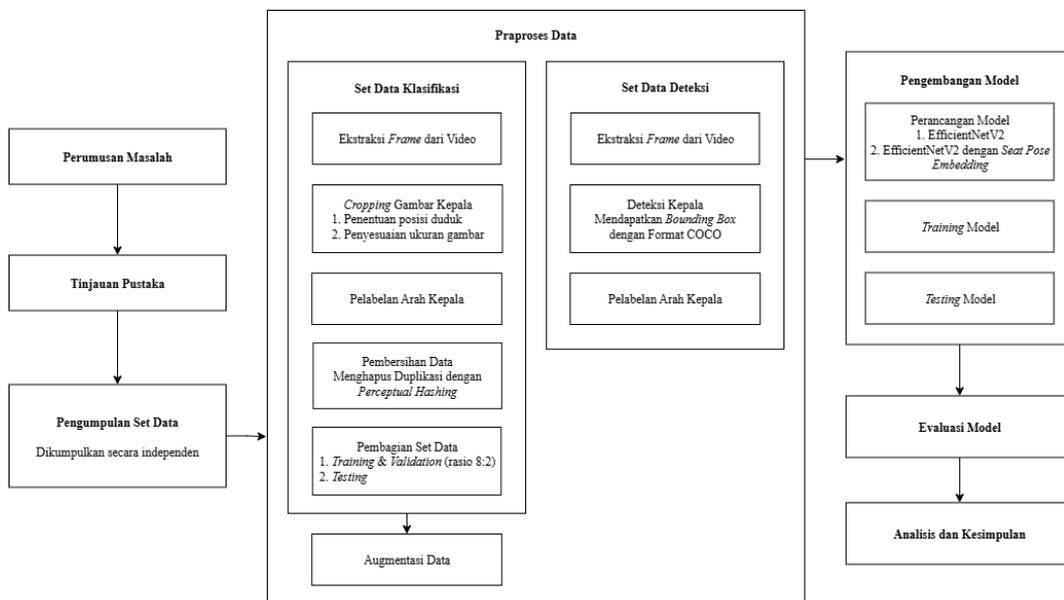


BAB III METODE PENELITIAN

3.1 Desain Penelitian

Desain penelitian dirancang untuk membuat perencanaan penelitian yang matang untuk mengatur pelaksanaan suatu penelitian agar penelitian terencana dan berjalan dengan baik secara sistematis. Setiap tahap memiliki peran penting dalam memastikan keberhasilan penelitian karena tahapan-tahapan tersebut akan menjelaskan bagaimana penelitian tersebut mengangkat suatu permasalahan dan bagaimana penulis mencapai kesimpulan dan solusi. Struktur desain penelitian yang digunakan mengacu pada metode (Hambali et al., 2023) dan dapat dilihat pada Gambar 3.1. Desain penelitian terdiri dari perumusan masalah, tinjauan pustaka, praproses data, pengembangan model, evaluasi model, serta analisis dan simpulan. Penjelasan detail mengenai setiap tahapan dijelaskan pada subbab berikut.



Gambar 3.1 Desain Penelitian

3.1.1 Perumusan Masalah

Pada tahap awal ini, penelitian dimulai dengan mengidentifikasi dan merumuskan suatu permasalahan sebagai objek penelitian yang akan dilakukan. Pada proses ini, penulis meneliti latar belakang masalah secara mendalam serta mencari solusi yang telah disebutkan dalam penelitian sebelumnya. Permasalahan

yang diangkat pada penelitian ini adalah bagaimana *input* posisi duduk siswa yang di diubah ke dalam bentuk *embedding* yang lalu digabungkan dengan citra kepala sebagai *input* untuk model klasifikasi arah pose kepala dapat mengurangi ambiguitas spasial sehingga meningkatkan akurasi model dalam melakukan klasifikasi arah pose kepala siswa di ruang kelas. Dengan melakukan perumusan masalah secara tepat, tujuan dan manfaat penelitian dapat diuraikan secara jelas. Tujuan pada penelitian ini adalah mencapai keberhasilan dalam menemukan solusi yang tepat untuk masalah yang diidentifikasi. Sementara itu, manfaat yang diharapkan penulis adalah bagaimana penelitian ini dapat memberikan kontribusi kepada dunia pendidikan.

3.1.2 Tinjauan Pustaka

Pada tahap ini, peneliti melakukan tinjauan terhadap literatur yang relevan dengan masalah penelitian yang telah dirumuskan dari berbagai sumber seperti jurnal, buku, dan artikel. Hal ini membantu peneliti dan pembaca untuk memahami konteks penelitian dengan memahami atau mengevaluasi bagaimana penelitian-penelitian terdahulu menyelesaikan masalah pada masalah penelitian yang dirumuskan, memperoleh wawasan mendalam tentang kemajuan pengetahuan dalam bidang yang diteliti, serta mengidentifikasi kesenjangan pengetahuan dan menegaskan relevansi dan urgensi pada penelitian ini. Pada bagian ini, teori-teori relevan yang diidentifikasi dapat memperkuat landasan yang kokoh dalam menyusun kerangka metodologi yang tepat untuk mengarahkan proses penelitian yang lebih teratur dan sistematis. Literatur yang dikumpulkan dan dipelajari pada penelitian ini mencakup teori, teknik, atau konsep dari *computer vision*, *head pose estimation*, *deep learning*, klasifikasi, serta topik lainnya.

3.1.3 Pengumpulan Set Data

Pengumpulan set data dalam penelitian ini dilakukan melalui pemanfaatan rekaman video dari kamera CCTV yang dipasang di ruang kelas saat proses pembelajaran berlangsung. Lokasi pengambilan data dilakukan di lingkungan kampus Fakultas Pendidikan Matematika dan Ilmu Pengetahuan Alam (FPMIPA), Universitas Pendidikan Indonesia. Video yang diperoleh selanjutnya digunakan sebagai sumber utama dalam penyusunan dua jenis set data, yaitu set data

klasifikasi dan set data deteksi, yang masing-masing digunakan untuk pelatihan dan pengujian sistem.

Secara umum, penyusunan set data klasifikasi dilakukan dengan mengekstraksi informasi visual dari video untuk menghasilkan potongan gambar kepala siswa, yang dilabeli secara manual berdasarkan arah pose kepala. Informasi tambahan seperti posisi duduk siswa di kelas juga disimpan. Di sisi lain, set data deteksi disusun dengan anotasi dalam format COCO menggunakan model deteksi kepala dan antarmuka pelabelan grafis Roboflow. Seluruh proses pengolahan data lebih lanjut dijelaskan secara rinci pada Subbab 3.1.4 mengenai praproses data.

3.1.4 Praproses Data

3.1.4.1 Set Data Klasifikasi

Praproses data pada penelitian ini dilakukan secara bertahap dengan tujuan menyiapkan set data yang representatif untuk keperluan *training* dan *validation* model klasifikasi arah pose kepala. Praproses dimulai dengan ekstraksi *frame* dari video rekaman CCTV yang diambil selama proses pembelajaran berlangsung di ruang kelas. *Frame* diekstraksi secara berkala dengan kecepatan *2 frame per second* (fps) untuk mengurangi redundansi visual dan mempercepat proses anotasi dan pengolahan data selanjutnya.

Setelah proses ekstraksi *frame*, dilakukan deteksi kepala secara otomatis menggunakan model deteksi kepala. Model ini digunakan untuk menghasilkan *bounding box* pada area kepala siswa dalam setiap *frame*. Informasi dari *bounding box* ini kemudian dimanfaatkan dalam dua hal. Pertama, *bounding box* digunakan untuk memotong gambar kepala (*cropping*) dari setiap *frame* yang terdeteksi. Kedua, koordinat *bounding box* digunakan untuk menentukan posisi duduk siswa di dalam kelas. Penentuan posisi ini dilakukan berdasarkan letak pusat *bounding box* terhadap lebar gambar, yang kemudian diklasifikasikan ke dalam tiga kategori: kiri (sepertiga kiri gambar), tengah (sepertiga tengah), dan kanan (sepertiga kanan).

Hasil potongan gambar kepala yang telah diperoleh kemudian disesuaikan ukurannya dan juga penambahan margin guna memastikan seluruh bagian kepala tercakup secara utuh. Selanjutnya, dilakukan proses pelabelan arah kepala secara manual. Kelas arah pose kepala yang digunakan terdiri dari lima kategori, yaitu

atas, bawah, depan, kanan, dan kiri. Pelabelan arah pose kepala dilakukan secara relatif terhadap konfigurasi ruang kelas, bukan berdasarkan arah absolut dalam gambar. Arah pandang seperti depan, kiri, kanan, atas, dan bawah ditentukan secara relatif terhadap posisi papan tulis dan meja guru sebagai titik acuan utama, bukan berdasarkan orientasi gambar. Hal ini memastikan bahwa label yang diberikan mencerminkan konteks aktual interaksi siswa dalam kegiatan pembelajaran di ruang kelas.

Setelah pelabelan selesai, dilakukan pembersihan data untuk mengurangi duplikasi gambar jika terdapat gambar yang terlalu mirip. Proses ini menggunakan teknik *perceptual hashing* dengan *threshold* kemiripan sebesar 5. Gambar-gambar yang memiliki nilai *hash* yang terlalu dekat akan dianggap sebagai duplikat dan dihapus untuk menjaga keragaman visual dalam set data. Langkah berikutnya adalah pembagian data. Data dari dua sesi kelas yang berbeda digunakan secara terpisah untuk memastikan pemisahan data pelatihan dan pengujian yang independen. Gambar dari satu kelas digunakan sebagai *data training* dan *validation* dengan rasio pembagian 8:2, sedangkan gambar dari kelas lainnya digunakan secara eksklusif sebagai *data testing*. Pembagian ini dilakukan untuk mengevaluasi performa model pada data yang belum pernah dilihat sebelumnya.

3.1.4.2 Augmentasi Data

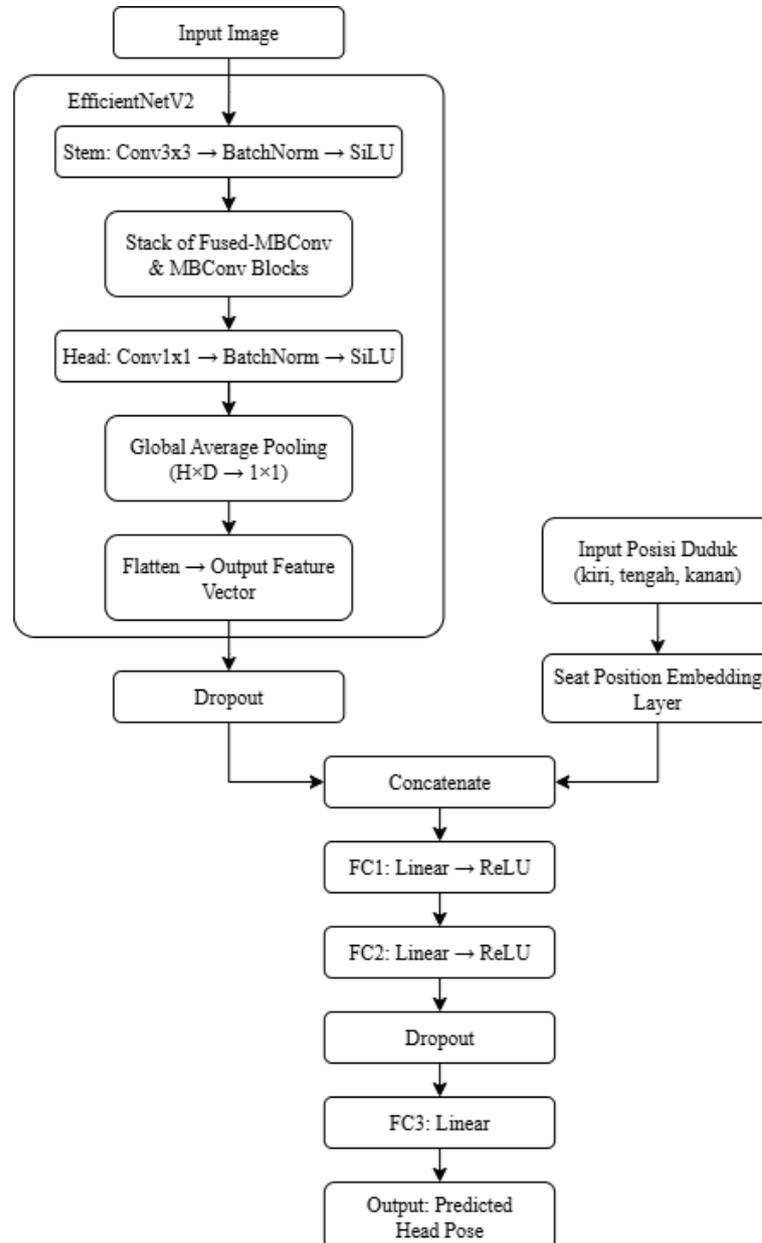
Untuk meningkatkan kapasitas generalisasi model terhadap variasi kondisi nyata, dilakukan proses augmentasi data pada *data training*. Seluruh gambar kepala yang telah dipotong dan disesuaikan ukurannya kemudian diproses melalui berbagai teknik augmentasi visual. Setelah proses augmentasi selesai, seluruh gambar diubah menjadi format *tensor* dan dinormalisasi menggunakan nilai rata-rata dan standar deviasi dari set data ImageNet. Proses augmentasi ini hanya diterapkan pada *data training* saja. Sementara itu, *data validation* dan *data training* hanya mengalami proses *resize* dan normalisasi tanpa augmentasi tambahan, agar evaluasi performa model dilakukan secara objektif dan tidak bias terhadap data hasil modifikasi.

3.1.4.3 Set Data Deteksi

Set data deteksi disusun dengan prosedur yang serupa dengan set data klasifikasi, yaitu menggunakan model deteksi kepala untuk menghasilkan anotasi awal dalam format JSON mengikuti skema COCO dengan satu kategori objek, yaitu *head*. Proses pelabelan arah pose kepala pada hasil deteksi dilakukan melalui antarmuka grafis pada *platform* Roboflow, dengan sistem pelabelan yang sama dengan set data klasifikasi. Setelah pelabelan awal, dilakukan proses koreksi manual terhadap hasil deteksi dengan tujuan memperbaiki atau menambahkan *bounding box* yang tidak akurat atau terlewat. Hasil akhir dari pelabelan ini juga disimpan dalam format COCO, dan digunakan untuk menguji model deteksi yang terintegrasi dengan model klasifikasi.

3.1.5 Pengembangan Model

3.1.5.1 Arsitektur EfficientNetV2



Gambar 3.2 Arsitektur EfficientNetV2 dengan *Seat Position Embedding* (SPE)

Arsitektur model yang diusulkan dalam penelitian ini, sebagaimana ditunjukkan pada Gambar 3.2, merupakan modifikasi dari arsitektur EfficientNetV2 dengan tujuan untuk menggabungkan informasi tambahan berupa posisi duduk sebagai konteks dalam klasifikasi arah pose kepala. EfficientNetV2 merupakan arsitektur CNN modern yang dikenal karena kemampuannya

Ananda Myzza Marhelio, 2025

DETEKSI FOKUS ATENSI VISUAL PESERTA DIDIK DI RUANG KELAS BERDASARKAN POSE KEPALA MENGGUNAKAN EFFICIENTNETV2 DENGAN SEAT POSITION EMBEDDING

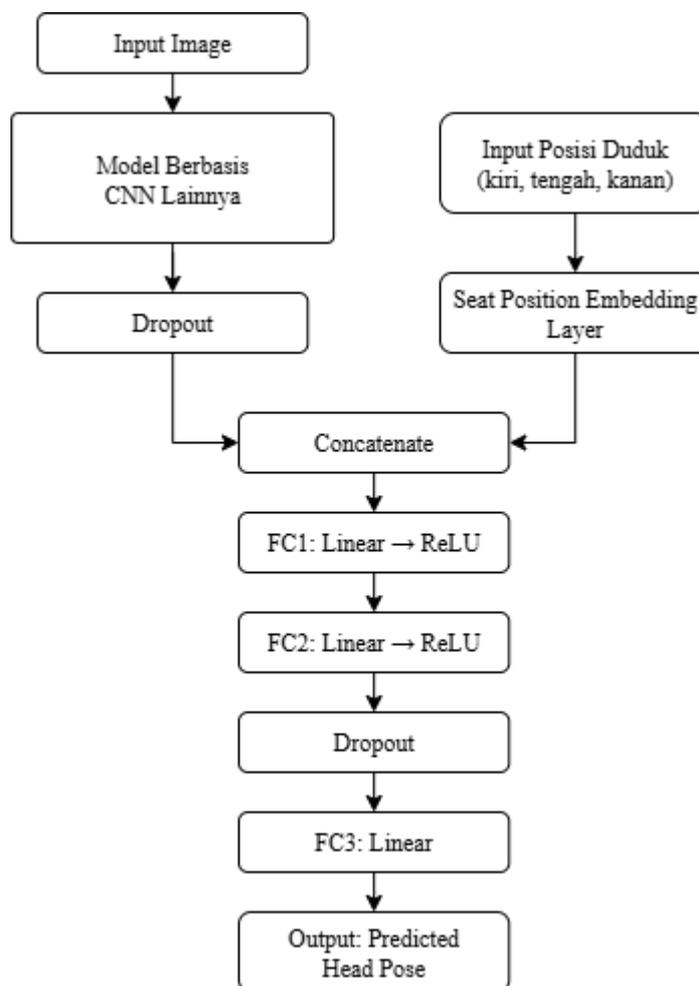
Universitas Pendidikan Indonesia | repository.upi.edu | perpustakaan.upi.edu

mengoptimalkan akurasi dan efisiensi parameter melalui pendekatan *compound scaling*, yaitu strategi yang secara bersamaan menyeimbangkan kedalaman, lebar, dan resolusi jaringan (Tan & Le, 2021). Dalam implementasi ini, bagian *classifier* dari EfficientNetV2 dihilangkan, sehingga memungkinkan integrasi fitur tambahan berupa *embedding* dari posisi duduk.

Pada arsitektur yang dimodifikasi ini, citra kepala yang menjadi *input* awal akan diproses menggunakan *backbone* EfficientNetV2 yang telah menggunakan *pretrained* pada set data ImageNet. Model ini berfungsi sebagai ekstraktor fitur visual, dan menghasilkan vektor fitur berdimensi 1280. Selanjutnya, vektor ini digabungkan (*concatenated*) dengan *embedding* posisi duduk siswa atau *Seat Position Embedding* (SPE), yang direpresentasikan dalam bentuk vektor berdimensi 8. Kategori posisi duduk (kiri, tengah, kanan) ditentukan berdasarkan letak horizontal pusat *bounding box* hasil deteksi kepala terhadap lebar gambar, yang memungkinkan model memahami konteks spasial dari arah pose kepala siswa. Gabungan fitur visual dan *embedding* posisi ini membentuk vektor berdimensi 1288, yang kemudian diproses lebih lanjut melalui beberapa lapisan *fully connected*. Penambahan lapisan-lapisan baru ini, setelah modifikasi pada bagian *classifier* asli, secara langsung memengaruhi jumlah total parameter pada arsitektur akhir yang digunakan dalam penelitian. Lapisan-lapisan ini dilengkapi fungsi aktivasi ReLU serta teknik regularisasi *dropout* guna mencegah *overfitting*. *Output* akhir dari model adalah prediksi arah kepala siswa ke dalam lima kelas: atas, bawah, depan, kiri, dan kanan.

Integrasi SPE dalam arsitektur model ini bertujuan untuk mengatasi ambiguitas spasial yang terjadi di ruang kelas, di mana arah pose kepala yang sama secara visual dapat memiliki makna atensi yang berbeda tergantung pada lokasi fisik siswa terhadap kamera. Dengan mengubah informasi posisi duduk menjadi *embedding* yang dapat dipelajari oleh model, sistem memperoleh konteks tambahan yang membantu dalam mengasosiasikan pola visual dengan informasi spasial yang relevan. Pendekatan ini dirancang agar tetap ringan dan terinterpretasi dengan baik, serta memberikan peningkatan akurasi dalam klasifikasi arah pose kepala tanpa menambah beban komputasi secara signifikan.

3.1.5.2 Arsitektur CNN Lainnya



Gambar 3.3 Arsitektur Berbasis CNN dengan *Seat Position Embedding* (SPE)

Sebagai bagian dari proses evaluasi komparatif, penelitian ini juga menerapkan beberapa arsitektur CNN lain untuk dibandingkan dengan model utama EfficientNetV2, baik dalam skenario tanpa maupun dengan integrasi SPE. Arsitektur yang digunakan meliputi ConvNeXt, MobileNetV3, dan ResNet50, yang seluruhnya merupakan model CNN populer dengan karakteristik efisiensi dan performa yang telah terbukti pada berbagai tugas klasifikasi citra. Setiap model digunakan sebagai *feature extractor* dengan struktur arsitektur yang mengikuti skema yang sama seperti pada EfficientNetV2, yaitu bagian *classifier* asli dihapus dan digantikan dengan blok klasifikasi kustom.

Pada masing-masing model, *output* akhir dari *backbone* CNN disesuaikan dimensinya dengan karakteristik model terkait, kemudian digabungkan (*concatenated*) dengan SPE. Vektor gabungan ini selanjutnya diproses oleh beberapa lapisan *fully connected* dengan aktivasi ReLU dan *dropout* sebelum menghasilkan prediksi arah pose kepala ke dalam lima kelas. Sama seperti pada EfficientNetv2, implementasi blok klasifikasi kustom yang seragam ini secara langsung memodifikasi jumlah total parameter akhir untuk setiap arsitektur berbasis CNN yang digunakan. Pendekatan ini memastikan bahwa perbandingan antar model dilakukan secara adil, di mana perbedaan performa model lebih merefleksikan kemampuan ekstraksi fitur masing-masing arsitektur, baik dalam kondisi hanya menggunakan informasi visual maupun ketika ditambahkan konteks posisi duduk.

3.1.5.3 Eksperimen Konfigurasi Model

Untuk mengevaluasi pengaruh integrasi informasi kontekstual berupa posisi duduk terhadap performa klasifikasi arah pose kepala, dilakukan eksperimen dengan dua konfigurasi *input* pada masing-masing arsitektur CNN yang digunakan. Konfigurasi pertama menggunakan *input* berupa citra kepala hasil *cropping* dari deteksi, sedangkan konfigurasi kedua menambahkan informasi posisi duduk siswa (kiri, tengah, kanan) yang direpresentasikan dalam bentuk SPE sebagai *input* tambahan ke dalam model klasifikasi.

Keempat arsitektur CNN yang diimplementasikan dilatih pada kedua konfigurasi *input* tersebut dengan parameter pelatihan yang seragam untuk memastikan validitas perbandingan. Seluruh model dilatih selama 100 *epoch* dengan ukuran *batch* sebesar 16 dan *learning rate* awal sebesar 0,0001. Untuk menghindari stagnasi dalam proses pembelajaran, digunakan mekanisme *learning rate scheduler* yang secara otomatis menurunkan nilai *learning rate* sebesar 0,5 apabila *validation loss* tidak menunjukkan penurunan selama tiga *epoch* berturut-turut. Selain itu, diterapkan *early stopping* dengan *patience* selama 10 *epoch* untuk menghentikan pelatihan lebih awal jika *validation loss* tidak mengalami penurunan signifikan dalam periode tersebut, sebagai langkah pencegahan terhadap *overfitting*. Mengingat adanya ketidakseimbangan distribusi data antar kelas arah

pose kepala seperti yang ditunjukkan pada Tabel x, pendekatan *class weighting* diterapkan selama pelatihan model guna meminimalkan bias klasifikasi dan meningkatkan generalisasi model terhadap kelas minoritas

3.1.6 Evaluasi Model

Evaluasi performa model dilakukan berdasarkan jenis tugas yang dijalankan, yakni klasifikasi arah pose kepala dan deteksi kepala. Untuk model klasifikasi, metrik yang digunakan meliputi akurasi, *precision*, *recall*, dan *f1-score* untuk masing-masing kelas arah pose kepala. Selain itu, *macro f1-score* digunakan sebagai metrik utama untuk mengevaluasi performa keseluruhan model terhadap ketidakseimbangan antar kelas (Sokolova & Lapalme, 2009). Visualisasi *confusion matrix* juga digunakan untuk memberikan gambaran tentang distribusi prediksi dan kesalahan antar kelas.

Sementara itu, untuk model deteksi kepala yang sekaligus mengklasifikasikan arah pose kepala ke dalam lima kelas, evaluasi dilakukan menggunakan metrik *mean Average Precision* (mAP) pada ambang batas *Intersection over Union* (IoU) sebesar 0.5, yang dikenal sebagai AP@0.5. Dalam konteks ini, model tidak hanya dituntut untuk mendeteksi keberadaan objek (kepala), tetapi juga mengklasifikasikan arah pose kepala secara akurat sebagai bagian dari tugas deteksi multi-kelas. Oleh karena itu, mAP dihitung sebagai rata-rata dari nilai *Average Precision* (AP) untuk setiap kelas arah kepala. Evaluasi ini mengikuti standar yang digunakan dalam kompetisi *Common Objects in Context* (COCO) dan PASCAL VOC, di mana mAP telah menjadi metrik utama untuk mengukur performa sistem deteksi objek multi-kelas (Everingham et al., 2010; Lin et al., 2014). Penilaian dilakukan berdasarkan anotasi berformat COCO yang telah dikoreksi secara manual menggunakan platform pelabelan.

3.1.7 Analisis dan Kesimpulan

Analisis dalam penelitian ini dilakukan secara bertahap untuk memastikan bahwa perbandingan antar pendekatan dilakukan pada konfigurasi model yang optimal. Sebelum membandingkan dua konfigurasi utama, yaitu model klasifikasi dengan dan tanpa SPE, terlebih dahulu dilakukan serangkaian eksperimen internal guna menentukan konfigurasi terbaik untuk setiap kombinasi pendekatan dan

arsitektur yang digunakan. Hasil dari analisis ini digunakan untuk menetapkan *baseline* akurasi terbaik yang akan dijadikan dasar dalam perbandingan akhir antar pendekatan.

Setelah konfigurasi optimal diperoleh untuk setiap skenario, konfigurasi terbaik tersebut kemudian digunakan dalam tahap evaluasi akhir untuk membandingkan efektivitas pendekatan dengan dan tanpa integrasi SPE. Evaluasi dilakukan menggunakan metrik akurasi dan *f1-score* untuk model klasifikasi, lalu untuk model deteksi kepala yang juga mengklasifikasikan arah pose kepala secara langsung, evaluasi dilakukan berdasarkan metrik mAP sesuai standar COCO. Hasil evaluasi deteksi akan dianalisis untuk melihat kemampuan model dalam mendeteksi serta membedakan arah pose kepala dalam skenario multi-kelas. Seluruh hasil evaluasi ini akan digunakan untuk menyimpulkan model terbaik, baik dari sisi arsitektur CNN maupun konfigurasi *input*, serta untuk mengkaji efektivitas integrasi informasi spasial dalam meningkatkan pemahaman model terhadap arah pose kepala siswa dalam konteks kelas.

3.2 Kebutuhan Perangkat

Penelitian menggunakan perangkat keras (*hardware*) dan perangkat lunak (*software*) sebagai sarana untuk proses penelitian. Berikut merupakan daftar dan spesifikasi perangkat yang akan digunakan selama penelitian berlangsung.

1. Perangkat Keras berupa Laptop.
 - A. CPU Intel® Core™ i5-10500H @ 2.50GHz
 - B. NVIDIA GeForce GTX 1650 with Max-Q Design
 - C. RAM 24 GB DDR4
 - D. SSD 512 GB
2. Perangkat Lunak
 - A. Sistem Operasi Microsoft Windows 11 Home Single Language
 - B. Jupyter Notebook
 - C. Python
 - D. Streamlit
 - E. Visual Studio Code
 - F. Roboflow