik evaluasi untuk setiap kelas tanpa memperhitungkan ketidakseimbangan kelas, ini adalah rata-rata dari metrik evaluasi (*precision*, *recall*, *f1-score*, dan lain-lain) untuk setiap kelas secara terpisah, tanpa memperhatikan distribusi kelas. *Macro avg precision* adalah 0.5347, *recall* adalah 0.5344, dan *f1-score* adalah 0.5302. *Weighted average* adalah rata-rata yang diberi bobot berdasarkan dukungan (jumlah sampel) untuk setiap kelas ini juga merupakan rata-rata dari metrik evaluasi untuk setiap kelas, namun mempertimbangkan distribusi kelas dalam *dataset*. Dalam *weighted avg precision* adalah 0.5373, *recall* adalah 0.5305, dan *f1-score* adalah 0.5096. *Weighted avg* memberikan lebih banyak bobot pada kelas yang lebih besar, sehingga memberikan gambaran tentang performa model secara keseluruhan dengan mempertimbangkan distribusi kelas. Selanjutnya adalah Tabel 4.7 Hasil klasifikasi *TestData*.

Tabel 4.7 Hasil klasifikasi *Test Data*

	<i>Precision</i>	<i>Recall</i>	<i>f1-score</i>	<i>Support</i>
0	0.5710	0.9087	0.7013	230
1	0.9261	0.6262	0.7472	420
<i>Accuracy</i>			0.7262	650
<i>Macro avg</i>	0.7485	0.7674	0.7243	650
<i>Weighted avg</i>	0.8004	0.7262	0.7309	650

Berdasarkan Tabel 4.7 Hasil klasifikasi *Test Data* maka dapat dijelaskan *Precision* mengukur seberapa banyak dari prediksi positif yang sebenarnya benar-benar positif. Kasus ini, untuk kategori 0 (kelas normal) dengan *precision* adalah 0.5710 yang berarti dari semua gambar yang diprediksi sebagai normal, sekitar 57.10% benar-benar normal. Selanjutnya kategori 1 (kelas *Cracked*), *precision* adalah 0.9261, yang berarti dari semua gambar yang diprediksi sebagai retak, sekitar 92.61% benar-benar retak.

Recall (atau *sensitivity*) mengukur seberapa banyak dari semua kelas positif yang diprediksi dengan benar oleh model. Kategori 0, *recall* adalah 0.9087, yang berarti model berhasil mendeteksi sekitar 90.87% dari semua gambar yang sebenarnya normal. Kategori 1, *recall* adalah 0.6262, yang berarti model hanya berhasil mendeteksi sekitar 62.62% dari semua gambar yang sebenarnya retak.

F1-score adalah ukuran gabungan dari *precision* dan *recall*. Ini adalah *harmonic mean* dari kedua metrik tersebut, memberikan nilai yang seimbang antara keduanya. *Support* adalah jumlah kemunculan sampel dengan masing-masing kelas dalam data uji, untuk kelas 0 berjumlah 230 data, untuk kelas 1 berjumlah 420 sehingga total datanya berjumlah 650 data. *Accuracy* adalah ukuran keseluruhan kinerja model, yaitu persentase prediksi yang benar secara keseluruhan, akurasi adalah 0.7262 atau 72.62% yang berarti model memprediksi label kelas dengan benar untuk sekitar 72.62% dari total sampel dalam *dataset* uji.

Macro average adalah rata-rata dari metrik evaluasi untuk setiap kelas tanpa memperhitungkan ketidakseimbangan kelas, ini adalah rata-rata dari metrik evaluasi (*precision*, *recall*, *f1-score*, dan lain-lain) untuk setiap kelas secara terpisah, tanpa memperhatikan distribusi kelas. *Macro avg precision* adalah 0.7485, *recall* adalah 0.7674, dan *f1-score* adalah 0.7243. *Weighted average* adalah rata-

rata yang diberi bobot berdasarkan dukungan (jumlah sampel) untuk setiap kelas ini juga merupakan rata-rata dari metrik evaluasi untuk setiap kelas, namun mempertimbangkan distribusi kelas dalam *dataset*. Dalam *weighted avg precision* adalah 0.8004, *recall* adalah 0.7262, dan *f1-score* adalah 0.7309. *Weighted avg* memberikan lebih banyak bobot pada kelas yang lebih besar, sehingga memberikan gambaran tentang performa model secara keseluruhan dengan mempertimbangkan distribusi kelas. Berikut merupakan perhitungan manual untuk *macro average* dan *weighted average*:

1. *Macro average*

Macro average untuk presisi adalah 0,7485

Macro average untuk *recall* adalah 0,7674

Macro average untuk *f1-score* adalah 0,7243

$$\begin{aligned} \text{Macro Avg (Presisi)} &= \frac{\text{Jumlah Nilai Presisi}}{\text{Jumlah Kelas}} \\ &= \frac{0,5710 + 0,9261}{2} \\ &= \frac{1,4971}{2} \\ &= 0,7485 \end{aligned}$$

$$\begin{aligned} \text{Macro Avg (Recall)} &= \frac{\text{Jumlah Nilai Recall}}{\text{Jumlah Kelas}} \\ &= \frac{0,9087 + 0,6262}{2} \\ &= \frac{1,5349}{2} \\ &= 0,7674 \end{aligned}$$

$$\begin{aligned} \text{Macro Avg (F1 Score)} &= \frac{\text{Jumlah Nilai F1 Score}}{\text{Jumlah Kelas}} \\ &= \frac{0,7013 + 0,7472}{2} \\ &= \frac{1,4485}{2} \\ &= 0,7243 \end{aligned}$$

2. *Weighted average*

Weighted average untuk *presisi* adalah 0.8004

Weighted average untuk *recall* adalah 0.7262

Weighted average untuk *f1-score* adalah 0.7309

$$\begin{aligned}
 \text{Weight Avg (Presisi)} &= \frac{\text{Presisi}_0 \times \text{Support}_0 + \text{Presisi}_1 \times \text{Support}_1}{\text{Total Support}} \\
 &= \frac{0,5710 \times 230 + 0,9261 \times 420}{650} \\
 &= \frac{131,33 + 388,962}{650} \\
 &= \frac{520,292}{650} \\
 &= 0,8004
 \end{aligned}$$

$$\begin{aligned}
 \text{Weight Avg (Recall)} &= \frac{\text{Recall}_0 \times \text{Support}_0 + \text{Recall}_1 \times \text{Support}_1}{\text{Total Support}} \\
 &= \frac{0,9087 \times 230 + 0,6262 \times 420}{650} \\
 &= \frac{209,001 + 263,004}{650} \\
 &= \frac{472,005}{650} \\
 &= 0,7262
 \end{aligned}$$

$$\begin{aligned}
 \text{Weight Avg (F1 Score)} &= \frac{\text{F1 Score}_0 \times \text{Support}_0 + \text{F1 Score}_1 \times \text{Support}_1}{\text{Total Support}} \\
 &= \frac{0,7013 \times 230 + 0,7472 \times 420}{650} \\
 &= \frac{161,299 + 313,824}{650} \\
 &= \frac{475,123}{650} \\
 &= 0,7310
 \end{aligned}$$

4.6 Hasil Perbandingan

Hasil perbandingan yang dilakukan untuk melihat seberapa bagus prediksi yang dilakukan, perbandingan dilakukan dari mulai banyaknya lapisan dan *epoch*,

rasio dataset yang di *splitting*, jenis optimasi algoritma, dan *learning rate*. Berikut merupakan hasil Tabel 4.8 Hasil Akurasi Model Klasifikasi Cacat Ban Untuk Beberapa Nilai *Layer* dan *Epoch*.

Tabel 4.8 Hasil Akurasi Model Klasifikasi Cacat Ban Untuk Beberapa Nilai *Layer* dan *Epoch*

<i>Layer</i>	<i>Epoch</i>		
	20	50	100
3	0,6523 (65,23%)	0,6262 (62,62%)	0,7231 (72,31%)
4	0,6862 (68,62%)	0,7246 (72,46%)	0,7262 (72,62%)
5	0,7 (70%)	0,6754 (67,54%)	0,6862 (68,62%)
6	0,6308 (63,08%)	0,6908 (69,08%)	0,6554 (65,54%)

Berdasarkan Tabel 4.8 Hasil Akurasi Model Klasifikasi Cacat Ban Untuk Beberapa Nilai *Layer* dan *Epoch* maka dapat dijelaskan perbandingan yang dilakukan berdasarkan jumlah *Layer* mulai dari 3 *layer*, 4 *layer*, 5 *layer*, dan 6 *layer* serta dikombinasikan dengan jumlah *epoch* mulai dari 20 *epoch*, 50 *epoch*, dan 100 *epoch*. Model menggunakan *rasio dataset* 80% untuk *training* data, 20% untuk *validation* data, optimasi algoritma adam dengan *learning rate* 0,001. Kesimpulan model evaluasi yang didapatkan bahwa hasil prediksi tertinggi ada pada jenis model penggunaan 4 *layer* dan penggunaan 100 *epoch* dengan hasil akurasi sebesar 0,7262 atau 72,62 %.

Selanjutnya setelah didapatkan hasil tertinggi berdasarkan jumlah *Layer* dan jumlah *epoch*, berikutnya akan diteruskan untuk dilakukan perbandingan berdasarkan *rasio dataset* yang di *splitting*. Berikut merupakan hasil Tabel 4.9 Hasil Akurasi Model Klasifikasi Cacat Ban Untuk Rasio *Dataset*.

Tabel 4.9 Hasil Akurasi Model Klasifikasi Cacat Ban Untuk Rasio *Dataset*

No	Jumlah Rasio <i>Dataset</i>			
	0,80:0,20	0,60:0,20:0,20	0,70:0,20:0,10	0,80:0,10:0,10
1	0,7262 (72,62%)	0,7883 (78,83%)	0,8155 (81,55%)	0,8252 (82,52%)

Berdasarkan Tabel 4.9 Hasil Akurasi Model Klasifikasi Cacat Ban Untuk Rasio *Dataset*, maka dapat dijelaskan di mana hasil kelanjutan untuk perbandingan berdasarkan jumlah *rasio dataset* mulai dari 0,80:0,20 (80% *training* data dan 20% *validation* data), 0,60:0,20:0,20 (60% *training* data, 20% *validation* data, dan 20% *testing* data), 0,70:0,20:0,10 (70% *training* data, 20% *validation* data, dan 10% *testing* data), dan 0,80:0,10:0,10 (80% *training* data, 10% *validation* data, dan 10%

testing data). Model menggunakan optimasi algoritma adam dengan *learning rate* 0,001. Kesimpulan yang di dapatkan bahwa hasil prediksi tertinggi ada pada jenis model penggunaan rasio *dataset* 0,80:0,10:0,10 (80% *training* data, 10% *validation* data, dan 10% *testing* data) dengan hasil akurasi sebesar 0,8252 (82,52%).

Selanjutnya setelah didapatkan hasil tertinggi berdasarkan jumlah rasio *dataset*, berikutnya akan diteruskan untuk dilakukan perbandingan berdasarkan optimasi algoritma. Berikut merupakan hasil Tabel 4.10 Hasil Akurasi Model Klasifikasi Cacat Ban Untuk Optimasi Algoritma.

Tabel 4.10 Hasil Akurasi Model Klasifikasi Cacat Ban Untuk Optimasi Algoritma

No	Optimasi Algoritma		
	SGD	RMSPROP	ADAM
1	0,5485(54,85%)	0,7767 (77,67%)	0,8252 (82,52%)

Berdasarkan Tabel 4.10 Hasil Akurasi Model Klasifikasi Cacat Ban Untuk Optimasi Algoritma, maka dapat dijelaskan bahwa hasil kelanjutan untuk perbandingan berdasarkan optimasi algoritma mulai dari *Stochastic Gradient Descent* (SGD) merupakan varian dari *gradian descent* yang bekerja dengan memperbarui parameter menggunakan sampel acak dari *dataset* pada setiap iterasi, *Root Mean Square Propagation* (RMSPROP) merupakan bertujuan untuk mengatasi masalah penurunan lambat dan tidak stabil dari SGD dengan menggunakan *moving average* (meratakan fluktuasi data dalam rangka mengidentidiaksi tren yang mendasarinya) dari kuadrat gradien untuk menormalisasi pembaruan parameter, *Adaptive Moment Estimation* (ADAM) merupakan menggabungkan ide-ide dari RMSPROP dan SGD dengan menyimpan *moving avarage* dan kuadrat gradien, dan menggunakan keduanya untuk memperbarui parameter. Model menggunakan *learning rate* 0,001. Kesimpulan yang di dapatkan bahwa hasil prediksi tertinggi ada pada jenis model penggunaan optimasi algoritma *Adaptive Moment Estimation* (ADAM) dengan hasil akurasi sebesar 0,8252 (82,52%).

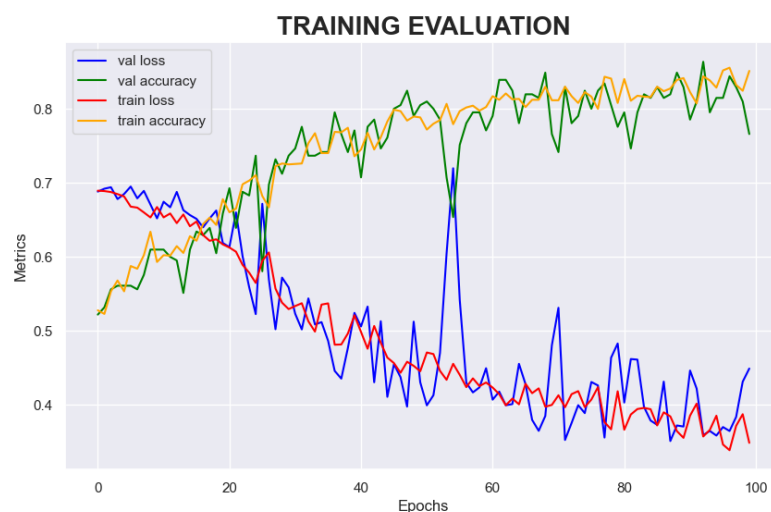
Selanjutnya setelah didapatkan hasil tertinggi berdasarkan optimasi algoritma *dataset*, berikutnya akan diteruskan untuk dilakukan perbandingan berdasarkan *learning rate* (salah satu hyperparameter penting dalam algoritma pembelajaran mesin, khususnya dalam konteks optimasi menggunakan algoritma

yang menentukan ukuran langkah pembaruan parameter model pada setiap iterasi selama pelatihan dan mengontrol seberapa cepat atau lambat model belajar dari data). Berikut merupakan hasil Tabel 4.11 Hasil Akurasi Model Pelatihan Klasifikasi Cacat Ban Untuk *Learning Rate*.

Tabel 4.11 Hasil Akurasi Model Klasifikasi Cacat Ban Untuk *Learning Rate*

No	<i>Learning Rate</i>		
	0,001	0,0001	0,00001
1	0,8252 (82,52%)	0,7233 (72,33%)	0,5777 (57,77%)

Berdasarkan Tabel 4.11 Hasil Akurasi Model Klasifikasi Cacat Ban Untuk *Learning Rate*, maka dapat dijelaskan di mana hasil kelanjutan untuk perbandingan berdasarkan *learning rate* mulai dari 0,001, 0,0001, dan 0,00001. Model menggunakan 4 layer, 100 epoch, rasio dataset 0,80:0,10:0,10 (80% training data, 10% validation data, dan 10% testing data), dan optimasi algoritma ADAM. Kesimpulan akhir dari hasil perbandingan yang ada di dapatkan bahwa hasil prediksi tertinggi ada pada jenis model penggunaan *learning rate* 0,001 dengan hasil akurasi sebesar 0,8252 (82,52%) menghasilkan akurasi tertinggi di antara tiga kecepatan yang diuji, menunjukkan bahwa model dilatih secara efektif dan mencapai keseimbangan yang baik antara kecepatan dan akurasi konvergensi (kondisi di mana proses pelatihan model mencapai titik stabilitas dalam nilai fungsi *loss* dan parameter yang diperbarui model tidak lagi mengalami perubahan signifikan). Berikutnya melihat hasil model pelatihan yang terjadi pada hasil terbaik pada Gambar 4.3 *Training Evaluation*.



Gambar 4.3 Training Evaluation

Berdasarkan Gambar 4.3 *Training Evaluation* bisa dijelaskan plot yang ditampilkan manampilkan performa pelatihan dan validasi model selama 100 *epoch*, dengan keterangan metriknya untuk *validation loss* (biru), *validation accuracy* (hijau), *traininig loss* (merah), *training accuracy* (oranye). Tren yang terjadi untuk *validation loss* (biru) dimulai dari 0,6880 (68,80%) sampai 0,4491 (44,91%) umumnya menurun tetapi dengan fluktuasi yang signifikan ini menunjukkan bahwa model mengalami kesulitan dalam stabilitas pada validasi data mungkin karena *overfitting* atau variasi dalam validasi data, dari stabilitasnya ada beberapa lonjakan yang menunjukkan momen di mana performa model pada set validasi memburuk untuk sementara. Tren yang terjadi untuk *validation accuracy* (hijau) meningkat seiring berjalannya waktu, mulai dari 0,5219 (52,19%) dan mencapai tertinggi sekitar 0,84 (84%) pada *epoch* 96 namun menurun kembali sampai akhirnya naik sekitar 0,7658 (76,58%) di *epoch* 100. Stabilitasnya ada beberapa fluktuasi, ini menunjukkan tren peningkatan secara keseluruhan serupa dengan akurasi pelatihan.

Tren yang terjadi untuk *traininig loss* (merah) menurun secara bertahap seiring berjalannya waktu dari 0,6893 (68,93%) pada awalnya hingga 0,3487 (34,87%) pada akhir pelatihan, menunjukkan bahwa model sedang mempelajari dan meminimalkan kesalahan pada data pelatihan. Stabilitasnya terdapat fluktuasi, namun tren secara keseluruhan menunjukkan penurunan. Tren yang terjadi untuk *training accuracy* (oranye) meningkat terus-menerus dari waktu ke waktu, dimulai dari 0,5277 (52,77%) sampai meningkat mencapai sekitar 0,8511 (85,11%) pada *epoch* 100, ini menunjukkan bahwa model semakin baik dalam memprediksi kelas pada data pelatihan. Stabilitasnya menunjukkan peningkatan yang konsisten dengan fluktuasi kecil.

Nilai akurasi yang terjadi terlihat berfluktuasi cenderung meningkat sehingga bisa disimpulkan performa model juga mengalami tidak konsisten, sedangkan nilai *loss* yang terjadi terlihat berfluktuasi yang cenderung menurun sehingga bisa disimpulkan karena model mungkin karena variasi yang terjadi dalam data. Menunjukkan bahwa model mungkin mengalami *overfitting* atau kesulitan dalam generalisasi, meskipun model dapat belajar dengan baik pada data pelatihan.

Kesimpulan akhir pada model akhir yang sudah dibandingkan dengan komponen 4 *layer*, jumlah *epoch* 100, rasio *dataset* 0,80:0,10:0,10 (80% *training* data, 10% *validation* data, dan 10% *testing* data), optimasi algoritma menggunakan *Adaptive Moment Estimation* (ADAM), *learning rate* 0,001. Akurasi *training* 0,8511 (85,11%), akurasi *validation* 0,7658 (76,58%), akurasi *testing* 0,8252 (82,52%). Sehingga perbedaan akurasi antara *training* dan *testing* berada pada 0,0259 (2,59%).

DAFTAR PUSTAKA

- Adel, A. (2022). Future of industry 5.0 in society: human-centric solutions, challenges and prospective research areas. In *Journal of Cloud Computing* (Vol. 11, Issue 1). Springer Science and Business Media Deutschland GmbH. <https://doi.org/10.1186/s13677-022-00314-5>
- Akintoye, S. B., Han, L., Lloyd, H., Zhang, X., Dancey, D., Chen, H., & Zhang, D. (2023). Layer-wise partitioning and merging for efficient and scalable deep learning. *Future Generation Computer Systems*, 149, 432–444. <https://doi.org/10.1016/j.future.2023.07.043>
- Alem, A., & Kumar, S. (2022). End-to-End Convolutional Neural Network Feature Extraction for Remote Sensed Images Classification. *Applied Artificial Intelligence*, 36(1). <https://doi.org/10.1080/08839514.2022.2137650>
- Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M. A., Al-Amidie, M., & Farhan, L. (2021). Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data*, 8(1). <https://doi.org/10.1186/s40537-021-00444-8>
- Benbarrad, T., Salhaoui, M., Kenitar, S. B., & Arioua, M. (2021). Intelligent machine vision model for defective product inspection based on machine learning. *Journal of Sensor and Actuator Networks*, 10(1). <https://doi.org/10.3390/jsan10010007>
- Breitenbach, J., Eckert, I., Mahal, V., Baumgartl, H., & Buettner, R. (2022). Automated Defect Detection of Screws in the Manufacturing Industry Using. *Proceedings of the 55th Hawaii International Conference on System Sciences*, 1226–1235.
- Burgueño, L., Cabot, J., Li, S., Sébastien Gérard, Spain, I., & Gérard, S. (2022). A Generic LSTM Neural Network Architecture to Infer Heterogeneous Model Transformations. *Journal on Software and Systems Modeling*, 1–18.
- Chlap, P., Min, H., Vandenberg, N., Dowling, J., Holloway, L., & Haworth, A. (2021). A review of medical image data augmentation techniques for deep

- learning applications. In *Journal of Medical Imaging and Radiation Oncology* (Vol. 65, Issue 5, pp. 545–563). John Wiley and Sons Inc. <https://doi.org/10.1111/1754-9485.13261>
- Chen, Y., & Chen, G. (2023). Optimization of the Intelligent Asset Management System Based on WSN and RFID Technology. *Journal of Sensors*. <https://doi.org/10.1155/2022/3436530>
- Ching, N. T., Ghobakhloo, M., Iranmanesh, M., Maroufkhani, P., & Asadi, S. (2022). Industry 4.0 applications for sustainable manufacturing: A systematic literature review and a roadmap to sustainable development. *Journal of Cleaner Production*, 334. <https://doi.org/10.1016/j.jclepro.2021.130133>
- Churcher, A., Ullah, R., Ahmad, J., Ur Rehman, S., Masood, F., Gogate, M., Alqahtani, F., Nour, B., & Buchanan, W. J. (2021). An experimental analysis of attack classification using machine learning in IoT networks. *Sensors* (Switzerland), 21(2), 1–32. <https://doi.org/10.3390/s21020446>
- Fahle, S., Prinz, C., & Kuhlenkötter, B. (2020). Systematic review on machine learning (ML) methods for manufacturing processes - Identifying artificial intelligence (AI) methods for field application. *Procedia CIRP*, 93, 413–418. <https://doi.org/10.1016/j.procir.2020.04.109>
- Fingscheidt, T., Gottschalk, H., & Houben Sebastia. (2022). *Deep Neural Networks and Data for Automated Driving Robustness, Uncertainty Quantification, and Insights Towards Safety*. Cham: Springer.
- Habib, G., & Qureshi, S. (2022). Optimization and acceleration of convolutional neural networks: A survey. In *Journal of King Saud University - Computer and Information Sciences* (Vol. 34, Issue 7, pp. 4244–4268). King Saud bin Abdulaziz University. <https://doi.org/10.1016/j.jksuci.2020.10.004>
- Hassan, N. M., Hamdan, A., Shahin, F., Abdelmaksoud, R., & Bitar, T. (2023). An artificial intelligent manufacturing process for high-quality low-cost production. *International Journal of Quality and Reliability Management*, 40(7), 1777–1794. <https://doi.org/10.1108/IJQRM-07-2022-0204>.
- Hernita, H., Surya, B., Perwira, I., Abubakar, H., & Idris, M. (2021). Economic business sustainability and strengthening human resource capacity based on

- increasing the productivity of small and medium enterprises (SMES) in Makassar city, Indonesia. *Sustainability (Switzerland)*, 13(6), 1–37. <https://doi.org/10.3390/su13063177>
- Hossain, E. (2023). *Machine Learning Crash Course for Engineers*. Cham: Springer.
- Id, I. D. (2021). *Machine Learning Teori, Studi Kasus dan Implementasi Menggunakan Python* (Vol. 1). Pekanbaru: UR PRESS.
- Isaza, A. A., & Jimenez, L. M. (2022). Data Augmentation and Transfer Learning for Brain Tumor Detection in Magnetic Resonance Imaging. *IEEE Access*, 10, 23217–23233. <https://doi.org/10.1109/ACCESS.2022.3154061>
- Javaid, M., Haleem, A., Singh, R. P., & Suman, R. (2021). Substantial capabilities of robotics in enhancing industry 4.0 implementation. In *Cognitive Robotics* (Vol. 1, pp. 58–75). KeAi Communications Co. <https://doi.org/10.1016/j.cogr.2021.06.001>
- Junaidi, S., Beno, I. S., Farkhan, M., Supartha, I. K. D. G., Pasaribu, A. A., Kmurawak, R. M., Supiyanto, Sroyer, A. M., Reba, F., Fitriyanto, R., Syafiqoh, U., & Rizal, A. A. (2024). *Buku Ajar Machine Learning* (Vol. 1). PT Sonpedia Publishing Indonesia.
- Job, M. S., Bhateja, P. H., Gupta, M., Bingi, K., & Prusty, B. R. (2022). Fractional Rectified Linear Unit Activation Function and Its Variants. *Mathematical Problems in Engineering*, 2022. <https://doi.org/10.1155/2022/1860779>
- Kiruthikaa R, Pon Saravanan R, Samyuktha R, Swathi P, & Vinush S. (2023). Tire Quality Inspection System Based on Deep Learning. *International Journal of Engineering Technology and Management Sciences*, 7(4), 143–148. <https://doi.org/10.46647/ijetms.2023.v07i04.022>
- Kneusel, R. T. (2021). *Practical Deep Learning A Python-Based Introduction* (Vol. 1). San Francisco: No Starch Press.
- Ko, D., Kang, S., Kim, H., Lee, W., Bae, Y., & Park, J. (2021). Anomaly segmentation based on depth image for quality inspection processes in tire manufacturing. *Applied Sciences (Switzerland)*, 11(21), 1–18. <https://doi.org/10.3390/app112110376>

- Kuric, I., Klarák, J., Bulej, V., Sága, M., Kandra, M., Hajdučík, A., & Tucki, K. (2022). Approach to Automated Visual Inspection of Objects Based on Artificial Intelligence. *Applied Sciences (Switzerland)*, 12(2), 1–19. <https://doi.org/10.3390/app12020864>
- Kuric, I., Klarák, J., Sága, M., Císar, M., Hajdučík, A., & Wiecek, D. (2021). Analysis of the possibilities of tire-defect inspection based on unsupervised learning and deep learning. *Sensors*, 21(21). <https://doi.org/10.3390/s21217073>
- Lee, S. K., Lee, H., Back, J., An, K., Yoon, Y., Yum, K., kim, S., & Hwang, S. U. (2021). Prediction of tire pattern noise in early design stage based on convolutional neural network. *Applied Acoustics*, 172, 1–18. <https://doi.org/10.1016/j.apacoust.2020.107617>.
- Li, X., & wang, F. (2021). *Artificial Intelligence oceanography* (Vol. 1). Qiangdao: Springer.
- Li, Y., Fan, B., Zhang, W., & Jiang, Z. (2021). *TireNet*: A high *recall* rate method for practical application of tire defect type classification. *Future Generation Computer Systems*, 125, 1–9. <https://doi.org/10.1016/j.future.2021.06.009>.
- Lillicrap, T. P., Santoro, A., Marris, L., Akerman, C., & Hinton, G. (2020). Backpropagation and the brain. *Nature Reviews Neuroscience*, 1–32.
- Lin, S. L. (2023). Research on tire crack detection using image deep learning method. *Scientific Reports*, 13(1). <https://doi.org/10.1038/s41598-023-35227-z>.
- Ma, K., Tang, C., Zhang, W., Cui, B., Ji, K., Chen, Z., & Abraham, A. (2023). DC-CNN: Dual-channel Convolutional Neural Networks with attention-pooling for fake news detection. *Applied Intelligence*, 53(7), 8354–8369. <https://doi.org/10.1007/s10489-022-03910-9>
- Madenda, S. (2015). *Pengolahan Citra & Video Digital Teori, Aplikasi, dan Pemrograman Menggunakan Matlab* (Vol. 1). Jakarta: Penerbit Erlangga.
- Mercioni, M. A., & Holban, S. (2020). The Most Used Activation Functions: Classic Versus Current. *15th International Conference on DEVELOPMENT AND APPLICATION SYSTEMS*, 141–145.

- Nguyen, H. T., Yu, G. H., Shin, N. R., Kwon, G. J., Kwak, W. Y., & Kim, J. Y. (2021). Defective product classification system for smart factory based on deep learning. *Electronics (Switzerland)*, 10(7).
<https://doi.org/10.3390/electronics10070826>
- Norhikmah, Lutfhi, A., & Rumini. (2022). INTERNATIONAL JOURNAL ON INFORMATICS VISUALIZATION The Effect of Layer Batch Normalization and Dropout of CNN model Performance on Facial Expression Classification. INTERNATIONAL JOURNAL ON INFORMATICS VISUALIZATION, 6(2), 481–488. www.joiv.org/index.php/joiv
- Oborski, P., & Wysocki, P. (2022). Intelligent Visual Quality Control System Based on Convolutional Neural Networks for Holonic Shop Floor Control of Industry 4.0 Manufacturing Systems. *Advances in Science and Technology Research Journal*, 16(2), 89–98. <https://doi.org/10.12913/22998624/145503>
- Prayogi. (2018). *Pemeliharaan Sasis dan Pemindahan Tenaga Kendaraan Ringan (C3) Kompetensi Keahlian: Teknik Kendaraan Ringan*. Jakarta: PT Gramedia Widiasarana Indonesia.
- Rane, M. (2023). Face Recognition Using Convolutional Neural Network (CNN). *SSmart Trends in Computing and Communications. SMART 2023*, 645, 203–214.
- Raschka, S., Patterson, J., & Nolet, C. (2020). Machine learning in python: Main developments and technology trends in data science, machine learning, and artificial intelligence. In *Information (Switzerland)* (Vol. 11, Issue 4). MDPI AG. <https://doi.org/10.3390/info11040193>
- Saleh, hyatt. (2020). *The deep Learning with Pyorch Workshop* (Vol. 1). Mumbai: Pack Publishing.
- Saleem, M. A., Senan, N., Wahid, F., Aamir, M., Samad, A., & Khan, M. (2022). Comparative Analysis of Recent Architecture of Convolutional Neural Network. *Mathematical Problems in Engineering*, 2022. <https://doi.org/10.1155/2022/7313612>
- Setiawan, W. (2020). *Deep Learning Menggunakan Convolutioanl Neural Network Teori dan Aplikasi* (Vol. 1). Malang: Media Nusa Creative.

- Seo, H., Back, S., Lee, S., Park, D., Kim, T., & Lee, K. (2020). Intra- and inter-epoch temporal context network (IITNet) using sub-epoch features for automatic sleep scoring on raw single-channel EEG. *Biomedical Signal Processing and Control*, 61. <https://doi.org/10.1016/j.bspc.2020.102037>
- Shidiq, A. L. A., Suhartono, E., & Saidah, S. (2022). Klasifikasi Kecacatan Ban Untuk Mengendalikan Kualitas Produk Menggunakan Model CNN Dengan Arsitektur VGG-16. *E-Proceeding of Engineering*, 8(6), 3216–3225.
- Siahaan, V., & Sianipar, R. H. (2020). *Pengantar Pengolahan Citra Digital* (Vol. 1). Sumatera Utara: Balige Publishing.
- Sivamani, C., Rajeswari, M., Julie, E. G., Robinson, Y. H., Shanmuganathan, V., Kadry, S., & Nam, Y. (2021). Tyre inspection through multi-state convolutional neural networks. *Intelligent Automation and Soft Computing*, 27(1), 1–13. <https://doi.org/10.32604/iasc.2021.013705>
- Sudipa, I. G. I., Udayana, I. P. agus E. D., Kharisma, P. I., Efitri, Ariana, A. A. G. B., & Rachman, A. (2023). *Metode Penelitian Bidang Ilmu Informatika (Teori & refernsi Berbais Studi Kasus)* (Vol. 1). Kota Jambi: PT. Sonpedia Publish Indonesia.
- Sundaram, S., & Zeid, A. (2023). *Artificial Intelligence-Based Smart Quality Inspection for Manufacturing. Micromachines*, 14(3). <https://doi.org/10.3390/mi14030570>
- Swaminathan, N., & Parente Alessandro. (2023). *Machine Learning and Its Application to Reacting Flows ML and Combustion* (Vol. 44). Cham: Springer.
- Tamborski, M., Rojek, I., & Mikołajewski, D. (2023). Revolutionizing Tire Quality Control: AI's Impact on Research, Development, and Real-Life Applications. In *Applied Sciences (Switzerland)* (Vol. 13, Issue 14). Multidisciplinary Digital Publishing Institute (MDPI). <https://doi.org/10.3390/app13148406>
- Tong, Z., Gao, J., & Tong, S. (2020). A convolutional neural network approach for visual recognition in wheel production lines. *International Journal of Advanced Robotic Systems*, 17(3), 1–13. <https://doi.org/10.1177/1729881420926879>
- Tulbure, A. A., Tulbure, A. A., & Dulf, E. H. (2022). A review on modern defect

- detection models using DCNNs – Deep convolutional neural networks. In *Journal of Advanced Research* (Vol. 35, pp. 33–48). Elsevier B.V. <https://doi.org/10.1016/j.jare.2021.03.015>
- Taye, M. M. (2023). Theoretical Understanding of Convolutional Neural Network: Concepts, Architectures, Applications, Future Directions. In *Computation* (Vol. 11, Issue 3). MDPI. <https://doi.org/10.3390/computation11030052>
- Vasan, V., Sridharan, N. V., Prabhakaranpillai Sreelatha, A., & Vaithiyanathan, S. (2023). Tire Condition Monitoring Using Transfer Learning-Based Deep Neural Network Approach. *Sensors*, 23(4), 1–18. <https://doi.org/10.3390/s23042177>
- Wang, R., Guo, Q., Lu, S., & Zhang, C. (2019). Tire Defect Detection Using Fully Convolutional Network. *IEEE Access*, 7, 43502–43510. <https://doi.org/10.1109/ACCESS.2019.2908483>
- Wang, Y., Li, Y., Song, Y., & Rong, X. (2020). The influence of the activation function in a convolution neural network model of facial expression recognition. *Applied Sciences (Switzerland)*, 10(5). <https://doi.org/10.3390/app10051897>
- Wu, Z., Jiao, C., Sun, J., & Chen, L. (2020). Tire Defect Detection Based on Faster R-CNN. *International Conference on Robotics and Rehabilitation Intelligence*, 1(1), 203–218.
- Yahya. (2022). *Data Mining* (Vol. 1). Suka Bumi: CV Jejak.
- Yan, Y., Ren, J., Sun, H., & Williams, R. (2024). Nondestructive Quantitative Measurement for Precision Quality Control in Additive Manufacturing Using Hyperspectral Imagery and Machine Learning. *IEEE Transactions on Industrial Informatics*. <https://doi.org/10.1109/TII.2024.3384609>
- Yang, L., & Jiang, H. (2021). Weld defect classification in radiographic images using unified deep neural network with multi-level features. *Journal of Intelligent Manufacturing*, 32(2), 459–469.
- Yaqub, M., Jinchao, F., Zia, M. S., Arshid, K., Jia, K., Rehman, Z. U., & Mehmood, A. (2020). State-of-the-art CNN optimizer for brain tumor segmentation in magnetic resonance images. *Brain Sciences*, 10(7), 1–19.

<https://doi.org/10.3390/brainsci10070427>

- Zebua, R. sandra, Khairunnisa, Hartatik, Pariyadi, Wahyuningtyas, D. P., Thantawi, A. M., Sudipa, I. G. I., Prayitno, H., Sumakul, G. C., Sepriano, & Kharisma, L. P. I. (2023). *Fenomena Artificial Intelligence (AI)* (Vol. 1). Kota Jambi: PT.Sonpedia Publishing Indonesia.
- Zhang, Y., Cui, X., Liu, Y., & Yu, B. (2018). Tire Defects Classification Using Convolution Architecture for Fast Feature Embedding. *International Journal of Computational Intelligence Systems*, 11, 1056–1066.
- Zhang, C., & Lu, Y. (2021). Study on artificial intelligence: The state of the art and future prospects. *Journal of Industrial Information Integration*, 23. <https://doi.org/10.1016/j.jii.2021.100224>
- Zheng, Z., Zhang, S., Yu, B., Li, Q., & Zhang, Y. (2020). Defect Inspection in Tire Radiographic Image Using Concise Semantic Segmentation. *IEEE Access*, 8, 112674–112687. <https://doi.org/10.1109/ACCESS.2020.3003089>