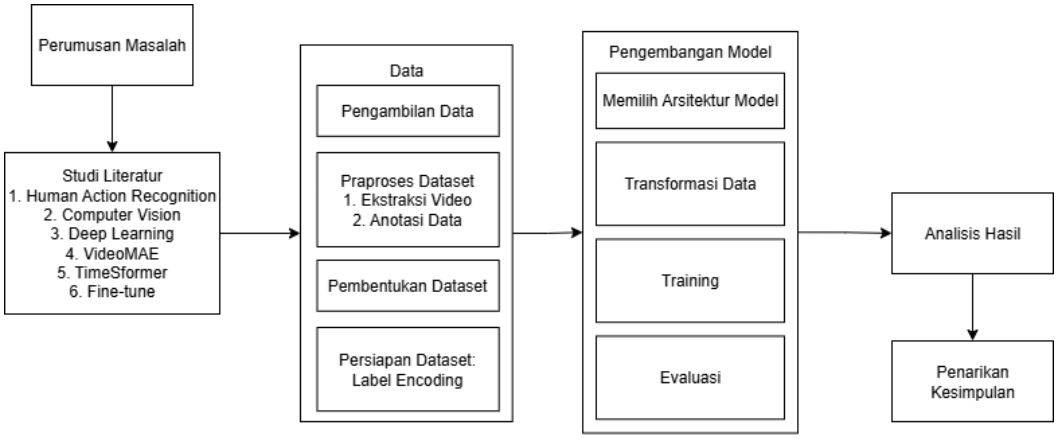
BAB III
METODE PENELITIAN

3.1 Desain Penelitian

Desain Penelitian adalah kerangka kerja yang digunakan sebagai perencanaan dan gambaran prosedur selama penelitian. Desain penelitian diilustrasikan dengan diagram pada Gambar 3.1 di bawah.



Gambar 3.1 Desain Penelitian

Terdapat tujuh tahapan utama yang akan dilakukan pada penelitian ini, tahap-tahap tersebut adalah sebagai berikut:

3.1.1 Perumusan Masalah

Penelitian ini diawali dengan tahap perumusan masalah, yaitu proses mengidentifikasi permasalahan yang terjadi di lapangan dan menentukan pendekatan untuk menyelesaikannya. Masalah yang diangkat dalam penelitian ini adalah mengenali dan mengklasifikasikan aksi peserta didik di dalam ruang kelas secara otomatis. Untuk menjawab permasalahan tersebut, digunakan pendekatan berbasis deep learning dengan memanfaatkan model VideoMAE. Model TimeSformer turut digunakan sebagai model *baseline* atau pembanding dalam mengevaluasi efektivitas VideoMAE dalam tugas klasifikasi tersebut.

3.1.2 Studi Literatur

Studi literatur dilakukan dengan mengumpulkan sumber referensi dan teori dari buku, artikel, penelitian, jurnal, dan konferensi yang telah dilakukan mengenai penelitian terdahulu dalam ruang lingkup yang sama dengan penelitian, *Human Action Recognition*. *Computer Vision*, *Deep Learning,* VideoMAE, TimeSformer, dan *Fine-tune*.

3.1.3 Data

Beberapa tahap yang dilakukan berhubungan data adalah sebagain berikut:

a. Pengambilan Data

Pengambilan data dilakukan dengan merekam kegiatan belajar mengajar dengan latar ruang kelas, sesuai dengan konteks penelitian. Proses perekaman dilakukan menggunakan beberapa perangkat kameran dengan sudut pengambilan bervariasi, yaitu:

- Kamera CCTV, diletakkan di atas papan tulis atau di tengah ruang kelas
- Handycam, diletakkan di sudut kanan depan kelas
- Kamera Ponsel, diletakkan di sudut kiri depan kelas

Data diambil dalam berbagai sudut pandang dengan tujuan untuk menangkap keragaman visual dari aksi peserta didik selama proses pembelajaran, sehingga model dapat mengenali aksi dari perspektif yang berbeda dan mempelajari kondisi nyata kegiatan dalam ruang kelas.

b. Praproses Data

Sebelum digunakan sebagai masukan ke dalam model, data mentah perlu melalui dua tahap praproses untuk memastikan bahwa data yang digunakan cocok untuk pelatihan model. Tahapan praproses tersebut meliputi:

- Ekstraksi Klip Video, Video mentah yang memiliki durasi panjang dan menampilkan banyak peserta didik dalam satu *frame* dipotong menjadi klip pendek. Hasil ekstraksi klip video hanya menampilkan satu peserta didik dalam frame mencakup satu peserta didik dalam

frame. Durasi klip juga disesuaikan agar hanya mencakup interval waktu di mana aksi yang relevan terjadi, sekitar 3-9 detik. Hal ini dilakukan agar model tidak bingung ketika mempelajari label

- Anotasi data, Setiap video klip hasil ekstraksi melalui anotasi data dengan diberi label sesuai dengan aksi yang ditampilkan oleh peserta didik. Proses anotasi dilakukan secara manual berdasarkan pengamatan terhadap isi video. Aksi peserta didik diklasifikasikan ke dalam lima kategori kelas, yaitu menunduk, mengangkat tangan, menggunakan ponsel, menopang kepala di atas meja, dan mengangguk.

c. Pembentukan Dataset

Setelah melalui seluruh tahap praproses, data video disusun ke dalam format dataset yang siap digunakan untuk proses pelatihan dan evaluasi model. Data dikelola menggunakan struktur dataset *folder-based*, yaitu metode pengelompokan data yang lazim digunakan dalam tugas klasifikasi di bidang *computer vision*. Pada pendekatan ini, seluruh klip video disimpan dalam sebuah folder utama yang diberi nama "dataset" atau sesuai konteks dari isi dataset tersebut. Folder utama kemudian memiliki subfolder yang merepresentasikan kelas aksi peserta didik.

Setiap subfolder kelas berisi kumpulan video yang hanya menampilkan satu jenis aksi tertentu, sehingga model dapat belajar mengenali pola visual yang konsisten untuk setiap aksi. Dalam penelitian ini, subfolder dibagi menjadi lima kategori dan diberi nama sesuai aksi, yaitu "mengangguk", "mengangkat_tangan", "menggunakan_hp", "menopang_kepala", dan "menunduk".

Jika seluruh klip video sudah ditempatkan sesuai dengan subfoldernya, maka dilakukan pembagian dataset (*split dataset*). *Split dataset* ke dalam tiga subset mengikuti rasio 80% untuk pelatihan (*training*), 10% untuk validasi (*validation*), dan 10% untuk pengujian (*testing*).

d. Persiapan Dataset

Persiapan dataset bertujuan menyiapkan data agar siap digunakan oleh model. Pada tahap ini, dataset yang sudah terbagi menjadi train, validation, dan test akan dibaca dari folder masing-masing. Setiap video diambil path-nya dan dicatat bersama label kelasnya ke dalam *dataframe* yang berisi kolom *path* dan label.

Selanjutnya dilakukan *label encoding*, yaitu proses mengubah label yang semula berbentuk teks (misalnya "mengangguk", "menggunakan_hp") menjadi angka. Proses ini penting karena model hanya dapat memproses label dalam bentuk numerik. Hasil *encoding* disimpan dalam bentuk *dictionary.*

3.1.4 Pengembangan Model

Dalam pengembangan model ada beberapa langkah di dalamnya, antara lain:

a. Pemilihan Arsitektur Model

Pemilihan arsitektur model pada penelitian ini didasarkan pada kebutuhan untuk melakukan klasifikasi aksi peserta didik di dalam kelas secara akurat menggunakan data video. Model utama yang digunakan adalah VideoMAE, yang memiliki kemampuan representasi video yang baik melalui strategi *masked autoencoding* sehingga dapat bekerja secara efektif pada dataset berukuran terbatas. Sebagai model pembanding *baseline*, digunakan TimeSformer, yang merupakan salah satu model video transformer pertama dan termasuk state-of-the-art pada dataset Kinetics-400 dan Kinetics-600. Kedua model dipilih karena sama-sama dikembangkan dari arsitektur ViT. Perbandingan keduanya bertujuan untuk mengevaluasi kinerja arsitektur berbasis transformer pada data video yang direkam di lingkungan kelas.

Model yang digunakan dalam penelitian ini adalah model *pre-trained* yang telah dilatih pada dataset Kinetics-400, salah *benchmark* pada tugas *human action recognition*. Dataset ini dikembangkan oleh DeepMind dan berisi sekitar 400 kelas aksi manusia dengan total lebih dari 300 ribu klip video yang diambil dari YouTube. Setiap klip video

berdurasi sekitar 10 detik dan menampilkan satu jenis aksi, seperti olahraga, aktivitas sehari-hari, hingga interaksi sosial.

Kedua model *pre=trained* dapat diakses publik pada *platform* HuggingFace. Variasi model yang digunakan untuk VideoMAE adalah "MCG-NJU/videomae-base" dan untuk TimeSformer adalah "facebook/timesformer-base-finetuned-k400". Pada masing-masing model *pre-trained* sudah ada konfigurasi parameter model dan prosesornya. Perbandingan konfigurasi model *pretrained* VideoMAE dan TimeSformer dapat dilihat pada Tabel 3.1.

Tabel 3.1 Perbanding Konfigurasi Awal VideoMAE dan TimeSformer

Parameter	VideoMAE	TimeSformer
image_size	224	224
initializer_range	0.02	0.02
intermediate_size	3072	3072
num_attention_heads	12	12
num_channels	3	3
num_frames	16	8
num_hidden_layers	12	12
patch_size	16	16
tubelet_size	2	2
hidden_act	gelu	gelu
decoder_hidden_size	384	-
decoder_intermediate_size	1536	-

decoder_num_attention_heads	6	-
decoder_num_hidden_layers	4	-
use_mean_pooling	True	-
Trainable Parameters	**94.2M**	**121M**

b. Transformasi Data

Data melalui proses transformasi terlebih dahulu sebelum dijadikan input model karena model hanya menerima input data dalam bentuk *tensor* atau. tahap-tahap yang dilewati data dalam prosesor adalah sebagai berikut:

- *Frame Sampling*

  Frame sampling adalah tahap pengambilan sejumlah frame tertentu dari setiap video agar setiap input memiliki jumlah frame yang konsisten. Sesuai dengan "num_frames" pada Tabel 3.1, jumlah frame yang diambil disesuaikan dengan kebutuhan masing-masing model. Untuk VideoMAE diambil 16 frame per video, sedangkan untuk TimeSformer diambil 8 frame per video. Pengambilan frame dapat dilakukan secara berurutan dengan interval tertentu atau secara acak agar tetap mewakili keseluruhan durasi video.

- *Resize*

  Pada tahap *resize* dilakukan pengubahan ukuran frame video ke 224 piksel, menjaga agar aspek rasio tidak terdistorsi.

- *Normalize*

  Pada tahap ini dilakukan standarisasi nilai piksel menggunakan mean dan standar deviasi (std). Pada penelitian ini, nilai yang digunakan adalah nilai default dari kedua model, yaitu mean dengan nilai [0.485, 0.456, 0.406] dan std dengan nilai [0.229,

0.224, 0.225]. Rumus perhitungan yang digunakan untuk menghitung normalisasi Z-score per channel adalah:

$$x' = \frac{x - \mu}{\sigma}$$

Keterangan:

μ = mean

σ = standar deviasi channel RGB

- Ubah ke Tensor

  Output akhir dari transformasi model merupakan tensor dengan format (num_frames, channel, height, width) dengan keterangan num_frames adalah jumlah frame masukan model, pada penelitian ini num_frames VideoMAE adalah 16 dan TimeSformer adalah 8. *Channel* adalah saluran warna, penelitian ini menggunakan saluran warna RGB (*Red Green Blue*). Height dan width adalah tinggi dan lebar daru data, yaitu 224. Maka, output akhir dari transformasi data adalah (16, 3, 224,224) untuk videoMAE dan (8, 3, 224,224) untuk TimeSformer.

Data kemudian diubah ke dalam bentuk Tensor dengan mengintegrasikan fungsi transformasi dengan *dataframe* yang sudah berisi *path* video dan label numerik untuk membuat objek dataset yang bisa dibaca oleh *data loader* pada saat *training*, validasi, dan pengujian.

c. Training

Proses *training* dilakukan dengan fine-tuning pada kombinasi dari parameter *epoch, batch size, learning rate, weight decay,* dan *learning rate scheduler* untuk menyesuaikan bobot model *pretrained* terhadap karakteristik dataset. Untuk mendapatkan konfigurasi parameter baru yang optimal, dilakukan beberapa skenario eksperimen kombinasi parameter. Skenario eksperimen tersebut adalah:

1. Eksperimen Model Zoo

   Rancangan skenario eksperimen ini dilakukan dengan mengacu pada konfigurasi parameter awal sebagaimana digunakan

oleh penulis asli dari masing-masing model. Konfigurasi awal masing-masing model dijelaskan pada bagian desain penelitian. Tabel 3.2 memuat parameter yang digunakan untuk masing-masing model saat proses pelatihan. Penyesuaian parameter tertentu dilakukan berdasarkan keterbatasan lingkungan komputasi yang digunakan dalam pelatihan, yaitu Google Colab. Spesifikasi fitur yang disediakan Google Colab dijelaskan pada sub bab 3.2. Hasil dari eksperimen ini akan menjadi tolak ukur awal untuk penentuan parameter eksperimen selanjutnya. Parameter yang telah disesuaikan dengan lingkungan komputasi dapat dilihat pada tabel di bawah.

ID Eksperimen	Model	Epoch	Batch Size	Learning Rate
1	VideoMAE	50	8	1e-3
2	TimeSformer	15	4	5e-3

Tabel 3.2 Parameter Eksperimen Model Zoo

2. Eksperimen Penyesuaian Parameter

Setelah melihat hasil dari eksperimen dengan parameter zoo model, dilakukan penyesuaian parameter yang cocok untuk dataset kecil dengan tujuan mendapatkan akurasi dan robustness model yang lebih baik. Parameter juga disesuaikan dengan keterbatasan lingkungan komputasi. Parameter yang digunakan adalah epoch sebesar 20, learning rate 1e-5 & 5e-5, dan batch size sebesar 4. Dalam penelitian ini digunakan dua nilai learning rate dengan pertimbangan bahwa pemilihan nilai learning rate seringkali menjadi kunci utama dalam fine-tuning model transformer. Kombinasi parameter yang akan diimplementasikan pada rancangan eksperimen dapat dilihat pada Tabel 3.3.

Tabel 3.3 Parameter Eksperimen Penyesuaian Parameter

ID Eksperimen	Model	Epoch	Batch Size	Learning Rate
3	VideoMAE	20	4	1e-5
4	VideoMAE	20	4	5e-5
5	TimeSformer	20	4	1e-5
6	TimeSformer	20	4	5e-5

3. Eksperimen Regularisasi & Scheduler

Setelah melihat hasil dari eksperimen sebelumnya, dilakukan penyesuaian upaya dalam mengurangi terjadinya *overfitting* dengan menerapkan regularisasi dan scheduler. Fungsi dari weight decay adalah memberikan penalti pada bobot model yang terlalu besar, sehingga model terdorong untuk memiliki bobot yang lebih sederhana dan tidak terlalu bergantung pada pola spesifik di data latih. Dengan begitu, generalisasi ke data validasi dapat lebih baik. Nilai weight decay 0.01 dipilih karena nilai ini umum digunakan sebagai titik awal yang seimbang. Digunakan juga learning rate scheduler untuk mengatur perubahan laju pembelajaran seiring berjalannya epoch. Tanpa scheduler, learning rate tetap sama dari awal hingga akhir pelatihan, yang bisa menyebabkan model kesulitan mencapai titik optimal karena langkah pembelajaran terlalu besar di akhir atau terlalu kecil di awal. Pada eksperimen ini digunakan scheduler tipe linear. Linear scheduler bekerja dengan menurunkan learning rate secara bertahap dan linier (garis lurus) dari nilai awal hingga mendekati nol pada akhir pelatihan. Pemilihan linear dilakukan karena sifatnya sederhana dan stabil untuk menjaga

agar model tidak melakukan update bobot yang terlalu agresif di akhir pelatihan.

Parameter yang digunakan adalah epoch yang ditetapkan pada angka 50, learning rate 1e-5 & 5e-5, batch size 4, weight decay 0.01, dan learning rate scheduler tipe “linear”. Epoch dinaikan ke angka 50 agar model mendapatkan kesempatan lebih lama untuk mempelajari data. Namun, peningkatan epoch juga berisiko menyebabkan overfitting. Untuk mengantisipasi hal tersebut, diterapkan early stopping pada metrik akurasi validasi dengan patience sebesar 10. Jika dalam 10 epoch berturut-turut akurasi validasi tidak mengalami peningkatan yang signifikan, maka proses pelatihan akan dihentikan lebih awal. Pendekatan ini dipilih agar model tetap memiliki kesempatan untuk mencapai performa optimal tanpa harus melanjutkan pelatihan yang tidak memberikan perbaikan berarti, sehingga dapat menghemat waktu komputasi dan mengurangi risiko overfitting. Kombinasi parameter yang akan diimplementasikan pada rancangan eksperimen dapat dilihat pada Tabel 3.4. Eksperimen ini dilakukan dengan tujuan mendapatkan akurasi dan robustness model yang lebih baik.

Tabel 3.4 Parameter Eksperimen Regularisasi dan Scheduler

ID Eksperi-men	Model	Epoch	Batch Size	Lear-ning Rate	Weight Decay	Learning Rate Scheduler
7	VideoMAE	50	4	1e-5	0.01	linear
8	VideoMAE	50	4	5e-5	0.01	linear
9	TimeSformer	50	4	1e-5	0.01	linear
10	TimeSformer	50	4	5e-5	0.01	linear

d. Evaluasi

Dilakukan *evaluasi* performa model hasil *training* dengan *fine-tuning* menggunakan data pada *test set*. Evaluasi dilakukan dengan mengukur matriks evaluasi utama, yaitu accuracy, precision, recall, dan f1-score. Perhitungan matriks evaluasi dijelaskan pada subbab 2.7

3.1.5 Analisis Hasil

Dalam tahap ini dilakukan analisis kualitatif terhadap kinerja masing-masing model untuk mengidentifikasi kelebihan dan kekurangan masing-masing arsitektur model. Analisis kuantitatif juga dilakukan dengan membandingkan matriks evaluasi.

3.1.6 Kesimpulan

Tahap terakhir adalah menarik kesimpulan berdasarkan hasil evaluasi dan analisis performa model. Kesimpulan akan dikaitkan kembali dengan rumusan masalah yang telah ditetapkan di awal penelitian guna menjawab tujuan penelitian secara menyeluruh. Selain itu, pada tahap ini juga direncanakan penyampaian rekomendasi untuk pengembangan penelitian selanjutnya serta potensi penerapan model dalam konteks praktis di lingkungan pendidikan.

3.2 Alat dan Bahan Penelitian

Alat dan bahan penunjang yang digunakan selama penelitian berlangsung berupa perangkat keras dan perangkat lunak. Seluruh proses pelatihan model dilakukan dalam lingkungan komputasi *cloud* pada perangkat lunak Google Colaboratory dan Kaggle Notebook. Untuk mengakses perangkat lunak, digunakan perangkat keras berupa laptop. Pengambilan data digunakan menggunakan tiga kamera. Adapun keterangan perangkat keras dan perangkat lunak tersebut adalah sebagai berikut:

1. Perangkat Keras
    a) Laptop ASUS Vivobook S, dengan spesifikasi:
        - Processor 12th Gen Intel Core i7-12700H,
        - RAM 16 GB,
        - SSD 512 GB,

- Sistem operasi Windows 11 Home Single Language.

b) Kamera CCTV, dengan spesifikasi:
   - Resolusi 1920 x 1080 piksel
   - 30 FPS

c) Kamera Ponsel, dengan spesifikasi:
   - Resolusi 1280 × 720 piksel
   - 30 FPS

d) Kamera Digital *Handycam*
   - Resolusi 1920 x 1080 piksel
   - 30 FPS

2. Perangkat Lunak:
   a) ClipChamp,
   b) Google Chrome,
   c) Google Colaboratory, dengan spesifikasi:
      - GPU NVDIA 4T 12 GB,
      - RAM 12 GB.
   d) HuggingFace Transformers,
   e) Kaggle Notebook, dengan spesifikasi:
      - GPU T4 15 GB,
      - RAM 30 GB.
   f) Microsoft Edge,
   g) Python,
   h) Pytorchvideo,
   i) Scikit-learn,
   j) Torchvision.