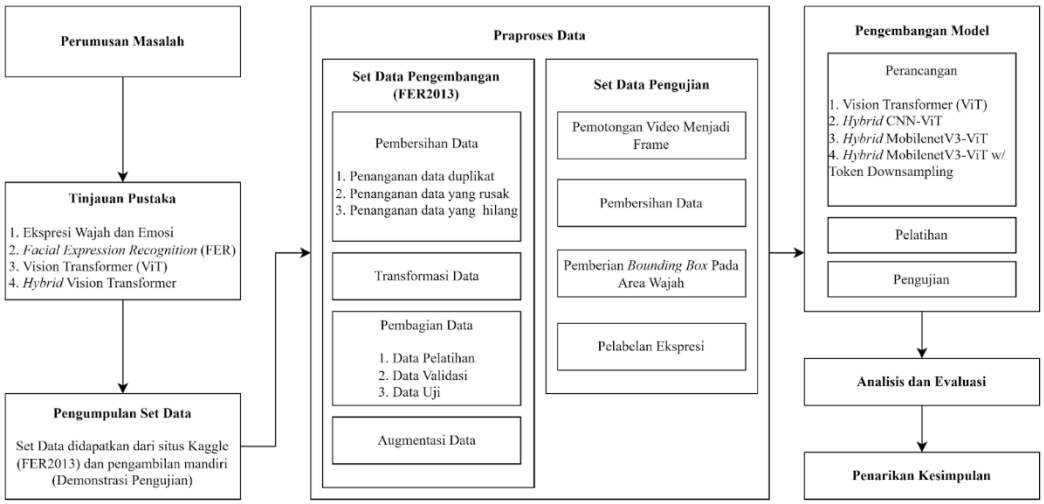
BAB III
METODE PENELITIAN

3.1 Desain Penelitian

Desain penelitian menggambarkan tahapan-tahapan kerja dari penulis untuk mencapai kesimpulan dan solusi terhadap permasalahan yang diangkat. Pada penelitian ini digunakan desain penelitian yang dikembangkan oleh (Santoso et al., 2020) yang melibatkan tujuh tahapan pengerjaan, yaitu perumusan masalah, tinjauan pustaka, pengumpulan data, praposes data, pengembangan model, analisis dan evaluasi, serta penarikan kesimpulan. Seluruh tahapan penelitian tersebut diilustrasikan pada Gambar 3.1. Tiap poin tahapan pada desain penelitian tersebut akan dijelaskan pada sub-bab selanjutnya.



Gambar 3. 1 Desain Penelitian

3.1.1 Perumusan Masalah

Tahap ini merupakan langkah awal di mana penulis merumuskan permasalahan yang akan diselesaikan dalam penelitian. Proses identifikasi masalah dilakukan dengan menelaah penelitian-penelitian terdahulu guna menemukan celah atau kesenjangan penelitian yang dapat dijadikan dasar bagi penelitian ini. Dari hasil perumusan masalah tersebut, diperoleh landasan penelitian berupa latar belakang, rumusan masalah utama, tujuan penelitian, serta metode yang akan digunakan.

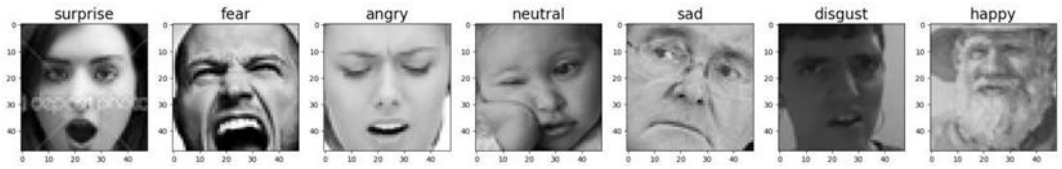
3.1.2 Tinjauan Pustaka

Tinjauan pustaka dilakukan dengan mengumpulkan bacaan-bacaan yang berasal dari jurnal, buku, atau artikel guna menelaah permasalahan yang diangkat pada penelitian ini serta kontribusi kajian-kajian terdahulu dalam menyelesaikan masalah tersebut. Dengan menggali literatur dari berbagai sumber, penulis memperoleh wawasan yang mendalam tentang perkembangan pengetahuan dalam bidang yang diteliti. Langkah ini tidak hanya memungkinkan identifikasi kesenjangan pengetahuan, tetapi juga menegaskan relevansi dan urgensi penelitian ini dalam konteks akademik dan praktis. Tinjauan pustaka juga dilakukan untuk mengidentifikasi teori-teori yang relevan terhadap penelitian ini dan memperkuat kerangka konseptual penelitian.

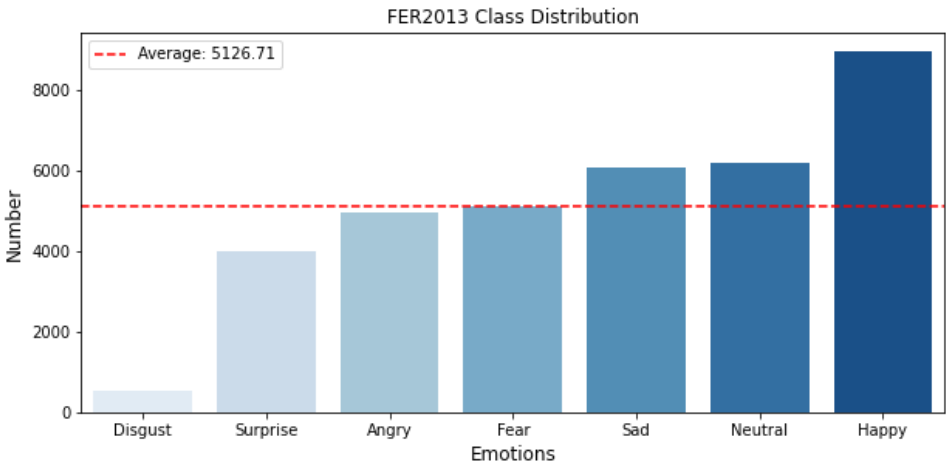
3.1.3 Pengumpulan Set Data

Dalam penelitian ini, set data FER-2013 digunakan pada tahap pengembangan awal dan pelatihan model. Set data ini merupakan set data publik yang tersedia melalui platform *Kaggle* dan pertama kali diperkenalkan oleh Pierre-Luc Carrier dan Aaron Courville dalam kompetisi "Challenges in Representation Learning" pada tahun 2013, yang diselenggarakan oleh ICML. FER-2013 dibangun dengan mengumpulkan gambar wajah dari berbagai sumber daring menggunakan *Google Image Search* API, berdasarkan kata kunci yang berhubungan dengan berbagai ekspresi wajah seperti “blissful”, “eraged”, dan lainnya (Goodfellow et al., 2015). Set data ini berisi gambar pose wajah yang dilabeli manual ke dalam tujuh kelas ekspresi, yaitu 0 : *angry* (marah), 1 : *disgust* (muak), 2 : *fear* (takut), 3 : *happy* (senang), 4 : *neutral* (netral), 5 : *sad* (sedih), dan 6 : *surprise* (terkejut) dengan ukuran masing-masing gambar sebesar 48x48 piksel pada skala abu abu (*grayscale*). Secara keseluruhan, set data FER-2013 terdiri dari 35.887 gambar yang dibagi ke dalam tiga subset, yaitu *training set* sebanyak 28.709 gambar, *public test set* sebanyak 3.589 gambar, dan *private test set* sebanyak 3.589 gambar. Pratinjau data gambar FER-2013 dapat dilihat pada Gambar 3.2. Dari gambar tersebut juga dapat dilihat bahwa terdapat berbagai variasi pada set data FER-2013, seperti variasi demografis (usia, jenis kelamin, dan etnis), posisi kepala, kualitas gambar, pencahayaan, dan variasi lainnya. Meskipun demikian, seperti yang terlihat pada

Gambar 3.3, salah satu keterbatasan utama dari FER-2013 adalah distribusi kelas yang tidak seimbang (*imbalance*). Kelas seperti "disgust" memiliki jumlah sampel yang jauh lebih sedikit dibandingkan dengan kelas lainnya seperti "happy" atau "neutral". Ketidakseimbangan ini berpotensi menyebabkan model bias terhadap kelas mayoritas dan kesulitan mengenali ekspresi yang jarang muncul.



Gambar 3. 2 Pratinjau Set Data FER-2013



Gambar 3. 3 Distribusi Data Per Label Set Data FER-2013

Selain set data untuk pengembangan model, dilakukan juga pengumpulan data untuk membangun set data pengujian demonstrasi. Pengumpulan data ini diperlukan untuk menguji performa model pada konteks nyata yaitu pada lingkungan kelas, lingkungan di mana model diimplementasikan nantinya. Selain itu, sebagian data yang dikumpulkan juga dimaksudkan untuk keperluan *fine-tuning* model, agar model dapat beradaptasi lebih baik dengan karakteristik data yang spesifik pada konteks pembelajaran di kelas.

Dilakukan pengumpulan data berupa pengambilan video berisi berbagai macam ekspresi peserta didik pada lingkungan pembelajaran di kelas. Video direkam secara mandiri dengan melibatkan mahasiswa Program Studi Ilmu Komputer Universitas Pendidikan Indonesia sebagai subjek. Seluruh subjek yang

terlibat memiliki karakteristik khas wajah orang Indonesia, seperti warna kulit yang berkisar antara sawo matang hingga kuning kecokelatan, iris mata berwarna coklat gelap, rambut berwarna hitam hingga kecokelatan dengan tekstur lurus hingga ikal bergelombang, serta bentuk hidung yang lebar namun tidak terlalu menonjol.

Pengambilan data dilakukan menggunakan kamera yang diposisikan di tiga sudut kelas, yaitu di atas papan tulis (depan tengah), sisi kiri depan, dan sisi kanan depan. Setiap subjek akan diminta untuk menampilkan ekspresi tertentu sesuai instruksi, disertai contoh ekspresi yang sesuai dengan *Facial Action Coding System* (FACS). Proses perekaman mencakup tujuh adegan, di mana setiap subjek diminta untuk memperagakan satu jenis ekspresi dari tujuh kelas ekspresi yang ada. Dalam setiap adegan, subjek diberikan waktu selama 10 detik untuk menampilkan ekspresi tersebut secara alami.

3.1.4 Praproses Set Data Pengembangan (FER-2013)

Pada tahap ini, peneliti melakukan pra-pemrosesan terhadap set data yang meliputi beberapa langkah utama, yaitu pembersihan data, pembagian data, augmentasi, dan transformasi. Setiap langkah dirancang untuk meminimalkan *noise*, meningkatkan kualitas representasi data, serta memastikan distribusi data tetap terjaga secara konsisten selama pelatihan model. Penjelasan lebih lanjut mengenai masing-masing tahapan dijabarkan sebagai berikut.

3.1.4.1 Pembersihan Data

Tahap pertama dalam pra-pemrosesan set data FER-2013 adalah pembersihan data (*data cleaning*), yang bertujuan untuk memastikan kualitas dan validitas data yang digunakan dalam pelatihan model. Pada tahap ini, peneliti melakukan identifikasi terhadap gambar-gambar yang tidak sesuai, seperti gambar yang tidak menampilkan wajah dengan jelas, buram, atau terlalu gelap hingga menyulitkan proses ekstraksi fitur wajah. Gambar-gambar tersebut kemudian dihapus agar tidak memengaruhi performa model. Selain itu, peneliti juga melakukan pemeriksaan terhadap data duplikat, yaitu gambar yang muncul lebih dari satu kali dalam set data. Data duplikat ini dihapus untuk mencegah bias dalam pelatihan model. Selanjutnya, dilakukan pengecekan terhadap data yang memiliki atribut tidak lengkap, seperti label ekspresi yang kosong (*null*). Proses pembersihan

ini penting untuk memastikan bahwa data yang masuk ke tahap selanjutnya adalah data yang berkualitas dan representatif untuk proses pelatihan model FER.

3.1.4.2 Transformasi Data

Transformasi data merupakan tahap penting dalam praproses yang bertujuan untuk menyesuaikan format data agar kompatibel dan optimal untuk digunakan oleh arsitektur model yang akan dilatih. Dalam penelitian ini, proses transformasi dilakukan melalui beberapa langkah utama. Pertama, mengubah jumlah saluran (*channel*) pada gambar. Karena set data FER-2013 disimpan dalam format *grayscale* (satu *channel*), sedangkan model *Hybrid Vision Transformer* dan MobileNetV3 yang digunakan dalam penelitian ini dirancang untuk menerima *input* dengan tiga saluran warna (RGB), maka setiap gambar *grayscale* perlu direplikasi ke dalam tiga *channel* agar dapat disesuaikan dengan *input layer* model. Proses ini dilakukan dengan menyalin nilai piksel *grayscale* ke tiga saluran secara bersamaan. Kedua, penyesuaian ukuran gambar. Ukuran asli gambar FER-2013 adalah 48x48 piksel, namun ukuran ini terlalu kecil untuk dimanfaatkan secara efektif oleh model *deep learning* modern, terutama ViT yang memproses citra dalam bentuk *patch*. Oleh karena itu, setiap gambar diubah ukurannya (*resize*) menjadi ukuran yang lebih besar, misalnya 224x224 piksel, agar sesuai dengan kebutuhan arsitektur *backbone* model. Ketiga, gambar dikonversi ke dalam format *tensor* dan tipe data yang sesuai untuk *input* model. Terakhir, dilakukan normalisasi nilai piksel gambar. Nilai piksel gambar yang semula berada dalam rentang 0 hingga 1 diubah menjadi nilai yang berada pada rentang -1 hingga 1.

3.1.4.3 Pembagian Data

Untuk menjaga validitas dan konsistensi proses pelatihan model, pada penelitian ini dilakukan pembagian data ke dalam tiga subset, yakni *training set*, *validation set*, dan *testing set* dengan rasio pembagian 80:10:10 sesuai dengan *default* dari set data FER-2013 itu sendiri. Pembagian ini dilakukan dengan metode *stratified splitting*, yaitu teknik pembagian data yang mempertahankan proporsi jumlah sampel dari setiap kelas ekspresi wajah pada setiap subset. Dengan demikian, setiap subset data tetap merepresentasikan distribusi label yang seimbang, yang penting untuk menghindari bias dalam pelatihan dan evaluasi

model, terutama karena FER-2013 memiliki ketidakseimbangan kelas yang cukup signifikan, seperti jumlah label “disgust” yang jauh lebih sedikit dibanding kelas lainnya.

3.1.4.4 Augmentasi Data

Tahap selanjutnya dalam praproses data pengembangan adalah augmentasi data (*data augmentation*), yang merupakan proses penting untuk meningkatkan generalisasi model serta mengurangi risiko *overfitting*. Augmentasi data bertujuan untuk menghasilkan variasi baru dari data pelatihan yang sudah ada dengan cara melakukan transformasi-transformasi tertentu, sehingga model dapat belajar dari distribusi data yang lebih luas tanpa perlu menambah data secara manual. Proses augmentasi data dilakukan pada setiap *batch* gambar saat pelatihan model berlangsung, atau yang dikenal dengan *online data augmentation*. Namun, augmentasi data ini hanya diterapkan pada data pelatihan saja dan tidak pada data validasi dan pengujian agar nantinya evaluasi tetap akurat dan mewakili kondisi nyata.

3.1.5 Praproses Set Data Pengujian Demonstrasi (Ruang Kelas)

Tahap awal dari praproses pada set data pengujian demonstrasi dimulai dengan konversi video mentah menjadi *frame* atau gambar diam (*still image*). Video yang dikumpulkan dari ruang kelas dipotong secara periodik berdasarkan interval waktu tertentu, misalnya 10 detik, untuk memastikan bahwa setiap *frame* yang diambil mewakili momen-momen penting dari ekspresi wajah peserta didik. Pendekatan ini dirancang untuk menangkap dinamika ekspresi yang terjadi secara alami di kelas tanpa harus mengambil seluruh *frame* dari video, sehingga efisien secara penyimpanan dan pemrosesan data.

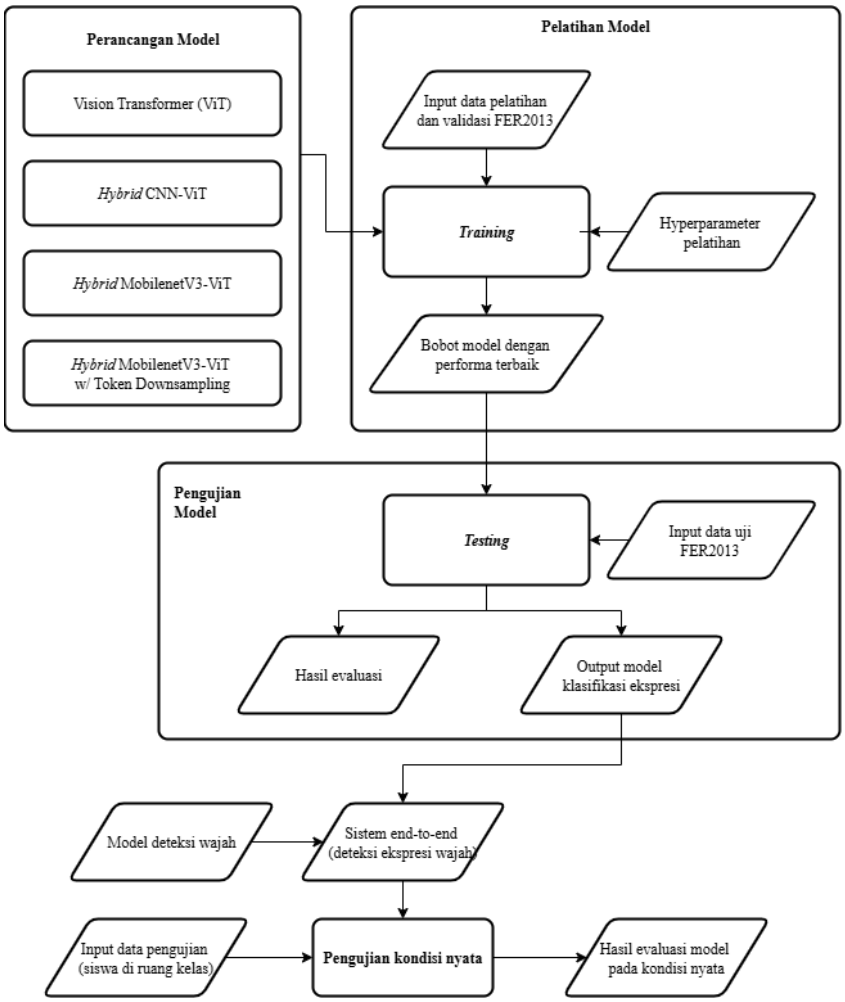
Setelah diperoleh kumpulan gambar diam dari video, dilakukan proses pengecekan kualitas data untuk memastikan hanya gambar yang relevan dan layak diproses lebih lanjut. Pada tahap ini, gambar yang memiliki kualitas rendah, seperti gambar buram, terlalu gelap, atau wajah tidak tampak jelas yang dapat menyulitkan dalam proses pelabelan ekspresi nantinya, dihapus dari set data. Selain itu, gambar yang diambil saat peserta didik belum menampilkan ekspresi sesuai instruksi atau kondisi kelas belum stabil (misalnya ketika peserta didik masih menunduk,

berpaling, atau belum siap) juga dieliminasi, agar hanya ekspresi yang valid dan sesuai konteks yang dianalisis.

Langkah selanjutnya adalah deteksi wajah dan pemberian *bounding box* pada setiap wajah yang terdapat dalam gambar. Deteksi ini dilakukan secara otomatis menggunakan model deteksi wajah yang telah terbukti akurasinya, seperti RetinaFace. Setiap wajah yang terdeteksi dalam gambar akan diberi *bounding box* secara otomatis. Setelah proses deteksi selesai, dilakukan evaluasi manual terhadap hasil deteksi tersebut untuk memastikan bahwa setiap *bounding box* benar-benar mencakup wajah secara presisi. Jika ditemukan ketidaksesuaian, seperti *bounding box* yang meleset atau kurang rapi, maka koreksi dilakukan secara manual. Untuk kasus wajah yang tidak berhasil dideteksi oleh model, *bounding box* akan ditambahkan secara manual.

Tahapan terakhir dari praproses ini adalah pemberian label ekspresi pada setiap wajah yang telah dibatasi dengan *bounding box*. Label ekspresi diberikan berdasarkan kerangka kerja *Facial Action Coding System* (FACS), yang merupakan metode ilmiah dalam mengkodekan ekspresi wajah berdasarkan gerakan otot-otot wajah. Dengan menggunakan referensi FACS, pelabelan menjadi lebih objektif dan konsisten, serta dapat merepresentasikan tujuh ekspresi dasar secara lebih akurat, yaitu marah (*angry*), muak (*disgust*), takut (*fear*), bahagia (*happy*), netral (*neutral*), sedih (*sad*), dan terkejut (*surprise*). Hasil anotasi kemudian disimpan ke dalam format COCO.

3.1.6 Pengembangan Model



Gambar 3. 4 Rancangan Pengembangan Model

Tahap ini merupakan bagian inti dari penelitian, di mana peneliti merancang, melatih, dan menguji model untuk tugas FER. Sebagaimana yang terlihat pada Gambar 3.4, tahapan pengembangan model ini dimulai dengan merancang model yang akan digunakan. Pada penelitian ini model yang digunakan adalah *Vision Transformer* (ViT) sebagai *baseline* model yang selanjutnya dilakukan eksperimen dengan mengganti *patchify* pada ViT dengan konvolusi sebagai *backbone* untuk ekstraksi fiturnya sehingga menghasilkan model *Hybrid*. Dilakukan juga penerapan metode regularisasi jika nantinya ditemukan bawah model kurang baik dalam melakukan generalisasi. Penjelasan lebih rinci terkait eksperimen model dijelaskan pada sub-bab selanjutnya.

Selanjutnya, tiap skenario model dilatih menggunakan data pelatihan dan divalidasi menggunakan data validasi dari set data FER-2013. Pelatihan dilakukan dengan pengaturan *hyperparameter* tertentu yang disesuaikan untuk memperoleh kinerja optimal, seperti *learning rate*, *batch size*, jumlah *epoch*, dan teknik *early stopping* untuk menghindari *overfitting*. Pada akhir proses pelatihan, dipilih bobot model terbaik berdasarkan performa validasi, yang kemudian digunakan untuk menguji performa akhir model pada data uji dari set data FER-2013. Pengujian ini bertujuan untuk mengevaluasi seberapa baik model mampu menggeneralisasi prediksi ekspresi wajah terhadap data yang belum pernah dilihat sebelumnya.

Setelah model klasifikasi ekspresi berhasil dikembangkan, tahap berikutnya adalah mengintegrasikan model klasifikasi tersebut ke dalam sistem yang lebih utuh dengan menggabungkannya dengan model deteksi wajah. Integrasi ini diperlukan agar sistem mampu mengenali ekspresi secara otomatis dari gambar atau video yang mengandung banyak wajah, seperti yang terjadi pada lingkungan kelas. Untuk keperluan ini, digunakan model deteksi wajah seperti RetinaFace, untuk mendeteksi lokasi wajah pada gambar atau video, kemudian setiap wajah yang terdeteksi diproses oleh model klasifikasi ekspresi untuk mengidentifikasi emosi yang ditampilkan.

Sebagai langkah akhir, dilakukan pengujian sistem pengenalan ekspresi wajah secara keseluruhan menggunakan data pengujian demonstrasi yang diperoleh dari lingkungan nyata, yakni data video atau gambar hasil perekaman di ruang kelas. Evaluasi dilakukan untuk menilai seberapa baik sistem dapat mengenali ekspresi wajah peserta didik secara akurat dan efisien dalam kondisi yang mencerminkan penggunaan nyata (*in-the-wild*), seperti pencahayaan yang tidak konsisten, pose wajah beragam, serta kualitas gambar yang tidak ideal. Hasil dari pengujian ini akan menjadi dasar dalam menilai keberhasilan model yang diusulkan dalam menjawab tantangan nyata dalam pengenalan ekspresi wajah di lingkungan pendidikan.

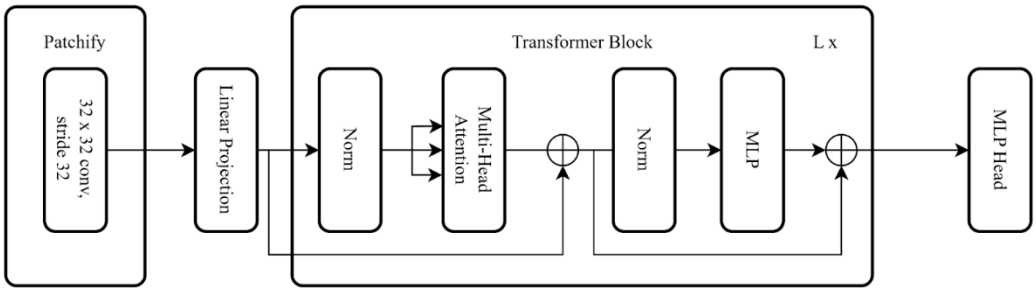
3.1.6.1 Vision Transformer (ViT)

Pada penelitian ini digunakan model ViT-B/32 dengan parameter sebagaimana tercantum pada Tabel 3.1. Dengan konfigurasi tersebut, jumlah total

parameter model mencapai 87.460.615 (87 M). Dua jenis pendekatan model akan digunakan, yaitu model yang dilatih dari awal (*from scratch*) dan model yang menggunakan pra-pelatihan (*pretrained*). Model pra-pelatihan ini mengacu pada hasil penelitian sebelumnya (Ping, 2024), yang diklaim mampu mencapai akurasi sebesar 69.60%. Bobot pra-pelatihan yang digunakan berasal dari model *pretrained* yang tersedia di pustaka PyTorch. Selain itu, dilakukan pula eksperimen dengan melatih model pada data tanpa augmentasi dan dengan augmentasi, guna mengevaluasi performa model saat dihadapkan pada keterbatasan jumlah data latih. Arsitektur dari ViT dapat dilihat pada Gambar 3.5.

Tabel 3. 1 Konfigurasi Parameter Model ViT

Patch size	*Input channel*	*Embedding dims*	*Num heads*	*MLP size*	*Transformer layers*
32	3	768	12	3072	12



Gambar 3. 5 Arsitektur *Vision Transfromer* (ViT)

3.1.6.2 *Hybrid* CNN-ViT

Pada skenario ini, arsitektur *Vision Transformer* (ViT) yang semula menggunakan metode *patchify*, yakni pemotongan gambar *input* menjadi beberapa bagian kecil (*patch*) berukuran tetap, kemudian diproyeksikan menjadi *token input* bagi blok *transformer*, dimodifikasi dengan mengganti proses *patchify* tersebut menggunakan modul konvolusi berbasis *Convolutional Neural Network* (CNN). Modifikasi ini merujuk pada pendekatan dalam (Xiao et al., 2021) di mana CNN digunakan sebagai *backbone* utama untuk mengekstraksi fitur visual dari gambar. Tujuan utama dari penggunaan CNN sebagai pengganti *patchify* adalah untuk

mengatasi kelemahan ViT dalam mengekstraksi informasi lokal pada tahap awal, terutama ketika jumlah data pelatihan terbatas, seperti yang umum terjadi pada set data dengan distribusi tidak seimbang atau ukuran kecil. CNN memiliki kemampuan yang lebih baik dalam menangkap informasi lokal seperti tepi, pola tekstur, dan bentuk objek secara hierarkis, sehingga cocok dijadikan penguat tahap awal sebelum *token* dikirimkan ke blok *transformer*.

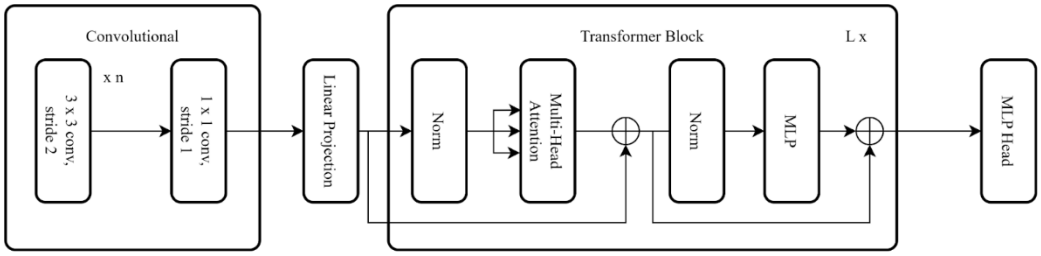
Arsitektur lengkap dari model *Hybrid* CNN-ViT dapat dilihat pada Gambar 3.6, dengan rincian konfigurasi *layer* CNN ditampilkan pada Tabel 3.2. Alur pemrosesan pada arsitektur ini dimulai dengan gambar *input* yang pertama kali diproses oleh jaringan CNN. CNN bertindak sebagai *feature extractor*, di mana *input* gambar dilewatkan melalui beberapa lapisan konvolusi, aktivasi *nonlinear*, dan normalisasi, untuk menghasilkan representasi spasial tingkat tinggi berupa *feature map*. *Feature map* ini secara efektif menggambarkan fitur visual penting dari gambar, menggantikan fungsi *patchify* pada ViT standar.

Ukuran *output feature map* dari modul CNN ditentukan agar sesuai dengan dimensi *patch* pada ViT-16, yaitu 14x14 piksel. Selain itu, jumlah *channel* akhir dari *feature map* juga disesuaikan dengan dimensi *embedding* yang dibutuhkan oleh ViT, misalnya 768 dimensi, agar dapat langsung diteruskan sebagai *patch token* ke dalam blok *transformer*. Proses ini dilakukan dengan cara melakukan *flattening* pada *feature map* menjadi deretan *token* vektor satu dimensi. Token-token ini kemudian dimasukkan ke dalam blok *transformer* ViT, di mana *self-attention* digunakan untuk memodelkan relasi spasial global antar bagian gambar. Dengan desain ini, diharapkan model dapat memanfaatkan kekuatan CNN dalam mengekstraksi fitur lokal dan kekuatan ViT dalam memahami konteks global, sehingga menghasilkan representasi yang lebih kaya untuk tugas pengenalan ekspresi wajah.

Tabel 3. 2 *Convolutional Stem* pada *Hybrid* CNN-ViT

Layer	*Input Size*	*Kernel Size*	*Stride*	*Padding*	*Output Channel*
Conv1 + BN	224 x 224	3 x 3	2	1	24

Layer	*Input Size*	*Kernel Size*	*Stride*	*Padding*	*Output Channel*
Conv2 + BN	112 x 112	3 x 3	2	1	48
Conv3 + BN	56 x 56	3 x 3	2	1	96
Conv4 + BN	28 x 28	3 x 3	2	1	192
Final Conv (1 x 1)	14 x 14	3 x 3	1	0	*embedding*



Gambar 3. 6 Arsitektur *Hybrid* CNN-ViT

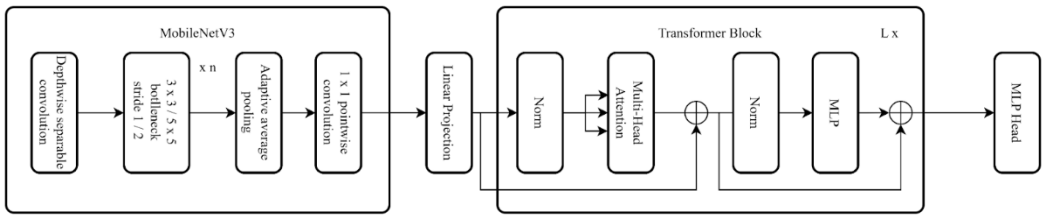
3.1.6.3 *Hybrid* MobileNetV3-ViT

Pada skenario ini, pengembangan model diarahkan pada peningkatan akurasi arsitektur *hybrid* dengan mengganti modul konvolusional (CNN) yang sebelumnya digunakan sebagai *backbone* pada *Vision Transformer* (ViT), menjadi arsitektur yang sudah *establish* yakni MobileNetV3. MobileNetV3 (Howard et al., 2019) merupakan arsitektur *Convolutional Neural Network* (CNN) yang dirancang secara khusus untuk keperluan inferensi cepat dan efisien pada perangkat dengan keterbatasan sumber daya, seperti perangkat *mobile* atau *edge device*. MobileNetV3 menggabungkan berbagai teknik efisiensi tinggi, yang menjadikannya sangat ringan namun tetap mempertahankan kinerja yang kompetitif pada tugas-tugas visi komputer.

Dalam penelitian ini, dilakukan eksperimen terhadap dua varian dari MobileNetV3, yaitu MobileNetV3-Small yang memiliki jumlah parameter sekitar 2.5 juta dan MobileNetV3-Large. MobileNetV3-Small yang memiliki sekitar 5.4 juta parameter. Keduanya diinisialisasi dengan bobot awal dari model pra-pelatihan

(*pretrained weights*) yang tersedia secara resmi dalam pustaka PyTorch, guna mempercepat konvergensi pelatihan dan meningkatkan generalisasi awal. Tujuan dari eksperimen ini adalah untuk mengevaluasi *trade-off* antara akurasi dan efisiensi komputasi.

Arsitektur dari model *Hybrid* MobileNetV3-ViT digambarkan pada Gambar 3.7. Model ini dirancang untuk menggabungkan keunggulan arsitektur konvolusional yang efisien, yaitu MobileNetV3, dengan kekuatan pemodelan spasial global dari *Vision Transformer*. Proses alur kerja model ini dimulai dari citra wajah berwarna (RGB) dengan resolusi tetap, misalnya 224x224 piksel, yang telah dipraproses melalui tahapan normalisasi dan penyesuaian ukuran. Gambar tersebut kemudian dimasukkan ke dalam MobileNetV3, yang bertindak sebagai *backbone* konvolusional untuk mengekstraksi fitur spasial lokal. MobileNetV3 memanfaatkan teknik efisiensi seperti *depthwise separable convolution*, *inverted residual block*, dan *squeeze-and-excitation* untuk menghasilkan representasi fitur yang kaya namun tetap ringan secara komputasi. Hasil dari ekstraksi ini berupa *feature map* berdimensi spasial, misalnya 7x7 dengan kedalaman saluran (*channel*) yang tergantung pada varian MobileNetV3 yang digunakan (576 untuk MobileNetV3-Small dan 960 untuk MobileNetV3-Large).



Gambar 3. 7 Arsitektur *Hybrid* MobileNetV3-ViT

Selanjutnya sebelum *feature map* masuk dan diproses oleh blok *transformer*, *feature map* akan dialirkan ke dalam *projection layer* terlebih dahulu, yakni sebuah layer konvolusi dengan besar *kernel* 1x1 (*pointwise convolutional*). Tujuannya adalah untuk menyesuaikan *channel* dari *feature map* agar kompatibel dengan besar *embedding* yang dapat diproses oleh blok *transformer*, misalnya sebesar 768. *Feature map* tersebut kemudian di-*flatten* menjadi sekumpulan *token* dan ditambahkan *token* khusus bernama *class* atau CLS *token* di awal urutan *token*,

yang berfungsi sebagai representasi global citra dan akan digunakan pada tahap klasifikasi akhir. Selain itu, *positional encoding* juga ditambahkan ke setiap *token* untuk mempertahankan informasi posisi spasial, karena ViT tidak memiliki kemampuan implisit untuk mengenali struktur spasial. Urutan *token* ini kemudian dimasukkan ke dalam *transformer encoder* dan terakhir representasi vektor dari *token* CLS diambil sebagai hasil representasi global citra. *Token* ini kemudian dilewatkan ke dalam *layer* klasifikasi berupa *fully connected layer* terakhir yang menghasilkan probabilitas untuk masing-masing kelas ekspresi wajah.

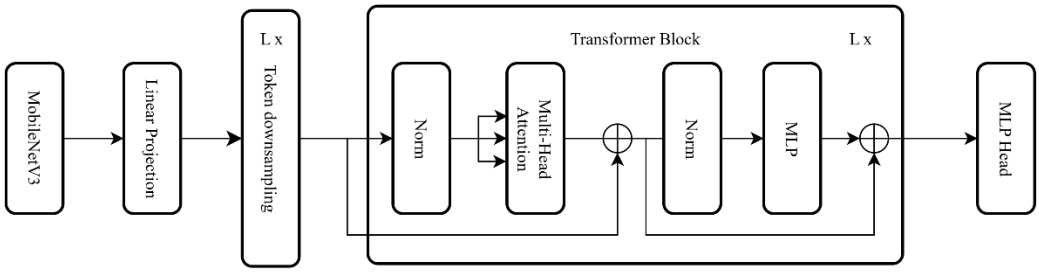
3.1.6.4 *Hybrid* MobileNetV3-ViT dengan *Token Downsampling*

Untuk mengatasi permasalahan utama *Vision Transformer* (ViT) yang memerlukan beban komputasi tinggi akibat jumlah *token* yang besar, penelitian ini mengadopsi pendekatan *Token Downsampling*. Teknik ini bertujuan untuk mengurangi jumlah *token* yang diproses oleh blok-blok *transformer*, tanpa mengorbankan informasi penting yang dibawa oleh *token* tersebut. Dengan berkurangnya jumlah *token*, kompleksitas komputasi terutama dalam operasi *self-attention*, yang skalanya kuadratik terhadap jumlah *token*, dapat ditekan secara signifikan, sehingga model menjadi lebih efisien dan dapat digunakan di lingkungan dengan sumber daya terbatas.

Secara umum, seperti yang terlihat pada Gambar 3.8, modul *Token Downsampling* diletakkan di antara blok *transformer*. Kemudian sebuah parameter tambahan, yakni *retention* ditentukan. Parameter ini yang mengatur banyaknya *token* yang akan dipertahankan untuk blok *transformer* selanjutnya dengan nilai rentang 0 (artinya hanya *token* CLS yang dipertahankan) hingga total jumlah *token*. Jumlah parameter *retention* disesuaikan dengan jumlah blok *transformer*. *Token* yang telah diproses oleh *transformer* awal kemudian diseleksi untuk menentukan subset *token* yang paling representatif, yang akan diteruskan ke blok *transformer* selanjutnya. Dalam penelitian ini, dua teknik *downsampling* digunakan dan dibandingkan, yaitu metode *merging* dengan pendekatan *clustering-based downsampling* dan metode *pruning* dengan pendekatan *score-based downsampling*.

Pendekatan pertama, *cluster-based token downsampling*, mengimplementasikan algoritma k-medoids untuk mengelompokkan *token*

berdasarkan kemiripan representasi vektornya. Proses ini diawali dengan memilih sejumlah *token* secara acak sebagai *medoid* awal, jumlahnya sama dengan target *token output* yang diinginkan. Kemudian, token-token lainnya dikelompokkan berdasarkan jarak Euclidean terhadap *medoid* di ruang *embedding*, di mana token-token yang memiliki representasi serupa akan membentuk *cluster* yang sama. Setelah proses iteratif pengelompokan selesai, *medoid* dari masing-masing *cluster* dipertahankan sebagai representasi dari kelompoknya dan digunakan sebagai *input* untuk blok *transformer* berikutnya. Pendekatan ini efektif untuk mempertahankan keragaman informasi karena setiap *cluster* mewakili kelompok *token* dengan fitur serupa.



Gambar 3. 8 Arsitektur *Hybrid* MobileNetV3-ViT dengan *Token Downsampling*

Pendekatan kedua, *score-based token downsampling*, menilai setiap *token* berdasarkan kontribusinya terhadap *attention* di dalam *transformer*. Nilai *significance score* dari setiap *token* dihitung dengan menjumlahkan bobot *attention* yang diterima sebuah *token* terhadap token-token lainnya. *Token* dengan skor paling tinggi dianggap paling penting dalam merepresentasikan informasi gambar, dan hanya token-token teratas yang dipertahankan. Jumlah *token* yang dipertahankan juga ditentukan sesuai target yaitu parameter *retention*, misalnya 15 yang artinya hanya dipertahankan 15 *token* teratas dari jumlah *token* awal. Berbeda dari pendekatan *clustering*, teknik *score-based* ini lebih eksplisit dalam memprioritaskan token-token yang benar-benar memiliki peran besar dalam konteks global perhatian model, meskipun bisa jadi kehilangan variasi informasi dari *token* minor.

Kedua teknik di atas memiliki kelebihan masing-masing: metode *cluster-based* lebih menjaga keragaman *token* dan hubungan antar fitur, sementara metode

score-based lebih selektif terhadap kontribusi aktual dari *token* dalam mekanisme *attention*. Oleh karena itu, eksperimen dalam penelitian ini membandingkan keduanya dalam hal performa klasifikasi dan efisiensi komputasi untuk menentukan pendekatan mana yang paling optimal.

3.1.6.5 Sistem *End-to-End* (Deteksi Ekspresi Wajah)

Setelah proses pengembangan model klasifikasi ekspresi wajah selesai dilakukan, langkah selanjutnya dalam penelitian ini adalah mengintegrasikan model tersebut ke dalam sistem deteksi ekspresi wajah secara *end-to-end*, yaitu dengan menggabungkan model klasifikasi ekspresi dan model deteksi wajah menjadi satu *pipeline* utuh. Tujuan dari sistem *end-to-end* ini adalah untuk memungkinkan model menerima *input* berupa gambar atau *frame* video mentah, kemudian secara otomatis mendeteksi lokasi wajah dan mengklasifikasikan ekspresinya tanpa memerlukan intervensi manual. Integrasi ini penting untuk penerapan sistem secara nyata di lingkungan dinamis seperti ruang kelas, di mana wajah peserta didik tidak selalu berada di posisi yang tetap dan jumlahnya bisa lebih dari satu dalam satu *frame*.

Dalam penelitian ini, dipilih model deteksi wajah yang sudah tersedia (*existing*) sebagai komponen deteksi wajah, dengan pertimbangan efisiensi waktu dan performa yang sudah terbukti di berbagai penelitian sebelumnya. Model yang digunakan adalah RetinaFace (Deng et al., 2019), yang merupakan salah satu model deteksi wajah berbasis CNN dengan performa tinggi, terbukti dari perolehan *Average Precision* (AP) sebesar 0.914 pada set data WIDER FACE, yang merupakan *benchmark* umum untuk tugas deteksi wajah. RetinaFace mampu mendeteksi wajah dengan akurat bahkan pada kondisi sulit seperti pose ekstrem, pencahayaan buruk, dan skala yang kecil, yang sangat relevan dengan kondisi nyata di ruang kelas.

Alur sistem *end-to-end* deteksi ekspresi wajah ini dimulai dari gambar *input*, baik berupa gambar tunggal maupun *frame* dari video. Gambar tersebut akan diproses terlebih dahulu oleh model deteksi wajah RetinaFace untuk menemukan *Region of Interest* (ROI), yaitu lokasi wajah dalam gambar beserta koordinat *bounding box*-nya. Setiap ROI yang terdeteksi kemudian akan diproses lebih lanjut oleh model klasifikasi ekspresi wajah dengan cara memotong area wajah yang

terdeteksi berdasarkan *bounding box*, kemudian diubah ukurannya dan diolah menjadi *input* yang sesuai dengan arsitektur klasifikasi. Model klasifikasi kemudian memprediksi kategori ekspresi dari setiap wajah yang terdeteksi.

Setelah *pipeline end-to-end* terbentuk, dilakukan pengujian sistem secara menyeluruh menggunakan set data pengujian demonstrasi, yaitu data yang dikumpulkan dari simulasi lingkungan kelas nyata. Pengujian ini bertujuan untuk mengevaluasi keandalan dan performa model dalam situasi nyata, termasuk aspek seperti variasi pencahayaan, sudut pandang wajah, ekspresi yang ambigu, serta perbedaan karakteristik wajah peserta didik. Pengujian ini juga mencerminkan aplikasi sebenarnya dari model dalam mendukung analisis emosi pada lingkungan pembelajaran.

Selain proses pengujian, penelitian ini juga berfokus pada adaptasi model agar performanya lebih sesuai dengan konteks ruang kelas. Terdapat dua pendekatan utama yang digunakan dalam proses adaptasi ini. Pertama adalah strategi *fine-tuning*, yaitu dengan mengambil model klasifikasi ekspresi wajah yang sebelumnya telah dilatih pada set data publik FER-2013, lalu dilakukan pelatihan lanjutan menggunakan data ruang kelas. Tujuan dari *fine-tuning* ini adalah untuk memperbarui bobot model agar lebih sensitif terhadap pola-pola visual khas lingkungan kelas, seperti ekspresi wajah peserta didik yang mungkin lebih halus atau terbagi dalam berbagai posisi kamera.

Pendekatan kedua adalah strategi pelatihan gabungan (*joint training*), di mana model dilatih sejak awal menggunakan gabungan dari FER-2013 dan set data ruang kelas. Dengan cara ini, model memiliki akses ke distribusi data yang lebih beragam dan luas, sehingga diharapkan dapat meningkatkan kemampuan generalisasi model terhadap berbagai variasi ekspresi dan kondisi visual. Data FER-2013 memberikan landasan yang kuat dalam mengenali ekspresi secara umum, sementara data ruang kelas memperkaya konteks spesifik aplikasi. Strategi ini dirancang untuk membuat model tidak hanya akurat dalam kondisi laboratorium, tetapi juga *robust* dan siap digunakan dalam implementasi nyata.

3.1.7 Analisis dan Evaluasi

Tahap evaluasi dan analisis performa merupakan bagian penting dalam proses penelitian ini karena bertujuan untuk menilai efektivitas dan keandalan model klasifikasi ekspresi wajah yang telah dikembangkan melalui berbagai skenario eksperimen. Pada tahap ini, seluruh hasil model yang berhasil dijalankan akan dianalisis secara menyeluruh baik dari sisi kuantitatif maupun kualitatif, untuk memberikan pemahaman yang utuh mengenai kekuatan dan kelemahan masing-masing model.

Evaluasi performa model klasifikasi ekspresi dilakukan menggunakan metrik evaluasi klasifikasi standar, yaitu akurasi dan F1-*score*. Selain itu, dilakukan juga analisis terhadap *confusion matrix* yang memberikan informasi detail tentang prediksi benar dan salah dari masing-masing kelas ekspresi, sehingga memudahkan dalam menilai sejauh mana model dapat membedakan tiap ekspresi dengan baik. Akurasi menunjukkan proporsi prediksi yang benar terhadap keseluruhan prediksi, sedangkan F1-*score* memberikan gambaran keseimbangan antara presisi dan *recall*, terutama berguna ketika distribusi data tidak seimbang antar kelas. Formula untuk metrik-metrik ini dirujuk pada Persamaan (3.1) hingga (3.4), yang menjelaskan secara matematis cara perhitungannya (Grandini et al., 2020). Selain metrik klasifikasi, nilai *loss* dari setiap model juga dianalisis, terutama *loss* pada data pelatihan dan validasi untuk setiap *epoch*. Grafik kurva *loss* ini digunakan untuk mengevaluasi proses pembelajaran model, apakah terjadi *overfitting* atau *underfitting*. Dengan demikian, analisis *loss* memberikan gambaran penting terkait kestabilan dan efektivitas pelatihan model.

Untuk model integrasi antara klasifikasi ekspresi dan deteksi wajah (sistem *end-to-end*), evaluasi dilakukan dengan pendekatan yang berbeda. Karena sistem ini bekerja untuk mendeteksi wajah dan langsung mengklasifikasikan ekspresinya, maka evaluasi dilakukan menggunakan metrik *Average Precision* (AP). AP digunakan untuk mengukur kualitas deteksi dan klasifikasi secara bersamaan, yakni apakah model dapat mendeteksi wajah dengan tepat dan sekaligus mengklasifikasikan ekspresi dengan benar. Nilai AP ini dihitung berdasarkan kurva *Precision-Recall*, yang formula matematisnya dijelaskan dalam Persamaan (3.5)

(Padilla et al., 2020). Semakin tinggi nilai AP, semakin baik performa sistem secara keseluruhan dalam konteks aplikasi dunia nyata.

Selain evaluasi kuantitatif, juga dilakukan analisis kualitatif untuk menilai performa model secara lebih mendalam. Analisis ini dilakukan dengan mengamati kualitas prediksi model terhadap sampel gambar dari masing-masing kelas ekspresi. Analisis kualitatif ini penting karena dapat memberikan wawasan tambahan yang tidak bisa ditangkap hanya dari angka metrik evaluasi.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.1}$$

$$Precision = \frac{TP}{TP + FP} \tag{3.2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3.3}$$

$$F1 - score = \frac{2\ x\ Precision\ x\ Recall}{Precision + Recall} \tag{3.4}$$

$$AP = \sum_{n} (R_{n+1} - R_n) P_{interp}(R_{n+1}) \tag{3.5}$$

3.1.8 Penarikan Kesimpulan

Tahap ini adalah tahap akhir dari penelitian, di mana kesimpulan ditarik berdasarkan hasil analisis yang dilakukan pada tahap evaluasi. Kesimpulan ini penting untuk melihat kinerja model yang telah dikembangkan dan mengevaluasi apakah tujuan dari penelitian telah tercapai. Penarikan kesimpulan juga mencakup perbandingan hasil dengan tujuan awal penelitian dan hipotesis yang diajukan. Proses penarikan kesimpulan didasarkan pada rumusan masalah yang telah ditetapkan sebelumnya.

3.2 Lingkungan Komputasi

Penelitian ini dilaksanakan dengan menggunakan perangkat keras dan perangkat lunak tertentu. Adapun spesifikasi lengkap dari perangkat yang digunakan dijabarkan sebagai berikut:

A. Perangkat keras (Laptop)
    a. CPU Intel(R) Core(TM) i7-11800H (2.30 GHz)
    b. GPU Nvidia GeForce RTX 3060 6 GB
    c. RAM DDR4 16 GB
    d. SSD 1.4 TB

B. Perangkat lunak
    a. Windows 11
    b. Jupyter Notebook
    c. CUDA *driver* 12.4
    d. Python 3.9.12