

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

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.

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.

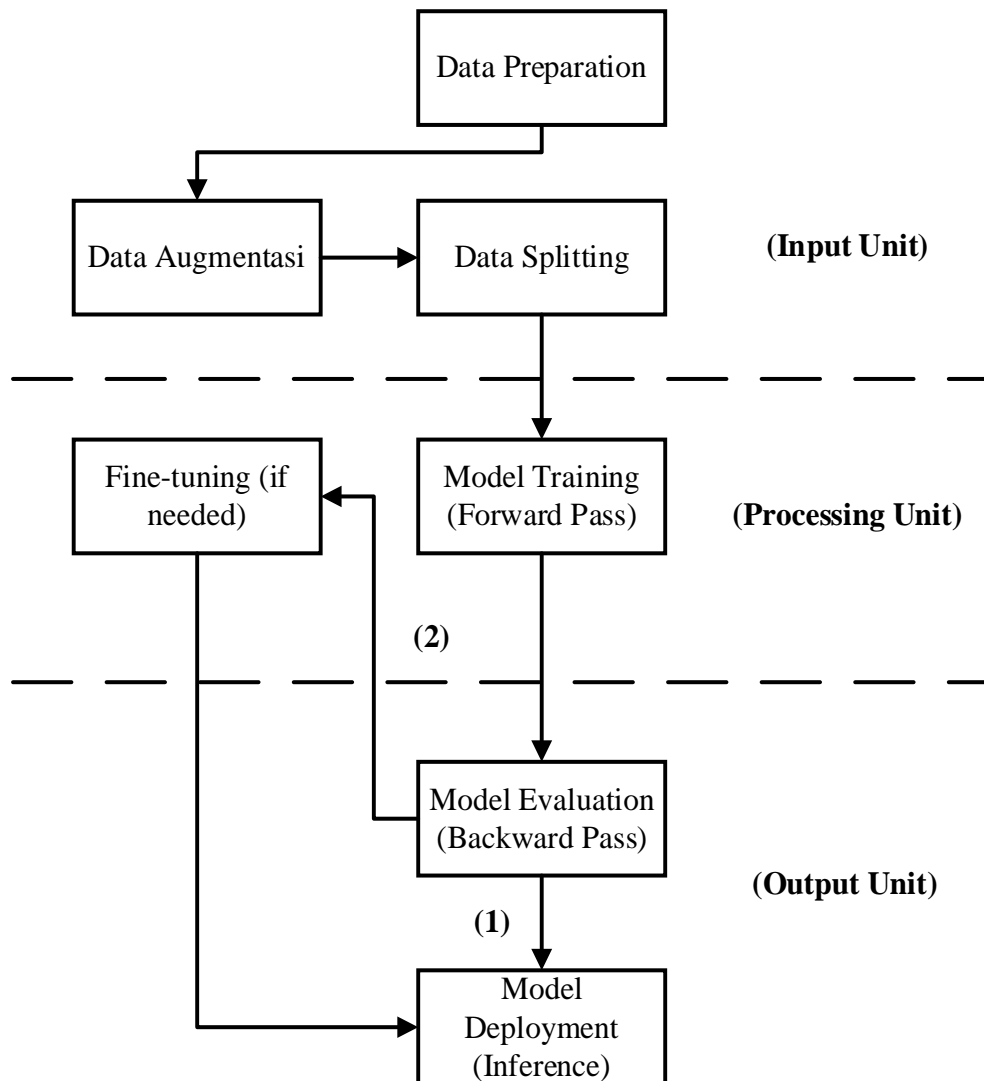
Pengembangan algoritma *Convolutional Neural Networks* (CNN) salah satunya dengan melakukan perbandingan pada komponen-komponen di dalamnya berdasarkan referensi sumber penelitian terdahulu sehingga membentuk perbandingan versi baru yang dibuat seperti melakukan perbandingan penggunaan *layer* yang sama (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).

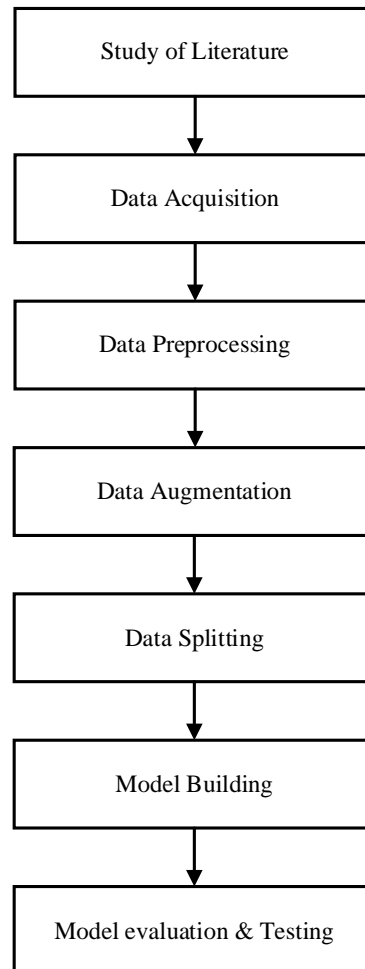
Bagian ketiga adalah unit keluaran yang bertindak ada proses model *evaluation* (*backward pass*, tahap di mana gradien (memperbarui parameter model

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.



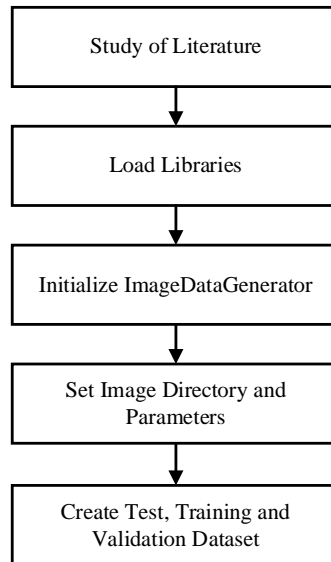
Gambar 3.2 Tahapan Penelitian

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

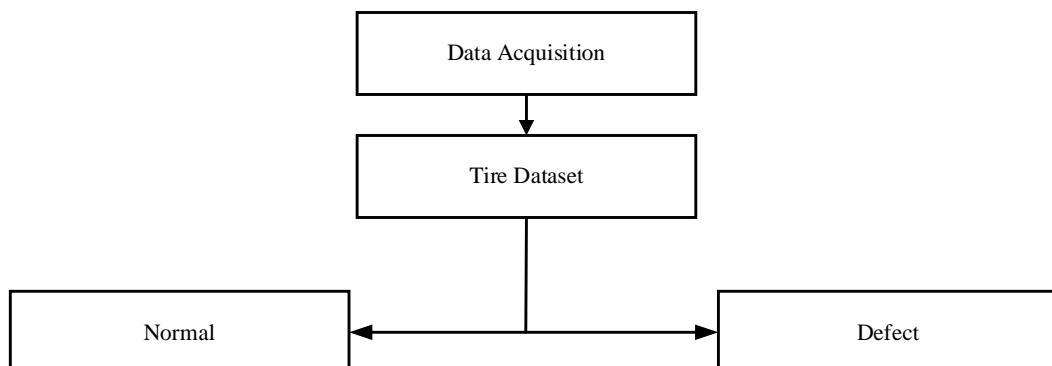
klasifikasi terhadap kondisi-kondisi produk ban. Sehingga dapat menemukan teknik terbaik yang dapat diaplikasikan pada masalah yang ada. Berikut merupakan Gambar 3.3 Tahapan *Study Literature*.



Gambar 3.3 Tahapan *Study Literature*

3.2.2 Data *Aquisition*

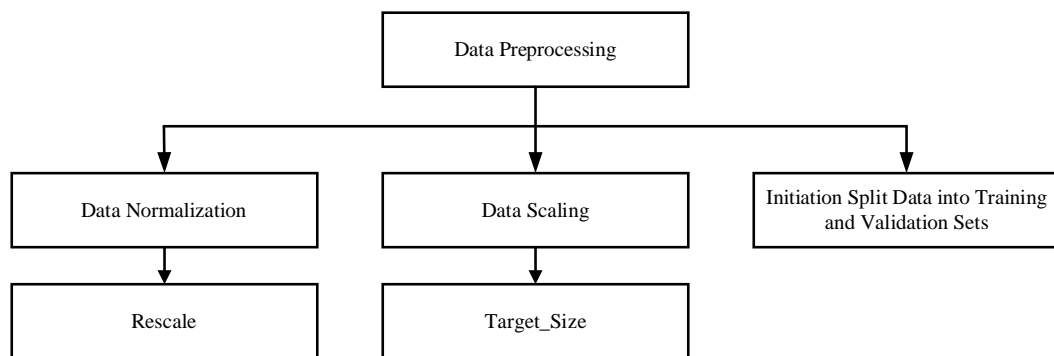
Tahap kedua adalah data *aquisition* dengan mengumpulkan kumpulan data sesuai tujuan penelitian dengan target untuk kumpulan data gambar ban untuk *training* data, *validation* data, dan *testing* data, memastikan bahwa kumpulan data tersebut memiliki varian yang secara akurat memang mewakili kondisi 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

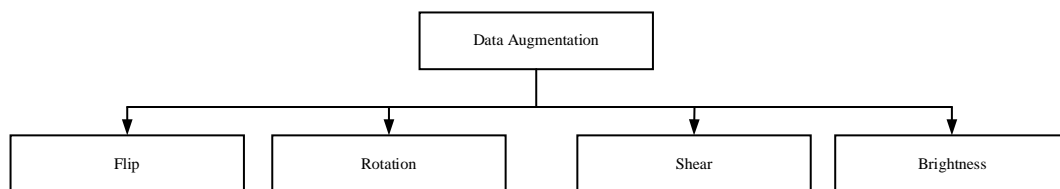
yang ada pada kenyataannya.

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.



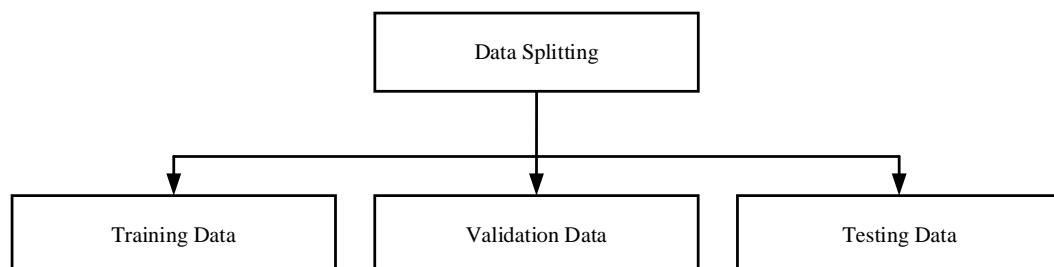
Gambar 3.6 Tahapan Data Augmentation

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.



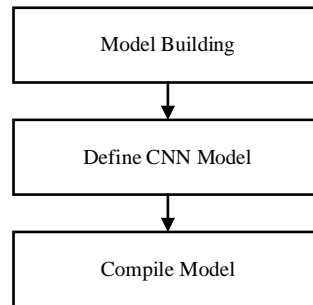
Gambar 3.7 Tahapan *Splitting* Data

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*

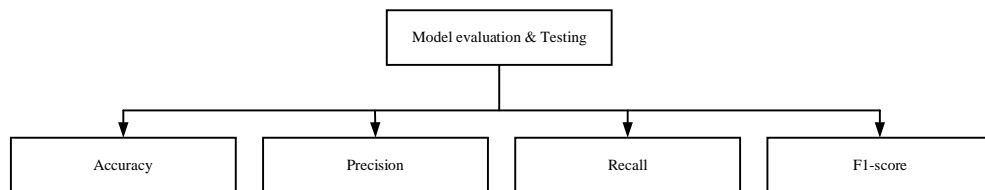
$$\begin{aligned} \text{Total Training data} &= \text{Jumlah Batch} \times \text{Batch Size} \times \text{Jumlah Epochs} \\ &= 36 \times 64 \times 100 \\ &= 230.400 \end{aligned}$$

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

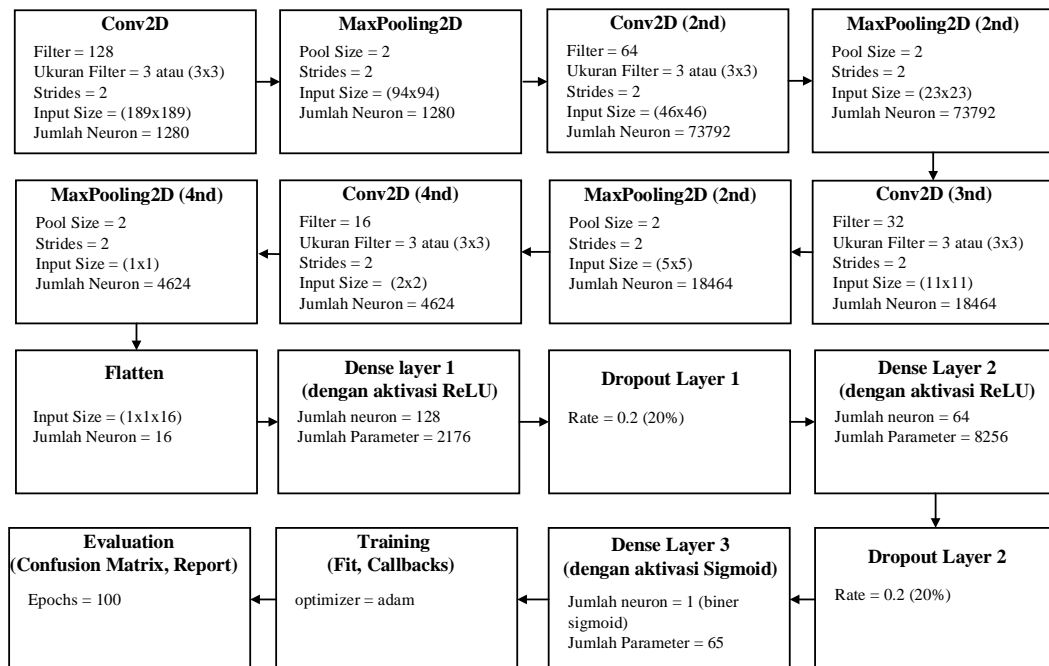
Tahap kedelapan adalah model *evaluation & testing* digunakan sebagai bahan terusan pada model *building* yang dibuat untuk melakukan evaluasi performanya dengan menggunakan bagian pengujian, dan parameter yang digunakan pada metrik evaluasi seperti akurasi, presisi, *recall*, dan *F1-score*. Berikut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

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

$$= \frac{44}{2} + 1$$

$$= 23$$

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.