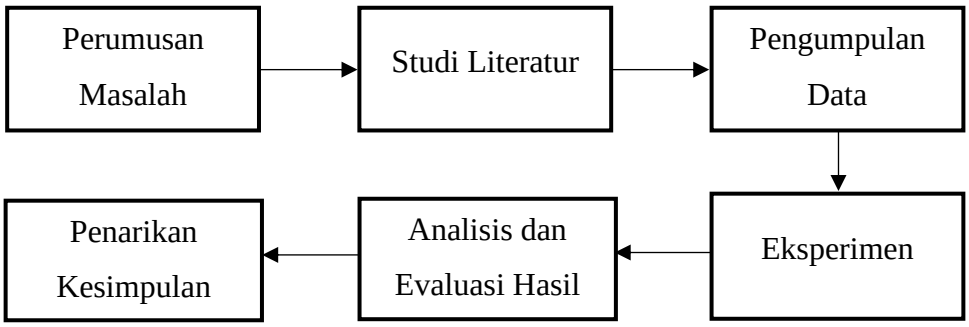
BAB III
METODOLOGI PENELITIAN

Pada bab ini dijelaskan mengenai rancangan sistem yang akan dibuat meliputi tahapan penelitian yang diawali dengan perumusan masalah sampai tahapan akhir penarikan kesimpulan dan juga kebutuhan perangkat untuk mendukung penelitian.

3.1 Desain Penelitian

Pada penelitian ini terdapat beberapa tahapan pengerjaan. Tahapan-tahapan ini dimulai dari merumuskan masalah, studi literatur, pengumpulan data, implementasi, evaluasi, dan tahapan terakhir yaitu penarikan kesimpulan. sebagaimana ditunjukkan pada Gambar 3.1.



Gambar 3.1 Desain Penelitian

Adapun penjelasan dari desain penelitian yang ada pada Gambar 3.1. adalah sebagai berikut.

1) Rumusan masalah, merupakan mempertanyakan bagaimana penelitian *action recognition* ini dilakukan dan bagaimana implementasi metode Transformer terhadap *ation recognition*, di mana nantinya jawaban dari pertanyaan-pertanyaan ini lah yang akan menjadi hasil penelitian.
2) Studi literatur adalah serangkaian kegiatan yang berkenaan dengan metode pengumpulan data pustaka, membaca dan mencatat, serta mengolah bahan penelitian (Zed, 2008). Studi literatur yang dikumpulkan dan dipelajari adalah mengenai topik *computer vision*, pengolahan citra digital, *action recognition*, dan metode Transformer. Selain itu terdapat konsep dan gagasan terhadap studi kasus yang

diangkat dari berbagai penelitian sebelumnya yang relevan dengan *action recognition*.

3) Pengumpulan data ini akan digunakan sebagai set data dalam penelitian *action recognition*. Dimana nantinya set data ini yang akan diproses melalui tahap *praprocessing*, dan dibagi menjadi data *training* dan data *testing* untuk menguji model yang sudah dibuat.
4) Eskperimen dilakukan sesuai dengan desain eksperimen yang sudah dirancang, mulai dari *praprocessing input* data, *construct model*, *training*, dan *testing*. Pada penelitian ini terdapat beberapa rancangan *construct* model berbeda yang akan diuji pada data *testing* yang telah dipersiapkan sebelumnya.
5) Analisis dan evaluasi hasil, pada tahap ini dilakukan analisis dan evaluasi terhadap hasil dari eksperimen yang telah didapat pada tahap implementasi. Model *action recognition* dengan metode Transformer yang merupakan hasil eksperimen dievaluasi berdasarkan tingkat hasil kerjanya sehingga diketahui model mana yang memiliki hasil terbaik.
6) Penarikan kesimpulan diambil berdasarkan hasil dari analisis pada tahap evaluasi. Kesimpulan ini dibuat bertujuan untuk dapat melihat kinerja sistem yang telah dibuat. Penarikan kesimpulan juga mengacu pada rumusan masalah yang dibuat tentang bagaimana penelitian *action recotion* dengan metode Transformer ini dilakukan, dan bagaimana unjuk kerjanya.

3.2 Data

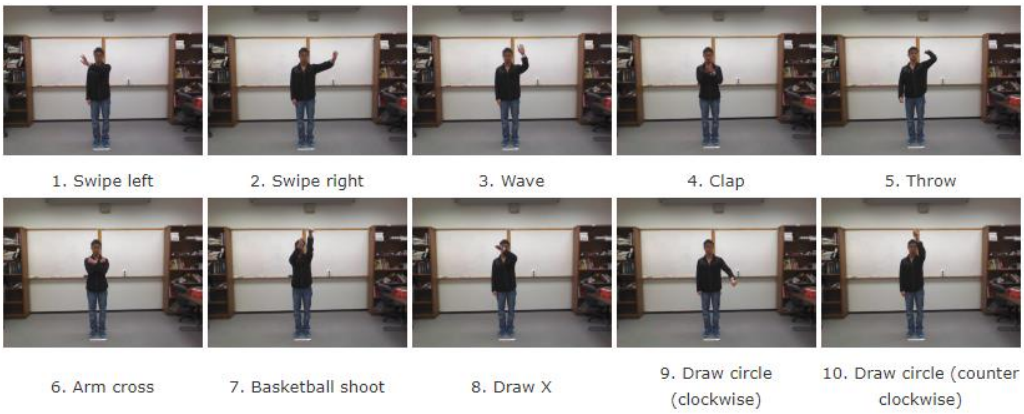
Pengumpulan Data pada penelitian ini menggunakan set data *action recognition* yang dapat diunduh secara publik. Set data ini diambil dari penelitian University of Texas at Dallas (UTD). Diberi nama UTD-MHAD, atau Multimodal Human Action Dataset yang terdiri dari 861 video.

Dataset ini dikumpulkan menggunakan sensor Microsoft Kinect dan Sensor Inersia. Dataset UTD-MHAD ini terdiri dari 27 aksi yang berbeda, dan dilakukan oleh 8 subjek (4 perempuan dan 4 laki-laki). Setiap subjek mengulangi setiap gerakan sebanyak 4 kali. Sehingga dataset yang terkumpul ada 864 video, namun ada 3 video yang rusak sehingga video yang diambil hanya 861 video. Tabel 3.1

menunjukkan 27 nama aksi di dalam video, dan Gambar 3.2 merupakan contoh set data UTD-MHAD.

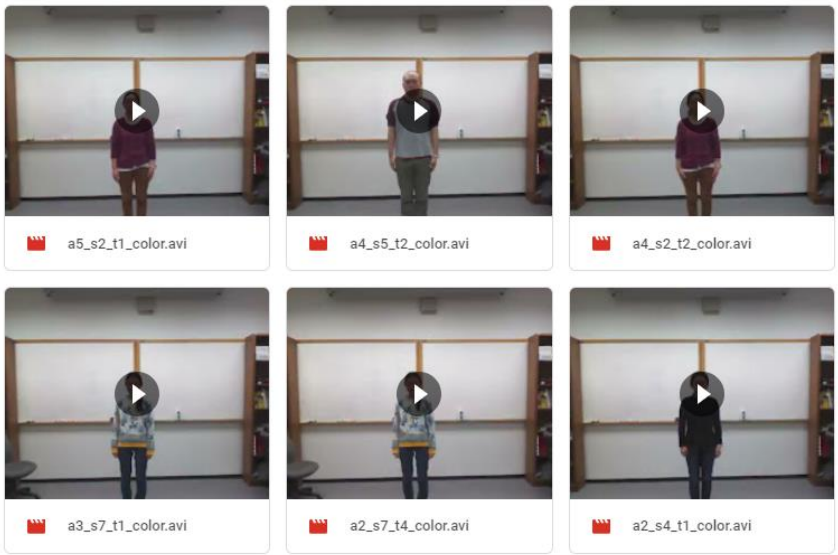
Tabel 3.1 Nama Aksi dalam Video

Nomor	**Aksi**
1	*Swipe left*
2	*Swipe right*
3	*Wave*
4	*Clap*
5	*Throw*
6	*Arm cross*
7	*Basketball shoot*
8	*Draw X*
9	*Draw circle (clockwise)*
10	*Draw circle (counter clockwise)*
11	*Draw triangle*
12	*Bowling*
13	*Boxing*
14	*Baseball swing*
15	*Tennis swing*
16	*Arm curl*
17	*Tennis serve*
18	*Push*
19	*Knock*
20	*Catch*
21	*Pickup and throw*
22	*Jog*
23	*Walk*
24	*Sit to stand*
25	*Stand to sit*
26	*Lunge*
27	*Squat*



Gambar 3.2 Contoh Set Data UTD-MHAD

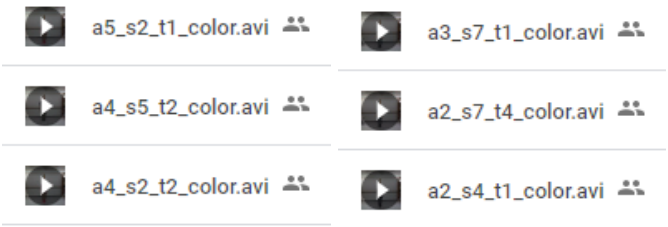
Set data UTD-MHAD ini memiliki multimodal data yang terdiri dari video *Red Green Blue* (RGB), *dept videos*, *skeleton positions* (posisi kerangka), dan sinyal inersia dari kamera Kinect dan sensor inersia. Namun pada penelitian *action recognition* ini set data yang diambil hanya jenis video RGB. Setiap video memiliki panjang yang berbeda-beda mulai dari 3 – 8 detik. Jenis file video ini disimpan dalam bentuk .avi. Berikut contoh data video yang telah dikumpulkan dapat dilihat pada Gambar 3.3.



Gambar 3.3 Contoh Set Data UTD-MHAD yang Terunduh

Video yang diunduh sudah berlabel di *title* video sehingga sudah siap untuk dilakukan proses berikutnya. Adapun konversi penamaan file adalah "ai_sj_tk_modality". Dimana “ai” adalah singkatan dari nomor tindakan, “sj”

adalah singkatan dari nomor subjek, dan “tk” adalah singkatan dari pengulangan keberapa. Sedangkan jenis modalitasnya ditandai dengan kata “color” yang menunjukkan video ini merupakan jenis video RGB. Adapun contoh penamaan file pada video dapat dilihat pada Gambar 3.4.

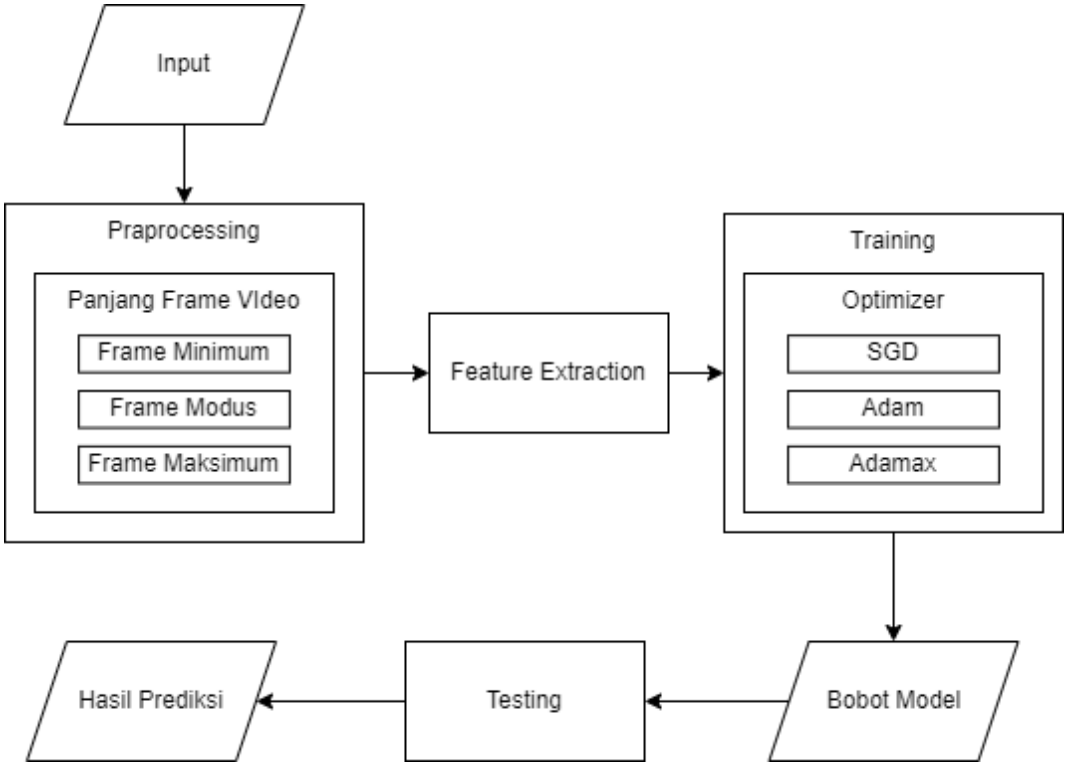


Gambar 3.4 Contoh Penamaan File pada Video

Pada penelitian ini label yang diambil hanya kode untuk aksinya saja. Sehingga yang diambil hanya angka paling depan dari nama *file*, yang nantinya akan ada angka 1 – 27. Angka 1 – 27 ini akan menunjukkan nama dari aksi yang ada di dalam video.

3.3 Eksperimen

Pada tahapan ini akan dilakukan eksperimen sesuai dengan desain eksperimen pada Gambar 3.5. Pada penelitian ini terdapat beberapa variasi arsitektur yang akan diuji pada set data yang telah dipersiapkan sebelumnya



Gambar 3.5 Desain Eksperimen

1) Input

Input data merupakan set data UTD-MHAD yang sudah diunduh sebelumnya. Set data UTD-MHAD ini merupakan set data Multimodal, yang memiliki 4 jenis data. Namun pada penelitian kali ini, set data yang digunakan hanya video dengan jenis RGB.

Sebelum masuk dalam tahap *praprocessing*, set data ini perlu dilihat panjang *frame*nya pada masing-masing video, karena setiap video dalam set data memiliki panjang yang berbeda-beda. Panjang *frame* yang diambil ada 3 kategori, yaitu panjang *frame* minimum, modus, dan maksimum.

2) *Praprocessing*

Kemudian pada tahap ini, dilakukan praproses pada set inputan supaya data mampu diproses dengan baik oleh model dan diharapkan mampu menghasilkan unjuk kerja yang lebih optimal. Adapun praproses yang akan dilakukan adalah menentukan panjang *frame*, *cropping*, dan *resizing*.

a) Penentuan Panjang Maksimal *Frame*

Tahap pertama pada tahap *praprocessing* adalah menentukan panjang *frame* set data yang akan diproses. Pada tahap persiapan data, telah ditentukan panjang *frame* yang akan digunakan, yaitu panjang *frame* minimum, modus, dan maksimum. Sehingga saat masuk pada proses selanjutnya jumlah frame setiap video akan sama.

b) Penyamarataan Panjang *Frame* Video Masukan

Setelah panjang *frame* video input ditentukan, langkah selanjutnya adalah membuat *padding* dan membuat pemotong panjang *frame*. *Padding* pada tahap *preprocessing* berfungsi sebagai menambah *frame* kosong untuk panjang *frame* yang kurang dari *variabel* panjang maksimal *frame* yang dimasukkan. Sedangkan proses pemotongan panjang *frame* digunakan untuk memotong panjang *frame* yang melebihi dari panjang maksimal *frame* yang dimasukkan.

c) *Cropping*

Pada tahapan ini dilakukan pemotongan pada sisi tepi kanan dan kiri citra untuk mengurangi dimensi latar belakang pada citra. Sehingga bentuk citra akan menjadi kotak atau *square*.

d) *Resizing*

Pada tahapan ini dilakukan perubahan ukuran pada citra untuk mempermudah dan mempercepat proses pengenalan wajah sesuai dengan kebutuhan *input* tanpa mengurangi kualitas fitur secara berarti. Citra akan dilakukan penurunan resolusi menjadi 128 x 128 piksel.

e) *Feature Extraction*

Pada proses berikutnya dilakukan proses *feature extraction*. *Feature extraction* adalah teknik pengambilan ciri/*feature* dari suatu bentuk yang nantinya nilai yang didapatkan akan dianalisis untuk proses selanjutnya. *Feature extraction* pada penelitian ini mengimplementasikan pre-trained model yang telah disediakan oleh Keras yaitu Dense Convolutional Networks (DenseNet). Model DenseNet dibangun dengan struktur dimana masing-masing layer akan terhubung dengan layer selanjutnya (G. Huang dkk., 2017).

Arsitektur DenseNet menggunakan *dense connection* untuk mengurangi terjadinya *overfitting* ketika jumlah datasetnya kecil. Model DenseNet yang digunakan pada penelitian ini adalah DenseNet121. Tahap ekstraksi fitur juga dilakukan baik pada data *training* dan data *testing*.

f) *Training*

Pada tahap *training* ini akan menghasilkan bobot model yang disimpan untuk digunakan pada proses testing. Model dibangun dengan metode yang digunakan pada penelitian ini yaitu metode Transformer. Pada proses ini, akan ada beberapa variasi atau *tuning* parameter yang dilakukan.

Tuning parameter yang dilakukan diantaranya adalah percobaan dengan beberapa *optimizer*, dan *activation function*. Adapun *optimizer* yang akan diujicobakan adalah SGD, Adam, dan Adamax. Sedangkan untuk *activation function* yang akan diujicobakan adalah Softmax dan ReLu.

g) *Testing*

Model dengan akurasi terbaik pada tahap *training* akan digunakan pada tahap *testing*, untuk mendapatkan hasil berupa *loss* dan akurasi pada penelitian ini. Data *testing* telah disiapkan sebelumnya, dan sudah melalui tahap *praprocessing*.

h) Hasil Prediksi

Hasil prediksi ini didapat dari proses *testing* menggunakan bobot model hasil dari *training* dengan data *testing*. Hasil prediksi ini berupa *loss* dan akurasi.

3.4 Evaluasi dan Analisis

Untuk mengevaluasi hasil prediksi pada permasalahan penelitian ini, digunakan suatu pengukuran yang dinamakan pengukuran *confusion matrix*. Pengukuran ini dilihat dari akurasi hasil prediksi pada model saat pengujian terhadap data test.

3.5 Lingkungan Komputasi

Lingkungan komputasi pada penelitian ini adalah seperangkat komputer dan beberapa perangkat lunak pendukung. Pada penelitian ini, spesifikasi perangkat keras komputer sebagai berikut:

1) RAM 8GB
2) SSD 120GB
3) *Processor* Intel Pentium N4200
4) *Mouse* dan *Keyboard*
5) Intel HD Graphics 505

Adapun perangkat lunak pada komputer tersebut sebagai berikut:

1) Microsoft Windows 10
2) Python 3.8
3) Google Colab