



**SISTEM PENGENALAN AKTIVITAS MANUSIA MENGGUNAKAN
LONG SHORT-TERM MEMORY DAN *MEDIAPIPE***

TUGAS AKHIR



**Program Studi
S1 TEKNIK KOMPUTER**

**UNIVERSITAS
Dinamika**

Oleh:

FILBERT DANIEL TANUGRAHA

18410200007

FAKULTAS TEKNOLOGI DAN INFORMATIKA

UNIVERSITAS DINAMIKA

2022

**SISTEM PENGENALAN AKTIVITAS MANUSIA MENGGUNAKAN
LONG SHORT TERM-MEMORY DAN *MEDIAPIPE***

TUGAS AKHIR

**Diajukan sebagai salah satu syarat untuk menyelesaikan
Program Sarjana Teknik**



UNIVERSITAS
Dinamika

Disusun Oleh:

Nama : Filbert Daniel Tanugraha
NIM : 18410200007
Program Studi : S1 Teknik Komputer

**FAKULTAS TEKNOLOGI DAN INFORMATIKA
UNIVERSITAS DINAMIKA**

2022

TUGAS AKHIR

SISTEM PENGENALAN AKTIVITAS MANUSIA MENGGUNAKAN *LONG SHORT-TERM MEMORY* DAN *MEDIAPIPE*

Dipersiapkan dan disusun oleh:

Filbert Daniel Tanugraha

NIM : 18410200007

Telah diperiksa, dibahas dan disetujui oleh Dewan Pembahas

Pada: 12 Januari 2022

Susunan Dewan Pembahas

Pembimbing:

I. Heri Pratikno, M.T., MTCNA., MTCRE.

NIDN: 0716117302

II. Musayyanah, S.ST., M.T.

NIDN: 0730069102

Pembahas:

Weny Indah Kusumawati, S.Kom., M.MT.

NIDN: 0721047201



Digitally signed by
Universitas Dinamika
Date: 2022.01.13 13:19:05
+07'00'



Digitally signed by Musayyanah
DN: cn=Musayyanah, o=Universitas
Dinamika, ou=SI Teknik Komputer,
email=musayyanah@dinamika.ac.id,
c=ID
Date: 2022.01.14 07:47:41 +07'00'
Adobe Acrobat Reader version:
2021.011.20039



Weny Indah Kusumawati
cn=Weny Indah Kusumawati,
o=Teknologi dan Informatika,
Undika, ou=Teknik Komputer,
email=weny@dinamika.ac.id,
c=ID
2022.01.14 08:07:14 +07'00'

Tugas Akhir ini telah diterima sebagai salah satu persyaratan

untuk memperoleh gelar Sarjana



Digitally signed by
Universitas Dinamika
Date: 2022.01.21
09:56:46 +07'00'

Tri Sagirani, S.Kom., M.MT.

NIDN: 0731017601

Dekan Fakultas Teknologi dan Informatika

UNIVERSITAS DINAMIKA



"Promise me you'll survive. That you won't give up, no matter what happens. No matter how hopeless."

UNIVERSITAS
Dinamika



Dipersembahkan untuk Ayah dan Mama yang telah memberi dukungan dan doa yang selalu diberikan kepada saya. Dan teruntuk semua orang yang juga selalu membantu secara langsung maupun tidak langsung agar saya tidak mudah menyerah dan menjadi orang yang lebih baik.

UNIVERSITAS
Dinamika

PERNYATAAN
PERSETUJUAN PUBLIKASI DAN KEASLIAN KARYA ILMIAH

Sebagai mahasiswa Universitas Dinamika, Saya :

Nama : **Filbert Daniel Tanugraha**

NIM : **18410200007**

Program Studi : **S1 Teknik Komputer**

Fakultas : **Fakultas Teknologi dan Informatika**

Jenis Karya : **Laporan Tugas Akhir**

Judul Karya : **SISTEM PENGENALAN AKTIVITAS MANUSIA
MENGUNAKAN *LONG-SHORT TERM MEMORY*
DAN *MEDIAPIPE***

Menyatakan dengan sesungguhnya bahwa :

1. Demi pengembangan Ilmu Pengetahuan, Teknologi dan Seni, Saya menyetujui memberikan kepada Universitas Dinamika Hak Bebas Royalti Non-Eksklusif (*Non-Exclusive Royalty Free Right*) atas seluruh isi/sebagian karya ilmiah Saya tersebut diatas untuk disimpan, dialihmediakan, dan dikelola dalam bentuk pangkalan data (*database*) untuk selanjutnya didistribusikan atau dipublikasikan demi kepentingan akademis dengan tetap mencantumkan nama Saya sebagai penulis atau pencipta dan sebagai pemilik Hak Cipta.
2. Karya tersebut diatas adalah hasil karya asli Saya, bukan plagiat baik sebagian maupun keseluruhan. Kutipan, karya, atau pendapat orang lain yang ada dalam karya ilmiah ini semata-mata hanya sebagai rujukan yang dicantumkan dalam Daftar Pustaka Saya.
3. Apabila dikemudian hari ditemukan dan terbukti terdapat tindakan plagiasi pada karya ilmiah ini, maka Saya bersedia untuk menerima pencabutan terhadap gelar kesarjanaan yang telah diberikan kepada Saya.

Surabaya, 12 Januari 2022



Filbert Daniel Tanugraha
NIM : 18410200007

ABSTRAK

Sistem pengenalan aktivitas manusia atau *Human Activity Recognition* memiliki banyak tantangan di saat ini, seperti mengenal gerakan aktivitas manusia. Salah satu aktivitas manusia dalam bidang kesehatan atau olahraga. Ada beberapa gerakan olahraga seperti *Tree Pose*, *T-Pose*, *Warrior II Pose* yang perlu dideteksi oleh *MediaPipe* agar gerakan tersebut dapat dikenali oleh komputer. *MediaPipe* menyediakan *keypoints* tubuh manusia secara *real-time* dengan akurasi tinggi bahkan bisa berjalan pada (*Central Processing Unit*) CPU. *MediaPipe* mendeteksi kerangka postur tubuh atau *body skeletons* sebanyak 33 *keypoints* pada *frame* video yang diamati. Untuk klasifikasi aktivitas manusia menggunakan *Long Short-Term Memory* (LSTM) dengan memanfaatkan *keypoints* dari urutan video tersebut. *Dataset* menggunakan *Berkeley Multimodal Human Action Database* (MHAD). Pada Tugas Akhir ini dibatasi hanya tiga gerakan yang baru yaitu pada HAR (*Human Activity Recognition*) dan telah berhasil dibuat yaitu (*T-Pose*, *Warrior II Pose* dan *Tree Pose*), pengujian klasifikasi gerakan dilakukan oleh empat subjek dengan lokasi yang sama menggunakan kamera *handphone* dalam pengambilan *video* menggunakan aplikasi *DroidCam* agar lebih maksimal dalam pendeteksian *keypoint*. Pada Tugas Akhir ini hasil *training* didapatkan akurasi sebesar 91% dan *loss* sebesar 0.29. Rata-rata FPS (*Frame per second*) didapatkan sebesar 10-30 FPS dengan rata-rata waktu komputasi untuk setiap proses *training* kurang lebih 10-12 menit.

Kata Kunci: *Long Short-Term Memory, Human Activity Recognition, Machine Learning, MediaPipe*

KATA PENGANTAR

Segala puji bagi Tuhan, karena dengan rahmatNya penulis dapat menyelesaikan penyusunan laporan Tugas Akhir yang berjudul “Sistem Pengenalan Aktivitas Manusia Menggunakan *Long Short Term-Memory* Dan *MediaPipe*”. Laporan Tugas Akhir ini disusun dalam rangka penulisan laporan untuk memperoleh gelar Sarjana Teknik pada Program Studi S1 Teknik Komputer Universitas Dinamika. Pada kesempatan ini, penulis ingin mengucapkan rasa terima kasih kepada pihak-pihak yang memberi dukungan dan masukan dalam menyelesaikan laporan Tugas Akhir ini. Oleh karena itu penulis ingin mengucapkan kepada:

1. Orang Tua dan Saudara-saudara saya tercinta yang telah memberikan dorongan dan bantuan baik moral maupun materi, sehingga penulis dapat menempuh dan menyelesaikan Tugas Akhir ini.
2. Ibu Tri Sagirani, S.Kom., M.MT., selaku Dekan Fakultas Teknologi dan Informatika (FTI) Universitas Dinamika.
3. Bapak Pauladie Susanto, S.Kom., M.T., selaku Ketua Program Studi S1 Teknik Komputer Universitas Dinamika.
4. Ibu Weny Indah Kusumawati, S.Kom., M.MT., selaku Dosen Pembahas atas saran dan masukannya, sehingga penulis dapat menyelesaikan laporan Tugas Akhir ini.
5. Bapak Heri Pratikno, M.T., MTCNA., MTCRE., selaku Dosen pembimbing I yang selalu memberi arahan dan bimbingan dalam menyelesaikan Tugas Akhir beserta laporan ini.
6. Ibu Musayyanah, S.ST., M.T., selaku Dosen Pembimbing II yang juga selalu memberi arahan dan bimbingan dalam menyelesaikan Tugas Akhir beserta laporan ini.
7. Seluruh dosen pengajar Porgram Studi S1 Teknik Komputer yang telah mendidik, memberi motivasi kepada penulis selama masa kuliah di Universitas Dinamika.

8. Teman-teman seperjuangan Teknik Komputer angkatan 2018 dan semua pihak yang terlibat namun tidak dapat penulis sebutkan satu persatu atas bantuan dan dukungannya.
9. Serta semua pihak lain yang tidak dapat disebutkan secara satu per satu, yang telah membantu dalam menyelesaikan Tugas Akhir ini baik secara langsung maupun tidak langsung.

Penulis menyadari bahwa Laporan Tugas Akhir ini jauh dari kata sempurna, masih banyak kekurangan dalam menyusun laporan ini. Oleh karena itu dalam kesempatan ini, penulis meminta maaf apabila dalam laporan Tugas Akhir ini masih banyak kesalahan baik dalam penulisan maupun bahasa yang digunakan. Penulis juga memerlukan kritik dan saran dari para pembaca yang sifatnya membangun untuk kesempurnaan laporan yang telah penulis susun.

Surabaya, 12 Januari 2022

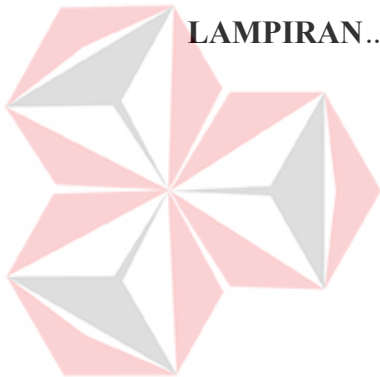


UNIVERSITAS
Dinamika
Penulis

DAFTAR ISI

	Halaman
ABSTRAK	vii
KATA PENGANTAR.....	viii
DAFTAR ISI.....	x
DAFTAR GAMBAR.....	xii
DAFTAR TABEL.....	xiv
DAFTAR LAMPIRAN	xv
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah	2
1.4 Tujuan	3
1.5 Manfaat	3
BAB II LANDASAN TEORI	4
2.1 Aktivitas Gerakan Olahraga.....	4
2.1.1 <i>Tree Pose</i>	4
2.1.2 <i>T-Pose</i>	5
2.1.3 <i>Warrior II Pose</i>	5
2.2 <i>MediaPipe</i>	6
2.3 <i>Dataset</i>	6
2.4 <i>Long Short-Term Memory</i>	6
2.4.1 <i>Forget Gate</i>	7
2.4.2 <i>Input Gate</i>	8
2.4.3 <i>Cell State</i>	8
2.4.4 <i>Output Gate</i>	9
2.5 <i>DroidCam</i>	9
2.6 <i>Google Colab</i>	9
2.7 <i>Jupyter Notebook (Anaconda3)</i>	10
BAB III METODOLOGI PENELITIAN	11
3.1 Blok Diagram Sistem.....	11
3.2 Pengumpulan <i>Keypoints</i>	11

3.3 Pengumpulan Klasifikasi Gerakan.....	12
3.4 <i>Flowchart Human Activity Recognition</i>	13
BAB IV HASIL DAN PEMBAHASAN.....	15
4.1 Pengujian Hasil Training <i>Long Short-Term Memory</i>	15
4.1.1 Perhitungan <i>Training</i> Dan <i>Step</i>	16
4.1.2 Accuracy Dan Validation_Accuracy.....	16
4.1.3 <i>Train_Loss, Validation_Loss, Dan Batch_Train_Loss</i>	19
4.2 Pengujian <i>Keypoints</i>	21
4.3 Pengujian Klasifikasi Gerakan.....	25
BAB V PENUTUP.....	31
5.1 Kesimpulan	31
5.2 Saran	31
DAFTAR PUSTAKA	33
LAMPIRAN.....	35



UNIVERSITAS
Dinamika

DAFTAR GAMBAR

	Halaman
Gambar 2.1 <i>Tree Pose</i>	4
Gambar 2.2 <i>T-Pose</i>	5
Gambar 2.3 <i>Warrior II Pose</i>	5
Gambar 2.4 Mediapipe untuk <i>Pose estimation</i>	6
Gambar 2.5 Komponen <i>Long Short-Term Memory</i>	7
Gambar 2.6 Aplikasi droidcam	9
Gambar 2.7 Tampilan <i>Google Colab</i>	10
Gambar 2.8 Tampilan <i>Jupyter</i>	10
Gambar 3.1 Blok diagram sistem program	11
Gambar 3.2 Index <i>keypoints</i> pada <i>Mediapipe</i>	12
Gambar 3.3 Flowchart <i>human activity recognition</i>	13
Gambar 4.1 Hasil grafik (a)underfitting, (b) overfitting, (c) ideal balance.....	15
Gambar 4.2 Hasil grafik plot <i>train_accuracy</i>	16
Gambar 4.3 Hasil grafik plot <i>val_acc</i>	17
Gambar 4.4 Hasil grafik plot <i>batch_train_acc</i>	17
Gambar 4.5 Hasil grafik plot <i>batch_val_acc</i>	18
Gambar 4.6. Hasil grafik plot pada <i>train_loss</i>	19
Gambar 4.7 Hasil grafik <i>batch_train_loss</i>	19
Gambar 4.8 Hasil grafik plot <i>val_loss</i>	20
Gambar 4.9 Hasil grafik <i>batch_val_loss</i>	20
Gambar 4.10 Subjek 1 melakukan gerakan <i>T-Pose</i>	27
Gambar 4.11 Subjek 1 melakukan gerakan <i>Tree Pose</i>	27
Gambar 4.12 Subjek 1 melakukan gerakan <i>Warrior II Pose</i>	27
Gambar 4.13 Subjek 2 melakukan gerakan <i>Warrior II Pose</i>	28
Gambar 4.14 Subjek 2 melakukan gerakan <i>Tree Pose</i>	28
Gambar 4.15 Subjek 2 melakukan gerakan <i>T-Pose</i>	28
Gambar 4.16 Subjek 3 melakukan gerakan <i>Warrior II Pose</i>	29
Gambar 4.17 Subjek 3 melakukan gerakan <i>T-Pose</i>	29
Gambar 4.18 Subjek 3 melakukan gerakan <i>Tree Pose</i>	29
Gambar 4.19 Subjek 4 melakukan gerakan <i>Tree Pose</i>	30

Gambar 4.20 Subjek 4 melakukan gerakan <i>T-Pose</i>	30
Gambar 4.21 Subjek 4 melakukan gerakan <i>Warrior II Pose</i>	30



UNIVERSITAS
Dinamika

DAFTAR TABEL

	Halaman
Tabel 4.1 Persen perbandingan accuracy training dan validasi_accuracy.....	18
Tabel 4.2 Regresi perbandingan training loss dan validation	21
Tabel 4.3 Pengujian keypoints.	21
Tabel 4.4 Keberhasilan klasifikasi gerakan	26



UNIVERSITAS
Dinamika

DAFTAR LAMPIRAN

	Halaman
Lampiran 1 Perbandingan hasil Training dan Validation	35
Lampiran 2 Source Code Program.....	43
Lampiran 3 Source Training Program.....	51
Lampiran 4 Pengujian Keseluruhan Klasifikasi Gerakan	53
Lampiran 5 Hasil Turnitin.....	73



UNIVERSITAS
Dinamika

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pengenalan aktivitas manusia, juga dikenal sebagai *Human Activity Recognition*, memainkan peran penting dalam interaksi manusia. Hal ini karena pengenalan perilaku tersebut memberikan informasi tentang kepribadian dan keadaan psikologis individu yang sulit diperoleh. Pengantar aktivitas manusia adalah salah satu topik ilmiah *Computer Vision* dan pembelajaran mesin. Sebuah studi oleh Michalis dkk menunjukkan bahwa sistem pengenalan aktivitas gerakan diperlukan untuk mengidentifikasi karakteristik perilaku manusia, seperti robot, sistem CCTV, dan interaksi manusia-komputer. (Michalis Vringkas, 2015)

Ada dua metode pengenalan aktivitas manusia yang telah dilakukan oleh penelitian sebelumnya yaitu pengenalan aktivitas manusia berdasarkan gambar visual dan berdasarkan *wearable* sensor (Jian Sun, 2018). Analisis pergerakan manusia dalam *Computer Vision* meliputi deteksi objek, *tracking*, pengenalan pergerakan manusia. Metode Kedua yaitu gambar visual dengan kamera namun klasifikasi dan *Keypoints* belum dapat diklasifikasikan secara *Real-Time* serta metode tersebut masih menggunakan *Convolutional neural network* (CNN). *Long Short-Term Memory* (LSTM) dirancang untuk bekerja secara berbeda dari CNN karena LSTM biasanya digunakan untuk memproses dan membuat prediksi yang diberikan urutan data (sebaliknya, CNN dirancang dalam data dan bekerja dengan baik pada gambar dan ucapan) (Geoffrey, 2019). Pada Tugas Akhir ini, menggunakan metode gambar visual dengan kamera, untuk mendeteksi aktivitas gerakan olahraga secara *Real Time* menggunakan *MediaPipe* dan klasifikasi tersebut menggunakan *Long Short-Term Memory* seperti gerakan *Warrior II Pose*, *T Pose*, dan *Tree Pose*.

Manfaat dari Tugas Akhir ini adalah membuat sistem pengenalan untuk gerakan olahraga yang digunakan saat di rumah, sehingga dapat dilihat gerakan olahraga yang benar. Sistem ini sangat tepat digunakan pada saat pandemi, dimana seseorang tidak perlu keluar rumah ke tempat gym, cukup dengan menggunakan sistem pengenalan ini.

Oleh karena itu, Tugas Akhir ini disampaikan sebagai berdasarkan latar belakang di atas, yaitu pengenalan Aktivitas Manusia (*Human Activity Recognition*) menggunakan *Long Short-Term Memory* dan *MediaPipe*. Langkah pertama dicapai dengan menggunakan kamera atau webcam yang digunakan, lalu *MediaPipe* menampilkan kerangka postur tubuh atau *body skeletons* sebanyak (33 *keypoints*). Setelah itu mengamati satu *frame* dalam sebuah video atau gambar. Kemudian menganalisis gerakan dari tubuh secara berubah dan membuat prediksi dilakukan dengan menggunakan *Long Short-Term Memory* (LSTM). *Keypoints* dari urutan *frame* dikirim ke LSTM untuk klasifikasi memprediksi aktivitas yang sesuai.

1.2 Rumusan Masalah

Berdasarkan latar belakang diatas, maka dapat dirumuskan masalah pada Tugas Akhir ini sebagai berikut:

1. Bagaimana mendeteksi *keypoints* badan manusia menggunakan *MediaPipe*?
2. Bagaimana mengklasifikasikan objek Label *Dataset* pada aktivitas gerakan badan manusia?
3. Bagaimana menentukan jumlah *Epochs* untuk meningkatkan akurasi pada proses *Training*?

1.3 Batasan Masalah

Dalam pembuatan Tugas Akhir ini, pembahasan masalah dibatasi beberapa hal berikut:

1. Dataset sistem pengenalan Aktivitas Manusia berasal dari *Berkeley Multimodal Human Action Database (MHAD)*.
2. Pembatasan hanya 3 gerakan yang menjadi *Dataset* yaitu *T-Pose*, *Warrior II Pose*, *Tree Pose*.
3. Pencahayaan menggunakan pencahayaan ruangan atau pencahayaan merata serta posisi sudut yang diambil harus linear dengan kamera agar dapat dikenali aktivitas yang dilakukan.

1.4 Tujuan

Berdasarkan latar belakang dan rumusan masalah diatas, mendapatkan tujuan pada Tugas Akhir ini sebagai berikut:

1. Mendeteksi *keypoints* badan manusia menggunakan *MediaPipe*.
2. Mengklasifikasikan aktivitas gerakan tubuh manusia menggunakan LSTM (*Long Short-Term Memory*).
3. Menentukan jumlah *Epochs* untuk meningkatkan meningkatkan akurasi Training.

1.5 Manfaat

Adapun dari Tugas Akhir ini dapat diperoleh manfaat sebagai berikut:

1. Dengan Menggunakan *MediaPipe* dapat membantu mengenali aktivitas tiap tubuh titik *keypoints* pergerakan di dalam setiap *Frame*.
2. Dapat mengklasifikasikan setiap gerakan video yang telah di *Training* dalam LSTM (*Long short-Term Memory*).
3. Hasil dari Tugas Akhir ini, dapat diterapkan untuk aktivitas gerakan olahraga yang berguna bagi masyarakat dalam kehidupan sehar-hari khususnya, di era pandemi dimana dapat dilakukan di rumah.
4. Dengan Tugas Akhir ini dapat memberi kontribusi terhadap perkembangan *Human Activity Recognition*.

BAB II

LANDASAN TEORI

2.1 Aktivitas Gerakan Olahraga

Dalam dunia olahraga, ada istilah yang disebut dengan *bodyweight exercise* atau latihan beban tubuh. Karena itu, olahraga yang mudah dilakukan di rumah bergantung pada berat dan gravitasi seseorang. Dengan melakukan olahraga teratur di rumah, seseorang dapat membangun kekuatan dan keseimbangan. Latihan ini bisa dilakukan di rumah dan bisa dilakukan oleh siapa saja, termasuk pemula yang baru memulai. (Azmi, 2020)

Olahraga didefinisikan sebagai aktivitas fisik yang sistematis dan terencana berdasarkan pada mematuhi aturan yang berlaku serta memperbaiki tubuh. Banyak gerakan olahraga yang dapat dilakukan tanpa bantuan peralatan khusus. Apalagi jika seseorang tinggal di rumah atau memiliki kondisi yang mengharuskan seseorang bekerja dari rumah (WFH). Dari semua gerakan yang dapat seseorang lakukan, seperti *push-up*, *squat*, *burpe*, *push-up bangku*. (Wicaksono, 2020)

2.1.1 Tree Pose

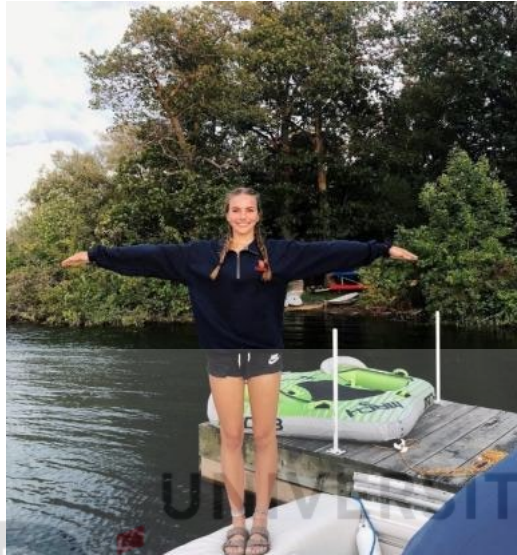
Tree Pose atau yang disebut *Vrksasana* adalah salah satu Pose berdiri di yoga dengan bertumpu 180 derajat di satu lutut dan 35 derajat di kanan atau 335 derajat kiri lutut, sebagaimana tampak pada gambar 2.1. (Fimela, 2014)



Gambar 2.1 *Tree Pose*
(Sumber: Fimela, 2014)

2.1.2 *T-Pose*

T-Pose atau *Pose* referensi *T-Pose* digunakan untuk sebagai pengganti animasi yang belum selesai, terutama dalam gerakan yoga cara yang dilakukan oleh gerakan *T-Pose* adalah bentangkan sikut 180 derajat kemudian untuk kedua bahu dibentangkan 90 derajat dan kedua lutut diluruskan 180 derajat, seperti terlihat pada gambar 2.2 di bawah ini. (Fimela, 2014)



Gambar 2.2 *T-Pose*
(Sumber: Fimela, 2014)

2.1.3 *Warrior II Pose*

Gerakan *Warrior II Pose* adalah gerakan yang digunakan untuk menguatkan kekuatan, kestabilan dan konsentrasi. Untuk memulai gerakan ini bisa dimulai dari sikut dibentangkan 180 derajat, kedua bahu lurus 90 derajat dan 180 derajat kaki kiri dan sekitar 90 derajat di sisi lutut lainnya, terlihat pada gambar 2.3 di bawah ini. (Fimela, 2014)



Gambar 2.3 *Warrior II Pose*
(Sumber: Fimela, 2014)

2.2 MediaPipe

Mediapipe adalah kerangka kerja yang terutama digunakan untuk menghasilkan audio atau video. Dengan bantuan *framework MediaPipe*, *pipeline Machine Learning* dapat dibuat untuk *instance model* inferensi seperti *TensorFlow*, *TFLite*, dan juga untuk fungsi pemrosesan media, bahkan tidak memerlukan GPU untuk menjalankan eksperimen dengan *MediaPipe*, karena grafik dan CPU terintegrasi saat ini bekerja dengan baik untuk solusi ini. Logikanya, FPS jauh lebih rendah daripada penggunaan GPU, tampak pada gambar 2.4 di bawah ini. (Pro, 2020)



Gambar 2.4 *Mediapipe* untuk *Pose estimation*
(Sumber: Pro, 2020)

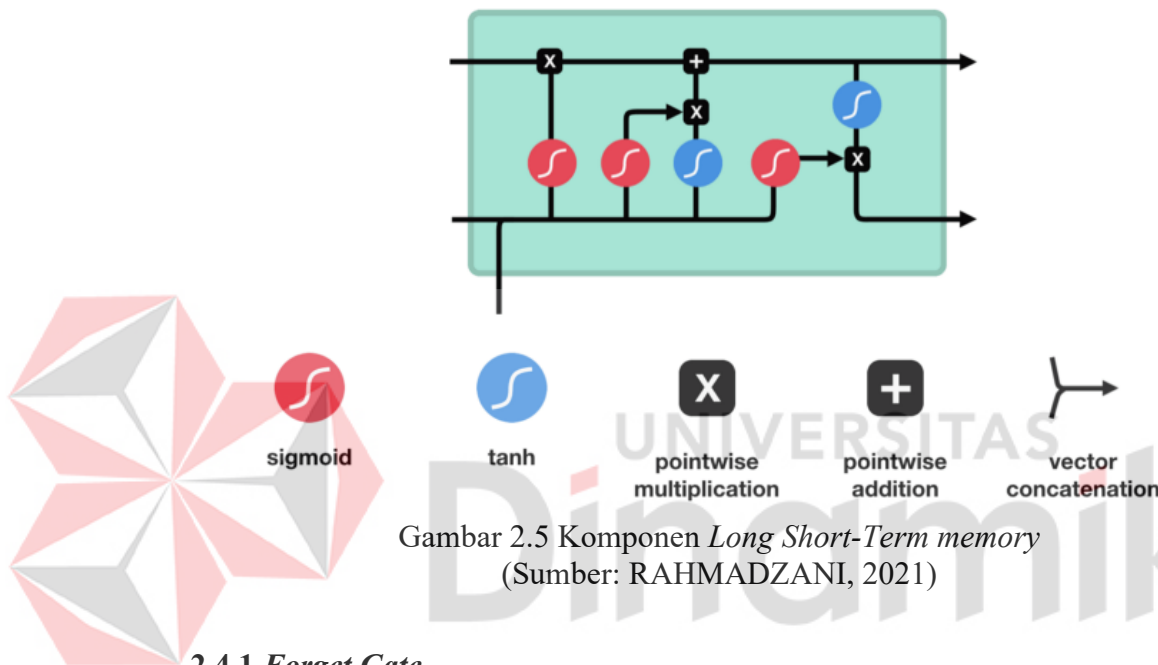
2.3 Dataset

Salah satu tugas utama untuk membangun model menggunakan *Machine Learning* adalah *dataset*. Sebelum memulai bekerja dengan algoritma, penting untuk dicatat bahwa setiap program perlu mengerti karakteristik data proses *Dataset Training*. Data ini nantinya dapat digunakan untuk melatih dan mengevaluasi model yang dihasilkan (Science, 2020). Untuk melatih model LSTM di Tugas Akhir ini menggunakan *Dataset RNN for Human Activity Recognition - 2D*. Ini terdiri dari deteksi *keypoint*, dibuat menggunakan model pembelajaran mendalam *OpenPose*, pada subset dari dataset *Berkeley Multimodal Human Action Database (MHAD)*. (Sebastian, 2021), serta membuat *training* gerakan yang baru dari dataset tersebut.

2.4 Long Short-Term Memory

Long Short-Term Memory (LSTM) adalah model jaringan saraf berulang (RNN) varian. LSTM terjadi karena dapat mengingat informasi jangka panjang

LSTM menggantikan node lapisan tersembunyi dari RNN dengan sel LSTM yang berfungsi untuk menyimpan informasi sebelumnya. LSTM memiliki tiga gerbang yang mengontrol penggunaan dan pembaruan informasi tekstual sebelumnya: *Forget Gate*, *Input Gate*, *Cell State*, dan *Output Gate*. Sel memori dan tiga gerbang untuk membaca, menyimpan, dan memperbarui informasi historis, berikut tampak pada gambar 2.5 di bawah ini. (RAHMADZANI, 2021)



2.4.1 *Forget Gate*

Tahap pertama dimulai melalui komponen *Forget Gate* (F_t). Pengolahan data masukan LSTM terdiri dari beberapa langkah. Pada bagian ini, informasi yang kurang penting atau tidak penting untuk kasus yang sedang ditangani dihilangkan menggunakan fungsi sigmoid. Data x_t adalah input (vektor input x pada langkah waktu t) dan h_{t-1} adalah vektor keadaan laten pada langkah waktu $t-1$ sebelumnya. Persamaan 1 menunjukkan perhitungan nilai parameter *Forget Gate*. (RAHMADZANI, 2021)

$$f_t = \sigma(W_f x_t + U_f h_{(t-1)} + b_f) \quad (1)$$

2.4.2 Input Gate

Pada langkah selanjutnya, informasi diproses melalui komponen *Input Gate* dengan menggunakan perhitungan pada persamaan (2). Proses ini menggunakan fungsi aktivasi sigmoid untuk mengurutkan dan menentukan informasi spesifik yang diperbarui di bagian status sel. Langkah ini juga menggunakan fungsi aktivasi tanh untuk menghasilkan kandidat vektor baru. Vektor ini ditambahkan ke bagian keadaan sel c_t menggunakan perhitungan dalam persamaan (3). Fungsi aktivasi ini berfungsi untuk mengubah suatu nilai atau menormalkan hasil dari fungsi ekstraksi sebelumnya. Cara kerja fungsi aktivasi ini: Jika ekstraksi fitur memiliki nilai negatif, maka dinormalisasi ke nilai 0. F (RAHMADZANI, 2021). Nilai dari suatu input hanya dapat disimpan ke dalam *cell state* hanya jika diijinkan oleh *input gate*. Perhitungan dari nilai pada input gate dan kandidat dari cell state dilakukan dengan menggunakan persamaan (1) dan (2). Dimana i_t adalah nilai dari *input gate*, W_i adalah bobot untuk nilai input pada waktu ke t , x_t adalah nilai input pada waktu ke t , U_i adalah bobot untuk nilai output dari waktu ke $t-1$, $h_{(t-1)}$ adalah nilai output dari waktu ke $t-1$ dan b_i adalah bias pada input gate dan σ adalah fungsi sigmoid. $C_t = \tanh(W_c x_t + U_i h_{(t-1)} + b_c)$

$$i_t = \sigma(W_i x_t + U_i h_{(t-1)} + b_i) \quad (2)$$

$$C_t = \tanh(W_c x_t + U_i h_{(t-1)} + b_c) \quad (3)$$

2.4.3 Cell State

Setelah memproses "*Cell State*" dan "*Input Gate*" masuk LSTM memperbarui nilai lama status sel c_{t-1} dengan status baru sel c_t . Persamaan 4. (RAHMADZANI, 2021). Dimana C_t adalah nilai kandidat cell state, W_c adalah bobot untuk nilai input pada cell ke c , x_t adalah nilai input pada waktu ke t , U_i adalah bobot untuk nilai output dari cell ke $c-1$, $h_{(c-1)}$ adalah nilai output dari cell ke $c-1$ dan b_c adalah bias pada cell ke c dan \tanh adalah fungsi hyperbolic tangent. Kemudian nilai dari forget gate dihitung dengan menggunakan persamaan (3).

$$C_t = i_t * C_t + f_t * C_{(t-1)} \quad (4)$$

2.4.4 Output Gate

Output Gate menjalankan sigmoid untuk menghasilkan *Output* tersembunyi dan atur status sel ke tanh. Setelah menghasilkan keluaran nilai sigmoidal dan nilai keluaran tanh, dikalikan kedua hasil aktivasi tersebut dan melanjutkan ke langkah berikutnya. *Output* dihitung sebagai persamaan 5 dan 6, kemudian mendapatkan nilai klasifikasi dari perhitungan keseluruhan proses LSTM. (RAHMADZANI, 2021)

$$o_t = \sigma(W_o x_t + U_o h_{(t-1)} + b_o) \quad (5)$$

$$h_t = o_t * \tanh(C_t) \quad (6)$$

2.5 DroidCam

DroidCam merupakan aplikasi digunakan untuk pengganti webcam pada PC atau laptop menggunakan *Smartphone* dengan ini aplikasi ini bisa menggantikan ketika webcam pada laptop kurang leluasa atau sebagai *IP Camera*. Fitur ini dapat digunakan mode Video 720p, seperti gambar 2.6 berikut ini. (Akbar, 2019)

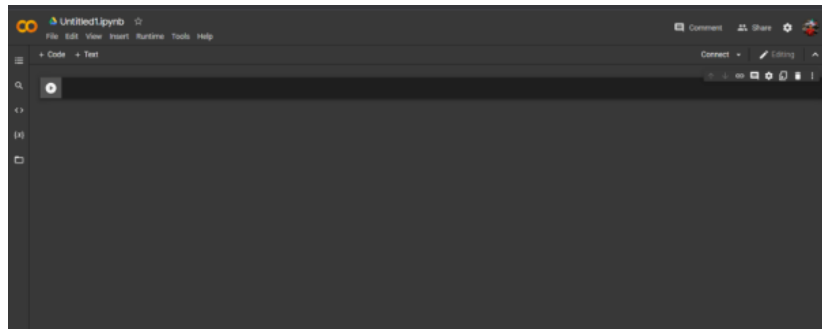


Gambar 2.6 Aplikasi *Droidcam*
(Sumber: Akbar, 2019)

2.6 Google Colab

Google Colab atau *Google Collaboratory* adalah dokumen yang dapat dieksekusi yang dapat digunakan untuk menyimpan, membuat, dan berbagi program yang ditulis melalui *Google Drive*. Perangkat lunak ini memungkinkan pengguna untuk menjalankan kode *Python* tanpa proses instalasi dan konfigurasi lainnya. Sebagai gantinya, semua pengaturan dan penyesuaian yang diperlukan

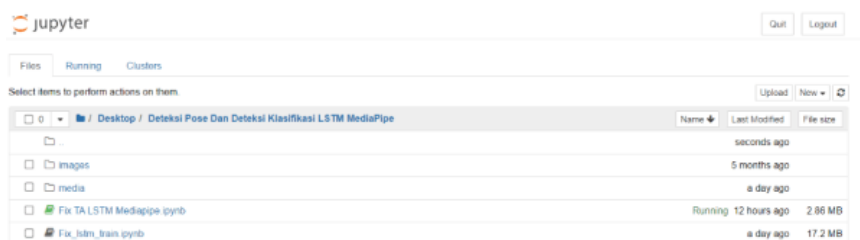
dikirim ke *cloud*. *Google Collaboratory* juga dikenal karena kemampuannya untuk kebutuhan kolaborasi tim, berikut di bawah ini gambar 2.7. (Oliver, 2021)



Gambar 2.7 Tampilan *Google Colab*
(Sumber: Oliver, 2021)

2.7 *Jupyter Notebook (Anaconda3)*

Jupyter Notebook adalah manipulasi data *Python* yang populer. *Notebook jupyter* memungkinkan seseorang untuk mengkonsolidasikan kode secara interaktif dengan *output* ke dalam satu dokumen. Cara memulai *Jupyter Notebook* adalah mengetikkan *jupyter notebook* di terminal. *Notebook Jupyter* diakses melalui *browser*. Ketika layanan dimulai, diarahkan ke halaman *browser http://localhost:8888/tree*. Halaman *Notebook Jupyter* muncul di folder tempat sintaks *Jupyter Notebook* yang dijalankan, seperti gambar 2.8 di bawah ini. (Galla, 2020)



Gambar 2.8 Tampilan *Jupyter*
(Sumber: Galla, 2020)

BAB III

METODOLOGI PENELITIAN

3.1 Blok Diagram Sistem

Pada dasarnya mengklasifikasikan suatu aktivitas, perlu diperhatikan dalam menemukan bagian tubuh di setiap *Frame* dan kemudian menganalisis pergerakan bagian tubuh dari waktu ke waktu. Langkah pertama, gerakan dideteksi menggunakan Kamera lalu *MediaPipe* memproses *Keypoints* digunakan Untuk menampilkan postur badan manusia (33 *Keypoints*), setelah mengamati satu *Frame* dalam sebuah video. Langkah kedua menganalisis gerakan tubuh dari waktu ke waktu dan membuat klasifikasi gerakan LSTM. Jadi, *Keypoints* dari dikirim ke LSTM untuk klasifikasi gerakan secara *Real-Time*, seperti yang ditunjukkan gambar 3.1.

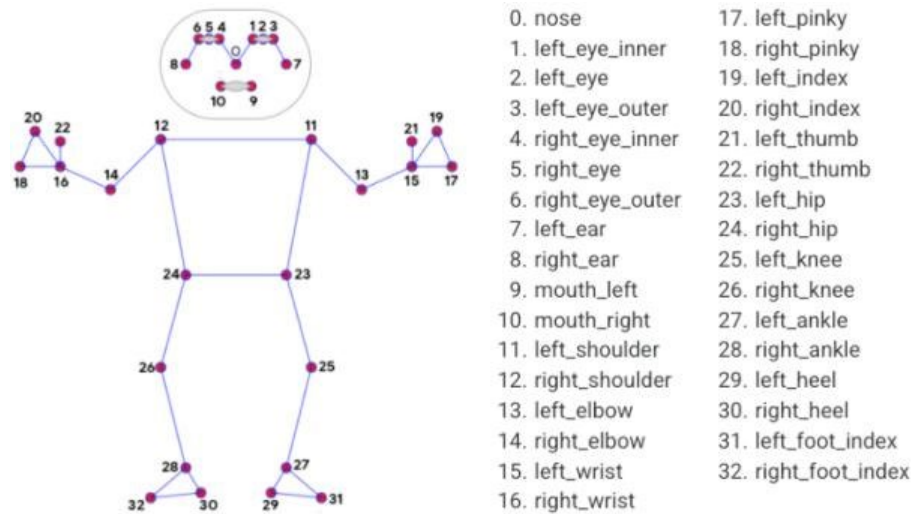


Gambar 3.1 Blok diagram sistem program

3.2 Pengumpulan *Keypoints*

MediaPipe juga mendukung mesin inferensi *TensorFlow*. Semua model *TensorFlow* di *MediaPipe*. Pada saat yang sama pada platform seluler dan tertanam, *MediaPipe* juga mendukung akselerasi GPU (*Graphical Processing Unit*) dan CPU (*Central Processing Unit*) perangkat itu sendiri. (Hossian, 2021)

Input *video* nanti mendeteksi sebanyak 33 *Keypoints* yang mewakili dtubuh badan manusia mulai dari bagian hidung sampai bagian kaki bawah, menggunakan kamera atau webcam tergantung alat yang dipakai dalam merekam nantinya pada penggunaan kali ini digunakan menggunakan CPU agar komputasi lebih efisien dan lebih cepat dalam proses pendeteksian nantinya, seperti gambar dibawah ini gambar 3.2.

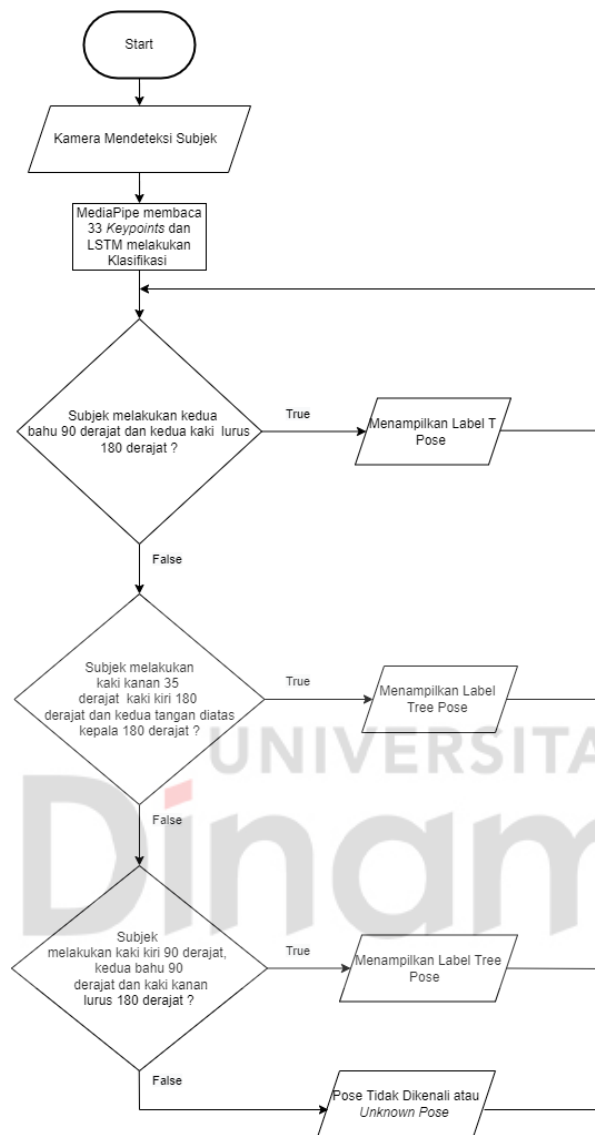


Gambar 3.2 Index *Keypoints* pada *Mediapipe*
(Sumber: MediaPipe, 2020)

3.3 Pengumpulan Klasifikasi Gerakan

Klasifikasi gerakan dikumpulkan dengan menggunakan software *OpenPose* dengan dataset baru yaitu *Warrior II Pose*, *T-Pose*, *Tree Pose*. Dataset ini nantinya diuji dengan 4 subjek melakukan 3 gerakan yang telah disebutkan didemokan secara *Real-Time* pada bagian hasil dan pembahasan.

3.4 Flowchart Human Activity Recognition



Gambar 3.3 Flowchart *human activity recognition*

Pada gambar 3.3 menjelaskan algoritma sistem yang dimulai dari Start, kemudian kamera mendeteksi gerakan. *MediaPipe* melakukan proses pembacaan 33 *keypoints* dan LSTM melakukan proses *Training Dataset* klasifikasi gerakan. Selanjutnya terjadi percabangan bersarang 3 kondisi klasifikasi gerakan, kondisi pertama yaitu subjek melakukan dengan *Pose* dengan kedua bahu searah 90° derajat dan kedua kaki lurus 180° derajat, maka kondisi bernilai *True* mengklasifikasi gerakan label *T Pose*. Sementara bila kondisi tersebut tidak terpenuhi atau bernilai *False*, maka proses berjalan pada kondisi berikutnya, maka melakukan kondisi

kedua yaitu subjek melakukan kaki kanan 35° derajat, kaki kiri 180° derajat dan kedua tangan di atas kepala 180° derajat, maka kondisi bernilai *True* mengklasifikasikan gerakan label *Tree Pose*. Sementara bila kondisi tersebut tidak terpenuhi atau bernilai *False*, maka proses berjalan pada kondisi terakhir. Kondisi terakhir yaitu subjek melakukan kaki kiri ditekuk 90° derajat, kedua bahu lurus 90° derajat dan kaki kanan lurus 180° derajat, maka program mengklasifikasi gerakan tersebut dengan label *Warrior II Pose*. Jika dari 3 kondisi tersebut tidak dapat terklasifikasi, maka disebut sebagai *Unknown Pose* atau *Pose* yang tidak dikenali, jika tidak ada proses pada percabangan, maka program mengulang pembacaan 33 *Keypoints* pada *MediaPipe* mengulang lagi sampai dari 3 kondisi tersebut. *Flowchart* berfungsi sebagai cara kerja alur bagan dari sistem program gerakan Aktivitas manusia yang terdeteksi oleh saluran inferensi yang dijelaskan pada video nantinya berupa gerakan.



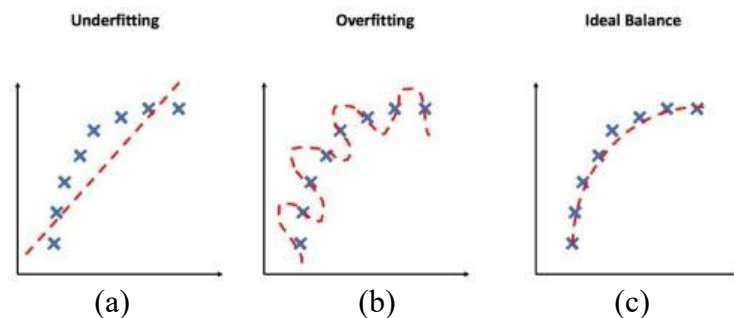
UNIVERSITAS
Dinamika

BAB IV

HASIL DAN PEMBAHASAN

4.1 Pengujian Hasil Training *Long Short-Term Memory*

Training pada *Dataset* dilakukan dengan 400 *Epochs* dimana setiap *Epochs* memiliki ukuran batch sebanyak 512 dan memiliki iterasi atau step sebanyak 57 setiap 1 *Epochs*. Proses *Training* menggunakan optimizer “Adam” dengan *Learning Rate* bernilai 0.001 menggunakan “Adam” karena mudah diterapkan efisien secara komputasi dan sesuai untuk gradien *noise* yang tinggi. Hasil dari *training* memiliki masing-masing grafik yaitu: *train_acc*, *train_loss*, *val_acc*, *val_loss* *batch_train_acc*, *batch_val_acc* *batch_train_loss*, *batch_val_loss* dan *Epochs*. Pada dasarnya yang diambil untuk training adalah *Train_loss*, *Train_acc*, *val_loss*, *val_acc*. Selama *Training* menghabiskan durasi selama rata-rata 10 menit sampai 12 menit *Training* juga dilakukan selama 4 kali percobaan. *Epochs* diambil 1 *Epochs* Terakhir yaitu mulai 390-399 *Epochs* sebagai hasil perbandingan dimulai dari waktu 10 menit dikarenakan Metode *Long Short-Term Memory* menggunakan *Transfer Learning* untuk menghasilkan grafik yang lebih *smooth* dibandingkan dengan menggunakan metode CNN. Hasil Pengujian keseluruhan tabel dapat dilihat pada lampiran T L1.1.



Gambar 4.1 Hasil grafik (a) *underfitting*, (b) *overfitting*, (c) *ideal balance*
(Sumber: Karthik Ramasubramanian, 2019)

Pada gambar 4.1 menjelaskan bahwa *Overfitting* (kondisi yang biasa memiliki loss rendah dan akurasi rendah) maupun *Underfitting* (kondisi yang biasa memiliki loss yang tinggi dan akurasi tinggi). *Ideal Balance* memiliki loss yang rendah dan akurasi yang tinggi.

4.1.1 Perhitungan *Training Dan Step*

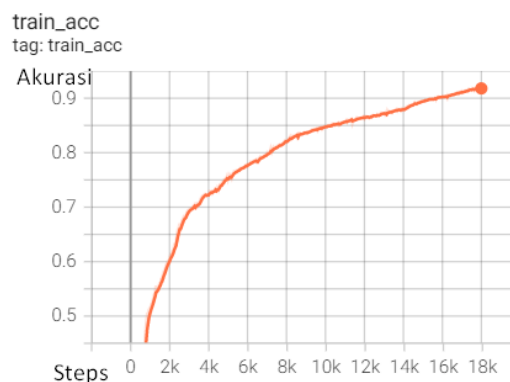
Sebelum menggunakan proses *Training* di dalam *Dataset* ada parameter yang harus diuji dari grafik yang didapatkan yaitu menentukan *Epoch*, digunakan dalam proses training, mewakili penggunaan dari *step* semua data pelatihan untuk perhitungan gradien dan optimisasi (*Training model*), sedangkan *step* digunakan untuk penggunaan suatu ukuran *Epochs* pelatihan untuk melatih model. Jadi jumlah batch sama dengan jumlah iterasi untuk satu epoch (Jalil, 2020), sehingga jika dari pengujian diatas menggunakan *Epochs* sebanyak 400, selama proses *training* didapatkan 7200000 data, maka perhitungan didapatkan sebagai berikut:

$$\text{Steps} = \frac{\text{total_number_of_training_examples}}{\text{Epochs}} \quad (1)$$

$$\frac{7.200.000}{400} = 18.000 \text{ Steps}$$

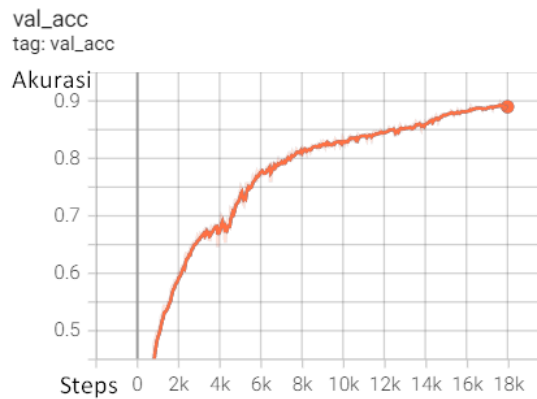
4.1.2 Accuracy Dan Validation_Accuracy

Gambar 4.2 sampai gambar 4.4 merupakan hasil *training* plot grafik semua akurasi, yang digunakan untuk menghitung perhitungan dan validasi akurasi dari data training model *batch* sendiri, untuk digunakan mengontrol akurasi *error gradient* dalam proses *Training*.



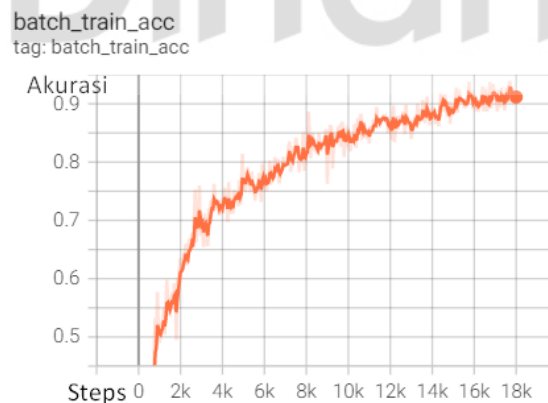
Gambar 4.2 Hasil grafik plot *train_accuracy*

Gambar 4.2 merupakan hasil *Training* plot grafik *Train_Acc* digunakan untuk menghitung perhitungan nilai grafik semua akurasi *function Training Dataset*. Hasil *Train_Acc* dari grafik plot tersebut menghasilkan 29% di titik akhir grafik tersebut.



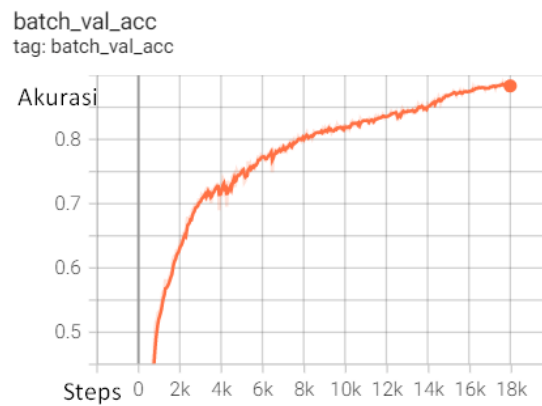
Gambar 4.3 Hasil grafik plot *val_acc*

Gambar 4.3 *Val_Acc* adalah mengukur kinerja model dengan menghitung segala bentuk tingkat kebenaran prediksi pada model. *Val_Acc* digunakan untuk menghitung validasi akurasi mencocokkan yang sesuai hasil training hasil dari *Val_Acc* dari grafik plot tersebut menghasilkan akurasi sebesar 91%.



Gambar 4.4 Hasil grafik plot *batch_train_acc*

Gambar 4.4 Hasil *Batch_Train_Acc* ini digunakan untuk Ukuran batch akurasi menentukan berapa banyak contoh pelatihan yang diproses secara paralel untuk *Training* /inferensi. Ukuran batch pada waktu pelatihan dapat memengaruhi seberapa cepat dan seberapa baik pelatihan. Hasil dari *Batch_Train_Acc* dari grafik plot tersebut menghasilkan loss sebesar 93%.



Gambar 4.5 Hasil grafik plot *batch_val_acc*

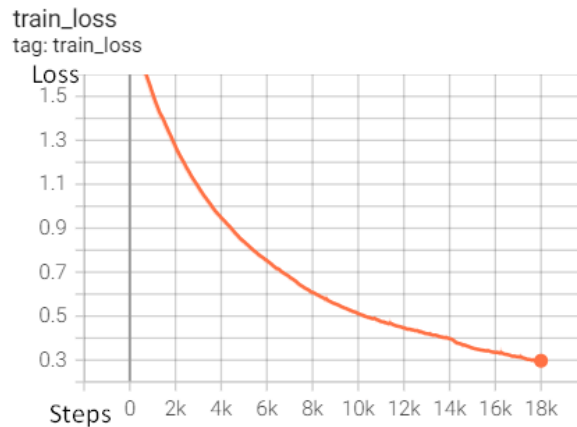
Gambar 4.5 *Batch_Val_Acc* adalah digunakan untuk mengontrol ukuran batch grafik jumlah lintasan akurasi melalui *dataset Training*. Hasil dari *Batch_Val_Acc* dari grafik plot tersebut menghasilkan loss sebesar 88%,

Tabel 4.1 Persen perbandingan *accuracy training* dan *validasi_accuracy*

<i>Epochs</i>	<i>Batch Train Acc (%)</i>	<i>Batch Val Acc (%)</i>	<i>Train Acc (%)</i>	<i>Val Acc (%)</i>
390	92	88	91	88
391	91	88	91	89
392	91	89	91	90
393	90	89	91	89
394	90	88	91	88
395	89	88	91	89
396	92	88	91	88
397	94	87	91	88
398	90	88	92	89
399	93	88	91	89

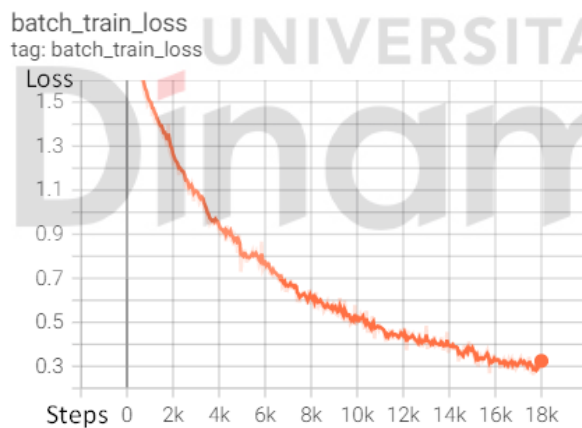
Tabel 4.1 merupakan empat variabel selama proses training yaitu *Batch_Train_Acc*, *Batch_Val_Acc*, *Train_Acc*, *Val_Acc* pada data tabel tersebut didapatkan selama proses *Training* yang diambil dari *Epochs* terakhir yaitu 390-399 tujuan dari pengambilan dari angka tersebut untuk menganalisis hasil akhir yang terbaik dan tidak ada terjadinya data yang *Overfitting* maupun *Underfitting* selama proses *Training*. Tabel ini digunakan untuk mengklasifikasikan setiap keberhasilan gerakan pada saat program tersebut berjalan.

4.1.3 *Train_Loss, Validation_Loss, Dan Batch_Train_Loss*



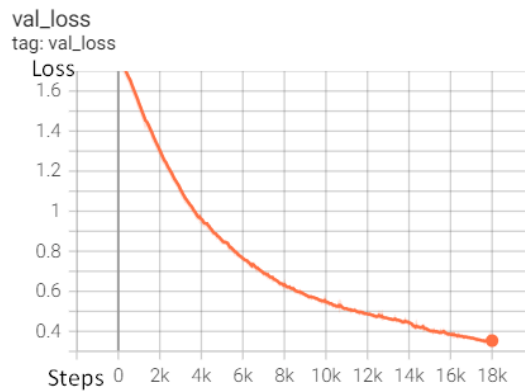
Gambar 4.6. Hasil grafik plot pada *train_loss*

Gambar 4.6 merupakan hasil *training* plot grafik *Train_Loss* digunakan untuk menghitung perhitungan nilai dari *loss function* training dataset. Hasil *Train_Loss* dari grafik plot tersebut menghasilkan 29% di titik akhir grafik tersebut.



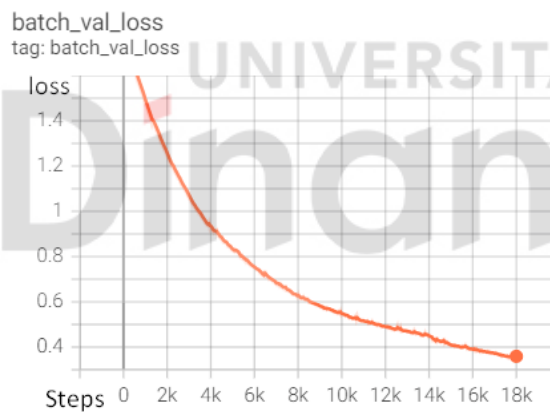
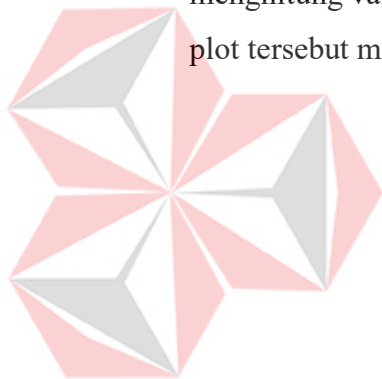
Gambar 4.7 Hasil grafik *batch_train_loss*

Gambar 4.7 hasil *Batch_Train_Loss* ukuran batch adalah penurunan gradien yang mengontrol jumlah sampel pelatihan untuk dikerjakan sebelum parameter internal model diperbarui. Hasil dari *Batch_Train_Loss* dari grafik plot tersebut menghasilkan loss sebesar 33%.



Gambar 4.8 Hasil grafik plot *val_loss*

Gambar 4.8 *Val_Loss* adalah mengukur kinerja model dengan menghitung segala bentuk tingkat kesalahan prediksi pada model. *Val_Loss* digunakan untuk menghitung validasi loss yang sesuai hasil training. Hasil dari *Val_Loss* dari grafik plot tersebut menghasilkan loss sebesar 35%



Gambar 4.9 Hasil grafik *batch_val_loss*

Gambar 4.9 *Batch_Val_Loss* adalah digunakan untuk mengontrol ukuran batch grafik jumlah lintasan lengkap melalui *dataset Training*. Hasil dari *Batch_Val_Loss* dari grafik plot tersebut menghasilkan loss sebesar 36%.

Tabel 4.2 Regresi perbandingan *training loss* dan *validation*

<i>Epochs</i>	<i>Train Loss</i>	<i>Val Loss</i>	<i>Batch Train Loss</i>	<i>Batch Val loss</i>
390	0.30	0.35	0.28	0.35
391	0.29	0.35	0.31	0.36
392	0.30	0.35	0.30	0.35
393	0.29	0.35	0.31	0.35
394	0.29	0.35	0.31	0.35
395	0.29	0.35	0.31	0.35
396	0.29	0.36	0.30	0.36
397	0.30	0.37	0.26	0.38
398	0.30	0.34	0.31	0.35
399	0.29	0.35	0.33	0.36

Tabel 4.2 merupakan hasil empat variabel selama proses training yaitu *Batch_Train_Loss*, *Batch_Val_Loss*, *Train_Loss*, *Val_Loss*. Data tabel tersebut didapatkan selama proses *Training* yang diambil dari *Epochs* terakhir yaitu 390-399 sekelompok perhitungan nilai *loss_function* training dataset yang memiliki nilai *Batch_Train_Loss* terkecil hasil terbaik di yaitu 0.33. Dibuktikan juga bahwa *validasi loss* memiliki 0.35 yang dimana masih memiliki nilai yang stabil dan cukup ideal dan tidak *underfitting*.

4.2 Pengujian Keypoints

Pengujian *Keypoints* dilakukan dengan menggunakan kamera handphone *XiaoMi8* dengan memakai aplikasi *DroidCam*. Dilakukan dengan 4 subjek, tujuan dari pengujian ini adalah menguji keberhasilan atau tidaknya hasil *Real-Time Keypoints* sebanyak 33 *Keypoint*. Hasil Pengujian bisa dilihat pada gambar 4.8 sampai gambar 4.20.

Tabel 4.3 Pengujian *keypoints*.

Subjek	Index	Nama Keypoint	Keberhasilan Keypoint
1	0	<i>Nose</i>	Terdeteksi
	1	<i>left_eye inner</i>	Terdeteksi
	2	<i>left_eye</i>	Terdeteksi
	3	<i>left_eye outer</i>	Terdeteksi
	4	<i>right_eye inner</i>	Terdeteksi
	5	<i>right_eye</i>	Terdeteksi

Subjek	Index	Nama Keypoint	Keberhasilan Keypoint
	6	<i>right_eye_outer</i>	Terdeteksi
	7	<i>left_ear</i>	Terdeteksi
	8	<i>right_ear</i>	Terdeteksi
	9	<i>mouth_left</i>	Terdeteksi
	10	<i>mouth_right</i>	Terdeteksi
	11	<i>left_shoulder</i>	Terdeteksi
	12	<i>right_shoulder</i>	Terdeteksi
	13	<i>left_elbow</i>	Terdeteksi
	14	<i>right_elbow</i>	Terdeteksi
	15	<i>left_wrist</i>	Terdeteksi
	16	<i>right_wrist</i>	Terdeteksi
	17	<i>left_pinky</i>	Terdeteksi
	18	<i>right_pinky</i>	Terdeteksi
	19	<i>left_index</i>	Terdeteksi
	20	<i>right_index</i>	Terdeteksi
	21	<i>left_thumb</i>	Terdeteksi
	22	<i>right_thumb</i>	Terdeteksi
	23	<i>left_hip</i>	Terdeteksi
	24	<i>right_hip</i>	Terdeteksi
	25	<i>left_knee</i>	Terdeteksi
	26	<i>right_knee</i>	Terdeteksi
	27	<i>left_ankle</i>	Terdeteksi
	28	<i>right_ankle</i>	Terdeteksi
	29	<i>left_heel</i>	Terdeteksi
	30	<i>right_heel</i>	Terdeteksi
	31	<i>left_foot_index</i>	Terdeteksi
	32	<i>right_foot_index</i>	Terdeteksi
2	0	<i>Nose</i>	Terdeteksi
	1	<i>left_eye_inner</i>	Terdeteksi
	2	<i>left_eye</i>	Terdeteksi
	3	<i>left_eye_outer</i>	Terdeteksi
	4	<i>right_eye_inner</i>	Terdeteksi
	5	<i>right_eye</i>	Terdeteksi
	6	<i>right_eye_outer</i>	Terdeteksi
	7	<i>left_ear</i>	Terdeteksi
	8	<i>right_ear</i>	Terdeteksi
	9	<i>mouth_left</i>	Terdeteksi
	10	<i>mouth_right</i>	Terdeteksi
	11	<i>left_shoulder</i>	Terdeteksi
	12	<i>right_shoulder</i>	Terdeteksi
	13	<i>left_elbow</i>	Terdeteksi

Subjek	Index	Nama Keypoint	Keberhasilan Keypoint
3	14	<i>right_elbow</i>	Terdeteksi
	15	<i>left_wrist</i>	Terdeteksi
	16	<i>right_wrist</i>	Terdeteksi
	17	<i>left_pinky</i>	Terdeteksi
	18	<i>right_pinky</i>	Terdeteksi
	19	<i>left_index</i>	Terdeteksi
	20	<i>right_index</i>	Terdeteksi
	21	<i>left_thumb</i>	Terdeteksi
	22	<i>right_thumb</i>	Terdeteksi
	23	<i>left_hip</i>	Terdeteksi
	24	<i>right_hip</i>	Terdeteksi
	25	<i>left_knee</i>	Terdeteksi
	26	<i>right_knee</i>	Terdeteksi
	27	<i>left_ankle</i>	Terdeteksi
	28	<i>right_ankle</i>	Terdeteksi
	29	<i>left_heel</i>	Terdeteksi
	30	<i>right_heel</i>	Terdeteksi
	31	<i>left_foot_index</i>	Terdeteksi
	32	<i>right_foot_index</i>	Terdeteksi
	0	<i>Nose</i>	Terdeteksi
	1	<i>left_eye_inner</i>	Terdeteksi
	2	<i>left_eye</i>	Terdeteksi
	3	<i>left_eye_outer</i>	Terdeteksi
	4	<i>right_eye_inner</i>	Terdeteksi
	5	<i>right_eye</i>	Terdeteksi
	6	<i>right_eye_outer</i>	Terdeteksi
	7	<i>left_ear</i>	Terdeteksi
	8	<i>right_ear</i>	Terdeteksi
	9	<i>mouth_left</i>	Terdeteksi
	10	<i>mouth_right</i>	Terdeteksi
	11	<i>left_shoulder</i>	Terdeteksi
	12	<i>right_shoulder</i>	Terdeteksi
	13	<i>left_elbow</i>	Terdeteksi
	14	<i>right_elbow</i>	Terdeteksi
	15	<i>left_wrist</i>	Terdeteksi
	16	<i>right_wrist</i>	Terdeteksi
	17	<i>left_pinky</i>	Terdeteksi
	18	<i>right_pinky</i>	Terdeteksi
	19	<i>left_index</i>	Terdeteksi
	20	<i>right_index</i>	Terdeteksi
	21	<i>left_thumb</i>	Terdeteksi

Subjek	Index	Nama Keypoint	Keberhasilan Keypoint
	22	<i>right_thumb</i>	Terdeteksi
	23	<i>left_hip</i>	Terdeteksi
	24	<i>right_hip</i>	Terdeteksi
	25	<i>left_knee</i>	Terdeteksi
	26	<i>right_knee</i>	Terdeteksi
	27	<i>left_ankle</i>	Terdeteksi
	28	<i>right_ankle</i>	Terdeteksi
	29	<i>left_heel</i>	Terdeteksi
	30	<i>right_heel</i>	Terdeteksi
	31	<i>left_foot_index</i>	Terdeteksi
	32	<i>right_foot_index</i>	Terdeteksi
4	0	<i>Nose</i>	Terdeteksi
	1	<i>left_eye_inner</i>	Terdeteksi
	2	<i>left_eye</i>	Terdeteksi
	3	<i>left_eye_outer</i>	Terdeteksi
	4	<i>right_eye_inner</i>	Terdeteksi
	5	<i>right_eye</i>	Terdeteksi
	6	<i>right_eye_outer</i>	Terdeteksi
	7	<i>left_ear</i>	Terdeteksi
	8	<i>right_ear</i>	Terdeteksi
	9	<i>mouth_left</i>	Terdeteksi
	10	<i>mouth_right</i>	Terdeteksi
	11	<i>left_shoulder</i>	Terdeteksi
	12	<i>right_shoulder</i>	Terdeteksi
	13	<i>left_elbow</i>	Terdeteksi
	14	<i>right_elbow</i>	Terdeteksi
	15	<i>left_wrist</i>	Terdeteksi
	16	<i>right_wrist</i>	Terdeteksi
	17	<i>left_pinky</i>	Terdeteksi
	18	<i>right_pinky</i>	Terdeteksi
	19	<i>left_index</i>	Terdeteksi
	20	<i>right_index</i>	Terdeteksi
	21	<i>left_thumb</i>	Terdeteksi
	22	<i>right_thumb</i>	Terdeteksi
	23	<i>left_hip</i>	Terdeteksi
	24	<i>right_hip</i>	Terdeteksi
	25	<i>left_knee</i>	Terdeteksi
	26	<i>right_knee</i>	Terdeteksi
	27	<i>left_ankle</i>	Terdeteksi
	28	<i>right_ankle</i>	Terdeteksi
	29	<i>left_heel</i>	Terdeteksi

Subjek	Index	Nama Keypoint	Keberhasilan Keypoint
	30	<i>right_heel</i>	Terdeteksi
	31	<i>left_foot_index</i>	Terdeteksi
	32	<i>right_foot_index</i>	Terdeteksi

Tabel 4.3 telah dilakukan pengujian *Keypoints* dengan 4 subjek secara keseluruhan dan dihasilkan bahwa pengujian dapat dilakukan dengan pengambilan sudut gambar kamera harus stabil dan lurus dikarenakan jika kamera tidak stabil dan subjek kurang terkena cahaya kamera, maka proses deteksi mengalami kesulitan mengenali setiap *keypoints*. Pada tabel 4.3 menggunakan aplikasi *DroidCam* untuk mendapatkan hasil index 33 *keypoints* yang maksimal dan menghasilkan rata-rata 10-30 FPS (*Frame per Second*).

4.3 Pengujian Klasifikasi Gerakan

Pengujian klasifikasi gerakan dilakukan dengan 4 subjek 5 kali percobaan di lokasi dan waktu yang sama. Gerakan tersebut direkam oleh kamera handphone *XiaoMi 8* memakai aplikasi *DroidCam* untuk mengambil sudut *Keypoints* tubuh manusia secara maksimal. Dalam hal ini *Keypoints* dari urutan *frame* dikirim ke dalam *Long Short-Term Memory* (LSTM) untuk klasifikasi dan memprediksi aktivitas yang sesuai. Pengujian klasifikasi gerakan digunakan untuk mengklasifikasi 3 gerakan yaitu: *T-Pose*, *Warrior II Pose*, *Tree Pose*. Pada lokasi ini juga mempengaruhi dalam pengambilan video karena harus memiliki ruangan yang cukup luas tidak terhalang oleh benda apapun didepan subjek. Diantara kedua gerakan *T-Pose* dan *Warrior II Pose* harus memiliki kedua lengan lurus dan bahu harus pada sudut tertentu. Program mendeteksi dan mengklasifikasi gerakan apakah kedua lengan lurus dan bahu harus berada di sudut lurus. Perbedaan dari kedua gerakan tersebut adalah jika *Warrior II Pose* satu kaki lurus dan kaki lainnya ditekuk pada sudut yang diperlukan, sedangkan *T-Pose* hanya mengecek jika kedua kaki lurus saja dan kedua tangan lurus 180 ° derajat.

Tabel 4.4 Keberhasilan klasifikasi gerakan

Subjek	<i>Warrior II Pose</i>					<i>T-Pose</i>					<i>Tree Pose</i>				
	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
1	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	-	✓
2	-	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
3	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	-	✓	✓	✓
4	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓	✓	✓
Rata-Rata Keberhasilan Gerakan (%)	85%					100%					80%				

Catatan: Terklasifikasi = ✓

Belum Terklasifikasi = -

$$\text{Rata-Rata Gerakan} = \frac{\text{Gerakan Terklasifikasi} * \text{Total Percobaan}}{\text{Banyaknya Subjek}} * 100\%$$

$$\begin{aligned} \text{Gerakan } \textit{Warrior II Pose} &= \frac{17 * 20}{4} * 100\% \\ &= 85\% \end{aligned}$$

$$\begin{aligned} \text{Gerakan } \textit{T - Pose} &= \frac{20 * 20}{4} * 100\% \\ &= 100\% \end{aligned}$$

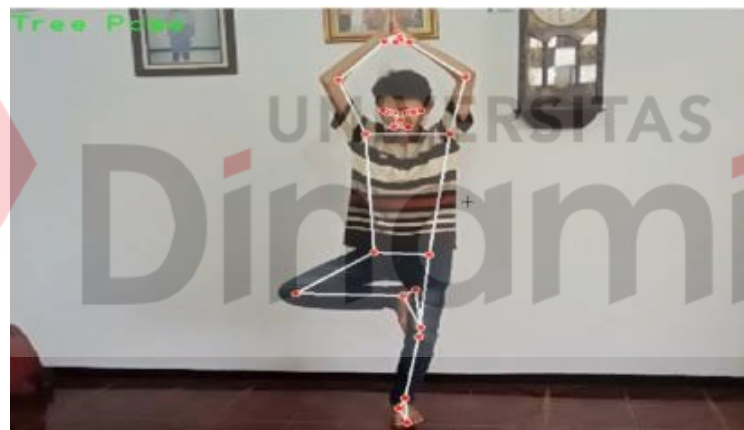
$$\begin{aligned} \text{Gerakan } \textit{Tree Pose} &= \frac{16 * 20}{4} * 100\% \\ &= 80\% \end{aligned}$$

Tabel 4.4 diatas menunjukkan bahwa dalam pengambilan video subjek tersebut menggunakan *Tripod*. 4 subjek dengan 3 gerakan klasifikasi dilakukan pada tempat lokasi yang sama serta diulang selama 5 kali percobaan, sehingga didapatkan 20 kali percobaan dari 4 subjek tersebut. Keberhasilan dan kegagalan dalam mengklasifikasi masing-masing gerakan dipengaruhi oleh beberapa faktor yaitu: Gerakan *Warrior II Pose* posisi kaki kiri yang harus menekuk pada sudut 90 derajat dan kaki kanan lurus 180 derajat, Gerakan *T-Pose* kedua tangan dan kedua kaki yang terkadang belum lurus 180 derajat dan Gerakan *Tree Pose* posisi kaki dengan bertumpu 180 derajat di satu lutut dan 35 derajat di kanan atau 335 derajat kiri lutut. Gerakan menghasilkan akurasi sebesar *Warrior II Pose* 85%, *T-Pose*

100%, dan *Tree Pose* 80%. Hasil pengujian keseluruhan gambar dapat dilihat pada lampiran gambar L4.1 sampai gambar L4.60. Berikut gambar 4.9 sampai gambar 4.20 merupakan percobaan ke-1 dari masing-masing subjek.



Gambar 4.10 Subjek 1 melakukan gerakan *T-Pose*



Gambar 4.11 Subjek 1 melakukan gerakan *Tree Pose*



Gambar 4.12 Subjek 1 melakukan gerakan *Warrior II Pose*



Gambar 4.13 Subjek 2 melakukan gerakan *Warrior II Pose*



Gambar 4.14 Subjek 2 melakukan gerakan *Tree Pose*



Gambar 4.15 Subjek 2 melakukan gerakan *T-Pose*



Gambar 4.16 Subjek 3 melakukan gerakan *Warrior II Pose*



Gambar 4.17 Subjek 3 melakukan gerakan *T-Pose*



Gambar 4.18 Subjek 3 melakukan gerakan *Tree Pose*



Gambar 4.19 Subjek 4 melakukan gerakan *Tree Pose*



Gambar 4.20 Subjek 4 melakukan gerakan *T-Pose*



Gambar 4.21 Subjek 4 melakukan gerakan *Warrior II Pose*

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil pengujian dan analisis yang dilakukan, diperoleh beberapa kesimpulan yaitu:

1. Klasifikasi gerakan "*Warrior II Pose*" berhasil terklasifikasi 85%. Gerakan belum berhasil terklasifikasi dengan benar dikarenakan pencahayaan yang kurang merata, posisi kedua tangan belum sejajar dan belum searah 180° derajat. Gerakan "*Warrior II Pose*" ini ditunjuKkan dengan kedua lengan harus lurus dan bahu harus pada sudut tertentu serta salah satu kaki harus ditekuk pada sudut tertentu agar *Training* gerakan dapat terklasifikasi.
2. Klasifikasi gerakan "*T-Pose*" sudah berhasil terklasifikasi 100%, adapun gerakan dari "*T-Pose*" adalah membentangkan kedua tangan sejajar searah 180° derajat dan kedua kaki lurus ditempat.
3. Klasifikasi "*Tree Pose*" 80%, bentuk "*Tree Pose*" adalah salah satu kaki berdiri tegak sedangkan kaki yang lainnya membentuk sudut yang diperlukan serta diperlukan keseimbangan dalam melakukan gerakan tersebut.
4. Hasil dari *Training Dataset* menggunakan *Long Short-Term Memory* menghasilkan nilai akurasi *Training* sebesar 91% untuk validasi menghasilkan nilai akurasi sebesar 89% untuk nilai *loss* didapatkan sebesar 0.29, sedangkan untuk validasi *loss* didapatkan sebesar 0.35 *Epochs* yang digunakan adalah 400 *Epochs*.

5.2 Saran

Mengenai saran untuk perbaikan pengembangan Tugas Akhir ini, ada beberapa saran yaitu:

1. Untuk pengembangan kedepannya sebaiknya bisa mengklasifikasikan dan mendeteksi *Keypoints* secara *Multiple Recognition*. *Multiple Recognition* memiliki kelebihan yaitu dapat mendeteksi *keypoints* lebih dari satu subjek. Sedangkan untuk metode klasifikasinya bisa menggunakan LSTM *MultiLayer Perceptron*, yang bisa menghindari penggunaan banyak *Epochs* selama satu

iterasi. Pendeteksian *keypoints* dapat menggunakan *PoseNet*, *PIPF* dan lain-lain yang sejenis.

2. Penelitian ini dapat dikembangkan dengan mengenali gerakan aktivitas keseharian seperti gerakan berjalan, berlari, duduk, jongkok dan sebagainya.



UNIVERSITAS
Dinamika

DAFTAR PUSTAKA

- Akbar, D. (2019). *Webcam Rusak tapi Butuh Buat Video Call? Nih Solusi Mudah, Cuma Pakai Smartphone dan Koneksi Internet*. Retrieved from <https://pemmzchannel.com/>:
<https://pemmzchannel.com/2020/05/16/webcam-rusak-tapi-butuh-buat-video-call-nih-solusi-mudah-cuma-pakai-smartphone-dan-koneksi-internet/>
- Azmi, N. (2020). *5 Gerakan Olahraga di Rumah yang Cocok Untuk Pandemi Bagi Wanita*. Retrieved December 21, 2020, from <https://hellosehat.com/kebugaran/kekuatan/gerakan-olahraga-di-rumah-bagi-wanita/>
- Fimela. (2014, June 22). *Macam-Macam Yoga*. Retrieved from Fimela: <https://www.fimela.com/beauty/read/3844095/cara-mengecilkan-betis-dengan-tree-Pose-dalam-yoga>
- Galla, F. (2020, October 26). *Konsep Dasar Jupyter Notebook Untuk Python*. Retrieved from IGLab.tech: <https://iglab.tech/konsep-dasar-jupyter-notebook-untuk-python/>
- Geoffrey. (2019, March 29). *Should We Abandon LSTM for CNN?* Retrieved from AI/ML At Symantec: <https://medium.com/ai-ml-at-symantec/should-we-abandon-lstm-for-cnn-83accaeb93d6>
- Hossian, Z. (2021, July 21). *a Few Lines of Code and Detect Faces, Draw Landmarks from Complex Images ~MediaPipe*. Retrieved from TowardsDataScience: <https://towardsdatascience.com/write-a-few-lines-of-code-and-detect-faces-draw-landmarks-from-complex-images-mediapipe-932f07566d11>
- Jalil. (2020). *Penentuan Epoch dan Batch size pada klasifikasi CNN*. Retrieved from kotakode.com: <https://kotakode.com/pertanyaan/2416/Penentuan-Epoch-dan-Batch-size-pada-klasifikasi-CNN>
- Jian Sun, Y. F. (2018). Sequential Human Activity Recognition Based on Deep Convolutional Network and Extreme Learning Machine Using Wearable Sensors. *Hindawi*, 2018, 1. Retrieved September 27, 2018, from <https://www.hindawi.com/journals/js/2018/8580959/>
- Karthik Ramasubramanian, J. M. (2019). Applied Supervised Learning with R. In J. M. Karthik Ramasubramanian, *Applied Supervised Learning with R* (p. 502). Packt.
- MediaPipe. (2020). *MediaPipe Pose*. Retrieved from google.github.io: <https://google.github.io/mediapipe/solutions/Pose>

- Michalis Vringkas, C. N. (2015). Frontiersin Robot Machine Vision and AI Deep Learning. *Frontiers in Robotics and AI*, 2, 1. Retrieved November 16, 2015, from <https://www.frontiersin.org/articles/10.3389/frobt.2015.00028/full#B158>
- Oliver, A. (2021). *Mengenal Google Colab, Aplikasi Penunjang Keperluan Machine Learning*. Retrieved June 22, 2021, from <https://glints.com/id/lowongan/google-colab-adalah/>
- Pro, I. (2020). *Ichi pro.com*. Retrieved from <https://ichi.pro/id/detectron2-tutorial-ujung-ke-ujung-yang-sederhana-99819572924643>
- RAHMADZANI, R. F. (2021). *Cara Kerja Long Short-Term Memory (LSTM) | Catatan Penelitian*. Retrieved February 21, 2021, from <https://rifqifai.com/cara-kerja-long-short-term-memory-lstm/>
- Science. (2020). *Dataset dalam Machine Learning*. Retrieved October 25, 2020, from <https://www.dqlab.id/pelajari-3-data-set-machine-learning-bersama-dqlab>
- Sebastian, B. (2021). *Human Action Recognition using Detectron2 and LSTM*. Retrieved July 26, 2021, from https://learnopencv.com/human-action-recognition-using-detectron2-and-lstm/?ck_subscriber_id=873080389#conclusion
- Wicaksono, B. D. (2020). *Tanpa Peralatan Khusus, 9 Gerakan Gym Ini Bisa Kamu Lakukan di Rumah!* Retrieved March 21, 2020, from <https://www.idntimes.com/health/fitness/bayu/gerakan-gym-untuk-di-rumah-tanpa-peralatan/9>