

BAB III

METODE PENELITIAN

3.1. Objek Penelitian

Penelitian ini bertujuan mengembangkan Algoritma *Convolutional Neural Network* untuk meningkatkan presisi dan efektivitas dalam prosedur pemeriksaan kualitas. Fokus utama adalah pada sistem pendeteksian cacat roda gigi sentris di lingkungan manufaktur dengan memanfaatkan data visual dari foto produk untuk mengidentifikasi kerusakan atau cacat. Pendekatan ini diharapkan mampu mengatasi kelemahan inspeksi konvensional yang cenderung memerlukan banyak tenaga kerja dan rentan terhadap kesalahan.

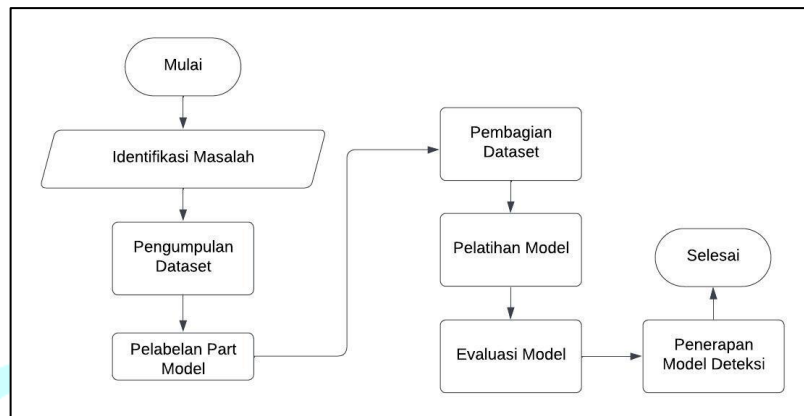
Dataset yang disertakan dalam penelitian ini mencakup foto-foto roda gigi sentris yang berada dalam kondisi baik dan diambil menggunakan kamera DSLR dan kotak foto. Setiap komponen roda gigi harus sepenuhnya terdeteksi agar dapat diklasifikasikan sebagai *Good*. Jika tidak, roda gigi akan diklasifikasikan sebagai *Not Good* (NG) atau beberapa bagian tidak ditemukan.

Penelitian ini dilaksanakan selama empat bulan dari Januari hingga April, dan melibatkan beberapa tahapan berikut:

Tabel 3. 1 Tabel Pelaksanaan Penelitian

NO	Deskripsi	Januari				Februari				Maret				April			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1.	Identifikasi Masalah																
2	Pengumpulan Data																
3.	Pengolahan Data																
4.	Evaluasi Model																
5.	Analisis dan Dokumentasi																

3.2. Prosedur Penelitian



Gambar 3.1. Tahapan Penelitian

3.3. Identifikasi Masalah

Menemukan masalah dengan identifikasi cacat pada roda gigi sentrik dalam lingkungan manufaktur adalah langkah pertama. Masalah ini adalah kekurangan teknik konvensional dalam mengidentifikasi cacat secara akurat, yang menjadi dasar untuk menciptakan solusi dengan menggunakan model CNN.

3.4. Pengumpulan Dataset

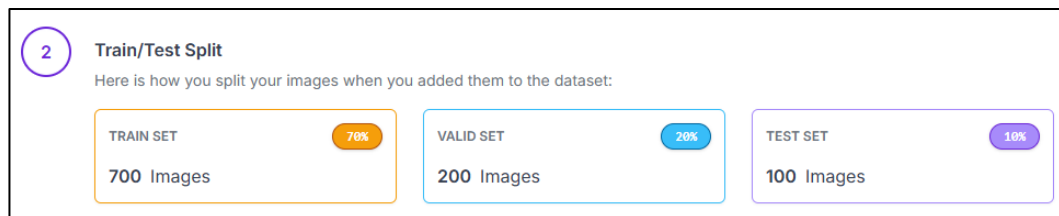
Data dikumpulkan melalui pengambilan foto sebanyak 1000 gambar menggunakan kamera DSLR dan Box Foto. Proses ini melibatkan penggunaan kamera untuk mengambil gambar produk atau objek yang sedang diinspeksi. Dengan mengandalkan teknologi pengambilan gambar, informasi visual yang diperoleh dapat dianalisis secara lebih rinci.

3.5. Pelabelan Part Model

Pelabelan part membantu sistem dalam mengenali dan membedakan setiap komponen agar proses deteksi berjalan akurat serta pelatihan model menjadi lebih optimal karena data yang digunakan sudah jelas dan terstruktur dengan baik. Model dapat mempelajari pola dari setiap part yang berbeda dan hasil deteksi lebih mudah dipahami. Sistem juga mampu mengenali part yang saling tumpang tindih dan mendeteksi kerusakan atau bentuk yang tidak sesuai. Pelabelan mempermudah pelacakan part dalam proses produksi dan mendukung analisis performa sistem secara menyeluruh.

3.6. Pembagian Dataset

Dalam penyusunan model data, gambar-gambar akan dibagi menjadi tiga set, yakni *train set*, *test set*, dan *val set*. Pada tahap awal, sebelum dilakukan augmentasi dan *preprocessing*, data direncanakan dibagi dengan proporsi 70% untuk *train set*, 20% untuk *validation set*, dan 10% untuk *test set*.



Gambar 3.2. Pembagian Dataset

Setelah proses augmentasi dan *preprocessing* dilakukan, jumlah gambar bertambah menjadi 7.359, sehingga proporsi pembagian data berubah menjadi 88% untuk *train set*, 8% untuk *validation set*, dan 4% untuk *test set*. Proses ini dilakukan menggunakan platform Roboflow, yang memudahkan pengelolaan dataset secara otomatis dan seimbang. Pelabelan objek dilakukan secara manual pada setiap gambar, dengan penyesuaian label sesuai dengan kelas keluaran yang telah ditetapkan, mengikuti standar pelabelan yang digunakan dalam arsitektur YOLO. Selain itu, penggunaan Roboflow juga membantu menghasilkan *file* konfigurasi *data.yaml* yang memuat semua daftar kelas yang diperlukan untuk keperluan pelatihan model.

3.7. Pelatihan Model

Memaksimalkan kecepatan dan kinerja deteksi, prosedur pelatihan jaringan YOLOv8 dimulai dengan pemilihan model berdasarkan ukuran (YOLOv8s, YOLOv8m, YOLOv8i, atau YOLOv8x). Dataset berlabel Roboflow digunakan untuk pelatihan, dengan penekanan pada kelas berat. Menginstal kerangka kerja *Ultralytics*, termasuk penyesuaian model misalnya segmentasi, merupakan langkah yang diperlukan dalam tahap persiapan sebelum proses pelatihan dimulai. Proses pelatihan memerlukan waktu yang cukup lama, karena model harus mempelajari pola dari ribuan data gambar. Setelah proses pelatihan selesai, sistem akan menyimpan model terbaik ke dalam file bernama *best.pt* yaitu *file* yang berisi

bobot (*weights*) dari model dengan kinerja terbaik berdasarkan hasil evaluasi selama pelatihan.

3.8. Evaluasi Model

Salah satu cara untuk mengevaluasi keberhasilan suatu jaringan adalah dengan menggunakan *Confusion Matrix*. *Confusion Matrix* adalah suatu ringkasan dari hasil prediksi dalam konteks permasalahan klasifikasi (Amwin, n.d.) *Confusion Matrix* membandingkan jumlah klasifikasi yang benar terjadi, termasuk yang diprediksi dengan benar (*True Positive*) dan yang diprediksi tidak terjadi (*True Negative*), dengan jumlah klasifikasi yang salah terjadi, baik yang diprediksi terjadi padahal tidak (*False Positive*) maupun yang diprediksi tidak terjadi padahal terjadi (*False Negative*). Dengan memanfaatkan nilai *True Positive* (TP), *False Positive* (FP), *True Negative* (TN), dan *False Negative* (FN), dapat dirumuskan persamaan sebagai berikut.

$$1. \text{ Accuracy} = \frac{\text{(Total True Positive)}}{\text{2. (Total Data)}} \quad 3. (1)$$

$$4. \text{ Precision} = \frac{TP}{(TP+FN)} \quad 5. (2)$$

$$6. \text{ Recall} = \frac{TP}{(TP+FN)} \quad 7. (3)$$

Accuracy mengukur tingkat keakuratan model dalam mendeteksi objek yang sedang diinvestigasi. *Error Classification* mencerminkan nilai kesalahan yang muncul dalam proses klasifikasi objek. *Precision* menggambarkan nilai presisi yang diperoleh dari perbandingan positif yang diprediksi dengan benar oleh sistem. *Recall* merupakan rasio dari total data positif yang berhasil diklasifikasikan dengan benar, dibagi dengan total sampel positif. *F1 Score* dirancang untuk

menghitung kombinasi antara presisi dan *recall*. *Average Precision* adalah perhitungan yang diperoleh dari evaluasi *Recall* dan *Precision*, yang selanjutnya dihitung dengan menggunakan grafik *Interpolated Precision* (Sanubari & Puriyanto, 2022).



3.9. Penerapan Model Deteksi

Setelah melalui proses pelatihan dan evaluasi, model YOLOv8m-seg diimplementasikan untuk mendeteksi cacat pada komponen roda gigi sentrik. Model menganalisis gambar input dan mengklasifikasikan hasil deteksi menjadi dua kategori: **Good**, jika semua bagian terdeteksi dengan benar, dan **NG**, jika ada bagian yang tidak terdeteksi atau cacat.

Penerapan model dilakukan menggunakan *Visual Studio Code (VS Code)*, dengan memasukkan file model hasil pelatihan terbaik (**best.pt**) ke dalam program deteksi. Model kemudian diuji pada data uji untuk memperoleh hasil deteksi yang akurat.

