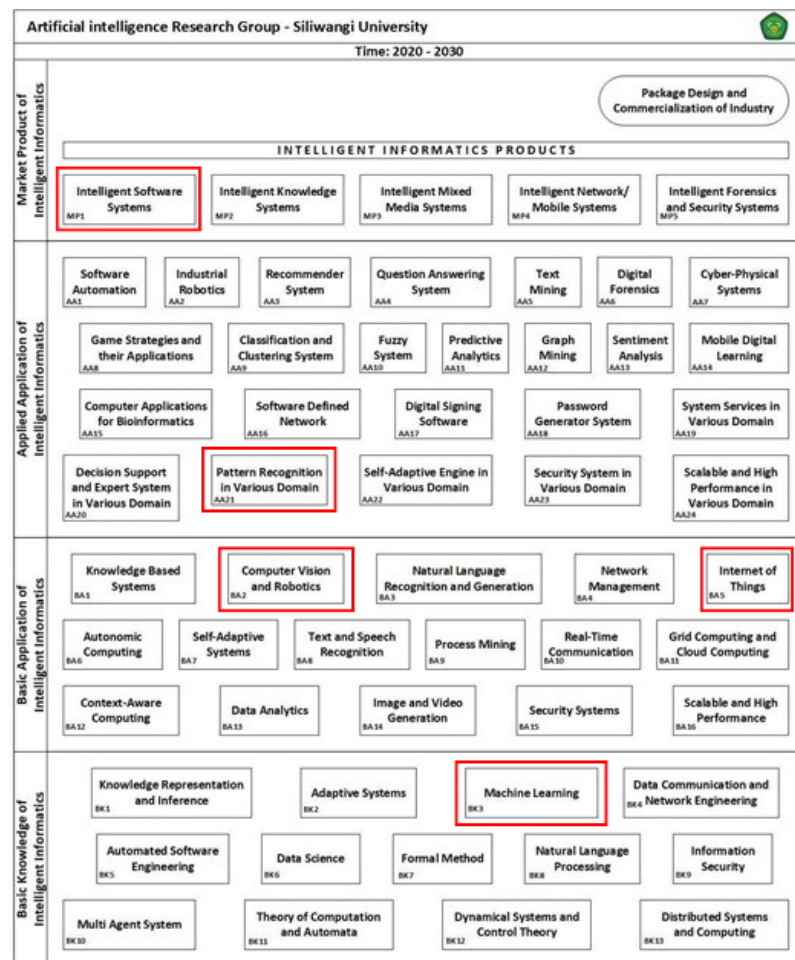


BAB III

METODOLOGI

3.1 Peta Jalan (*Road Map*) Penelitian

Mengacu dari *roadmap* penelitian untuk ranah grup penelitian Informatika dan Sistem Intelijen di Universitas Siliwangi seperti yang ada pada Gambar 3.1.1, penelitian ini bersinggungan dengan beberapa topik.



Gambar 3.1.1 Roadmap Penelitian

Secara spesifik, penelitian ini berfokus pada pengembangan model YOLOv11 dengan kuantisasi dan *hyperparameter tuning* untuk mendeteksi objek ritel dari berbagai sudut dan latar belakang, sehingga berkaitan erat dengan *Machine Learning* (BK3), *Computer Vision and Robotics* (BA2) dan *Pattern Recognition in Various Domain* (AA21).

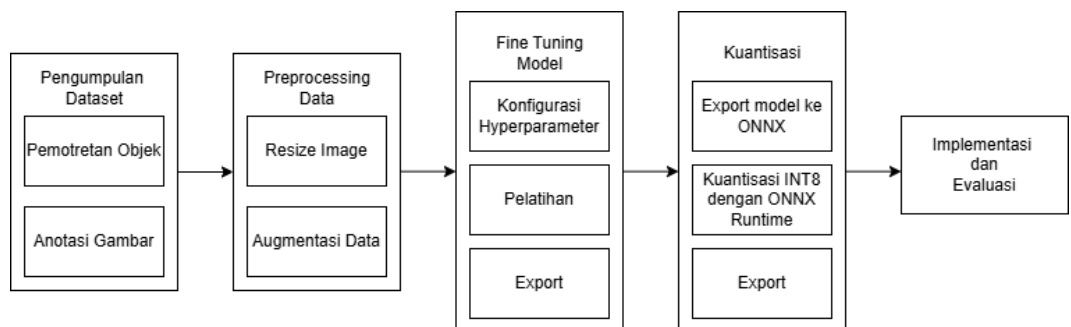
Kategori *Machine Learning* mencakup optimasi model melalui kuantisasi dan *hyperparameter tuning*, yang bertujuan untuk meningkatkan akurasi serta mempercepat waktu inferensi. Sementara *Computer Vision and Robotics* menjadi ranah utama karena penelitian ini berfokus pada pengolahan citra digital untuk mendeteksi objek ritel. Selain itu, proses pelatihan model YOLOv11 yang melibatkan pembelajaran pola gambar dari dataset menjadikan penelitian ini relevan dalam ranah *Pattern Recognition in Various Domain*.

Penelitian ini memiliki potensi untuk dikembangkan lebih lanjut ke dalam ranah *Internet of Things* (BA5) dan *Intelligent Software Systems* (MP1). Model YOLOv11 yang telah dioptimalkan berpeluang untuk dapat diintegrasikan ke dalam sistem IoT, memungkinkan deteksi produk secara langsung melalui *edge device* seperti kamera kasir atau *smart shelves* di ritel modern. Dengan dukungan sistem informasi kedepannya, data hasil deteksi dapat dikelola untuk pengambilan keputusan otomatis, seperti pembaruan stok produk secara *real-time*. Pada tahap selanjutnya, penelitian ini dapat dikembangkan menjadi produk jadi seperti mesin *checkout* otomatis berbasis AI yang tidak hanya meningkatkan efisiensi transaksi tetapi juga

mengurangi ketergantungan terhadap tenaga kasir, sejalan dengan tren otomatisasi dalam industri ritel.

3.2 Tahapan Penelitian

Penelitian ini dirancang dengan pendekatan eksperimental yang terdiri dari beberapa tahap dengan rincian seperti pada Gambar 3.2.1



Gambar 3.2.1 Tahap Penelitian

3.2.1. Pengumpulan Dataset

Dataset diambil pada waktu siang hari untuk memastikan kondisi pencahayaan yang memadai sehingga objek terlihat jelas. Namun pencahayaan tidak diatur secara spesifik karena variasi pencahayaan ditangani pada tahap preprocessing.

Pemotretan objek diambil dari berbagai sudut dengan variasi posisi objek seperti tegak, terbalik, atau menyamping. Pengambilan gambar dilakukan dengan dua skema yaitu satu gambar satu objek sebagai data untuk diolah dan satu gambar banyak objek untuk data uji pada proses inferensi model.

Mempertimbangkan biaya komputasi saat melakukan *fine-tuning*, dikarenakan semakin banyak data yang dipakai maka akan semakin mahal biaya komputasinya (Hamir dkk., 2024), sehingga dataset yang akan digunakan hanya berjumlah 10 produk dengan total keseluruhan 1.000 gambar seperti referensi penelitian sebelumnya agar proses *fine-tuning* tidak terlalu membebani perangkat.

Pemilihan metode menggunakan dataset custom dilakukan agar memberikan fleksibilitas dan kontrol penuh terhadap data terutama untuk memfasilitasi proses modifikasi seperti penambahan atau pengurangan gambar jika diperlukan, hal ini dilakukan untuk memastikan kualitas data yang sesuai dengan kebutuhan analisis. Selain itu dataset yang tersedia secara publik banyak memiliki kondisi visual yang kurang variatif seperti dataset yang keseluruhan latar belakangnya putih sehingga kurang merepresentasikan kondisi nyata di lapangan serta dapat menimbulkan *overfitting* (Tang dkk., 2021).

3.2.2. *Preprocessing Data*

Setelah pengambilan dataset dilakukan, seluruh gambar diunggah untuk diproses di situs Roboflow. Roboflow adalah *tools* berbasis web yang difokuskan untuk pelabelan, *preprocessing*, dan melatih model *computer vision*. Penelitian ini hanya menggunakan fitur pelabelan *preprocessing* dan *data augmentation* pada situs ini.

Dataset sebelumnya yang masih berupa gambar mentah, diberikan *bounding box* dalam proses pelabelannya, yang dilakukan secara manual satu persatu serta dibagi menjadi 60% (pelatihan) : 20% (validasi) : 20% (pengujian).

Selain itu *data augmentation* juga ditambahkan di dalam *pipeline* Roboflownya untuk mengantisipasi agar model tidak *overfitting* (Hamir dkk., 2024). hasil augmentasi disimpan langsung di dalam penyimpanan agar tidak membebani perangkat dalam proses pelatihan. Setelah itu data diekspor dengan opsi format YOLOv11 yang sudah disediakan pada Roboflow dengan format zip berisi data gambar, data koordinat *bounding box* dan satu data yang berisi informasi dataset dengan format yaml.

3.2.3. *Fine-Tuning* Model YOLOv11

Pada penelitian ini menggunakan *pre-trained* model YOLOv11 dari Ultralytics yang sudah dilatih sebelumnya, disini pendekatan eksperimental dilakukan, peneliti akan bereksperimen dengan melakukan berbagai macam *hyperparameter tuning* berdasarkan penelitian sebelumnya untuk mendapatkan model yang terbaik. *Hyperparameter tuning* acuan yang digunakan mengambil referensi dari (Hamir dkk., 2024) dan (Kusumawardhani dkk., 2024) sebagai model 1, referensi dari (Chhabra & Goyal, 2024) sebagai model 2, dan kombinasi dari keduanya sebagai model 3.

Fine-tuning akan dilakukan menggunakan Google Colab karena keterbatasan spesifikasi perangkat lokal. Versi YOLOv11 yang dipakai yaitu versi S (*Small*), dikarenakan ukurannya lebih kecil dan lebih ringan untuk dijalankan dikarenakan parameternya lebih sedikit. Bantuan GPU dari Google Colab juga digunakan untuk mempercepat proses *fine-tuning*, GPU yang digunakan ialah NVIDIA Tesla T4.

Jika selama pelatihan ditemukan indikasi *overfitting* atau nilai *loss* yang tidak kunjung turun secara signifikan, maka jumlah epochs dapat disesuaikan atau dihentikan lebih awal dengan penerapan teknik *early stopping* untuk mencegah *overfitting* (Bai dkk., 2021).

Pada tahap ini ketiga model diatas dievaluasi terlebih dahulu manakah yang memiliki performa terbaik sebelum dilanjutkan ke tahap kuantisasi.

3.2.4. Kuantisasi

Jika proses pelatihan telah selesai dilakukan, tahap selanjutnya ialah mengekspor model pytorch ke dalam format ONNX untuk dikuantisasi, model siap diproses dengan *library* dari ONNX Runtime untuk perubahan bit secara dinamis dari default model YOLOv11 yang memiliki ukuran *bit* sebesar Float 16 (FP16) diubah menjadi Integer 8 (INT8). Kuantisasi dinamis dipilih untuk menjaga keseimbangan antara efisiensi komputasi dan performa model. Dengan pendekatan ini, kuantisasi yang dilakukan bisa mengurangi

ukuran model dan mengefisienkan inferensi serta potensi penurunan akurasi juga lebih minimal dalam satu waktu.

3.2.5. Implementasi dan Evaluasi

Proses implementasi dilakukan di 3 platform yaitu web, laptop, dan Google Colab. Proses ini dibutuhkan untuk menguji performa model dan memperoleh data hasil sebagai pertimbangan evaluasi akhir. Platform web dipilih karena sifatnya yang universal dan mudah diakses dari berbagai perangkat, dengan mengimplementasikannya di web, model ini dapat diuji dan digunakan secara lebih luas, mendekati kondisi dunia nyata. Pengujian di perangkat yang berbeda dilakukan untuk memahami bagaimana performa model di lingkungan dengan kondisi eksekusi yang bervariasi.

Pada proses evaluasi, dilakukan penilaian kinerja model berdasarkan 4 metrik utama yaitu :

1. *Mean Average Precision (mAP)*

mAP digunakan untuk mengevaluasi akurasi deteksi objek dari model, dengan memperhatikan hubungan antara *precision* dan *recall* untuk berbagai nilai ambang batas.

2. Waktu Inferensi

Waktu inferensi dihitung untuk mengetahui efisiensi performa model dalam mendeteksi objek terhadap data uji, baik sebelum maupun setelah proses kuantisasi.

3. Ukuran Model

Kuantisasi menyebabkan ukuran model mengecil, namun diperlukan perbandingan juga untuk mengukur seberapa besar model ukuran model berkurang.

4. *Confusion Matrix*

Confusion Matrix digunakan untuk mengevaluasi jumlah prediksi yang benar dan salah serta performa model dalam mendeteksi masing-masing kelas objek.