# BAB III
# METODE PENELITIAN

## 3.1 Objek Penelitian

Objek penelitian ini adalah sistem inspeksi cacat pada produk hasil proses *stamping* baja yang dikembangkan untuk membantu operator *Quality Control* (QC) dalam melakukan pemeriksaan visual terhadap kondisi produk. Fokus penelitian diarahkan pada pemanfaatan metode *deep learning* YOLOv8 untuk mendeteksi dan mengklasifikasikan cacat berdasarkan karakteristik visual yang terdapat pada permukaan produk.

Produk yang menjadi objek penelitian merupakan part hasil proses *stamping* baja yang memiliki bentuk dan karakteristik permukaan tertentu sesuai dengan kebutuhan produksi. Pemeriksaan dilakukan terhadap bagian permukaan produk yang berpotensi mengalami cacat setelah proses *stamping*. Jenis cacat yang menjadi fokus dalam penelitian ini terdiri atas tiga kelas, yaitu *black spot*, *dent*, dan *scratch*. Ketiga jenis cacat tersebut memiliki karakteristik visual yang berbeda, sehingga digunakan sebagai kelas objek yang harus dipelajari oleh model YOLOv8.

*Black spot* merupakan cacat yang secara visual tampak sebagai bercak atau titik berwarna gelap pada permukaan produk. *Dent* merupakan cacat berupa perubahan bentuk atau cekungan pada permukaan produk, sedangkan *scratch* merupakan cacat berupa goresan atau garis pada permukaan produk. Karakteristik visual dari masing-masing cacat tersebut menjadi dasar dalam proses anotasi dan pelatihan model.

Penelitian dilakukan pada lingkungan industri *stamping* baja dengan menggunakan citra produk yang diperoleh melalui proses akuisisi menggunakan kamera pada *box* inspeksi. Citra tersebut selanjutnya digunakan sebagai dataset untuk proses *pre-processing*, anotasi, pelatihan, validasi, pengujian, dan evaluasi model YOLOv8.

Selain model deteksi, objek penelitian juga mencakup sistem berbasis *web* yang digunakan untuk mengintegrasikan model YOLOv8 dengan kamera sehingga proses inspeksi dapat dilakukan secara *real-time*. Sistem tersebut dirancang sebagai alat bantu bagi operator QC, sehingga keputusan inspeksi tetap dapat dilakukan oleh operator dengan bantuan informasi hasil deteksi dari sistem.

Lingkup penelitian dibatasi pada proses deteksi cacat berdasarkan karakteristik visual pada citra produk. Penelitian tidak membahas secara mendalam parameter mekanis mesin *stamping*, komposisi material baja, maupun analisis penyebab terjadinya cacat dari sisi proses produksi. Fokus penelitian adalah bagaimana citra produk dapat digunakan sebagai masukan untuk model YOLOv8 dan bagaimana hasil deteksi tersebut diterapkan pada sistem inspeksi berbasis *web*.



Gambar 3. 1 Objek Penelitian

### 3.2 Prosedur Penelitian

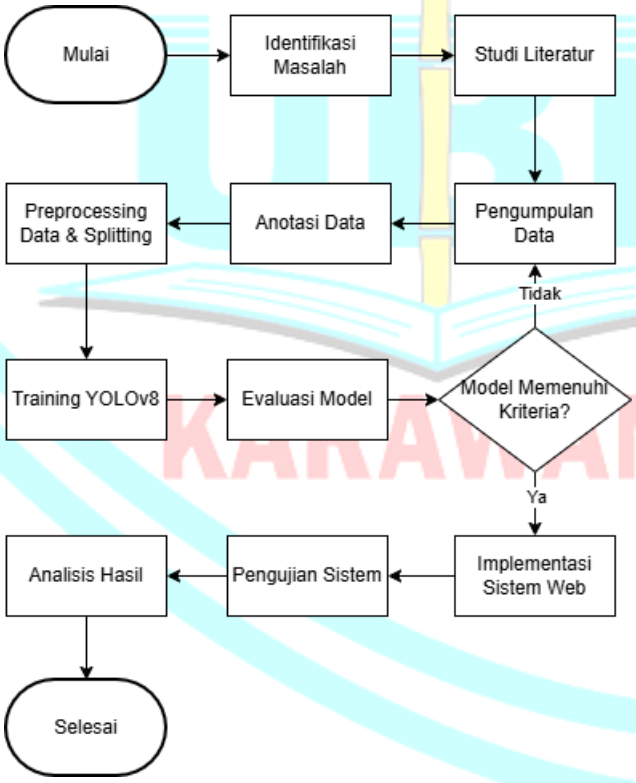
Prosedur penelitian merupakan tahapan yang dilakukan secara sistematis untuk mencapai tujuan penelitian, mulai dari identifikasi permasalahan hingga diperoleh sistem inspeksi cacat berbasis *web*. Tahapan penelitian tidak hanya mencakup proses pembangunan aplikasi, tetapi juga mencakup proses pembentukan dataset dan pengembangan model YOLOv8 sebagai komponen utama dalam proses deteksi.

Secara umum, penelitian dimulai dengan melakukan identifikasi permasalahan dan studi literatur untuk menentukan pendekatan yang sesuai dalam melakukan inspeksi cacat produk *stamping* baja. Selanjutnya dilakukan pengumpulan citra produk sebagai dataset awal. Dataset yang diperoleh kemudian melalui tahap pemeriksaan, *pre-processing*, anotasi, dan augmentasi sebelum digunakan untuk membangun model.

Dataset yang telah dipersiapkan kemudian dibagi menjadi data *training*, *validation*, dan *testing*. Data *training* digunakan untuk melatih model YOLOv8, sedangkan data *validation* digunakan untuk memantau performa model selama proses pelatihan. Setelah pelatihan selesai, model terbaik diuji menggunakan data *testing* yang tidak digunakan dalam proses pelatihan maupun validasi.

Hasil pengujian kemudian dianalisis menggunakan beberapa parameter evaluasi, seperti *Precision*, *Recall*, *mAP@50*, *mAP@50-95*, *F1-Score*, *Precision-Recall Curve*, *Confusion Matrix*, dan *Normalized Confusion Matrix*. Model terbaik selanjutnya diintegrasikan ke dalam aplikasi berbasis *web* menggunakan *Flask*. Tahap terakhir adalah melakukan pengujian terhadap sistem untuk mengetahui kemampuan sistem dalam melakukan deteksi secara *real-time*.

### 3.2.1 Diagram Alur Penelitian




Gambar 3. 2 Diagram Alur Penelitian

Alur penelitian menggambarkan tahapan yang dilakukan mulai dari identifikasi permasalahan hingga diperoleh hasil implementasi sistem. Diagram alur penelitian digunakan untuk memberikan gambaran hubungan antarproses sehingga tahapan penelitian dapat dipahami secara sistematis.

Tahapan penelitian dimulai dari identifikasi masalah dan studi literatur, kemudian dilanjutkan dengan pengumpulan data citra produk *stamping* baja. Data yang diperoleh selanjutnya diperiksa dan dipersiapkan melalui tahap *pre-processing*. Setelah itu dilakukan anotasi terhadap objek cacat dan proses augmentasi untuk meningkatkan variasi dataset.

Dataset yang telah dipersiapkan kemudian dibagi menjadi data *training*, *validation*, dan *testing*. Data *training* digunakan dalam proses pelatihan YOLOv8, sedangkan data *validation* digunakan untuk memantau perkembangan model selama proses pelatihan. Model terbaik yang dihasilkan kemudian diuji menggunakan data *testing* dan dievaluasi berdasarkan parameter yang telah ditentukan.

Setelah model dinyatakan dapat digunakan, model terbaik (*best.pt*) diintegrasikan ke dalam aplikasi berbasis *web* menggunakan *Flask*. Sistem kemudian diuji menggunakan produk dalam kondisi OK dan produk dengan cacat *black spot*, *dent*, serta *scratch* untuk mengetahui kemampuan implementasi sistem dalam melakukan inspeksi secara *real-time*.

### 3.2.2 Kebutuhan Data dan Pengumpulan Data

Data merupakan komponen utama dalam penelitian karena kualitas dan keberagaman data sangat berpengaruh terhadap kemampuan model YOLOv8 dalam mempelajari karakteristik cacat. Data yang digunakan berupa citra produk hasil stamping baja yang diperoleh secara langsung melalui proses akuisisi citra pada area inspeksi.

Pada tahap awal, diperoleh sebanyak **617 citra** produk *stamping* baja yang terdiri atas produk dalam kondisi normal atau OK sebanyak **143 citra** dan produk dalam kondisi NG sebanyak **474 citra**. Produk NG merupakan produk yang memiliki satu atau lebih jenis cacat yang menjadi fokus penelitian, yaitu *black spot*, *dent*, dan *scratch*.

Pengambilan citra dilakukan menggunakan kamera *webcam* Logitech C920 yang ditempatkan pada bagian atas *box* inspeksi. Jarak antara kamera dengan bagian bawah atau area peletakan produk adalah sekitar **30 cm**. Posisi tersebut digunakan agar keseluruhan area produk dapat masuk ke dalam bidang pandang

kamera dan citra yang dihasilkan memiliki ukuran serta komposisi yang relatif konsisten.

*Box* inspeksi digunakan untuk membatasi pengaruh kondisi lingkungan luar terhadap proses pengambilan citra. *Box* memiliki ukuran sekitar **30 cm × 30 cm × 40 cm**. Penggunaan *box* bertujuan menciptakan kondisi pengambilan gambar yang lebih terkontrol, khususnya dari sisi pencahayaan dan latar belakang.

Sistem pencahayaan menggunakan satu buah *LED Flood Light* In-Lite seri INF030 dengan daya **10 watt** dan warna cahaya *Cool Daylight*. Lampu ditempatkan pada bagian belakang *box* dengan ketinggian yang kurang lebih sejajar dengan posisi kamera. Arah lampu dibuat miring sekitar **120° ke bawah menuju bagian depan** area peletakan produk. Konfigurasi tersebut digunakan untuk memberikan pencahayaan yang cukup pada permukaan produk sehingga karakteristik cacat dapat terlihat dengan lebih jelas.

Penggunaan satu sumber lampu dilakukan agar kondisi pencahayaan selama proses pengambilan citra relatif konsisten. Lampu yang digunakan juga memiliki perlindungan **IP65**, sehingga memiliki ketahanan terhadap masuknya debu serta perlindungan terhadap air dari arah tertentu. Dalam penelitian ini, lampu berfungsi sebagai sumber pencahayaan terkontrol pada area inspeksi.

Bagian dalam *box* inspeksi menggunakan **latar belakang berwarna putih**. Pemilihan warna tersebut dilakukan berdasarkan pengamatan pada tahap awal pengambilan data. Kondisi awal *box* menggunakan latar belakang berwarna hitam, namun warna tersebut memiliki karakteristik visual yang menyerupai salah satu objek cacat yang menjadi fokus penelitian, yaitu *black spot*.

Apabila latar belakang berwarna hitam digunakan, terdapat kemungkinan model mengenali bagian latar belakang tertentu sebagai *black spot* karena memiliki kemiripan warna dan tingkat kegelapan. Kondisi tersebut dapat meningkatkan potensi *false positive* pada proses deteksi. Oleh karena itu, latar belakang diganti menjadi warna putih agar kontras antara latar belakang dengan objek *black spot* menjadi lebih jelas.

Penggunaan latar belakang putih juga membantu menghasilkan kondisi citra yang lebih seragam sehingga model dapat lebih fokus mempelajari karakteristik objek cacat pada permukaan produk daripada mempelajari pola dari latar belakang.

Proses pengambilan citra dilakukan dengan meletakkan produk pada area inspeksi di dalam *box*. Produk kemudian diposisikan di bawah kamera dan dilakukan pengambilan citra pada bagian permukaan yang akan diperiksa.

Untuk memperoleh variasi citra, produk diputar dengan perubahan posisi yang relatif kecil sebelum dilakukan pengambilan gambar berikutnya. Perubahan tersebut dilakukan secara terbatas agar karakteristik produk tetap berada pada area pengambilan citra, tetapi menghasilkan variasi posisi dan orientasi objek.

Pengambilan citra dilakukan pada bagian depan dan belakang produk sehingga karakteristik cacat pada kedua sisi dapat direpresentasikan dalam dataset. Dengan demikian, dataset tidak hanya berisi objek dengan satu orientasi tertentu, tetapi memiliki variasi posisi dan orientasi yang dapat membantu model mempelajari karakteristik cacat secara lebih baik.

Setelah seluruh citra terkumpul, dilakukan pemeriksaan terhadap data untuk menghindari penggunaan citra yang tidak sesuai, terlalu buram, tidak menampilkan objek secara jelas, atau tidak memiliki informasi visual yang diperlukan untuk proses anotasi.



Gambar 3. 3 Proses Pengambilan Data Citra

### 3.2.3 Anotasi Data

Setelah proses pengumpulan data citra selesai dilakukan, tahap berikutnya adalah melakukan anotasi terhadap citra produk *stamping* baja. Anotasi bertujuan untuk memberikan informasi kepada model mengenai lokasi dan jenis cacat yang terdapat pada setiap citra. Proses anotasi menjadi tahap penting karena hasil anotasi

digunakan sebagai *ground truth* atau acuan dalam proses pembelajaran model YOLOv8.

Proses anotasi dilakukan menggunakan *platform* Roboflow. Setiap citra diperiksa secara visual untuk mengidentifikasi keberadaan cacat pada permukaan produk. Apabila ditemukan cacat, dibuat *bounding box* yang mengelilingi area cacat tersebut. *Bounding box* ditempatkan sedekat mungkin dengan area cacat agar informasi lokasi objek yang diberikan kepada model sesuai dengan kondisi sebenarnya.

Pada penelitian ini digunakan tiga kelas cacat, yaitu:

1. *Black spot*
2. *Dent*
3. *Scratch*

Setiap objek cacat yang ditemukan pada citra diberikan label sesuai dengan kelasnya. Apabila dalam satu citra terdapat lebih dari satu objek cacat, maka setiap objek diberikan *bounding box* dan label masing-masing. Dengan demikian, satu citra dapat memiliki lebih dari satu anotasi apabila terdapat beberapa cacat pada permukaan produk.

Khusus untuk cacat *black spot*, proses anotasi dilakukan dengan memperhatikan batas area bercak yang terlihat pada permukaan produk. Hal tersebut dilakukan karena ukuran *black spot* pada beberapa citra relatif kecil sehingga ketepatan penentuan area *bounding box* menjadi penting agar model dapat mempelajari karakteristik cacat tersebut dengan baik. Hal yang sama diterapkan pada kelas *scratch* dan *dent*, yaitu dengan menentukan area cacat berdasarkan bagian permukaan produk yang menunjukkan karakteristik visual dari masing-masing kelas.

Setelah seluruh objek cacat diberikan *bounding box* dan label, dilakukan pemeriksaan kembali terhadap hasil anotasi. Pemeriksaan dilakukan untuk memastikan tidak terdapat objek cacat yang terlewat, kesalahan pemberian kelas, maupun *bounding box* yang terlalu luas atau tidak sesuai dengan posisi cacat. Dataset hasil anotasi kemudian disimpan dalam format yang kompatibel dengan YOLOv8.

Hasil anotasi selanjutnya menjadi dasar untuk tahap *pre-processing*, augmentasi, dan pembagian dataset. Dengan adanya anotasi yang sesuai, model dapat mempelajari hubungan antara karakteristik visual, lokasi, dan kelas cacat pada produk *stamping* baja.

### 3.2.4 *Pre-processing* dan Augmentasi Data

Setelah proses anotasi selesai, tahap berikutnya adalah *pre-processing* dan augmentasi data. Tahap ini dilakukan untuk mempersiapkan citra agar memiliki format dan ukuran yang sesuai dengan kebutuhan model YOLOv8 sekaligus meningkatkan keberagaman data yang digunakan dalam proses pelatihan.

Proses *pre-processing* dan pengelolaan dataset dilakukan menggunakan Roboflow. Tahap pertama dilakukan melalui proses *Auto Orient* untuk memastikan orientasi setiap citra berada pada posisi yang sesuai. Penyeragaman orientasi diperlukan agar perbedaan orientasi file citra tidak menyebabkan ketidakkonsistenan pada dataset.

Selanjutnya dilakukan proses *Resize* dengan ukuran **640 × 640 piksel**. Ukuran tersebut disesuaikan dengan *image size* yang digunakan pada proses pelatihan YOLOv8. Penyeragaman ukuran bertujuan agar seluruh citra memiliki dimensi masukan yang konsisten ketika diproses oleh model.

Setelah proses *resize*, dilakukan augmentasi data untuk meningkatkan variasi citra. Teknik augmentasi yang digunakan dalam penelitian ini meliputi *Horizontal Flip*, *Rotation*, *Brightness Adjustment*, *Blur*, *Noise*, dan *Random Crop*. Setiap teknik digunakan untuk menghasilkan variasi kondisi citra sehingga model tidak hanya mempelajari objek cacat berdasarkan satu kondisi visual tertentu.

*Horizontal Flip* digunakan untuk memberikan variasi orientasi objek secara horizontal. *Rotation* digunakan untuk menghasilkan variasi kemiringan atau orientasi produk. *Brightness Adjustment* digunakan untuk mensimulasikan perubahan intensitas pencahayaan. Sementara itu, *Blur* dan *Noise* digunakan untuk memberikan variasi terhadap kualitas citra yang mungkin terjadi pada proses akuisisi. *Random Crop* digunakan untuk menghasilkan variasi komposisi citra dengan tetap mempertahankan informasi objek yang diperlukan.

Penerapan augmentasi tersebut menghasilkan peningkatan jumlah data dari dataset awal sebanyak **518 citra menjadi 1.244 citra**. Penambahan variasi data

diharapkan dapat membantu model mengenali karakteristik cacat pada kondisi yang lebih beragam, khususnya terhadap perubahan posisi, orientasi, pencahayaan, dan kualitas citra.

### 3.2.5 Pembagian Dataset

Setelah proses anotasi, *pre-processing*, dan augmentasi selesai dilakukan, dataset kemudian dibagi menjadi tiga bagian, yaitu data *training*, *validation*, dan *testing*. Pembagian dataset dilakukan untuk memberikan fungsi yang berbeda pada masing-masing kelompok data selama proses pengembangan dan evaluasi model.

Data *training* digunakan sebagai data utama dalam proses pembelajaran YOLOv8. Data ini digunakan model untuk mempelajari karakteristik visual dari tiga kelas cacat, yaitu *black spot*, *dent*, dan *scratch*.

Data *validation* digunakan untuk memantau performa model selama proses *training*. Data validasi tidak digunakan secara langsung untuk memperbarui parameter model, tetapi digunakan untuk mengetahui kemampuan model ketika diberikan data yang berbeda dari data *training*. Hasil validasi juga menjadi salah satu pertimbangan dalam menentukan model dengan performa terbaik.

Sementara itu, data *testing* digunakan setelah proses pelatihan selesai. Data ini tidak digunakan dalam proses pembelajaran maupun validasi sehingga dapat digunakan untuk mengukur kemampuan model dalam melakukan deteksi terhadap data yang belum pernah digunakan sebelumnya.

Pada penelitian ini, dataset dibagi dengan komposisi **70% *training*, 20% *validation*, dan 10% *testing***.

Tabel 3. 1 Pembagian Dataset

| **Dataset** | **Persentase** |
|---|---|
| *Training* | 70% |
| *Validation* | 20% |
| *Testing* | 10% |

Pembagian tersebut dilakukan agar proses pelatihan dan evaluasi dapat dilakukan secara terpisah. Dengan demikian, performa model pada tahap *testing* dapat memberikan gambaran mengenai kemampuan model dalam melakukan generalisasi terhadap data yang belum digunakan selama proses pembelajaran.

### 3.2.6 Proses Pelatihan Model YOLOv8

Dataset yang telah melalui proses anotasi, *pre-processing*, augmentasi, dan pembagian dataset selanjutnya digunakan dalam proses pelatihan model YOLOv8. Pada tahap ini, data *training* digunakan untuk melatih model dalam mengenali karakteristik visual dari tiga kelas cacat, sedangkan data *validation* digunakan untuk memantau perkembangan performa model selama proses pelatihan.

Model yang digunakan adalah **YOLOv8n**. Pemilihan model tersebut disesuaikan dengan kebutuhan penelitian karena sistem ditujukan untuk melakukan deteksi secara *real-time* menggunakan perangkat dengan pemrosesan berbasis CPU.

Proses pelatihan dilakukan menggunakan **100 *epoch*** dengan ukuran citra **640 × 640 piksel** dan *batch size* sebesar **8**. Selama proses pelatihan, model melakukan pembelajaran terhadap karakteristik objek berdasarkan data yang telah dianotasi. Performa model kemudian dipantau berdasarkan hasil pada data validasi.

Dari proses pelatihan tersebut diperoleh model dengan performa terbaik yang disimpan sebagai **best.pt**. Model *best.pt* kemudian digunakan pada tahap *testing* dan evaluasi untuk mengetahui kemampuan model dalam mendeteksi ketiga jenis cacat pada data yang belum digunakan selama proses pembelajaran.

### 3.2.7 Pengujian (*Testing*) dan Evaluasi Model YOLOv8

Setelah proses pelatihan selesai, model *best.pt* diuji menggunakan *testing dataset*. Data pengujian merupakan data yang telah dipisahkan sebelumnya dan tidak digunakan dalam proses *training* maupun *validation*. Tahap ini dilakukan untuk mengetahui kemampuan model dalam mendeteksi dan mengklasifikasikan cacat pada data yang belum pernah digunakan selama proses pembelajaran.

Pada proses *testing*, setiap citra pada dataset pengujian diproses menggunakan model YOLOv8. Model menghasilkan prediksi berupa lokasi *bounding box*, kelas cacat, dan nilai *confidence*. Hasil prediksi tersebut kemudian dibandingkan dengan *ground truth* hasil anotasi untuk mengetahui kesesuaian antara hasil prediksi model dengan kondisi sebenarnya.

Berdasarkan hasil perbandingan tersebut diperoleh komponen *True Positive* (TP), *False Positive* (FP), dan *False Negative* (FN). Nilai tersebut kemudian digunakan untuk menghitung berbagai parameter evaluasi model.

Parameter yang digunakan dalam penelitian ini meliputi *Precision*, *Recall*, *F1-Score*, *Precision-Recall Curve*, *Confusion Matrix*, *Normalized Confusion Matrix*, *mAP@50*, dan *mAP@50-95*. *Precision* digunakan untuk mengetahui ketepatan prediksi objek yang dilakukan model, sedangkan *Recall* digunakan untuk mengetahui kemampuan model dalam menemukan objek cacat yang terdapat pada citra.

Selain itu, *mAP@50* digunakan untuk mengukur kemampuan deteksi dengan batas *Intersection over Union* (IoU) sebesar 0,5. Sementara itu, *mAP@50-95* digunakan untuk mengevaluasi performa model pada rentang IoU yang lebih ketat, yaitu dari 0,5 hingga 0,95.

Hasil evaluasi tersebut digunakan untuk mengetahui kemampuan model dalam mengenali *black spot*, *dent*, dan *scratch*. Model yang telah melalui proses pengujian dan evaluasi kemudian digunakan pada tahap implementasi sistem berbasis web.

### 3.2.8 Implementasi Sistem Berbasis Web

Model *best.pt* yang telah melalui proses *training*, *testing*, dan evaluasi kemudian diintegrasikan ke dalam aplikasi berbasis web menggunakan *framework Flask*. Implementasi ini bertujuan untuk mengubah model hasil penelitian menjadi sistem yang dapat digunakan sebagai alat bantu inspeksi oleh operator *Quality Control* (QC).

Sistem menggunakan kamera sebagai perangkat akuisisi citra. Citra produk yang diperoleh dari kamera diteruskan ke komputer pemrosesan dan diproses menggunakan OpenCV. Selanjutnya, setiap *frame* diproses oleh model YOLOv8 untuk mendeteksi keberadaan cacat pada permukaan produk.

Apabila model mendeteksi minimal satu objek cacat berupa *black spot*, *dent*, atau *scratch*, sistem memberikan status **NG**. Sebaliknya, apabila tidak ditemukan objek cacat pada citra, sistem memberikan status **OK**.

Hasil deteksi kemudian ditampilkan pada antarmuka web melalui Flask. Informasi yang ditampilkan meliputi *bounding box*, label kelas cacat, nilai *confidence*, nilai FPS, serta status inspeksi **OK** atau **NG**. Dengan demikian, operator dapat melihat hasil proses deteksi secara langsung melalui *web browser*.

Implementasi sistem tidak dimaksudkan untuk menggantikan pemeriksaan operator QC sepenuhnya, tetapi digunakan sebagai **alat bantu inspeksi visual**. Operator tetap bertanggung jawab dalam menempatkan produk pada area inspeksi, sedangkan sistem membantu melakukan identifikasi cacat berdasarkan citra secara otomatis dan memberikan informasi hasil deteksi secara *real-time*.

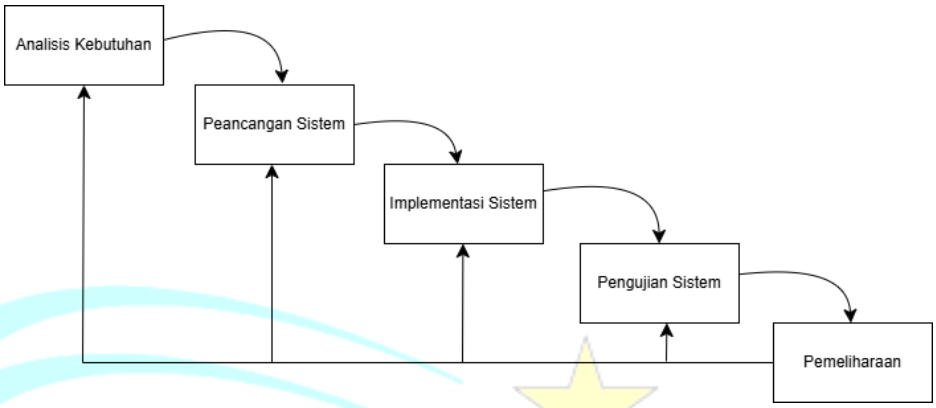
### 3.3 Metode Pengembangan Sistem

Pada penelitian ini, metode pengembangan sistem yang digunakan adalah metode *Waterfall*. Metode *Waterfall* dipilih karena memiliki tahapan pengembangan yang terstruktur dan berurutan, sehingga setiap tahapan dapat diselesaikan terlebih dahulu sebelum dilanjutkan ke tahap berikutnya. Pendekatan ini sesuai dengan kebutuhan pengembangan sistem inspeksi cacat produk *stamping* baja karena kebutuhan sistem, rancangan, implementasi, dan pengujian dapat ditentukan secara sistematis sejak awal.

Metode *Waterfall* yang digunakan dalam penelitian ini terdiri atas lima tahapan utama, yaitu analisis kebutuhan, perancangan sistem, implementasi sistem, pengujian sistem, dan pemeliharaan. Tahapan tersebut digunakan untuk mengembangkan aplikasi inspeksi cacat berbasis *web* yang mengintegrasikan kamera sebagai perangkat akuisisi citra, OpenCV sebagai pengolah dan penghubung citra dari kamera, model YOLOv8 sebagai metode deteksi cacat, serta Flask sebagai *framework* aplikasi berbasis *web*.

Secara umum, proses pengembangan dimulai dari identifikasi kebutuhan sistem berdasarkan proses inspeksi produk yang dilakukan oleh operator *Quality Control* (QC). Selanjutnya dilakukan perancangan alur kerja dan arsitektur sistem, implementasi model YOLOv8 ke dalam aplikasi, pengujian setiap fungsi sistem, dan pemeliharaan apabila ditemukan kesalahan atau diperlukan pengembangan lebih lanjut.

.




Gambar 3. 4 Metode Pengembangan Sistem

### 3.3.1 Analisis Kebutuhan

Tahap analisis kebutuhan merupakan tahap awal dalam pengembangan sistem. Tahap ini dilakukan untuk mengidentifikasi kebutuhan yang diperlukan agar sistem inspeksi cacat dapat menjalankan fungsi sesuai dengan tujuan penelitian. Analisis dilakukan dengan memperhatikan proses inspeksi produk *stamping* baja serta kebutuhan operator dalam melakukan pemeriksaan kondisi produk.

Analisis kebutuhan dibagi menjadi tiga bagian, yaitu kebutuhan perangkat keras, kebutuhan perangkat lunak, dan kebutuhan fungsional sistem.

**1. Kebutuhan Perangkat Keras**

Perangkat keras diperlukan untuk mendukung proses pengambilan citra, pemrosesan model, pencahayaan objek, dan menjalankan aplikasi. Perangkat utama yang digunakan adalah Mini PC Intel NUC dengan prosesor *Intel Core i5-1135G7*, RAM 8 GB, dan SSD 256 GB. Perangkat tersebut digunakan sebagai pusat pemrosesan sistem sekaligus menjalankan aplikasi berbasis *web* dan proses inferensi YOLOv8 menggunakan CPU.

Untuk proses akuisisi citra digunakan webcam Logitech C920. Kamera digunakan untuk mengambil citra permukaan produk yang ditempatkan di dalam *box* inspeksi. Sistem juga menggunakan satu unit LED *Flood Light* sebagai sumber pencahayaan tambahan. Seluruh

perangkat akuisisi ditempatkan pada *box* inspeksi untuk membantu menghasilkan kondisi pengambilan citra yang lebih terkendali.

**2. Kebutuhan Perangkat Lunak**

Perangkat lunak digunakan untuk mendukung proses pengolahan citra, deteksi objek, pengembangan aplikasi, serta pengelolaan dataset. Sistem menggunakan sistem operasi Windows sebagai lingkungan utama pengembangan dan implementasi.

Bahasa pemrograman yang digunakan adalah Python. Model YOLOv8 digunakan untuk melakukan proses deteksi objek cacat, sedangkan OpenCV digunakan untuk mengakses webcam, mengambil *frame* secara *real-time*, serta melakukan pengolahan dan integrasi citra dengan model deteksi. Aplikasi berbasis *web* dikembangkan menggunakan *framework* Flask.

Pada tahap pengelolaan dataset, Roboflow digunakan untuk proses anotasi, *pre-processing*, *augmentation*, dan pengelolaan dataset sebelum digunakan dalam proses pelatihan model.

**3. Kebutuhan Fungsional Sistem**

Berdasarkan kebutuhan proses inspeksi, sistem dirancang agar mampu menjalankan beberapa fungsi utama. Sistem harus mampu mengakses kamera dan menerima citra produk secara *real-time*. Citra yang diperoleh kemudian diproses menggunakan model YOLOv8 untuk mendeteksi tiga jenis cacat, yaitu *black spot*, *dent*, dan *scratch*.

Sistem juga harus mampu menampilkan hasil deteksi melalui antarmuka berbasis *web*. Informasi yang ditampilkan meliputi *bounding box*, label kelas cacat, nilai *confidence*, nilai *Frame Per Second* (FPS), serta status hasil inspeksi berupa **OK** atau **NG**.

Status **OK** diberikan ketika sistem tidak menemukan objek cacat yang termasuk dalam kelas yang digunakan pada penelitian. Sebaliknya,

status **NG** diberikan apabila sistem mendeteksi minimal satu objek cacat pada produk.

### 3.3.2 Perancangan Sistem

Tahap perancangan sistem dilakukan setelah kebutuhan sistem ditentukan. Tahap ini bertujuan untuk menentukan bagaimana setiap komponen perangkat keras dan perangkat lunak akan saling terintegrasi sehingga dapat menghasilkan proses inspeksi cacat secara *real-time*.

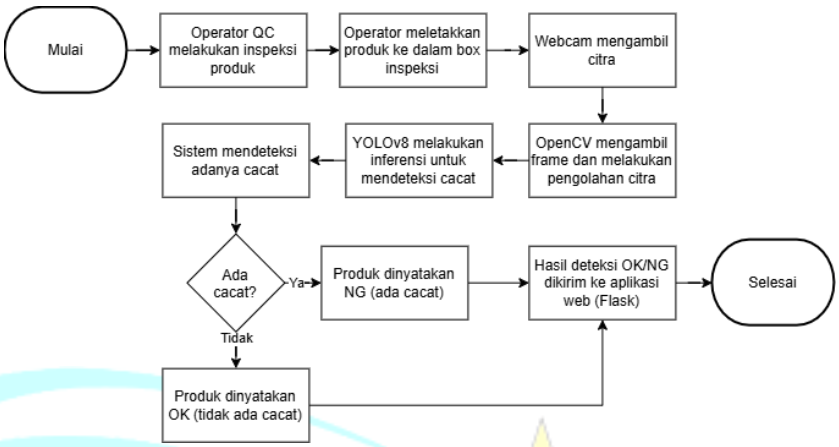
Perancangan sistem meliputi perancangan arsitektur sistem, alur proses inspeksi, integrasi kamera dengan OpenCV, integrasi model YOLOv8 dengan aplikasi Flask, serta rancangan antarmuka pengguna.

Pada proses inspeksi, operator QC terlebih dahulu menempatkan produk *stamping* baja ke dalam *box* inspeksi. Kamera kemudian mengambil citra produk secara *real-time*. Citra dari kamera diterima oleh OpenCV dan diteruskan ke model YOLOv8 untuk proses inferensi.

Hasil inferensi berupa lokasi objek, kelas cacat, dan nilai *confidence* kemudian diproses oleh aplikasi Flask. Sistem menentukan status produk berdasarkan hasil deteksi tersebut. Apabila tidak terdapat cacat, sistem memberikan status **OK**, sedangkan apabila ditemukan salah satu dari tiga kelas cacat, sistem memberikan status **NG**.

Hasil akhir kemudian ditampilkan melalui *web browser* sehingga operator dapat melihat kondisi produk tanpa harus berinteraksi langsung dengan proses pemrosesan model.

Perancangan juga memperhatikan kondisi lingkungan pengambilan citra. Produk ditempatkan pada *box* inspeksi dengan latar belakang berwarna putih. Penggunaan latar belakang putih dipilih untuk memberikan kontras terhadap produk sekaligus mengurangi kemungkinan permukaan latar belakang dianggap sebagai *black spot*. Hal tersebut menjadi pertimbangan karena *black spot* merupakan salah satu kelas cacat yang dideteksi oleh model.




Gambar 3. 5 Alur Perancangan Sistem

### 3.3.3 Implementasi Sistem

Tahap implementasi merupakan proses penerapan rancangan sistem menjadi aplikasi yang dapat digunakan untuk melakukan inspeksi. Implementasi dilakukan dengan mengintegrasikan model YOLOv8 yang telah dilatih dan menghasilkan model terbaik (*best.pt*) ke dalam aplikasi berbasis *web* menggunakan *framework* Flask.

Aplikasi dikembangkan menggunakan bahasa pemrograman Python. OpenCV digunakan untuk menghubungkan aplikasi dengan webcam dan mengambil *frame* secara berkelanjutan. Setiap *frame* yang diperoleh kemudian diteruskan ke model YOLOv8 untuk dilakukan proses inferensi.

Model YOLOv8 menghasilkan informasi berupa lokasi objek cacat dalam bentuk *bounding box*, kelas objek, serta nilai *confidence*. Informasi tersebut kemudian digunakan oleh aplikasi untuk menentukan kondisi produk.

Pada sisi input, webcam berfungsi sebagai perangkat akuisisi citra. OpenCV menerima *frame* dari kamera dan meneruskannya ke model YOLOv8. Model YOLOv8 melakukan proses deteksi terhadap tiga kelas cacat, yaitu *black spot*, *dent*, dan *scratch*. Hasil deteksi selanjutnya diproses oleh Flask untuk ditampilkan pada antarmuka aplikasi.

Antarmuka sistem dirancang untuk menampilkan citra hasil kamera beserta informasi hasil deteksi. Informasi tersebut meliputi *bounding box*, nama kelas cacat, nilai *confidence*, FPS, dan status produk.

Sistem juga menyediakan kontrol Start Camera dan Stop Camera untuk mengatur proses pengambilan citra. Dengan adanya fungsi tersebut, operator dapat memulai dan menghentikan proses inspeksi sesuai kebutuhan.

Dalam penerapannya, sistem tidak dimaksudkan untuk menggantikan operator QC. Operator tetap melakukan proses penempatan dan pemeriksaan produk, sedangkan sistem digunakan sebagai alat bantu untuk mengidentifikasi cacat secara visual secara lebih cepat dan konsisten.

### 3.3.4 Pengujian Sistem

Tahap pengujian dilakukan setelah implementasi sistem selesai. Pengujian bertujuan untuk memastikan bahwa sistem dapat menjalankan fungsi sesuai dengan kebutuhan yang telah ditentukan serta mampu menghasilkan keluaran deteksi yang sesuai dengan kondisi produk.

Pengujian sistem dilakukan menggunakan metode *Black Box Testing*. Metode ini digunakan dengan menguji fungsi sistem berdasarkan masukan (*input*) dan keluaran (*output*) tanpa melihat proses internal program.

Pengujian dilakukan terhadap beberapa fungsi utama sistem, yaitu:

1. Pengujian koneksi dan akuisisi kamera, untuk memastikan webcam dapat diakses dan menghasilkan citra secara *real-time*.
2. Pengujian proses deteksi, untuk memastikan model YOLOv8 dapat memproses citra dan mendeteksi objek cacat.
3. Pengujian klasifikasi status produk, untuk memastikan sistem memberikan status OK ketika tidak ditemukan cacat dan NG ketika ditemukan cacat.
4. Pengujian tampilan hasil deteksi, untuk memastikan *bounding box*, label kelas, *confidence*, dan FPS dapat ditampilkan dengan benar.
5. Pengujian kontrol kamera, untuk memastikan fungsi Start Camera dan Stop Camera berjalan sesuai dengan rancangan.
6. Pengujian antarmuka *web*, untuk memastikan halaman aplikasi dapat diakses dan menampilkan informasi hasil inspeksi dengan baik.

Selain pengujian fungsional menggunakan *Black Box Testing*, dilakukan pula pengujian implementasi menggunakan beberapa kondisi produk. Produk yang digunakan meliputi produk dalam kondisi **OK** serta produk dengan cacat *black spot*, *dent*, dan *scratch*. Setiap kondisi diuji sebanyak 20 kali dengan variasi posisi peletakan produk di dalam *box* inspeksi.

### 3.3.5 Pemeliharaan

Tahap pemeliharaan merupakan tahap setelah sistem selesai diimplementasikan dan diuji. Pemeliharaan dilakukan untuk menjaga agar sistem tetap dapat digunakan sesuai dengan fungsi yang telah dirancang serta memberikan ruang untuk perbaikan apabila ditemukan permasalahan selama penggunaan.

Pemeliharaan dapat dilakukan apabila terdapat *bug* pada aplikasi, kesalahan pada proses akuisisi citra, gangguan pada integrasi webcam, maupun permasalahan pada proses deteksi. Perbaikan dapat dilakukan dengan menyesuaikan kode program, konfigurasi perangkat, maupun parameter sistem.

Selain perbaikan teknis, pemeliharaan juga dapat dilakukan melalui pembaruan dataset. Apabila ditemukan kondisi cacat yang belum dapat dikenali dengan baik oleh model, citra tersebut dapat ditambahkan ke dalam dataset untuk digunakan pada proses pelatihan berikutnya. Dengan demikian, model dapat dikembangkan secara bertahap mengikuti variasi kondisi produk dan lingkungan inspeksi.

Tahap pemeliharaan juga dapat mencakup pengembangan fitur sistem, seperti penambahan jenis cacat baru, penyimpanan riwayat hasil inspeksi, pengembangan laporan hasil pemeriksaan, serta peningkatan antarmuka aplikasi sesuai dengan kebutuhan pengguna.

## 3.4 Perangkat Keras dan Perangkat Lunak

Pada penelitian ini, sistem inspeksi cacat produk *stamping* baja dibangun menggunakan kombinasi perangkat keras (*hardware*) dan perangkat lunak (*software*) yang saling terintegrasi. Perangkat keras digunakan untuk mendukung proses akuisisi citra, penyediaan kondisi pencahayaan, penempatan objek, serta pemrosesan data. Sementara itu, perangkat lunak digunakan untuk melakukan

pengolahan citra, pelatihan dan inferensi model YOLOv8, serta menyediakan antarmuka sistem inspeksi berbasis *web*.

Pemilihan perangkat keras dan perangkat lunak disesuaikan dengan kebutuhan penelitian, khususnya untuk menghasilkan citra produk yang konsisten, menjalankan model YOLOv8 pada perangkat berbasis CPU, serta menampilkan hasil inspeksi secara *real-time*. Konfigurasi perangkat juga dirancang agar proses inspeksi dapat dilakukan pada ruang inspeksi yang memiliki kondisi pencahayaan dan latar belakang yang relatif terkendali.

### 3.4.1 Perangkat Keras (*Hardware*)

Perangkat keras yang digunakan dalam penelitian terdiri atas **Mini PC Intel NUC, webcam Logitech C920, LED *flood light*, dan box inspeksi**. Masing-masing perangkat memiliki fungsi yang berbeda, tetapi saling terintegrasi dalam proses inspeksi cacat.

**1. Mini PC Intel NUC**

Mini PC Intel NUC digunakan sebagai pusat pemrosesan sistem (*processing unit*). Perangkat ini menjalankan seluruh proses komputasi yang diperlukan dalam sistem, mulai dari menerima citra yang diperoleh dari webcam, melakukan pengolahan citra menggunakan OpenCV, menjalankan model YOLOv8 untuk proses inferensi, hingga menjalankan aplikasi berbasis *web* menggunakan Flask.

Penggunaan Mini PC dipilih karena sistem penelitian dirancang untuk dapat melakukan proses deteksi tanpa menggunakan GPU (*Graphics Processing Unit*). Dengan demikian, proses inferensi YOLOv8 dijalankan menggunakan CPU. Kondisi tersebut juga menjadi pertimbangan dalam pemilihan model **YOLOv8n** yang memiliki ukuran model relatif ringan sehingga lebih sesuai untuk perangkat dengan sumber daya komputasi terbatas.

Mini PC yang digunakan dalam penelitian memiliki spesifikasi sebagai berikut:

Tabel 3. 2 Spesifikasi Mini PC Intel NUC

| Komponen | Spesifikasi |
|---|---|
| *Processor* | Intel Core i5-1135G7 |
| *RAM* | 8 GB DDR4 2666Mhz |
| *Storage* | SSD 256 GB |

Mini PC berperan sebagai pusat integrasi antara perangkat akuisisi citra dengan perangkat lunak deteksi. Citra dari webcam diterima oleh Mini PC dan selanjutnya diproses oleh aplikasi yang telah dibangun.



Gambar 3. 6 Mini PC Intel NUC

**2. Webcam Logitech C920**

Webcam Logitech C920 digunakan sebagai perangkat akuisisi citra untuk menangkap permukaan produk *stamping* baja. Kamera ditempatkan pada bagian atas atau sisi belakang box inspeksi dengan posisi yang mengarah ke area permukaan produk.

Webcam digunakan untuk memperoleh citra secara *real-time* yang kemudian diteruskan ke Mini PC untuk diproses menggunakan OpenCV dan model YOLOv8. Kamera Logitech C920 mampu menghasilkan citra dengan resolusi hingga **Full HD 1080p**, sehingga dapat memberikan kualitas citra yang cukup untuk menangkap karakteristik visual cacat pada permukaan produk.

Dalam penelitian ini, posisi kamera dibuat relatif tetap agar jarak antara kamera dan objek tidak berubah secara signifikan selama proses pengambilan citra. Jarak antara kamera dengan bagian dasar box inspeksi adalah sekitar **30 cm**. Pengaturan jarak tersebut dilakukan untuk menjaga ukuran objek pada citra tetap relatif konsisten sehingga model menerima masukan dengan karakteristik visual yang seragam.

Pada proses pengambilan dataset, kamera digunakan untuk mengambil citra bagian depan dan belakang produk. Produk diputar secara perlahan dengan perubahan posisi yang relatif kecil untuk memperoleh cakupan permukaan produk tanpa mengubah kondisi pengambilan citra secara ekstrem.




Gambar 3. 7 Webcam Logitech C920

**3. Sistem Pencahayaan LED**

Sistem pencahayaan digunakan untuk memberikan kondisi pencahayaan yang lebih terkendali selama proses akuisisi citra. Pencahayaan yang tidak konsisten dapat menyebabkan perubahan intensitas cahaya, bayangan, maupun pantulan pada permukaan produk yang berpotensi memengaruhi karakteristik citra.

Pada penelitian ini digunakan **satu buah lampu In-Lite LED Flood Light seri INF030 dengan daya 10 Watt**. Lampu menggunakan warna cahaya **Cool Daylight** dan memiliki tingkat perlindungan **IP65**. Lampu

ditempatkan di bagian belakang box inspeksi dengan posisi ketinggian yang kurang lebih sejajar dengan webcam.

Posisi lampu tidak diarahkan secara tegak lurus terhadap permukaan produk, tetapi dipasang dengan kemiringan sekitar **120° ke arah bawah menuju bagian depan box inspeksi**. Pengaturan tersebut digunakan untuk membantu memberikan pencahayaan pada area produk sekaligus mengurangi kondisi pencahayaan yang terlalu langsung pada permukaan objek.

Penggunaan satu sumber LED juga membuat konfigurasi pencahayaan lebih sederhana dan relatif konsisten selama proses pengambilan citra. Dengan posisi kamera dan lampu yang tetap, kondisi pencahayaan dapat dipertahankan agar perubahan visual pada citra lebih banyak berasal dari karakteristik produk dan cacat yang terdapat pada permukaannya.




Gambar 3. 8 LED *Flood Light*

**4. Box Inspeksi**

Box inspeksi digunakan sebagai ruang pengambilan citra produk *stamping* baja. Box berfungsi untuk membatasi pengaruh kondisi lingkungan luar terhadap proses akuisisi citra, terutama perubahan pencahayaan dan latar belakang.

Box yang digunakan memiliki ukuran sekitar **30 cm pada bagian lebar, 30 cm pada kedalaman, dan 40 cm pada tinggi**. Kamera dan lampu ditempatkan pada posisi tertentu di dalam atau pada bagian box sehingga area produk dapat diamati dengan sudut dan pencahayaan yang relatif tetap.

Penggunaan box inspeksi juga bertujuan untuk menciptakan lingkungan pengambilan citra yang lebih terkontrol. Dengan kondisi tersebut, perubahan pencahayaan dari lingkungan luar dapat diminimalkan sehingga citra yang digunakan sebagai dataset memiliki karakteristik yang lebih konsisten.



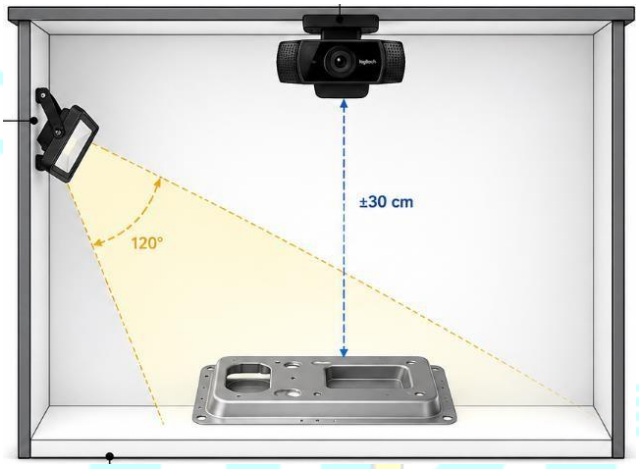
Gambar 3. 9 Box Inspeksi

**6. Konfigurasi Posisi Perangkat pada Box Inspeksi**

Konfigurasi perangkat pada box inspeksi dirancang agar kamera, sumber pencahayaan, dan objek memiliki posisi yang relatif tetap selama proses akuisisi citra. Webcam Logitech C920 ditempatkan pada posisi dengan jarak sekitar **30 cm dari bagian dasar box**. Sementara itu, satu buah LED In-Lite LED Flood Light ditempatkan pada bagian belakang dengan ketinggian yang kurang lebih sejajar dengan webcam.

LED diarahkan dengan kemiringan sekitar **120° ke bawah menuju bagian depan box**. Dengan konfigurasi tersebut, cahaya diarahkan ke area objek tanpa ditembakkan secara langsung dari posisi tegak lurus terhadap permukaan produk.

Konfigurasi ini digunakan untuk menjaga konsistensi kondisi pencahayaan dan posisi kamera selama pengambilan data. Konsistensi tersebut penting karena perubahan posisi kamera maupun pencahayaan yang terlalu besar dapat menghasilkan variasi citra yang tidak berkaitan dengan karakteristik cacat.




Gambar 3. 10 Konfigurasi Posisi Perangkat Pada Box Inspeksi

### 3.4.2 Perangkat Lunak (*Software* dan *Tools*)

Perangkat lunak yang digunakan dalam penelitian terdiri atas bahasa pemrograman, *framework*, model *deep learning*, pustaka pengolahan citra, serta perangkat bantu untuk pengelolaan dataset dan anotasi. Setiap perangkat lunak memiliki fungsi yang berbeda dalam keseluruhan tahapan penelitian.

**1. Python**

Python digunakan sebagai bahasa pemrograman utama dalam pengembangan sistem. Python digunakan untuk mengintegrasikan model YOLOv8 dengan aplikasi inspeksi, mengolah citra menggunakan OpenCV, serta membangun aplikasi berbasis *web* menggunakan Flask.

Penggunaan Python memungkinkan berbagai komponen sistem, mulai dari akuisisi citra hingga inferensi model, diintegrasikan dalam satu lingkungan pemrograman.

**2. Ultralytics YOLOv8**

Ultralytics YOLOv8 digunakan sebagai metode *deep learning* untuk melakukan deteksi objek cacat pada produk *stamping* baja. Model yang digunakan dalam penelitian adalah **YOLOv8n** (*nano*) yang dipilih karena memiliki ukuran model yang relatif ringan dan dapat dijalankan pada Mini PC berbasis CPU.

Model digunakan dalam dua tahap utama, yaitu proses pelatihan menggunakan dataset yang telah dianotasi dan proses inferensi ketika sistem melakukan inspeksi terhadap citra baru.

Dalam penelitian ini, model dilatih untuk mengenali tiga kelas cacat, yaitu:

1. *Black spot*
2. *Dent*
3. *Scratch*

Hasil inferensi model berupa lokasi objek dalam bentuk *bounding box*, kelas objek, serta nilai *confidence* yang selanjutnya digunakan oleh sistem untuk menentukan status inspeksi produk.

**3. OpenCV**

OpenCV digunakan sebagai pustaka *computer vision* untuk menangani proses akuisisi dan pengolahan citra. Pada sistem yang dikembangkan, OpenCV digunakan untuk mengakses webcam Logitech C920 dan mengambil *frame* secara *real-time*.

Citra yang diperoleh dari webcam kemudian diteruskan ke model YOLOv8 untuk proses deteksi. Setelah proses inferensi selesai, hasil deteksi dapat diproses kembali untuk ditampilkan pada antarmuka sistem.

Dengan demikian, OpenCV berperan sebagai penghubung antara perangkat kamera dengan proses deteksi YOLOv8.

**4. Flask**

Flask digunakan sebagai *framework* untuk membangun aplikasi berbasis *web*. Flask berfungsi sebagai bagian server yang menghubungkan proses deteksi YOLOv8 dengan antarmuka yang digunakan operator.

Melalui Flask, hasil pemrosesan citra dapat dikirimkan dan ditampilkan pada *web browser*. Informasi yang ditampilkan meliputi citra hasil inspeksi, *bounding box*, nama kelas cacat, nilai *confidence*, serta status produk.

Penggunaan Flask memungkinkan sistem inspeksi dijalankan melalui *web browser* sehingga operator tidak perlu berinteraksi langsung dengan kode atau program deteksi pada sisi pemrosesan.

**5. Roboflow**

Roboflow digunakan untuk membantu proses pengelolaan dataset dan anotasi citra. Pada tahap anotasi, Roboflow digunakan untuk membuat *bounding box* pada area cacat serta memberikan label berdasarkan kelas cacat.

Selain anotasi, Roboflow digunakan untuk membantu proses persiapan dataset, seperti *resize*, augmentasi, dan pengaturan dataset sebelum digunakan dalam proses pelatihan YOLOv8.

**6. NumPy**

NumPy digunakan sebagai pustaka pendukung untuk pengolahan data numerik dan struktur data berbasis array. Dalam sistem *computer vision*, data citra direpresentasikan dalam bentuk array sehingga NumPy dapat mendukung proses pengolahan data citra sebelum maupun setelah proses inferensi.

**7. PyTorch**

PyTorch digunakan sebagai *framework deep learning* yang menjadi salah satu komponen pendukung dalam proses pelatihan dan inferensi

model YOLOv8. PyTorch menangani proses komputasi model *deep learning* yang digunakan oleh YOLOv8.

### 3.5 Arsitektur Sistem

Arsitektur sistem menggambarkan hubungan antara perangkat keras, perangkat lunak, proses pengolahan citra, model YOLOv8, dan antarmuka pengguna. Sistem dirancang dengan alur utama berupa **input → proses → output**.

Secara keseluruhan, proses dimulai ketika operator QC meletakkan produk pada area inspeksi. Webcam menangkap citra produk, kemudian citra diteruskan ke Mini PC untuk diproses menggunakan OpenCV dan YOLOv8. Hasil deteksi selanjutnya diproses oleh aplikasi Flask dan ditampilkan melalui *web browser*.

#### 3.5.1 Input

Tahap *input* merupakan tahap masuknya objek dan data citra ke dalam sistem. Operator *Quality Control* (QC) meletakkan produk hasil *stamping* baja ke dalam box inspeksi yang telah memiliki kondisi pencahayaan dan background yang telah ditentukan.

Produk kemudian diamati menggunakan webcam Logitech C920. Proses pengambilan citra dilakukan pada bagian depan dan belakang produk. Untuk memperoleh cakupan permukaan yang lebih luas, produk diputar dengan perubahan posisi yang relatif kecil sehingga bagian permukaan yang berbeda dapat masuk ke dalam area pengamatan kamera.

Kondisi pencahayaan dijaga menggunakan satu buah LED In-Lite LED Flood Light INF030 10 Watt yang ditempatkan pada bagian belakang box. Penggunaan background putih dilakukan untuk memberikan perbedaan visual yang lebih jelas terhadap cacat *black spot* dan mengurangi kemungkinan background dikenali sebagai objek cacat.

Citra yang diperoleh webcam kemudian dikirimkan ke Mini PC untuk diproses lebih lanjut.

#### 3.4.2 Proses

Tahap proses merupakan bagian utama sistem yang bertugas mengubah citra hasil akuisisi menjadi informasi hasil inspeksi.

**a. Akuisisi Citra Menggunakan OpenCV**

Citra dari webcam diterima oleh aplikasi menggunakan OpenCV. OpenCV digunakan untuk menangkap *frame* kamera secara *real-time* dan menyediakan citra sebagai masukan bagi model YOLOv8.

**b. Inferensi Menggunakan YOLOv8**

Citra yang telah diterima kemudian diberikan kepada model **YOLOv8n** yang telah dilatih menggunakan dataset penelitian.

Model melakukan proses deteksi untuk mencari keberadaan tiga kelas cacat, yaitu:

- *Black spot*
- *Dent*
- *Scratch*

Apabila terdapat objek yang memenuhi nilai *confidence threshold* yang ditentukan, model menghasilkan informasi berupa koordinat *bounding box*, kelas cacat, dan nilai *confidence*.

**c. Pengolahan Hasil Deteksi**

Hasil prediksi YOLOv8 kemudian diproses oleh aplikasi. Sistem menampilkan lokasi objek cacat pada citra menggunakan *bounding box* dan memberikan label sesuai dengan kelas hasil deteksi.

Selanjutnya sistem melakukan penentuan kondisi produk berdasarkan hasil deteksi. Apabila tidak ditemukan ketiga jenis cacat tersebut, produk dinyatakan **OK**. Sebaliknya, apabila ditemukan minimal satu objek cacat berupa *black spot*, *dent*, atau *scratch*, produk dinyatakan **NG**.

Dengan mekanisme tersebut, klasifikasi OK dan NG tidak dilakukan sebagai kelas tambahan pada model YOLOv8, tetapi ditentukan berdasarkan **ada atau tidaknya objek cacat hasil deteksi**.

Ini penting banget untuk kamu tuliskan karena menjelaskan logika sistem secara jelas.

### 3.4.3 Output

Tahap *output* merupakan tahap penyajian hasil inspeksi kepada operator melalui aplikasi berbasis *web*.

Hasil proses deteksi dari model YOLOv8 diteruskan ke aplikasi Flask untuk ditampilkan pada *web browser*. Informasi yang ditampilkan meliputi:

1. **Citra hasil inspeksi**, yaitu citra produk yang sedang diperiksa.
2. **Bounding box**, yaitu kotak yang menunjukkan lokasi objek cacat.
3. **Label kelas**, yaitu jenis cacat yang terdeteksi.
4. **Confidence**, yaitu tingkat keyakinan model terhadap hasil prediksi.
5. **Status inspeksi**, yaitu **OK** atau **NG**.
6. **FPS**, apabila digunakan pada tampilan sistem untuk menunjukkan kecepatan pemrosesan.

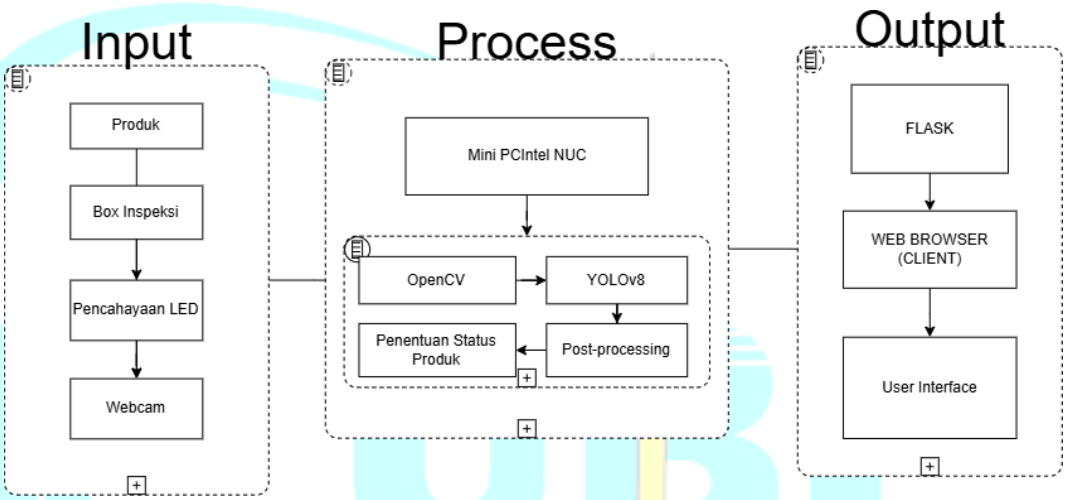
Status **OK** diberikan apabila sistem tidak menemukan objek cacat *black spot*, *dent*, maupun *scratch*. Sementara itu, status **NG** diberikan apabila sistem mendeteksi minimal satu dari ketiga jenis cacat tersebut.

Hasil tersebut ditampilkan secara *real-time* melalui *web browser* sehingga operator QC dapat menggunakan sistem sebagai **alat bantu inspeksi**, sedangkan keputusan dan proses pemeriksaan kualitas tetap berada pada operator.

### 3.4.4 Alur Integrasi Perangkat Keras dan Perangkat Lunak

Secara keseluruhan, integrasi perangkat keras dan perangkat lunak pada sistem inspeksi cacat dirancang agar setiap komponen dapat menjalankan fungsi sesuai dengan tahapan proses inspeksi. Proses diawali dari **kamera Logitech C920** yang berfungsi sebagai perangkat input untuk menangkap citra produk *stamping* baja secara *real-time*. Citra yang diperoleh kemudian diteruskan ke **Mini PC Intel NUC** sebagai pusat pemrosesan sistem.

Pada Mini PC, citra diproses menggunakan **OpenCV** untuk mengelola input kamera dan kemudian diteruskan ke model **YOLOv8** untuk melakukan proses deteksi terhadap objek cacat. Hasil inferensi YOLOv8 berupa lokasi *bounding box*, kelas cacat, dan nilai *confidence* selanjutnya diproses oleh aplikasi **Flask**. Aplikasi tersebut berfungsi mengintegrasikan hasil deteksi dengan antarmuka berbasis web sehingga hasil inspeksi dapat ditampilkan kepada operator melalui **web browser**.




Gambar 3. 11 Arsitektur Sistem

Dengan arsitektur tersebut, kamera berfungsi sebagai perangkat **input**, Mini PC berfungsi sebagai perangkat **pemrosesan**, sedangkan Flask dan *web browser* berfungsi sebagai bagian **penyajian output**. Keseluruhan komponen bekerja secara terintegrasi mulai dari proses pengambilan citra, pengolahan dan deteksi cacat, hingga penyajian hasil inspeksi kepada operator. Integrasi ini memungkinkan proses pemeriksaan dilakukan secara *real-time* dan memberikan informasi hasil deteksi berupa **OK** atau **NG** sebagai pendukung proses inspeksi kualitas produk *stamping* baja.