# HALAMAN JUDUL



**UNIVERSITAS DIPONEGORO**

**KLASIFIKASI *HUMAN ACTION RECOGNITION* (HAR) BERBASIS MEDIAPIPE DAN *LONG SHORT-TERM MEMORY* (LSTM)**

**TUGAS AKHIR**

**Diajukan sebagai salah satu syarat untuk memperoleh gelar**

**Sarjana Teknik**

**ICHSAN ARSYI PUTRA**

**21120118120029**

**FAKULTAS TEKNIK**

**DEPARTEMEN TEKNIK KOMPUTER**

**SEMARANG**

**2022**

# HALAMAN PENGESAHAN

Tugas Akhir ini diajukan oleh

Nama : Ichsan Arsyi Putra

NIM : 21120118120029

Jurusan/Program Studi : Teknik Komputer

Judul Tugas Akhir : Klasifikasi *Human Action Recognition* (HAR)

Berbasis MediaPipe dan *Long Short-Term Memory*

(LSTM)

Telah berhasil dipertahankan di hadapan Tim Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Teknik pada Jurusan/Program Studi Teknik Komputer, Fakultas Teknik, Universitas Diponegoro.

TIM PENGUJI

Pembimbing I : Dr. Oky Dwi Nurhayati S.T., M.T. ( )

Pembimbing II : Dania Eridani S.T., M.Eng. ( )

Ketua Penguji : ( )

Anggota Penguji : ( )

Semarang, 2022

Ketua Departemen Teknik Komputer

Dr. Adian Fatchur Rochim, S.T., M.T.

NIP. 197302261998021001

# HALAMAN PERNYATAAN ORISINALITAS

**Tugas Akhir ini adalah hasil karya saya sendiri, dan semua sumber baik yang dikutip maupun yang dirujuk telah saya nyatakan dengan benar**

Nama : Ichsan Arsyi Putra

NIM : 21120118120029



Tanda Tangan :

Tanggal : 2022

# HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI

**TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS**

Sebagai civitas akademika Universitas Diponegoro, saya yang bertanda tangan di bawah ini:

Nama : Ichsan Arsyi Putra

NIM : 21120118120029

Jurusan/Program Studi : Teknik Komputer

Fakultas : Teknik

Jenis Karya : Tugas Akhir

demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Diponegoro **Hak Bebas Royalti Noneksklusif** (*Non-exclusive Royalty Free Right*) atas karya ilmiah saya yang berjudul:

**Klasifikasi *Human Action Recognition* (HAR) Berbasis MediaPipe dan *Long Short-Term Memory* (LSTM)**

Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti/Noneksklusif ini, Universitas Diponegoro berhak menyimpan, mengalihmedia/formatkan, mengelola, dalam bentuk pangkalan data (*database*), merawat, dan mempublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik hak cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Semarang

Pada Tanggal : 2022

Yang menyatakan

Ichsan Arsyi Putra

# KATA PENGANTAR

Puji syukur kehadirat Allah SWT yang telah melimpahkan rahmat-Nya sehingga penulis dapat menyelesaikan laporan tugas akhir yang berjudul “Klasifikasi *Human Action Recognition* (HAR) Berbasis MediaPipe dan *Long Short-Term Memory* (LSTM)”.

Laporan tugas akhir ini merupakan salah satu syarat untuk menyelesaikan studi di Departemen Teknik Komputer Falkutas Teknik Universitas Diponegoro. Selain itu, penyusunan laporan tugas akhir ini juga diharapkan dapat memberikan manfaat dalam bidang pendidikan.

Dalam penyusunan laporan tugas akhir ini, penulis banyak mendapatkan dukungan, doa, bimbingan, dan dorongan dari berbagai pihak. Oleh karena itu, penulis menyampaikan ucapan terima kasih kepada:

1. Bapak Dr. Adian Fatchur Rochim, S.T., M.T. selaku Ketua Departemen Teknik Komputer yang telah menyediakan lingkungan akademis yang kondusif bagi mahasiswa yang menulis tugas akhir.
2. Ike Pertiwi Windasari, S.T., M.T. selaku Koordinator Tugas Akhir yang telah memberikan persetujuan dan bimbingan dalam pelaksanaan kegiatan tugas akhir.
3. Ibu Dr. Oky Dwi Nurhayati S.T., M.T. selaku dosen pembimbing I yang telah memberikan bimbingan dan petunjuk dalam pembuatan tugas akhir.
4. Ibu Dania Eridani S.T., M.Eng. selaku dosen pembimbing II yang telah memberikan bimbingan dan petunjuk dalam pembuatan tugas akhir.
5. Bapak dan Ibu dosen Departemen Teknik Komputer yang telah memberikan materi perkuliahan berharga dalam kehidupan akademis penulis.
6. Bapak Samiran dan Ibu Ngatni selaku kedua orang tua yang telah memberikan dukungan mental, moral, spiritual, material, dan finansial kepada penulis.
7. Diana Nur Auliasari selaku kakak penulis yang telah memberikan saran berharga dalam penulisan tugas akhir.
8. Teman-teman angkatan 2018 selaku rekan seperjuangan yang telah menyediakan sarana diskusi serta memberikan saran yang berharga.
9. Kakak-kakak tingkat angkatan 2017, 2016, dan 2015 yang telah memberikan nasihat dalam perkuliahan dan kegiatan kampus lainnya.
10. Adik-adik tingkat angkatan 2019, 2020, dan 2021 yang selalu memberikan inspirasi dan semangat dalam aktif berkarya.
11. Serta semua pihak yang tidak dapat Penulis sebutkan satu-persatu yang telah membantu penyelesaian Tugas Akhir ini.

Penulis menyadari bahwa laporan tugas akhir ini masih memerlukan perbaikan di berbagai aspek. Oleh sebab itu, penulis mengharapkan kritik, saran, dan masukan dari pembaca. Akhir kata, penulis mengucapkan terima kasih dan semoga laporan tugas akhir ini dapat bermanfaat bagi orang banyak.

Semarang, 2022

Ichsan Arsyi Putra

# DAFTAR ISI

[HALAMAN JUDUL 1](#_Toc101156503)

[HALAMAN PENGESAHAN 2](#_Toc101156504)

[HALAMAN PERNYATAAN ORISINALITAS 3](#_Toc101156505)

[HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI 4](#_Toc101156506)

[KATA PENGANTAR 5](#_Toc101156507)

[DAFTAR ISI 7](#_Toc101156508)

[DAFTAR GAMBAR 9](#_Toc101156509)

[DAFTAR TABEL 11](#_Toc101156510)

[ABSTRAK 12](#_Toc101156511)

[ABSTRACT 13](#_Toc101156512)

[BAB I PENDAHULUAN 14](#_Toc101156513)

[1.1 Latar Belakang 14](#_Toc101156514)

[1.2 Rumusan Masalah 15](#_Toc101156515)

[1.3 Tujuan Tugas Akhir 15](#_Toc101156516)

[1.4 Batasan Masalah 15](#_Toc101156517)

[1.5 Manfaat Penelitian 16](#_Toc101156518)

[1.6 Sistematika Penulisan 16](#_Toc101156519)

[BAB II KAJIAN PUSTAKA 17](#_Toc101156520)

[2.1 Penelitian Terdahulu 17](#_Toc101156521)

[2.2 *Human Action Recognition* (HAR) 22](#_Toc101156522)

[2.2.1 Representasi Aksi 22](#_Toc101156523)

[2.2.2 Klasifikasi Aksi 22](#_Toc101156524)

[2.2.3 *Deep Network* 23](#_Toc101156525)

[2.3 Klasifikasi 24](#_Toc101156526)

[2.4 *Deep Learning* 24](#_Toc101156527)

[2.5 Pustaka MediaPipe 25](#_Toc101156528)

[2.6 *Long Short-Term Memory* 26](#_Toc101156529)

[2.7 *Dataset* Weizmann 28](#_Toc101156530)

[2.8 *Data Preprocessing* 29](#_Toc101156531)

[2.8.1 *Feature Selection* 29](#_Toc101156532)

[2.8.2 *Data Labeling* 29](#_Toc101156533)

[2.9 *Data Augmentation* 29](#_Toc101156534)

[2.10 Model OSEMN 29](#_Toc101156535)

[2.11 *Confusion Matrix* 31](#_Toc101156536)

[BAB III METODE PENELITIAN 34](#_Toc101156537)

[3.1 Spesifikasi Perangkat Penelitian 34](#_Toc101156538)

[3.1.1 Spesifikasi Perangkat Keras 34](#_Toc101156539)

[3.1.2 Spesifikasi Perangkat Lunak 34](#_Toc101156540)

[3.2 Model OSEMN 35](#_Toc101156541)

[3.2.1 *Obtain*-1 35](#_Toc101156542)

[3.2.2 *Scrub*-1 37](#_Toc101156543)

[3.2.3 *Explore*-1 39](#_Toc101156544)

[3.2.4 *Obtain*-2 39](#_Toc101156545)

[3.2.5 *Scrub*-2 40](#_Toc101156546)

[3.2.6 *Explore*-2 40](#_Toc101156547)

[3.2.7 *Model* 41](#_Toc101156548)

[3.2.8 *Interpret* 43](#_Toc101156549)

[BAB IV HASIL DAN PEMBAHASAN 45](#_Toc101156550)

[4.1 Analisis dan Pembahasan 45](#_Toc101156551)

[4.1.1 Analisis *Training* dan *Validation* 45](#_Toc101156552)

[4.1.2 Evaluasi *Testing* 49](#_Toc101156553)

[4.2 Implementasi Model pada Video Demo 53](#_Toc101156554)

[BAB V PENUTUP 56](#_Toc101156555)

[5.1 Kesimpulan 56](#_Toc101156556)

[5.2 Saran 57](#_Toc101156557)

[DAFTAR PUSTAKA 58](#_Toc101156558)

[BIODATA MAHASISWA 65](#_Toc101156559)

[LAMPIRAN 66](#_Toc101156560)

# DAFTAR GAMBAR

[Gambar 2. 1 Ilustrasi hubungan antara *Artificial Intelligence*, *Machine Learning*, *Deep Learning*, dan *Deep Neural Network* [26] 25](#_Toc100637432)

[Gambar 2. 2 Contoh struktur *Deep Neural Network* sederhana [27] 26](#_Toc100637433)

[Gambar 2. 3 Deteksi objek menggunakan MediaPipe [28] 27](#_Toc100637434)

[Gambar 2. 4 *Pose Landmarks* [30] 27](#_Toc100637435)

[Gambar 2. 5 Ilustrasi algoritma LSTM [33] 28](#_Toc100637436)

[Gambar 2. 6 *Dataset* Weizmann 29](#_Toc100637437)

[Gambar 2. 7 Perbandingan ekspektasi dan realitas penggunaan model OSEMN [39] 31](#_Toc100637438)

[Gambar 3. 1 Alur kerja penelitian tugas akhir 36](#_Toc100637439)

[Gambar 3. 2 Jumlah video per kelas pada *dataset* Weizmann 38](#_Toc100637440)

[Gambar 3. 3 Satu detik awal (25 *frames*) pada video “ira\_bend.avi” sebelum (kiri) dan sesudah pemotongan durasi (kanan) 38](#_Toc100637441)

[Gambar 3. 4 Jumlah video per kelas pada *dataset* Weizmann hasil *data augmentation* 40](#_Toc100637442)

[Gambar 3. 5 Ilustrasi ekstraksi ciri video menggunakan pustaka MediaPipe: Pose 40](#_Toc100637443)

[Gambar 3. 6 Arsitektur Model 1 43](#_Toc100637444)

[Gambar 3. 7 Arsitektur Model 2 43](#_Toc100637445)

[Gambar 3. 8 Arsitektur Model 3 44](#_Toc100637446)

[Gambar 3. 9 Beberapa peragaan aksi pada video demo 45](#_Toc100637447)

[Gambar 4. 1 Grafik *loss* (a) dan *accuracy* (b) pada proses *training* dan *validation* terhadap Model 1 46](#_Toc100637448)

[Gambar 4. 2 Grafik *loss* (a) dan *accuracy* (b) pada proses *training* dan *validation* terhadap Model 2 47](#_Toc100637449)

[Gambar 4. 3 Grafik *loss* (a) dan *accuracy* (b) pada proses *training* dan *validation* terhadap Model 3 49](#_Toc100637450)

[Gambar 4. 4 *Confusion matrix* hasil proses *testing* terhadap Model 1 50](#_Toc100637451)

[Gambar 4. 5 *Confusion matrix* hasil proses *testing* terhadap Model 2 51](#_Toc100637452)

[Gambar 4. 6 *Confusion matrix* hasil proses *testing* terhadap Model 3 52](#_Toc100637453)

[Gambar 4. 7 Deteksi pjump oleh Model 1 (a), deteksi side oleh Model 2 (b), dan deteksi pjump oleh Model 3 (c) 55](#_Toc100637454)

[Gambar 4. 8 Deteksi wave1 oleh Model 1 (a) dan Model 2 (b) serta deteksi walk oleh Model 3 (c) 55](#_Toc100637455)

[Gambar 4. 9 Hasil deteksi aksi skip oleh Model 1 (a), Model 2 (b), dan Model 3 (c) 56](#_Toc100637456)

[Gambar 4. 10 Deteksi aksi wave2 (a) dan jack (b) oleh Model 1 56](#_Toc100637457)

# DAFTAR TABEL

[Tabel 2. 1 Perbandingan metode dan hasil penelitian terdahulu 17](#_Toc100638336)

[Tabel 2. 2 Jenis-jenis klasifikasi aksi 22](#_Toc100638337)

[Tabel 2. 3 *Confusion matrix* 31](#_Toc100638338)

[Tabel 2. 4 Rumus-rumus *confusion matrix* pada masalah klasifikasi banyak kelas 32](#_Toc100638339)

[Tabel 3. 1 Jumlah video tiap kelas pada *dataset* Weizmann 35](#_Toc100638341)

[Tabel 3. 2 Contoh hasil proses *data augmentation* 38](#_Toc100638342)

[Tabel 3. 3 Nama kelas dan kode angka 40](#_Toc100638343)

[Tabel 3. 4 Dimensi *array* pada data X 40](#_Toc100638344)

[Tabel 3. 5 Dimensi *array* pada data y 41](#_Toc100638345)

[Tabel 3. 6 Inisialisasi nilai *hyperparameter* yang sama pada ketiga model 41](#_Toc100638346)

[Tabel 3. 7 Perubahan *hyperparameter* pada Model 1 42](#_Toc100638347)

[Tabel 3. 8 Perubahan *hyperparameter* pada Model 2 43](#_Toc100638348)

[Tabel 4. 1 Hasil nilai TP, FP, TN, dan FN dari proses *testing* Model 1 50](#_Toc100638349)

[Tabel 4. 2 Hasil nilai TP, FP, TN, dan FN dari proses *testing* Model 2 51](#_Toc100638350)

[Tabel 4. 3 Hasil nilai TP, FP, TN, dan FN dari proses *testing* Model 3 52](#_Toc100638351)

[Tabel 4. 4 Hasil pengukuran kualitas klasifikasi tiap model 52](#_Toc100638352)

[Tabel 4. 5 Perubahan deteksi aksi oleh tiap model 53](#_Toc100638353)

# ABSTRAK

*Pengembangan teknologi* Human Action Recognition *(HAR) dalam domain* Computer Vision *masih terus dilakukan. HAR dapat diimplementasikan pada beragam aplikasi serta dengan bermacam-macam metode deep learning, seperti* ConvolutionalNeuralNetwork *(CNN),* HiddenMarkovModel *(HMM),* SupportVectorMachine *(SVM), dan* LongShort-TermMemory *(LSTM). Namun demikian, penggunaan metode-metode tersebut sering kali diikuti oleh proses operasi citra yang lama dan memerlukan spesifikasi alat yang cukup tinggi. Untuk mengatasi permasalahan tersebut, peneliti menggunakan data ekstraksi* landmarks *citra yang diperoleh melalui deteksi video oleh pustaka* MediaPipe*.*

*Penelitian dilaksanakan berdasarkan metode OSEMN yang terdiri atas tahap* Obtain*,* Scrub*,* Explore*,* Model*, dan* iNterpret*.* Dataset *yang digunakan dalam penelitian adalah* dataset *Weizmann sehingga terdapat 10 kelas aksi dalam bentuk video. Sebelum digunakan,* dataset *Weizmann dikenakan proses* data preprocessing *dan* data augmentation*. Ekstraksi ciri dilakukan terhadap video menggunakan pustaka* MediaPipe*:* Pose *dan menghasilkan data* landmarks*. Proses* training *dan* validation *dilakukan dengan tiga variasi arsitektur model* neural network *yang berfokus pada* layerLong Short-Term Memory *(LSTM). Evaluasi performa model dilakukan melalui interpretasi* confusion matrix *dan perhitungan nilai* accuracy*,* loss*,* precision*, dan* F1score*.*

*Hasil penelitian yang diperoleh berupa tiga model arsitektur dengan dengan nilai* accuracy *mencapai 84%. Berdasarkan hasil proses* testing*, klasifikasi terhadap aksi* run *dan* walk *sulit dilakukan dengan perolehan nilai* True Positive *mulai dari 3 hingga 6. Di sisi lain, klasifikasi yang baik terjadi pada aksi* skip*,* jack,jump*,* pjump*,* side*,* wave2*,* wave1*, dan* bend *dengan nilai* TruePositive *mulai dari 7 hingga 9.*

***Kata kunci:*** *Klasifikasi,* Deep Learning*,* Human Action Recognition*,* MediaPipe*,* Long Short-Term Memory

# ABSTRACT

*The development of Human Action Recognition (HAR) technology in the Computer Vision domain is still far from being done. HAR can be implemented in various applications as many as the number of deep learning methods, e.g. Convolutional Neural Network (CNN), Hidden Marcov Model (HMM), Support Vector Machine (SVM), and Long Short-Term Memory (LSTM). However, the research methods were often followed by a long imagery operation process and required high specification tools. To address this problem, we implemented image landmarks data extracted from MediaPipe library’s video detection.*

*This research was conducted under OSEMN methods consisting of five main processes namely Obtain, Scrub, Explore, Model, and iNterpret. The dataset used in this research was preprocessed Weizmann dataset with data preprocessing and data augmentation implementations. The video feature extracting process was done by MediaPipe: Pose library. Training and validation processes were conducted on three variants of the neural network model focusing on LSTM layers. The processes were finished by model performance evaluation based on confusion matrices interpretation and calculations of accuracy, error rate, precision, recall, and F1score.*

*This research yielded three model variants with the highest accuracy at 84%. From the testing process, we found that the classifications on actions run and walk were hard to be done proven by various True Positive (TP) values starting from 3 to 6. In contrast, the better classifications were done on actions skip, jack, jump, pjump, side, wave2, wave1, and bend proven by TP values starting from 7 to 9.*

***Keywords:*** *Classification, Deep Learning, Human Action Recognition, MediaPipe, Long Short-Term Memory*

# BAB I PENDAHULUAN

## Latar Belakang

Pengembangan teknologi komputer di bidang *Machine Learning* dan *Computer Vision* masih terus dilakukan. Dalam bidang *Computer Vision*, informasi dari data citra dengan berbagai sumber menjadi bahan yang menarik untuk diteliti. Salah satu informasi yang dapat diperoleh dari citra adalah Pengenalan Aksi Manusia atau *Human Action Recognition* (HAR). *Human* *Action Recognition* merupakan topik yang sangat penting karena pengaplikasiannya yang beragam, di antaranya pada video pengawas, interaksi manusia-mesin, dan perolehan informasi lain dari video [1]. Aksi manusia dikenali dengan berbagai macam ciri yang dimiliki citra dan diolah dengan beragam metode *deep learning*. Metode-metode yang pernah diajukan antara lain *Convolutional Neural Network* (CNN), *Hidden Markov Model* (HMM), *Support Vector Machine* (SVM), dan *Long Short-Term Memory* (LSTM) [2].

Namun demikian, penggunaan metode-metode tersebut sering kali diikuti oleh proses operasi citra yang lama dan memerlukan spesifikasi alat yang cukup tinggi [3]. Selain itu, pada beberapa kasus, aksi manusia hanya dapat dikenali berdasarkan kondisi yang berurutan secara temporal sehingga data dalam bentuk video sering digunakan. Hal ini tentu akan menambah beban komputasi dan memperlama durasi proses yang terjadi [4].

Untuk mengatasi permasalahan beban spesifikasi tersebut, peneliti mengusulkan penggunaan data ekstraksi *landmarks* citra yang diperoleh melalui deteksi video oleh pustaka MediaPipe [5]. Data yang diperoleh akan diurutkan berdasarkan waktu dan akan diproses melalui salah satu jenis *Artificial Neural Network* (ANN), yaitu *Long Short-Term Memory* (LSTM).

## Rumusan Masalah

Rumusan masalah yang dapat disusun berdasarkan latar belakang di atas adalah sebagai berikut.

1. Bagaimana metode *deep learning* dapat digunakan untuk klasifikasi *Human Action Recognition*?
2. Bagaimana susunan model *neural* *network* LSTM yang dihasilkan dapat digunakan dalam *Human Action Recoginition* berdasarkan ekstrasi ciri MediaPipe?

## Tujuan Tugas Akhir

Penelitian tugas akhir ini bertujuan untuk membuat model *Artificial Neural Network* pada masalah klasifikasi *Human Action Recognition* berdasarkan hasil ekstraksi ciri *landmarks* citra oleh pustaka MediaPipe dan jaringan *Long Short-Term Memory*.

## Batasan Masalah

Berdasarkan latar belakang masalah di atas, peneliti menentukan batasan-batasan masalah sebagai berikut.

1. Data penelitian yang digunakan dalam penelitian ini adalah *dataset* Weizmann [6].
2. Jenis atau kelas aksi manusia yang akan dideteksi berjumlah 10 kelas sesuai label yang terdapat pada *dataset* Weizmann [6], yaitu *Walk, Run, Jump, Gallop sideways, Bend, One-hand wave, Two-hands wave, Jump in place, Jumping* *Jack*, dan *Skip*.
3. Pustaka utama yang paling berperan yaitu MediaPipe [5] dan *Long Short-Term Memory Recurrent Neural Network* (RNN) berbasis pustaka Keras [7].
4. Dataset dibagi ke dalam tiga bagian, yaitu data *training, validation*, dan *testing* dengan rasio 60:20:20 [8].
5. Data yang diambil sejumlah 25 *frames* gambar pada satu detik pertama tiap video.

## Manfaat Penelitian

Manfaat yang diperoleh dari penelitian ini yaitu:

1. Mendapatkan model *neural network* LSTM untuk *Human Action Recognition* berdasarkan ekstraksi ciri MediaPipe.
2. Membantu peneliti lain sebagai referensi penelitian dengan tujuan, variabel, atau metode sejenis.
3. Menjadi sarana implementasi teori *artificial neural network* dalam masalah *Human Action Recognition.*

## Sistematika Penulisan

Laporan tugas akhir ini terdiri atas lima bab dengan sistematika penulisan berikut.

1. BAB I PENDAHULUAN

Bab I membahas mengenai latar belakang, rumusan masalah, tujuan, batasan masalah, manfaat tugas akhir, dan sistematika penulisan laporan tugas akhir.

1. BAB II KAJIAN PUSTAKA

Bab kedua ini berisi kajian mengenai penelitian-penelitian terdahulu dengan topik serupa serta teori-teori yang mendukung penelitian tugas akhir.

1. BAB III METODE PENELITIAN

Bab ini menguraikan tentang spesifikasi sistem dan jenis metode atau langkah kerja yang digunakan dalam penelitian tugas akhir ini.

1. BAB IV HASIL DAN PEMBAHASAN

Pada bab IV, peneliti menjelaskan implementasi, hasil uji, dan hasil evaluasi dari penelitian “Klasifikasi *Human Action Recognition* (HAR) Berbasis MediaPipe dan *Long Short-Term Memory* (LSTM)”.

1. BAB V PENUTUP

Sebagai penutup, peneliti memberikan kesimpulan yang diperoleh berdasarkan hasil penelitian dengan disertai saran-saran yang dapat digunakan dalam penelitian selanjutnya.

# BAB II KAJIAN PUSTAKA

## Penelitian Terdahulu

Berdasarkan literatur jurnal penelitian terdahulu, peneliti menemukan bahwa terdapat penelitian-penelitian yang berkaitan dengan penelitian tugas akhir ini. Tabel 2. 1 menunjukkan daftar penelitian terdahulu dengan disertai jenis metode dan hasil penelitiannya.

Tabel 2. 1 Perbandingan metode dan hasil penelitian terdahulu

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Judul Penelitian | Metode\* | Hasil Penelitian |
| 1. | *Deep Progressive Reinforcement Learning for Skeleton-based Action Recognition* | GCNN | Penelitian menggunakan data input berupa jumlah *frame*, jumlah titik sendi, dan 3 buah koordinat lokasi sendi berbentuk graf. Model *neural network* yang digunakan terdiri atas 3 *convolutional layers* dan 3 *max pooling layers*, dan 3 *fully-connected layes*. *Accuracy* hasil percobaan pada *dataset* NTU, SYSU, dan UT berturut-turut mencapai 89,8%, 76,9%, dan 98,5% [9]. |
| 2. | *Robust Human Action Recognition via Long Short-Term Memory* | LSTM | Penelitian dilakukan terhadap *dataset* KTH dengan teknik ekstraksi ciri menggunakan *histogram of optical flow* (HOF) *descriptor*. Model *neural network* tersusun atas 90 *input neuron*, 50 *memory blocks*, dan 6 *output neurons*. Proses *training* dilakukan dengan *learning rate* 0.0001 dan 2000 *epochs*. Keakuratan klasifikasi mencapai 90,7% [10]. |
| 3. | *Online Video Object Detection using Association* LSTM | LSTM | *Dataset* yang digunakan adalah *Youtube-Object* dan *MOTChallenge* yang diterapkan proses *data* *augmentation* dengan teknik *mirrorring*, *random* *crop*, dan *reversed sequence*. Proses ekstraksi fitur dilakukan dengan *Single shot multibox detector* (SSD). Model dilatih menggunakan konfigurasi arsitektur 2 LSTM *layers*, RMSProp *optimizer*, *decay rate* 0,9, dan 200 *epochs*. Hasil evaluasi berupa nilai *tracking performance* (mAP) yang mencapai 72,14 [3]. |
| 4. | *Automatic Parental Guide Scene Classification Menggunakan Metode Deep Convolutional Neural Network dan* LSTM | CNN dan LSTM | *Dataset* terdiri atas kumpulan video *Youtube* dan cuplikan film yang disusun secara manual dengan 6 klasifikasi (*violence* *and* *gore*, *sex and nudity, frightening, alcohol, smoke and drugs, profanity*, dan *none*). Klasifikasi *subtitle* menggunakan LSTM sedangkan klasifikasi audio dan video menggunakan CNN. Berdasarkan penelitian, diketahui penggunaan *batch normalization* layer dapat memberikan nilai akurasi dengan rata-rata 83,5% meskipun dengan jumlah data *training* yang relatif kecil [11]. |
| 5. | Deteksi Aktivitas Manusia Berdasarkan Data *Skeleton* dengan Menggunakan Modifikasi VGG16 | CNN | Penelitian menggunakan *dataset* *skeleton* yang diperoleh melalui perangkat Microsoft Kinect 2. Model *neural network* menggunakan arsitektur VGG16 dengan modifikasi *fully* *connected* *layer*, *average* *pooling* *layer*, dan *dropout*. Penggunaan CNN dengan representasi data *skeleton* dalam bentuk gambar dirasa kurang tepat. Model modifikasi VGG16 menghasilkan akurasi *training* rata-rata sebesar 90,22% dan akurasi testing hingga 54,59% [12]. |
| 6. | *Hierarchical Recurrent Neural Network for Skeleton Based Action Recognition* | BRNN dan LSTM | Arsitektur *neural network* terdiri atas beberapa *Bidirectional recurrent neural networks layers* (BRNN) yang diakhiri dengan LSTM *layer*. Varian model HBRNN-L memiliki akurasi percobaan berdasarkan *dataset* MSR Action3D, Berkeley MHAD, dan HDM05 secara berurutan sebesar 94,49%, 100%, dan 96,92% [13]. |
| 7. | *Human Action Recognition Using Support Vector Machines and* 3D *Convolutional Neural Networks* | 3D CNN dan SVM | Penelitian dilakukan terhadap *dataset* KTH. Ekstraksi ciri spasial dan temporal menggunakan 3D CNN dengan data berupa *frame* video dan *optical flow*. Proses klasifikasi menggunakan SVM menghasilkan performa yang baik dengan *accuracy* 90,34% [14]. |
| 8. | *Hierarchical* LSTM *with Adjusted Temporal Attention for Video Captioning* | CNN dan LSTM | *Neural network* disusun secara hierarkis antara CNN layer (pengolah vektor fitur), 2 LSTM *layers*, *attention layer*, dan MLP *layer*. Integrasi antara *hierarchical* LSTM dan automasi pemilihan fitur temporal mampu bekerja dengan baik pada *low-level video* *visual features* dan *language context information* dengan nilai performa sebesar 82,9% [15]. |
| 9. | *Human Activity Recognition Using Support Vector Machine for Automatic Security System* | SVM | Pengumpulan data dilakukan menggunakan Kinect *camera* yang menghasilkan data *skeleton*. Melalui penelitian ini, diketahui algoritma SVM mampu bekerja secara optimal pada data yang diperoleh dari jarak 1 meter dengan akurasi 100% sedangkan akurasi pada jarak 2 dan 3 meter sebesar 91% dan 81% [16]. |
| 10. | *Real-time Vernacular Sign Language Recognition using MediaPipe and Machine Learning* | SVM | Data dikumpulkan melalui pustaka MediaPipe *Hands* yang kemudian diklasifikasikan dengan algoritma SVM. Akurasi rata-rata 99% pada berbagai macam *sign language datasets* (*American*, *Indian*, *Italian*, dan *Turkey*). MediaPipe dapat menjadi alat yang efisien untuk mendeteksi gestur tangan yang kompleks dengan teliti [17]. |
| 12. | *Thai Finger Spelling Recognition: Investigating MediaPipe Hands Potentials* | ANN dan RB | *Dataset* diperoleh melalui deteksi pustaka MediaPipe *Hands* terhadap *single-hand dataset* dan *point-on-hand dataset*. Proses klasifikasi menggunakan pendekatan *Thai Finger Spelling* (TFS) VGG-*based*. Akurasi deteksi tangan statis dan dinamis sebesar 88,57%. Klasifikasi mencapai akurasi 77,78% dan 69,19% secara berurutan pada latar belakang polos dan kompleks [18]. |
| 13. | Sistem Pengenalan Aktivitas Manusia Menggunakan *Long Short-Term Memory* dan Mediapipe | LSTM | Penelitian dilakukan terhadap *Dataset* RNN *for Human Activitiy Recognition*-2D. Proses *training* dilakukan dengan sebanyak 400 *epochs*, *batch* *size* sebesar 512, dan *learning* *rate* 0,001 pada Adam *optimizer*. Hasil *training* memberikan nilai akurasi sebesar 91% dan *loss* sebesar 0.29% dengan rata-rata waktu *training* selama 10-12 menit [19]. |

\*Keterangan:

GCNN : *Graph-based Convolutional Neural Network*

LSTM : *Long Short-Term Memory*

BRNN : *Bidirectional Recurrent Neural Network*

CNN : *Convolutional Neural Network*

SVM : *Support Vector Machine*

ANN : *Artificial Neural Network*

RB : *Rule-Based*

Berdasarkan Tabel 2. 1, penelitian yang menggunakan metode SVM [14], [16], [17], CNN [11], [12], [14], BRNN [13], DPRL, dan GCNN [9] menunjukkan hasil dengan akurasi yang tinggi. Meskipun demikian, *dataset* yang diolah cukup besar dan memerlukan perangkat keras pendukung dengan spesifikasi tinggi [3]. Di sisi lain, penelitian menggunakan metode LSTM mampu menghasilkan akurasi yang baik [3], [10], [19], meskipun dengan video kualitas rendah [15] dan data *training* yang relatif kecil [11].

Dengan demikian, penelitian tugas akhir ini akan menerapkan metode LSTM [7] dengan data masukan yang diperoleh dari ekstraksi ciri *landmarks* citra yang dihasilkan melalui pustaka MediaPipe [5]. Penerapan metode dan pustaka tersebut diharapkan mampu menghasilkan model ANN berkemampuan cukup baik dan proses pengolahan yang tidak memerlukan perangkat berspesifikasi tinggi.

## *Human Action Recognition* (HAR)

*Human Action Recognition* merupakan kegiatan mengenali aksi manusia melalui video yang mengandung eksekusi lengkap dari suatu aksi [20]. Sedangkan aksi adalah urutan dari beberapa tindakan primitif untuk memenuhi tujuan yang sederhana, seperti melompat, berjalan, dan menendang [21].

Menurut Y. Kong dan Y. Fu [20], permasalahan pengenalan aksi secara umum terdiri atas dua komponen, yaitu representasi aksi dan klasifikasi aksi. Kedua komponen tersebut selanjutnya dapat dikombinasikan ke dalam suatu *deep network*.

### Representasi Aksi

Representasi aksi merupakan vektor fitur yang diperoleh dari hasil konversi video. Representasi aksi terbagi atas dua jenis, yaitu representasi lokal dan holistik [20]. Representasi lokal hanya mengidentifikasi area lokal yang memiliki informasi gerakan yang menonjol sedangkan representasi holistik menangkap seluruh informasi spasial dan temporal gerakan manusia secara dinamis.

### Klasifikasi Aksi

Komponen klasifikasi aksi terdiri atas tujuh jenis klasifikasi [20] seperti yang tertera pada Tabel 2. 2.

Tabel 2. 2 Jenis-jenis klasifikasi aksi

|  |  |  |
| --- | --- | --- |
| No. | Klasifikasi Aksi | Cara Kerja |
| 1. | *Direct classification* | Menghimpun vektor fitur dari suatu video kemudian melakukan klasifikasi secara langsung menggunakan *classifiers* yang sudah tersedia, seperti *Support Vercor Machine* (SVM) atau K-*Nearest Neighbor* (KNN). |
| 2. | *Sequential approaches* | Menghimpun perubahan penampilan atau pose di dalam video menggunakan *sequential model*, seperti *Hidden Marcov Model* (HMMs), *Structured Supprot Vector Machine* (SSVM), dan *Conditional Random Fields* (CRFs). |
| 3. | *Space-time approaches* | Bekerja seperti *direct classification* namun dengan penambahan data fitur berupa korelasi antarfitur lokal secara *spatio-temporal* dan distribusi global *spatio*-*temporal* dari *interest point*. Metode yang digunakan, seperti *Directional Pyramid Co-ocurrence Matrix* (DPCM), dan *Contex Dependent Graph Kernels* (CGKs). |
| 4. | *Part-based approaches* | Memodelkan aksi berdasarkan informasi gerakan dari tubuh secara keseluruhan dan secara per bagian. |
| 5. | *Manifold learning aproaches* | Mengklasifikasikan aksi berdasarkan variasi perubahan siluet tubuh manusia. |
| 6. | *Mid-level feature approaches* | Menggunakan fitur *mid-level* (antara fitur *low-level* dan *high-level*) untuk penanganan derau yang lebih baik dan variasi aksi yang lebih beragam. |
| 7. | *Feature fusion approaches* | Memanfaatkan gabungan beberapa jenis fitur yang diperoleh dari video. Contoh penerapan jenis klasifikasi ini terdapat pada *Multi-Task Sparse* *Learning* (MTSL) dan *Multi-feature Max-Margin Hierarchical Bayesian Model* (M3HBM). |

### *Deep Network*

*Deep network* dikembangkan untuk mengatasi masalah kesulitan ekstraksi fitur secara manual dan generalisasi pada *dataset* yang besar [20]. Terdapat tiga jenis *deep networks* yang digunakan dalam *action recognition*. Ketiga jenis *deep networks* tersebut, ialah *space-time networks*, *multi-stream networks*, dan *hybrid networks*.

*Space-time networks* merupakan pengembangan dari 2D ConvNets yang dapat memproses data *spatial* dan *temporal* menggunakan 3D *convolutions*. Contoh arsitektur yang menggunakan 3D *convolutions* adalah C3D. Jenis kedua, *multi-stream networks* memodelkan *convolutional networks* terhadap fitur tampilan dan fitur gerakan. Hasil dari keduanya digabungkan melalui sebuah *pooling layer* seperti ActionVLAD. Jenis ketiga, *hybrid networks* menggabungkan informasi *temporal* menggunakan *recurrent layer* di atas CNN. Contoh dari *recurrent layer* yang biasa digunakan adalah LSTM [20].

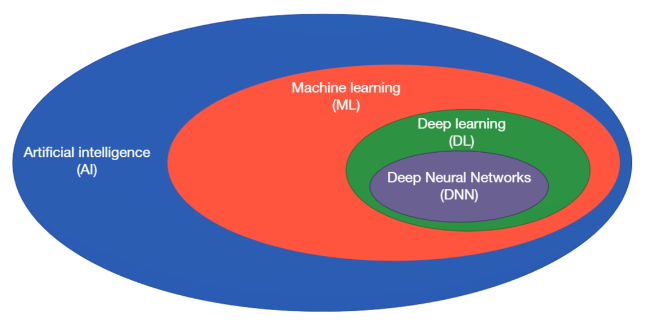
## Klasifikasi

Klasifikasi adalah sebuah teknik dalam *data mining* (*machine learning*) yang digunakan untuk memprediksi anggota kelompok atau kelas dari contoh-contoh data [22]. Secara proses, klasifikasi dapat dilakukan secara manual maupun dengan bantuan teknologi. Klasifikasi manual dilakukan langsung oleh manusia sedangkan klasifikasi dengan teknologi dapat dilakukan melalui beberapa cara atau algoritma, di antaranya *Naïve Bayes, Support Vector Machine* (SVM), *Decission Tree*, *Fuzzy*, dan *Artificial Neural Network* (ANN) [23].

Klasifikasi dilakukan dengan mengelompokkan vektor fitur ke dalam kelas-kelas yang sesuai. Vektor fitur pelatihan yang telah tersedia sebelumnya dan telah diketahui kelas-kelasnya kemudian dijadikan dasar untuk membangun pengklasifikasi. Dengan demikian klasifikasi disebut sebagai metode terbimbing (*supervised*) [24].

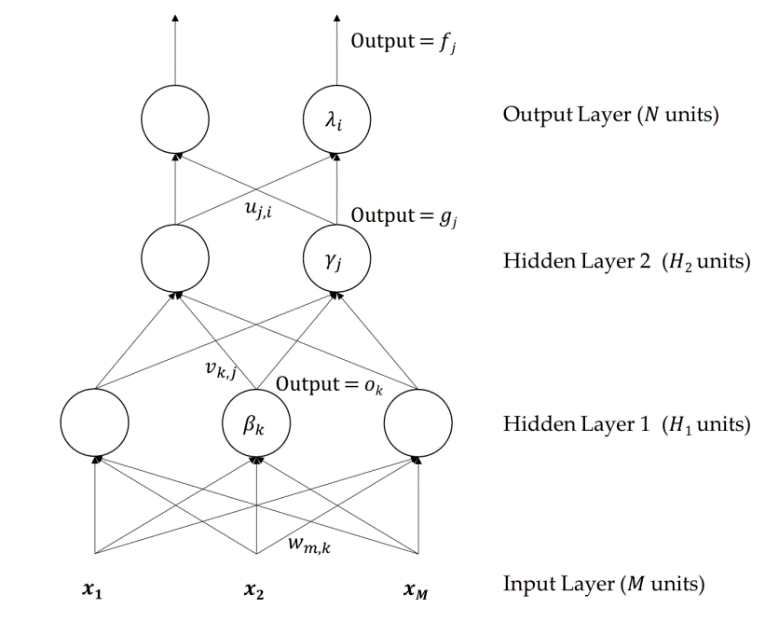
## *Deep Learning*

*Deep learning* adalah salah satu bagian dari algoritma *machine learning* yang tersusun atas banyak lapisan atau *layer* (unit komputasional). Masing-masing layer tersebut berperan dalam mempelajari cara merepresentasikan data input yang hasilnya akan dikombinasikan dengan layer berikutnya secara hierarkis. Bagian mendasar dari *deep learning* yaitu *Deep Neural Network* (DNN) yang terinspirasi dari neuron makhluk hidup [25]. Hubungan antara *Artificial Intelligence*, *Machine Learning*, *Deep Learning*, dan *Deep Neural Network* diilustrasikan melalui Gambar 2. 1.



Gambar 2. 1 Ilustrasi hubungan antara *Artificial Intelligence*, *Machine Learning*, *Deep Learning*, dan *Deep Neural Network* [25]

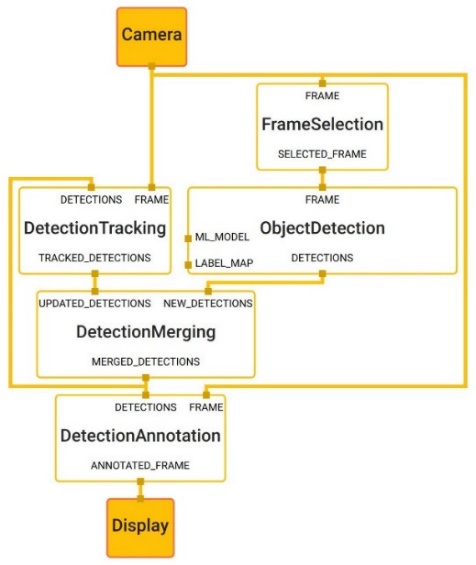
Secara umum, *deep neural nework* memiliki lebih dari 3 *layers* yang terdiri atas *input* *layer*, N >= 2 *hidden layers*, dan *output layer*. Susunan sederhana ketiga layers ini dapat dilihat pada Gambar 2. 2 Jumlah *layer* yang banyak tersebut menyebabkan *neural network* ini disebut “*deep*” (dalam). Salah satu kelebihan dari *deep neural network* dibandingkan dengan *neural network* yang lebar dan sedikit *layer* adalah kemampuannya dalam mengubah data dari *non-linearly separable* menjadi *linearly separabel* melalui proses transformasi di *hidden layers* [26].



Gambar 2. 2 Contoh struktur *Deep Neural Network* sederhana [26]

## Pustaka MediaPipe

MediaPipe merupakan sebuah kerangka kerja dalam pembuatan pipeline untuk melakukan prediksi terhadap data sensoris yang berubah-ubah. MediaPipe didesain untuk kepentingan praktisi *machine learning* (ML), seperti peneliti, pelajar, dan pengembang *software* yang mengimplementasikan aplikasi *machine* *learning* siap pakai, publikasi kode penyerta hasil penelitian, maupun pembuatan purwarupa dari suatu teknologi [27]. Alur kerja pustaka MediaPipe ditunjukkan dengan Gambar 2. 3.



Gambar 2. 3 Deteksi objek menggunakan MediaPipe [27]

Salah satu fitur yang dimiliki oleh MediaPipe adalah MediaPipe Pose yang dapat digunakan untuk keperluan aplikasi *human pose estimation* [5]. Fitur ini mampu mendeteksi 33 *landmarks* 3D seperti pada Gambar 2. 4 dan segmentasi latar belakang di seluruh tubuh dari *frame* video berformat warna RGB. MediaPipe Pose didasarkan pada penelitian BlazePose GHUM 3D [28] dan mendukung ML Kit Pose Detection API [29].

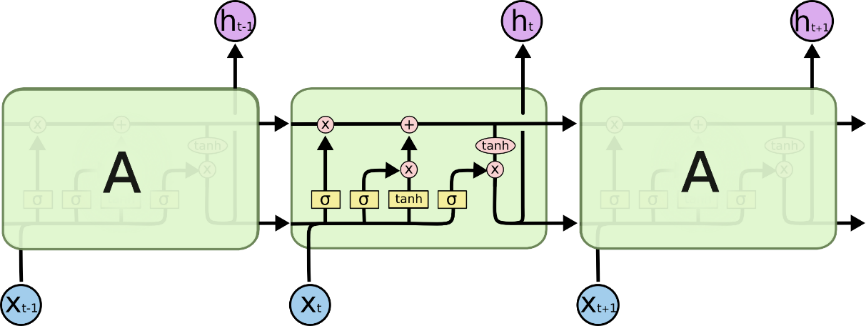


Gambar 2. 4 *Pose Landmarks* [29]

## *Long Short-Term Memory*

*Long Short-Term Memory* (LSTM) merupakan jenis arsitektur *Recurrent Neural Network* (RNN) yang didesain untuk model rangkaian temporal yang memungkinkan ketergantungan berjangkauan luas yang lebih akurat dari RNN tradisional [30]. LSTM pertama kali diperkenalkan oleh Hochreiter dan Schmidhuber [31] untuk mengatasi masalah *error back-flow* berupa *blow up* atau *vanish* pada metode *backpropagation*.

Secara visual, skema algoritma LSTM diilustrasikan melalui Gambar 2. 5 berikut.



Gambar 2. 5 Ilustrasi algoritma LSTM [32]

Secara matematis, LSTM memiliki kalkulasi dengan urutan sebagai berikut [33].

|  |  |  |
| --- | --- | --- |
|  |  | (2.1) |
|  |  | (2.2) |
|  |  | (2.3) |
|  |  | (2.4) |
|  |  | (2.5) |
|  |  | (2.6) |

dengan:

= *forget gate*

= fungsi sigmoid

= nilai *weight* untuk *forget gate*

= nilai *output* sebelum orde ke-t

= nilai input pada orde ke-t

= nilai bias pada *forget gate*

= input *gate*

= nilai *weight* untuk input *gate*

= nilai bias pada input *gate*

= nilai baru yang dapat ditambahkan ke *cell state*

= fungsi tanh

= nilai *weight* untuk *cell state*

= nilai bias untuk *cell state*

= *cell state*

= *cell state* sebelum orde ke-t

= *output* *gate*

= nilai *weight* untuk *output* *gate*

= nilai bias pada *output* *gate*

= nilai *output* orde ke-t

Sementara itu, nilai *weight* (W) diperoleh melalui persamaan (2.7).

|  |  |  |
| --- | --- | --- |
|  |  | (2.7) |

dengan:

= *weight*

= jumlah data

## *Dataset* Weizmann

*Dataset* Weizmann terdiri atas 10 aksi (kelas), yaitu “*run*,” “*walk*,” “*skip*,” “*jumping-jack*” (disingkat “*jack*”), “*jump-forward-on-two-legs*” (disingkat “*jump*”), “*jump-in-place-on-two-legs*” (disingkat “*pjump*”), “*gallop-sideways*” (disingkat “*side*”), “*wave-two-hands*” (disingkat “*wave2*”), “*wave-one-hand*” (disingkat “*wave1*”), dan “*bend*” yang diperagakan oleh 9 orang berbeda. Data yang disediakan berupa video dengan jumlah 93 video, resolusi 180x144, dan *frame rate* 25 fps [6]. Gambar 2. 6 menujukkan beberapa contoh cuplikan frame video pada setiap aksi.



Gambar 2. 6 *Dataset* Weizmann

## *Data Preprocessing*

*Data preprocessing* adalah proses mengubah data mentah menjadi format atau bentuk yang dapat digunakan secara efektif pada suatu model *machine learning* [34]. Metode-metode dalam *data* *preprocessing* di antaranya adalah *feature* *selection* dan *data* *labeling*.

### *Feature Selection*

*Feature selection* merupakan proses pembersihan data yang tidak relevan dengan tujuan data *mining*. Dengan demikian, *feature selection* bertujuan untuk menghindari *overfitting* dan meningkatkan performa model [35].

### *Data Labeling*

*Data labeling* adalah proses pemberian penanda atau label sebenarnya pada suatu data atau *record*. Proses ini umumnya ditemukan di dalam teknik *supervised* *learning*. Label dapat disematkan pada setiap data oleh peneliti, para ahli, maupun komunitas tertentu [36].

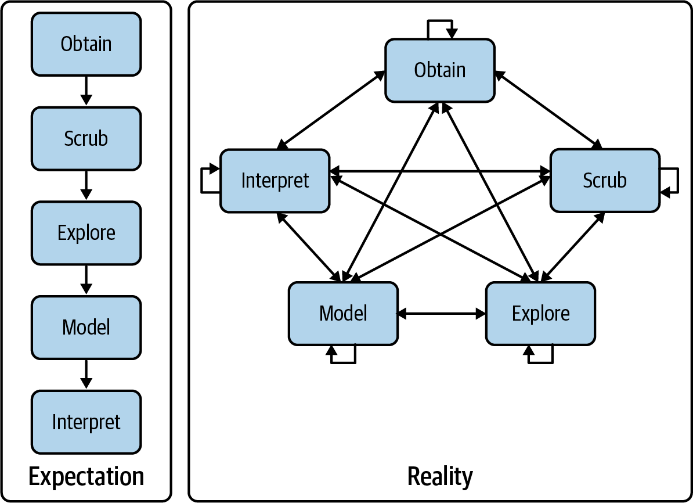
## *Data Augmentation*

*Data augmentation* merupakan suatu metode untuk meningkatkan jumlah data training dengan cara membuat beberapa data buatan dari data asil. Metode ini bertujuan untuk menambah kemampuan *naural network* dalam melakukan generalisasi dan mencegah *overfitting*. Data dalam bentuk citra dapat dikenakan proses *data augmentation* dengan teknik pembesaran, pergeseran, pencerminan, pemutaran, hingga pemotongan gambar. Teknik-teknik tersebut diterapkan sesuai dengan keperluan dan kondisi permasalahan [8].

## Model OSEMN

OSEMN (dibaca *awesome*) merupakan metode atau model dalam *data science* yang dikemukakan oleh Masson dan Wiggins [37]. OSEMN terdiri atas urutan langkah kronologis: ***O****btain*, ***S****crub*, ***E****xplore*, ***M****odel*, dan *i****N****terpret*.

Meskipun kelima langkah tersebut tersusun secara urut, penggunaan model OSEMN sangat umum mengalami perubahan urutan bahkan pelaksanaan beberapa langkah secara bersamaan [38] seperti pada Gambar 2. 7.



Gambar 2. 7 Perbandingan ekspektasi dan realitas penggunaan model OSEMN [38]

1. *Obtain*

Memperoleh data atau *dataset* (*obtain*) merupakan langkah awal yang harus dilakukan dalam proses kerja data *science*. Dataset dapat diperoleh melalui:

1. Mengunduh data dari tempat lain (contoh: *webpage* atau *server*)
2. Data *query* dari *database* atau API (contoh: MySQL atau Twitter)
3. Ekstraksi data dari *file* lain (contoh: *file* HTML atau *spreadsheet*)
4. Membuat data sendiri (contoh: hasil tangkapan sensor atau survei)
5. *Scrub*

Secara umum, data yang berhasil didapatkan masih memiliki kekurangan, seperti hilangnya nilai, inkonsistensi, dan galat (*error*). Oleh karena itu, proses *scrub* atau pembersihan data perlu dilakukan. Contoh proses *scrub* tersebut, meliputi pemilihan data, ekstraksi kolom atau kata, mengubah nilai, mengatasi nilai yang hilang atau terduplikasi, hingga mengonversi format data.

1. *Explore*

Setelah data berhasil dibersihkan, proses eksplorasi data dapat dilakukan. Pada proses ini, peneliti diharapkan mampu mengenali data yang akan ditangani dan siap diolah. Proses *eksplore* di antaranya adalah memahami data, menghitung nilai statistik, dan membuat visualisasi data.

1. *Model*

Tahap ini terdiri atas proses pembuatan model untuk keperluan penjelasan data hingga melakukan suatu prediksi. Berbagai teknik pembuatan model yang dapat dilakukan, antara lain *clustering*, *classification*, *regression*, dan *dimensionality reduction*.

1. *Interpret*

Langkah terakhir dan terpenting dari model OSEMN adalah interpretasi data. Langkah *interpret* ini meliputi penggambaran kesimpulan, evaluasi hasil, dan mengkomunikasikan hasil.

## *Confusion Matrix*

*Confusion matrix* merupakan salah satu metode pengukuran pilihan klasik di dalam *machine learning* terbimbing (*supervised*). Metode *confussion matrix* mampu menvisualisasikan tingkat kebingungan suatu algoritma dalam membedakan kelas dan tidak bergantung pada algoritma klasifikasi yang digunakan [39]. Secara umum, *confusion matrix* disajikan ke dalam bentuk Tabel 2. 3 berikut.

Tabel 2. 3 *Confusion matrix*

|  |  |  |
| --- | --- | --- |
| *Data class* | *Calssified as positive* | *Classified as negative* |
| *Positive* | *True positive ()* | *False negative ()* |
| *Negative* | *False positive ()* | *True negative ()* |

Pada masalah klasifikasi banyak kelas, *confusion matrix* dapat digunakan sebagai dasar dalam pengukuran performa dari klasifikasi tersebut. Untuk setiap kelas , evaluasi didefinisikan berdasarkan nilai , , , , , yang dihitung sebanyak jumlah kelas . Kualitas klasifikasi secara keseluruhan biasanya dievaluasi berdasarkan dua cara, yaitu pengukuran rata-rata dari nilai yang sama untuk (*macro-averaging* dengan *index*) atau penjumlahan dari hitungan , kemudian dilakukan penghitungan performa (*micro-averaging* dengan *index*) [40]. Rumus hitung evaluasi *confusion matrix* pada klasifikasi banyak kelas diuraikan pada Tabel 2. 4 dengan adalah jumlah kelas.

Tabel 2. 4 Rumus-rumus *confusion matrix* pada masalah klasifikasi banyak kelas

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Pengukuran | Rumus | Keterangan |
| 1. | *Average Accuracy* |  | Rata-rata efektivitas tiap kelas. |
| 2. | *Error Rate* |  | Rata-rata *error* tiap kelas. |
| 3. |  |  | Kecocokan label kelas data jika dihitung dari jumlah keputusan per data. |
| 4. |  |  | Efektivitas untuk mengidentifikasi label kelas jika dihitung dari jumlah keputusan per data. |
| 5. |  |  | Relasi antara label positif data dan label berdasarkan jumlah keputusan per data. |
| 6. |  |  | Rata-rata tunggal kecocokan tiap kelas dari label kelas data dengan label. |
| 7. |  |  | Rata-rata tunggal efektivitas tiap untuk mengidentifikasi label kelas. |
| 8. |  |  | Relasi antara label positif data dan label positif berdasarkan rata-rata per kelas. |

Nilai pada adalah nilai relatif yang mengatur seberapa penting nilai terhadap [41]. Rentang nilai adalah . Apabila nilai = 1, yang dihitung adalah yang merupakan nilai *harmonic means* antara dan . Jika , yang dihitung lebih bersifat *recall-oriented* sedangkan nilai digunakan untuk menghitung yang bersifat *precision-oriented* [42].

# BAB III METODE PENELITIAN

## Spesifikasi Perangkat Penelitian

Penelitian tugas akhir “Klasifikasi *Human Action Recognition* (HAR) Berbasis MediaPipe dan *Long Short-Term Memory* (LSTM)” dikerjakan dengan spesifikasi perangkat keras dan perangkat lunak berikut.

### Spesifikasi Perangkat Keras

Penelitian ini dilaksanakan pada perangkat keras dengan spesifikasi sebagai berikut.

SKU Perangkat : Lenovo ideapad 100-14IBD

Prosesor : Processor Intel® Core™ i3-5005U CPU @ 2.00GHz,

2000 Mhz, 2 *Cores*, 4 *Logical Processors*

RAM : 6 GB

VGA : Intel® HD Graphics 5500

Penyimpanan : 500 GB

### Spesifikasi Perangkat Lunak

Perangkat keras, sebagai mana telah disebutkan di atas didukung oleh spesifikasi perangkat lunak berikut.

Sistem operasi : Microsoft Windows 10 Pro 64-bit (10.0 Build 19044)

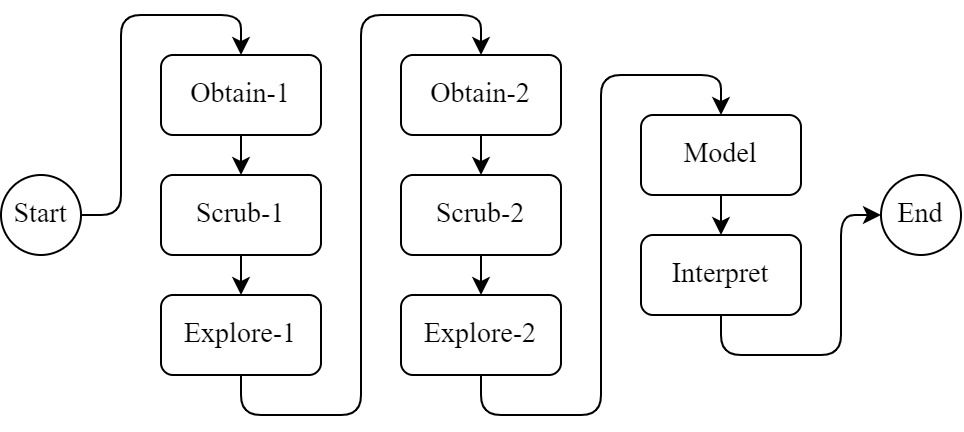
Peramban : Microsoft Edge Version *Official build* (64-bit)

IDE : Jupyter Notebook versi 6.4.5, Anaconda 3, python 3.9

Editor video : Kdenlive 21.12.3

## Model OSEMN

Berdasarkan skema model OSEMN yang umumnya mengalami perubahan urutan [38], peneliti merumuskan langkah kerja penelitian tugas akhir ini seperti Gambar 3. 1 berikut.



Gambar 3. 1 Alur kerja penelitian tugas akhir

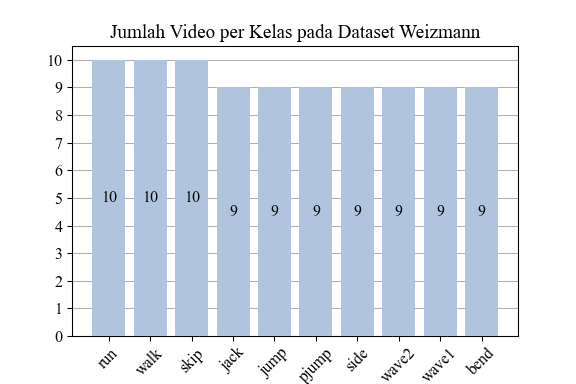
### *Obtain*-1

*Dataset* Weizmann diperoleh langsung melalui *website* resmi Weizmann di *link* <https://www.wisdom.weizmann.ac.il/~vision/SpaceTimeActions.html> pada bagian “*Classification Database*” dengan ukuran *file* terkompresi (ZIP) sebesar 336 MB atau sebesar 454 MB setelah didekompresi. Tabel 3. 1 berikut menunjukkan jumlah video tiap kelas pada *dataset* Weizmann.

Tabel 3. 1 Jumlah video tiap kelas pada *dataset* Weizmann

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Nama Kelas | Jumlah Video | Contoh Cuplikan Video |
| 1. | *run* | 10 |  |
| 2. | *walk* | 10 |  |
| 3. | *skip* | 10 |  |
| 4. | *jumping-jack* (*jack*) | 9 |  |
| 5. | *jump-forward-on-two-legs* (*jump*) | 9 |  |
| 6. | *jump-in-place-on-two-legs* (*pjump*) | 9 |  |
| 7. | *gallop-sideways* (*side*) | 9 |  |
| 8. | *wave-two-hands* (*wave2*) | 9 |  |
| 9. | *wave-one-hand* (*wave1*) | 9 |  |
| 10. | *bend* | 9 |  |

Berdasarkan Tabel 3. 1, *Dataset* Weizmann terdiri atas 10 kelas dengan total video sejumlah 93 video serta jumlah data video pada masing-masing kelasnya sebanyak 9 video (*jack, jump, pjump, side, wave2, wave1*, dan *bend*) atau 10 video (*run, walk*, dan *skip*). Gambar 3. 2 berikut merupakan visualisasi jumlah video pada *dataset* Weizmann.



Gambar 3. 2 Jumlah video per kelas pada *dataset* Weizmann

Detail lain mengenai video *dataset* adalah sebagai berikut:

1. Format video : AVI.
2. Durasi tiap video :1-3 detik.
3. *Framerate* : 25 fps.
4. *Frame* *width* x *height* : 180 x 144 piksel.
5. Jumlah aktor : 9 orang.

### *Scrub*-1

Pada batasan masalah, peneliti menentukan data yang akan digunakan berasal dari 25 *frames* awal pada setiap video. Namun, peneliti menemukan adanya video yang kurang representatif terhadap aksi yang dilakukan pada video “ira\_bend.avi” dalam satu detik awal. Oleh karena itu, peneliti melakukan proses *preprocessing* pada data tersebut berupa pemotongan durasi awal sebanyak 20 *frames* sehingga diperoleh satu detik awal video yang cukup representatif terhadap aksi yang diakukan. Gambar 3. 3 berikut contoh video yang dikenakan pemotongan durasi.



Gambar 3. 3 Satu detik awal (25 *frames*) pada video “ira\_bend.avi” sebelum (kiri) dan sesudah pemotongan durasi (kanan)

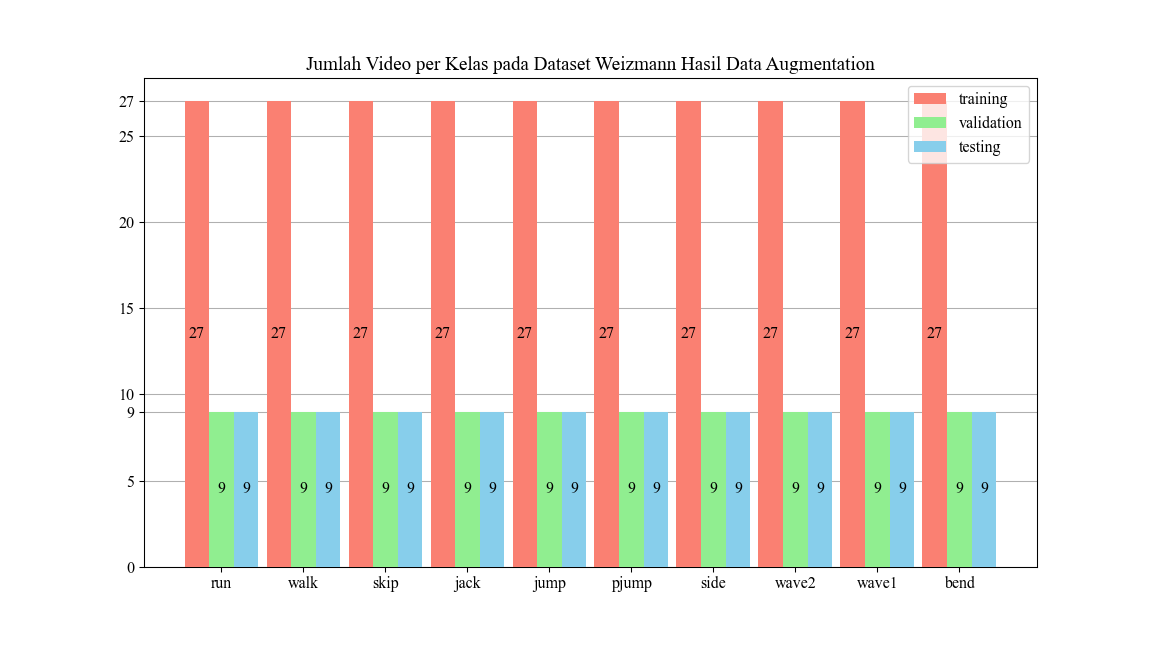
Berdasarkan Gambar 3. 2, *dataset* Weizmann diketahui memiliki 10 kelas dengan jumlah video tiap kelas sebanyak 9-10 video. Dengan jumlah data video yang terlalu sedikit (*small datasets*) [43] untuk rasio 60:20:20 dan distribusi data di setiap kelas yang kurang merata tersebut, peneliti memutuskan untuk melakukan proses *data augmentation* pada *dataset*. Proses *data augmentation* dilakukan melalui aplikasi Kdenlive dengan mengaplikasikan teknik pencerminan, pembesaran, translasi, dan kombinasi di antara ketiganya. Setiap video dikenakan proses *data augmentation* sebanyak 1, 2, atau 4 kali. Tabel 3. 2 berikut adalah beberapa hasil dari proses *data augmentation* yang berhasil diperoleh.

Tabel 3. 2 Contoh hasil proses *data augmentation*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Video Asli | Hasil Proses *Data* *Augmentation* | | | |
| Video #1 | Video #2 | Video #3 | Video #4 |
| 1. |  | Pembesaran: 150%  Translasi: (x,y) = (0,+20) | Pembesaran: 150%  Translasi: (x,y) = (+45,+20) | Pembesaran: 150%  Translasi: (x,y) = (+45,+20)  Pencerminan: horizontal | Pencerminan: horizontal |
| 2. |  | Pembesaran: 150%  Translasi: (x,y) = (-45, 0) | Pembesaran: 150%  Translasi: (x,y) = (-45, +10) |  |  |
| 3. |  | Pembesaran: 150% |  |  |  |

### *Explore*-1

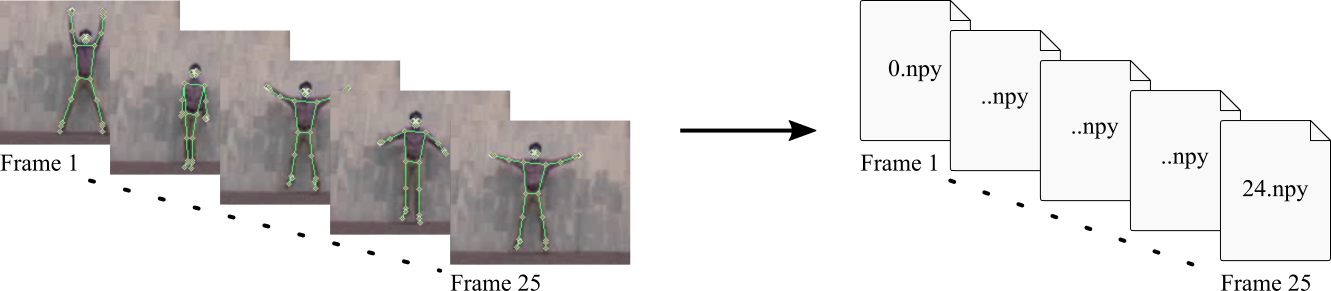
Setelah melakukan tahap *Scrub*-1, *dataset* yang diperoleh memiliki jumlah total 450 video (45 video per kelas) dengan ukuran *file* sebesar 461 MB. Dengan pembagian rasio data *training*:*validation*:*testing* sebesar 60:20:20 maka diperoleh data *training* sebanyak 270 video (27 video per kelas), data *validation* sebanyak 90 video (9 video per kelas), dan data *testing* dengan jumlah yang sama dengan data *validation*, yaitu 90 video (9 video per kelas). *Dataset* Weizmann hasil *data augmentation* dapat divisualisasikan melalui Gambar 3. 4.



Gambar 3. 4 Jumlah video per kelas pada *dataset* Weizmann hasil *data augmentation*

### *Obtain*-2

Tahap *Obtain*-2 terdiri atas proses ekstraksi ciri dari *dataset* video menggunakan pustaka MediaPipe: *Pose*. Data yang diperoleh berupa nilai x, y, z, dan *visibility* dari 33 posisi *landmarks* atau *keypoints* dikalikan dengan jumlah video. Data tersebut disimpan ke dalam format numpy *array* (.npy) sejumlah 25 *files* untuk setiap video. Gambar 3. 5 menunjukkan ilustrasi proses Obtain2.



Gambar 3. 5 Ilustrasi ekstraksi ciri video menggunakan pustaka MediaPipe: *Pose*

### *Scrub*-2

Data numpy yang didapatkan dari proses *Obtain*-2 diberikan label kelas untuk setiap 25 *frames* data. Label kelas yang dimaksud berjumlah 10 kelas dan telah dikodekan ke dalam angka 0-9 seperti pada Tabel 3. 3 berikut.

Tabel 3. 3 Nama kelas dan kode angka

|  |  |  |
| --- | --- | --- |
| No. | Nama Kelas | Kode Angka |
| 1. | *run* | 0 |
| 2. | *walk* | 1 |
| 3. | *skip* | 2 |
| 4. | *jack* | 3 |
| 5. | *jump* | 4 |
| 6. | *pjump* | 5 |
| 7. | *side* | 6 |
| 8. | *wave2* | 7 |
| 9. | *wave1* | 8 |
| 10. | *bend* | 9 |

### *Explore*-2

Data hasil proses *Scrub*-2 adalah *array* dengan lebih dari satu dimensi pada masing-masing data *training*, *validation*, dan *testing*. Data tersebut dibagi ke dalam 2 kelompok, yaitu X dan y. Tabel 3. 4 berikut merupakan nilai dimensi *array* dari data X.

Tabel 3. 4 Dimensi *array* pada data X

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Data | Dimensi | | |
| Video | *Frame* | *Keypoints* |
| 1. | X\_train | 270 | 25 | 132 |
| 2. | X\_valid | 90 | 25 | 132 |
| 3. | X\_test | 90 | 25 | 132 |

Sementara itu, data y yang dihasilkan memiliki dimensi *array* sebagaimana yang ditunjukkan pada Tabel 3. 5 berikut.

Tabel 3. 5 Dimensi *array* pada data y

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Data | Dimensi | |
| Video | Kelas |
| 1. | y\_train | 270 | 10 |
| 2. | y\_valid | 90 | 10 |
| 3. | y\_test | 90 | 10 |

### *Model*

Perancangan model *neural network* dilakukan dengan tiga variasi model. Ketiga model tersebut memiliki pengaturan *hyperparameter* yang sama pada variabel-variabel yang ditunjukkan pada tabel Tabel 3. 6.

Tabel 3. 6 Inisialisasi nilai *hyperparameter* yang sama pada ketiga model

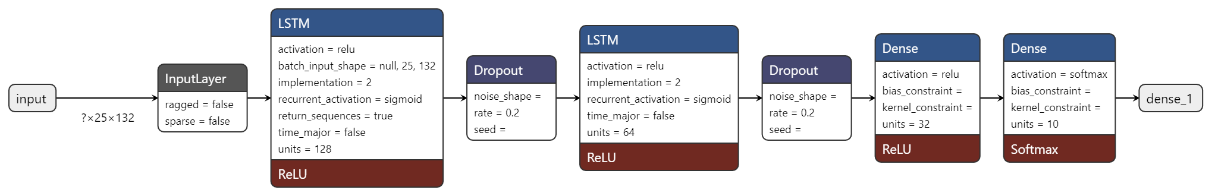
|  |  |  |
| --- | --- | --- |
| No. | *Hyperparameter* | Nilai |
| 1. | input\_shape | (25, 132) |
| 2. | optimizer | Nadam |
| 3. | loss | categorical\_crossentropy |
| 4. | metrics | categorical\_accuracy |
| 5. | epochs | 200 |
| 6. | shuffle | True |

Selain beberapa *hyperparameter* yang ditetapkan sama tersebut, peneliti juga memperhatikan rekomendasi penyusunan model berbasis LSTM, seperti LSTM *layer* yang minimal, penggunaan *dropout* (terutama *variational dropout*), dan ukuran *batch* yang kecil [44]. Agar didapatkan model yang optimal, peneliti menggunakan fungsi ModelCheckpoint [45] pada proses *training* untuk menyimpan model dengan *weight* terbaik dari setiap *epoch* berdasarkan nilai *validation* *categorical accuracy* (val\_categorical\_accuracy).

Dengan mengacu pada ketentuan-ketentuan di atas, peneliti merancang tiga buah model *neural network* sebagai berikut.

1. Model 1

Model ini terdiri atas 2 LSTM *layers*, 2 *Dropout layers*, dan 2 *Dense layers*. *Dropout layer* ditempatkan setelah masing-masing LSTM *layer* dengan tujuan untuk mencegah kemungkinan terjadinya *overfitting* [46]. Gambar 3. 6 menunjukkan pengaturan arsitektur dari Model 1 beserta *hyperparameter* yang diterapkannya.



Gambar 3. 6 Arsitektur Model 1

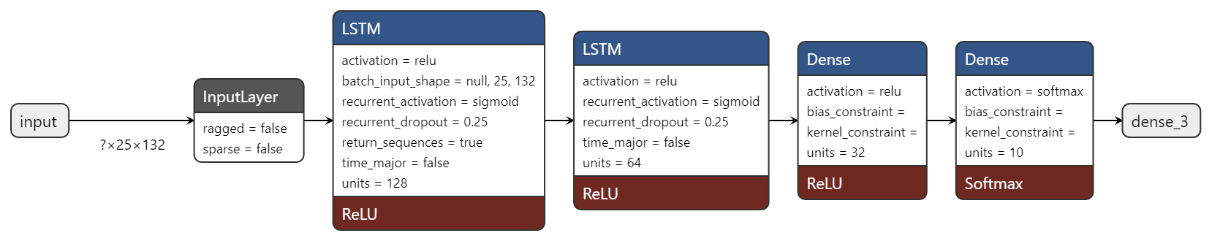
Pengaturan nilai *hyperparameter* lain yang diterapkan pada Model 1 terlihat pada Tabel 3.7.

Tabel 3. 7 Perubahan *hyperparameter* pada Model 1

|  |  |  |
| --- | --- | --- |
| No. | *Hyperparameter* | Nilai |
| 1. | learning\_rate | 0,0001 |
| 2. | batch\_size | 4 |

1. Model 2

Gambar 3. 7 menunjukkan Model 2 yang tersusun atas 2 LSTM *layers* dan 2 *Dense layers*. Nilai *dropout* yang digunakan berupa *variational dropout* melalui variabel recurrent\_dropout.



Gambar 3. 7 Arsitektur Model 2

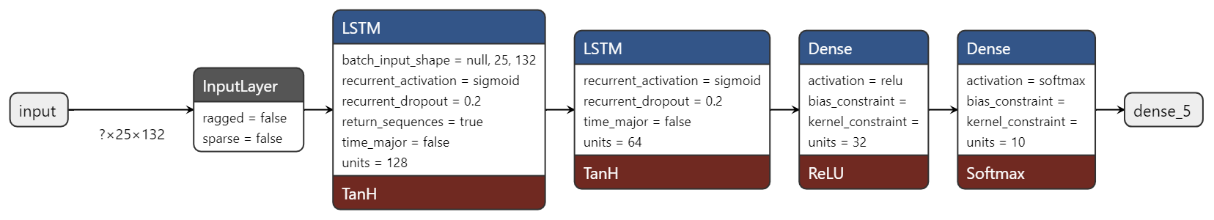
Pengaturan nilai *hyperparameter* lain yang diterapkan pada Model 2 selain terhadap arsitekturnya terlihat pada Tabel 3. 8 berikut.

Tabel 3. 8 Perubahan *hyperparameter* pada Model 2

|  |  |  |
| --- | --- | --- |
| No. | *Hyperparameter* | Nilai |
| 1. | learning\_rate | 0,000075 |
| 2. | batch\_size | 2 |

1. Model 3

Pada Model 3 ini, peneliti merancang arsitektur yang mirip dengan Model 2 dengan perbedaan pada *hyperparameter activation* di LSTM *layer* yang menggunakan nilai *default* (TanH) serta nilai recurrent\_dropout yang diperkecil menjadi 0,2. Gambar 3. 8 menunjukkan pengaturan arsitektur dari Model 3 beserta *hyperparameter* yang diterapkannya.



Gambar 3. 8 Arsitektur Model 3

*Hyperparameter* lainnya tidak mengalami perubahan sehingga masih sama dengan nilai yang tertera pada Tabel 3. 8.

### *Interpret*

Interpretasi dilakukan dengan membandingkan performa arsitektur model *neural network* berdasarkan *accuracy* dan *loss* pada proses *training* dan *validation* serta pada proses *testing*. Hasil proses *testing* dievaluasi menggunakan metode *confusion matrix* dan perhitungan *accuracy*, *error rate, precision,* *recall*, hingga .

Setelah itu, dilakukan pula implementasi prediksi sederhana pada video demo. Video demo memiliki resolusi sebesar 640x360 dengan *framerate* 25 fps. Aksi-aksi di dalam video diperagakan oleh seorang aktor dengan dua variasi gerakan untuk setiap aksinya. Setiap variasi gerakan diperagakan dalam durasi dua detik. Gambar 3. 9 berikut merupakan beberapa contoh cuplikan video demo.



Gambar 3. 9 Beberapa peragaan aksi pada video demo

Masing-masing model terlatih akan diimplementasikan secara bergantian. Kemudian, hasil prediksi atau deteksi aksi akan diperoleh berdasarkan nilai input *keypoints* MediaPipe sebanyak 25 *frames* (terbaru). Perubahan hasil deteksi yang diperoleh akan dicatat dan diinterpretasikan dalam bentuk narasi.

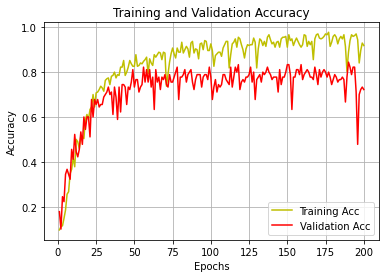
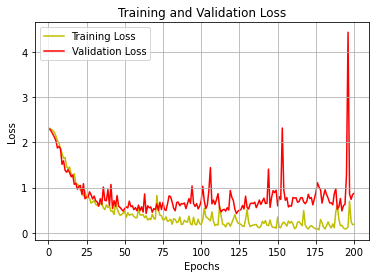
# BAB IV HASIL DAN PEMBAHASAN

## Analisis dan Pembahasan

Berdasarkan penelitian klasifikasi *Human Action Recognition* yang telah dilakukan terhadap *dataset* Weizmann menggunakan tiga macam model, peneliti memperoleh hasil berupa nilai akurasi dan *loss* pada masing-masing proses *training* dan *validation*. Model beserta *weight* (bobot) dari hasil proses *training* dan *validation* yang diperoleh telah disimpan secara lokal dengan ekstensi *file* .h5 dengan memperhatikan nilai *validation categorical accuracy* tertinggi di dalam 200 *epochs*.

### Analisis *Training* dan *Validation*

Proses *training* dan *validation* dataset pada arsitektur Model 1 ditunjukkan pada Gambar 4. 1. Selama 200 *epochs*, terlihat perkembangan nilai *loss* dan *accuracy* dari proses *training* dan *validation* pada arsitektur *neural network* yang terdiri atas 2 LSTM *layers*, 2 *Dropout layers*, dan 2 *Dense layers*.



(a) (b)

Gambar 4. 1 Grafik *loss* (a) dan *accuracy* (b) pada proses *training* dan *validation* terhadap Model 1

Berdasarkan Gambar 4. 1(a), nilai *training loss* menurun secara drastis dari *epoch* ke-1 sampai *epoch* ke-25. Setelah proses *training* melampaui *epoch* tersebut, penurunan *loss* terjadi lebih pelan secara bertahap hingga *epoch* ke-125. Mulai dari *epoch* ke-125, nilai *training* *loss* sudah mulai menunjukkan konvergensi pada titik optimumnya. Secara keseluruhan, perubahan nilai *training* *loss* cenderung stabil.

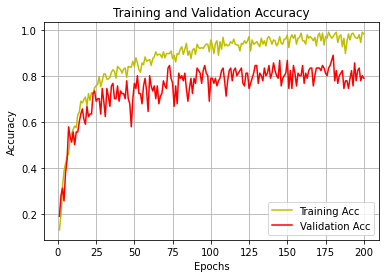
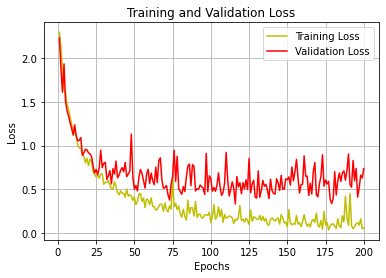
Perkembangan nilai *loss* pada *validation* *loss* menunjukkan kondisi yang berbeda jika dibandingkan dengan nilai *training loss*. Nilai *validation loss* terlihat mulai konvergen pada *epoch* ke-50. Setelah melampaui *epoch* tersebut, nilai *validation loss* mengalami beberapa kali divergensi mulai dari *epoch* ke-75 dan 125. Nilai *validation* *loss* ini juga diketahui mengalami *overshoot* beberapa kali pada *epoch* ke-106 (1,4414), *epoch* ke-144 (1,4076), *epoch* ke-153 (2,3185), dan *epoch* ke-196 (4,4361).

Apabila ditinjau dari sisi akurasi, perkembangan proses *training* dan *validation* model menunjukkan proses yang cukup cepat. Pada perkembangan nilai *traning accuracy*, peningkatan nilai terjadi secara signifikan dari awal *training* hingga *epoch* ke-50. Setelah itu, nilai *traning accuracy* meningkat secara bertahap hingga mencapai nilai konvergensi setelah *epoch* ke-125.

Di sisi lain, proses *validation* menunjukkan peningkatan signifikan nilai *accuracy* dari *epoch* ke-1 hingga *epoch* ke-50 kemudian nilai tersebut langsung berada pada kondisi yang konvergen setelahnya. Namun, kondisi *overshooting* nilai *validation* juga terlihat dari Gambar 4. 1(b), yaitu pada *epoch* ke-63 (0,6333), *epoch* ke-153 (0,6333), dan *epoch* ke-196 (0,4778).

Dengan bantuan fungsi ModelCheckpoint, proses *traning* dan *validation* pada model 1 menghasilkan *weight* terbaik pada *epoch* ke-190. Pada epoch tersebut, diketahui nilai *training* *loss* = 0,3005, *training categorical accuracy* = 0,8889, *validation loss* = 0,5747, dan *validation categorical accuracy* = 0,8444.

Percobaan selanjutnya adalah proses *training* dan *validation* pada Model 2 dengan arsitektur yang menerapkan nilai *variational dropout*. Dengan pengaturan *learning rate* sebesar 0,000075 dan *batch size* sebesar 2, perkembangan nilai *loss* dan *accuracy* ditunjukkan pada Gambar 4. 2.



(a) (b)

Gambar 4. 2 Grafik *loss* (a) dan *accuracy* (b) pada proses *training* dan *validation* terhadap Model 2

Perkembangan nilai *training loss* yang disajikan pada Gambar 4. 2(a) menunjukkan penurunan signifikan dari *epoch* ke-1 hingga *epoch* ke-25. Proses penurunan nilai *loss* tersebut kemudian melambat hingga kondisi konvergensi yang dimulai pada *epoch* ke-175. Namun hingga *epoch* ke-200, nilai *training loss* sempat mengalami sedikit *overshooting* pada *epoch* ke-188 (0,4235) dan *epoch* ke-191 (0,4561).

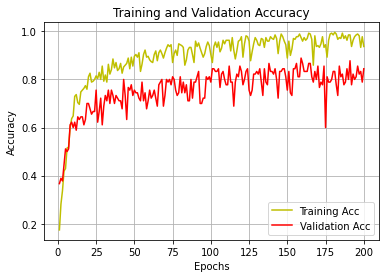
Kondisi nilai *validation loss* menunjukkan tren yang berbeda dari *training* *loss*. Penurunan nilai *validation* *loss* terjadi sangat cepat dari awal *training* hingga *epoch* ke-25. Setelah itu, kecepatan penurunan nilai *validation* *loss* melambat hingga titik mencapai titik konvergensi setelah *epoch* ke-75. Secara keseluruhan, grafik nilai *validation* *loss* tidak lebih stabil dari pada grafik nilai *training* *loss*. Hal tersebut dapat dilihat dari banyaknya *overshooting* nilai seperti yang terjadi pada *epoch* ke-48 (1,1309). Selain itu, grafik juga menunjukkan adanya divergensi mulai *epoch* ke-150 dan *epoch* ke-175.

Berdasarkan grafik accuracy pada Gambar 4. 2(b), perkembangan nilai *training accuracy* dan *validation accuracy* dapat diamati. Nilai *training accuracy* cenderung menunjukkan perubahan nilai yang stabil pada sepanjang proses *training*. Peningkatan nilai *training* *accuracy* terjadi secara signifikan dari awal *training* hingga *epoch* ke-25. Kenaikan nilai *accuracy* semakin melambat setelahnya hingga mencapai konvergensi pada titik optimal mulai *epoch* ke-175.

Pada sisi *validation accuracy*, peningkatan nilai terjadi sangat cepat pada *epoch* ke-1 hingga *epoch* ke-25. Kecepatan peningkatan nilai *validation* *accuracy* berangsur menurun hingga mencapai kondisi konvergen setelah *epoch* ke-125. Kestabilan perkembangan nilai *validation* *accuracy* tersebut juga kurang baik jika dibandingkan dengan nilai *training* *accuracy*. Gambar 4. 2(b) menunjukkan kondisi *overshooting* yang jelas pada *epoch* ke-48 (0,5778).

Nilai *weight* terbaik yang berhasil dicapai Model 2 terjadi pada *epoch* ke-180. Pada epoch tersebut, diketahui nilai *training* *loss* = 0,1074, *training categorical accuracy* = 0,9704, *validation loss* = 0,3952, dan *validation categorical accuracy* = 0,8889.

Pada percobaan terakhir, pelatihan Model 3 dilakukan dengan arsitektur yang identik dengan Model 2. Perubahan fungsi aktivasi dari ReLU ke TanH menyebabkan perubahan perkembangan nilai *loss* dan *accuracy* pada proses *training* dan *validation*. Gambar 4. 3 merupakan visualisasi dari kedua proses tersebut.



(a) (b)

Gambar 4. 3 Grafik *loss* (a) dan *accuracy* (b) pada proses *training* dan *validation* terhadap Model 3

Berdasarkan nilai *training* *loss* pada Gambar 4. 3(a), proses *training* menunjukkan tren perubahan nilai yang stabil. Nilai *training* *loss* menurun secara drastis dari awal *training* hingga *epoch* ke-25. Setelah itu, proses *training* dilanjutkan dengan menghasilkan penurunan nilai *loss* yang melambat hingga *epoch* ke-50. Nilai *training* *loss* yang dihasilkan setelah *epoch* ke-50 mengalami konvergensi hingga akhir *training*.

Penurunan nilai secara drastis juga terjadi pada nilai *validation loss* di awal proses *validation* hingga *epoch* ke-25. Penurunan nilai melambat secara perlahan hingga mencapai titik konvergen setelah *epoch* ke-125. Pada grafik nilai *validation* *loss* ini, kondisi *overshooting* terjadi beberapa kali. Kondisi tersebut terlihat jelas terutama pada *epoch* ke-175 (1,4094). Selain *overshooting*, kondisi divergensi nilai *validation loss* juga dapat diamati pada grafik setelah *epoch* ke-150. Apabila diamati secara keseluruhan, perubahan nilai *training* *loss* cenderung lebih stabil dari perubahan nilai *validation* *loss.*

Jika ditinjau dari nilai *accuracy* pada Gambar 4. 3(b), proses *training* dan *validation* terhadap Model 3 berhasil menghasilkan tren peningkatan nilai *accuracy*. Pada *training* *accuracy*, peningkatan nilai yang cukup signifikan terjadi pada awal *training* hingga *epoch* ke-25. Peningkatan ini berlanjut dengan lambat hingga *epoch* ke-150 yang merupakan posisi awal grafik mencapai kondisi konvergen.

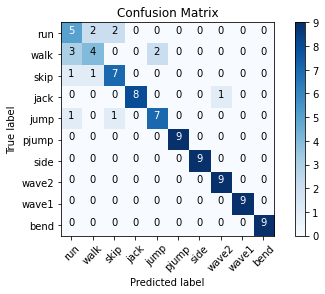
Peningkatan drastis dari nilai *accuracy* juga terlihat pada perkembangan nilai *validation* *accuracy* mulai awal proses *validation* hingga *epoch* ke-25. Peningkatan yang lambat ditunjukkan setelahnya hingga terjadi kondisi konvergen setelah *epoch* ke-125. Setelah kondisi konvergen tercapai, proses *validation* mengalami kondisi *overshooting* pada *epoch* ke-175 yang menyebabkan nilai *validation* *accuracy* turun drastis hingga mencapai angka 0,6.

Setelah proses *training* dan *validation* selesai pada *epoch* ke-200, Model 3 memperoleh nilai *weight* terbaik pada *epoch* ke-159 dengan nilai *training* *loss* = 0,0745, *training categorical accuracy* = 0,9741, *validation loss* = 0,3467, dan *validation categorical accuracy* = 0,8889.

### Evaluasi *Testing*

Model 1, Model 2, dan Model 3 dengan *weight* yang berhasil didapatkan dari proses *training* dan *validation* selanjutnya diuji berdasarkan data *testing* (proses *testing*). Hasil dari proses *testing* ini kemudian dievaluasi dengan *confusion matrix* beserta perhitungan nilai *average* *accuracy*, *error* *rate*, *precision, recall*, dan *fscore* baik secara *micro* maupun *macro*.

Jika ditinjau dari perolehan nilai TP tiap kelas pada Gambar 4. 4, proses *testing* terhadap Model 1 menghasilkan klasifikasi yang cukup baik pada kelas *skip, jack, jump, pjump, side, wave2, wave1*, dan *bend*. Sementara itu, dua kelas lainnya, yaitu *run* dan *walk* memberikan hasil klasifikasi yang kurang baik.



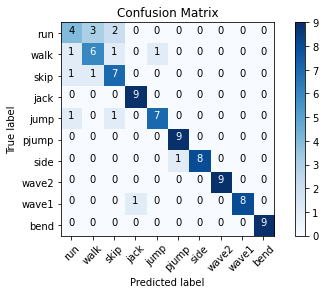
Gambar 4. 4 *Confusion matrix* hasil proses *testing* terhadap Model 1

Selain nilai TP, *confusion matrix* pada Gambar 4. 4 juga menunjukkan nilai FP, TN, dan FN. Daftar nilai tersebut disajikan secara lengkap melalui Tabel 4. 1 berikut.

Tabel 4. 1 Hasil nilai TP, FP, TN, dan FN dari proses *testing* Model 1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No. | Nama Kelas | Evaluasi Klasifikasi | | | |
| TP | FP | TN | FN |
| 1. | *run* | 5 | 5 | 76 | 4 |
| 2. | *walk* | 4 | 3 | 78 | 5 |
| 3. | *skip* | 7 | 3 | 78 | 2 |
| 4. | *jack* | 8 | 0 | 81 | 1 |
| 5. | *jump* | 7 | 2 | 79 | 2 |
| 6. | *pjump* | 9 | 0 | 81 | 0 |
| 7. | *side* | 9 | 0 | 81 | 0 |
| 8. | *wave2* | 9 | 1 | 80 | 0 |
| 9. | *wave1* | 9 | 0 | 81 | 0 |
| 10. | *bend* | 9 | 0 | 81 | 0 |

Model 2 menghasilkan kondisi *confusion matrix* dengan nilai TP per kelas yang sedikit lebih merata jika dibandingkan dengan *confusion matrix* pada Model 1. Gambar 4. 5 menunjukkan penurunan nilai TP pada kelas *run*, *side*, dan *wave1* sedangkan kelas *walk* dan *jack* mengalami kenaikan nilai TP.



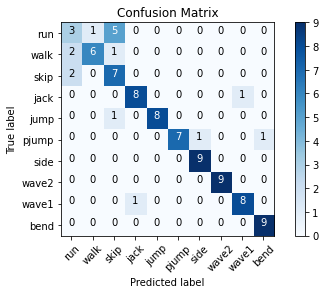
Gambar 4. 5 *Confusion matrix* hasil proses *testing* terhadap Model 2

Nilai TP dari proses *testing* Model 2 yang bervariasi juga diikuti dengan nilai FP, TN, dan FN yang bervariasi pula. Perincian nilai TP, FP, TN, dan FN tersebut dapat dilihat pada Tabel 4. 2 di bawah ini.

Tabel 4. 2 Hasil nilai TP, FP, TN, dan FN dari proses *testing* Model 2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No. | Nama Kelas | Evaluasi Klasifikasi | | | |
| TP | FP | TN | FN |
| 1. | *run* | 4 | 3 | 78 | 5 |
| 2. | *walk* | 6 | 4 | 77 | 3 |
| 3. | *skip* | 7 | 4 | 77 | 2 |
| 4. | *jack* | 9 | 1 | 80 | 0 |
| 5. | *jump* | 7 | 1 | 80 | 2 |
| 6. | *pjump* | 9 | 1 | 80 | 0 |
| 7. | *side* | 8 | 0 | 81 | 1 |
| 8. | *wave2* | 9 | 0 | 81 | 0 |
| 9. | *wave1* | 8 | 0 | 81 | 1 |
| 10. | *bend* | 9 | 0 | 81 | 0 |

Hasil *confusion matrix* Model 3 pada Gambar 4. 6 menunjukkan hasil proses *testing* yang cukup bervariasi jika dilihat dari perolehan nilai TP dari tiap kelas aksi. Klasifikasi sangat baik pada aksi *side*, *wave2*, dan *bend* dengan perolehan TP = 9. Klasifikasi kelas aksi *jack*, *jump*, dan *wave* juga cukup baik dengan nilai TP sebayak 8. Nilai TP = 7 terlihat diperoleh pada kelas *skip* dan *pjump*. Sementara itu, aksi *walk* mendapat nilai TP = 6 dan aksi *run* mendapat nilai TP = 3.



Gambar 4. 6 *Confusion matrix* hasil proses *testing* terhadap Model 3

Selain nilai TP, terdapat juga nilai FP, TN, dan FN dari tiap kelas yang dihasilkan dari proses *testing* Model 3. Nilai-nilai tersebut diuraikan dalam Tabel 4. 3 berikut.

Tabel 4. 3 Hasil nilai TP, FP, TN, dan FN dari proses *testing* Model 3

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No. | Nama Kelas | Evaluasi Klasifikasi | | | |
| TP | FP | TN | FN |
| 1. | *run* | 3 | 4 | 77 | 6 |
| 2. | *walk* | 6 | 1 | 80 | 3 |
| 3. | *skip* | 7 | 7 | 74 | 2 |
| 4. | *jack* | 8 | 1 | 80 | 1 |
| 5. | *jump* | 8 | 0 | 81 | 1 |
| 6. | *pjump* | 7 | 0 | 81 | 2 |
| 7. | *side* | 9 | 1 | 80 | 0 |
| 8. | *wave2* | 9 | 0 | 81 | 0 |
| 9. | *wave1* | 8 | 1 | 80 | 1 |
| 10. | *bend* | 9 | 1 | 80 | 0 |

Semua nilai evaluasi klasifikasi berupa nilai TP, FN, TN, dan FN diolah hingga menghasilkan nilai *average accuracy*, *error rate*, *precision*, *recall*, dan *fscore*. Tabel 4. 4 berikut merupakan perbandingan hasil perhitung evaluasi terhadap Model 1, Model 2, dan Model 3.

Tabel 4. 4 Hasil pengukuran kualitas klasifikasi tiap model

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Pengukuran | Nilai | | |
| Model 1 | Model 2 | Model 3 |
| 1. | *Average Accuracy* | 0,84 | 0,84 | 0,82 |
| 2. | *Error Rate* | 0,16 | 0,16 | 0,18 |
| 3. |  | 0,84 | 0,84 | 0,82 |
| 4. |  | 0,84 | 0,84 | 0,82 |
| 5. |  | 0,84 | 0,84 | 0,82 |
| 6. |  | 0,84 | 0,85 | 0,84 |
| 7. |  | 0,84 | 0,84 | 0,82 |
| 8. |  | 0,84 | 0,84 | 0,82 |

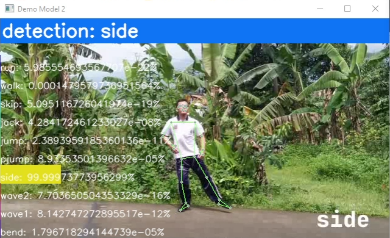
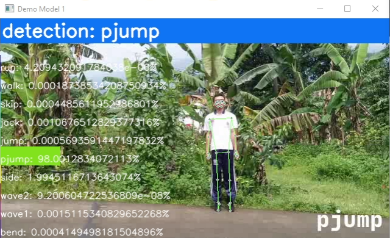
## Implementasi Model pada Video Demo

Model yang telah melalui proses *training* dan *validation* menghasilkan *weight* yang bisa digunakan sebagai acuan dalam mengklasifikasikan aksi. Dengan menggunakan video yang direkam secara mandiri, peneliti menggunakan program *python* sederhana di dalam Jupyter Notebook untuk menerapkan uji klasifikasi terhadap Model 1, Model 2, dan Model 3. Pengujian tersebut memberikan hasil sesuai dengan Tabel 4. 5 berikut.

Tabel 4. 5 Perubahan deteksi aksi oleh tiap model

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Aksi | Urutan Perubahan Deteksi | | |
| Model 1 | Model 2 | Model 3 |
| 1. | *run* | *walk,* ***run****, wave1,* ***run*** | *walk,* ***run****, bend* | *walk,* ***run****, side* |
| 2. | *walk* | *bend,* ***walk****, bend* | ***walk****, bend* | ***walk****, bend,* ***walk****, bend,* ***walk****, side* |
| 3. | *skip* | *run* | *walk, run, bend* | *walk, run, pjump* |
| 4. | *jack* | *skip,* ***jack*** | *wave1,* ***jack*** | *side,* ***jack*** |
| 5. | *jump* | *run, walk, run, skip,* ***jump*** | *walk, run, skip, bend* | *side, walk, run, skip, bend* |
| 6. | *pjump* | ***pjump*** | ***pjump****, side* | ***pjump*** |
| 7. | *side* | ***side****, pjump,* ***side****, pjump* | ***side*** | ***side****, pjump,* ***side****, pjump* |
| 8. | *wave2* | *jack, wave1, jack,* ***wave2****, jack* | *jack, side,* ***wave2****, jack* | *jack,* ***wave2****, jack* |
| 9. | *wave1* | ***wave1****, pjump,* ***wave1****, jack, pjump* | *jack,* ***wave1****, jack, side, jack, side* | *jack,* ***wave1****, jack,* ***wave1****, jack* |
| 10. | *bend* | *wave1,* ***bend*** | *walk,* ***bend*** | *wave1, side, pjump,* ***bend*** |

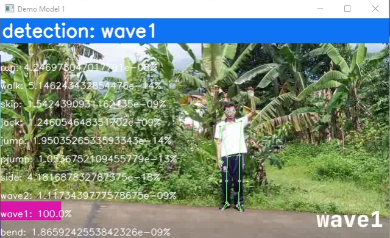
Implementasi 3 model yang berbeda menghasilkan perbedaan deteksi pula terhadap aksi di dalam video demo. Berdasarkan hasil implementasi, Model 1, Model 2, dan Model 3 masing-masing paling konsisten dalam mendeteksi aksi *pjump, side*, dan *pjump* yang cuplikannya dapat dilihat pada Gambar 4. 7.



(a) (b) (c)

Gambar 4. 7 Deteksi *pjump* oleh Model 1 (a), deteksi *side* oleh Model 2 (b), dan deteksi *pjump* oleh Model 3 (c)

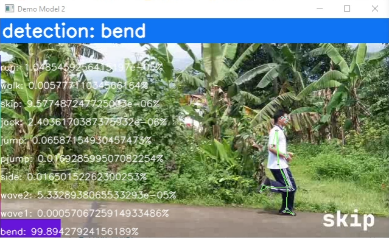
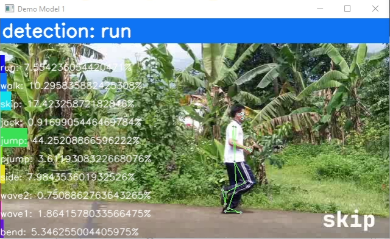
Konsistensi yang kurang baik (terjadi banyak perubahan deteksi) diketahui terjadi pada deteksi aksi *wave1* untuk Model 1 dan Model 2. Pada Model 3, kondisi konsistensi yang kurang baik terjadi pada aksi *walk*. Gambar 4. 8 berikut adalah contoh hasil deteksi terhadap aksi-aksi tersebut.



(a) (b) (c)

Gambar 4. 8 Deteksi *wave1* oleh Model 1 (a) dan Model 2 (b) serta deteksi *walk* oleh Model 3 (c)

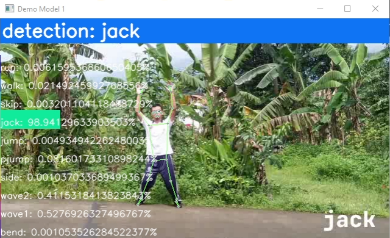
Sementara itu, Model 1 diketahui paling kesulitan dalam mendeteksi aksi *skip*. Pada Model 2, kesulitan deteksi terjadi pada aksi *skip* dan *jump*. Kondisi serupa terjadi pula pada Model 3. Dengan kata lain, skip merupakan aksi yang paling sulit dideteksi. Kesalahan deteksi aksi *skip* dapat dilihat pada Gambar 4. 9.



(a) (b) (c)

Gambar 4. 9 Hasil deteksi aksi *skip* oleh Model 1 (a), Model 2 (b), dan Model 3 (c)

Pada ketiga model, aksi *wave2* diketahui sering terdeteksi sebagai aksi *jack*. Di sisi lain, aksi *jack* merupakan kelas aksi yang paling konsisten terdeteksi karena hanya mengalami sedikit perubahan. Contoh hasil deteksi aksi *wave2* dan *jack* oleh Model 1 ditunjukkan pada Gambar 4. 10.



(a) (b)

Gambar 4. 10 Deteksi aksi *wave2* (a) dan *jack* (b) oleh Model 1

# BAB V PENUTUP

## Kesimpulan

Berdasarkan tujuan dan hasil penelitian tugas akhir yang berjudul “Klasifikasi *Human Action Recognition* (HAR) Berbasis MediaPipe dan *Long Short-Term Memory* (LSTM)” ini, peneliti memperoleh beberapa kesimpulan yaitu:

1. Diperoleh tiga buah model *Artificial Neural Network* pada masalah klasifikasi *Human Action Recognition* berdasarkan hasil ekstraksi ciri *landmarks* citra oleh pustaka MediaPipe dan jaringan *Long Short-Term Memory.*
2. Nilai *accuracy* proses *testing* dari Model 1, Model 2, dan Model 3 masing-masing secara berurutan adalah 84%, 84%, dan 82%.
3. Evaluasi *confusion matrix* menunjukkan bahwa klasifikasi terhadap aksi *run* dan *walk* cukup sulit dilakukan oleh ketiga model yang dibuktikan dengan angka TP yang rendah (3 hingga 6). Di sisi lain, klasifikasi terhadap aksi *skip*, *jack*, *jump*, *pjump*, *side*, *wave2*, *wave1*, dan *bend* menunjukkan hasil yang cukup baik dengan angka TP yang tinggi (7 hingga 9).
4. Hasil deteksi aksi pada tahap implementasi video demo menghasilkan klasifikasi yang berbeda dan cenderung berubah-ubah jika dibandingkan dengan hasil *testing*. Kondisi ini menunjukkan adanya perbedaan perilaku model ketika melakukan klasifikasi di antara data statis pada data *training, validation,* dan *testing* dengan data dinamis yang diperoleh melalui video. Dengan demikian, generalisasi serta konsistensi deteksi yang dilakukan oleh ketiga model belum cukup baik.

## Saran

Saran-saran yang dapat penulis berikan berdasarkan penelitian tugas akhir ini adalah sebagai berikut.

1. Hendaknya dilakukan perbandingan di antara beberapa *dataset* sekaligus agar pengaruh penggunaan jenis *dataset* terhadap hasil *training* model. Salah satu *dataset* yang berpotensi sebagai pembanding yaitu *dataset* KTH [47], [48].
2. Sebaiknya jumlah data per kelas di dalam *dataset* dapat ditambah sehingga diharapkan generalisasi model menjadi lebih baik. Penambahan data tersebut juga termasuk pada variasi posisi aktor untuk setiap aksi.
3. Penggunaan parameter pustaka MediaPipe hendaknya lebih diperhatikan sehingga *keypoints* yang diperoleh bisa lebih optimal dan akurat terutama pada aksi yang cenderung mirip. Contoh parameter yang dapat disesuaikan antara lain pemilihan 4 *keypoints values* (x, y, z, dan *visibility*) dan nilai min\_detection\_confidence [5].
4. Proses *data augmentation* terhadap data video membutuhkan waktu yang cukup lama. Oleh karena itu, sebaiknya teknik *data augementation* tersebut dapat dilakukan secara otomatis.
5. Latar belakang pada video demo sebaiknya berupa tanah lapang atau dinding yang polos sehingga tidak mengganggu deteksi oleh pustaka MediaPipe.
6. Penelitian ini menentukan model hasil *training* terbaik dengan pendekatan *maximum accuracy* yang menyebabakan nilai *loss* hanya dapat dikendalikan dengan pembatasan jumlah *epoch*. Oleh karena itu, sebaiknya perlu dilakukan pula proses *training* dengan pendekatan *minimum loss* terhadap model [26].

# DAFTAR PUSTAKA

[1] G. Cheng, Y. Wan, A. N. Saudagar, K. Namuduri, and B. P. Buckles, “Advances in Human Action Recognition: A Survey,” *Computing Research Repository (CoRR)*, Jan. 2015, [Online]. Available: http://arxiv.org/abs/1501.05964

[2] S. Zhang, Z. Wei, J. Nie, L. Huang, S. Wang, and Z. Li, “A Review on Human Activity Recognition Using Vision-Based Method,” *Journal of Healthcare Engineering*, vol. 2017, 2017, doi: 10.1155/2017/3090343.

[3] Y. Lu, C. Lu, and C.-K. Tang, “Online Video Object Detection Using Association LSTM,” in *2017 IEEE International Conference on Computer Vision (ICCV)*, Oct. 2017, pp. 2363–2371. doi: 10.1109/ICCV.2017.257.

[4] I. Jegham, A. ben Khalifa, I. Alouani, and M. A. Mahjoub, “Vision-based human action recognition: An overview and real world challenges,” *Forensic Science International: Digital Investigation*, vol. 32. Elsevier Ltd, Mar. 01, 2020. doi: 10.1016/j.fsidi.2019.200901.

[5] Google LLC, “MediaPipe Pose,” 2020. https://google.github.io/mediapipe/solutions/pose.html (accessed Jan. 19, 2022).

[6] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri, “Actions as Space-Time Shapes,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 12, pp. 2247–2253, Dec. 2007, doi: 10.1109/TPAMI.2007.70711.

[7] Keras, “LSTM layer.” https://keras.io/api/layers/recurrent\_layers/lstm/ (accessed Jan. 20, 2022).

[8] V. Verdhan, *Computer Vision Using Deep Learning*, 1st ed. Berkeley, CA: Apress, 2021. doi: 10.1007/978-1-4842-6616-8.

[9] Y. Tang, Y. Tian, J. Lu, P. Li, and J. Zhou, “Deep Progressive Reinforcement Learning for Skeleton-Based Action Recognition,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun. 2018, pp. 5323–5332. doi: 10.1109/CVPR.2018.00558.

[10] A. Grushin, D. D. Monner, J. A. Reggia, and A. Mishra, “Robust Human Action Recognition via Long Short-Term Memory,” in *The 2013 International Joint Conference on Neural Networks (IJCNN)*, Aug. 2013, pp. 1–8. doi: 10.1109/IJCNN.2013.6706797.

[11] R. Gunawan and Y. Kristian, “Automatic Parental Guide Scene  Classification Menggunakan Metode Deep Convolutional Neural Network dan LSTM,” *Journal of Intelligent System and Computation*, vol. 2, no. 2, pp. 86–90, Oct. 2020, doi: 10.52985/insyst.v2i2.124.

[12] D. Subroto, “Deteksi Aktivitas Manusia Berdasarkan Data Skeleton dengan Menggunakan Modifikasi VGG16,” *Jurnal Infra*, vol. 9, no. 1, 2021, Accessed: Jan. 20, 2022. [Online]. Available: https://publication.petra.ac.id/index.php/teknik-informatika/article/view/10933

[13] Yong Du, W. Wang, and L. Wang, “Hierarchical Recurrent Neural Network for Skeleton Based Action Recognition,” in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2015, pp. 1110–1118. doi: 10.1109/CVPR.2015.7298714.

[14] M. Latah, “Human Action Recognition Using Support Sector Machines and 3D Convolutional Neural Networks,” *International Journal of Advances in Intelligent Informatics*, vol. 3, no. 1, pp. 47–55, Mar. 2017, doi: 10.26555/ijain.v3i1.89.

[15] J. Song, Z. Guo, L. Gao, W. Liu, D. Zhang, and H. T. Shen, “Hierarchical LSTM with Adjusted Temporal Attention for Video Captioning,” *Computing Research Repository (CoRR)*, Jun. 2017, [Online]. Available: http://arxiv.org/abs/1706.01231

[16] T. B. Supriyatna, S. M. Nasution, and R. A. Nugraheni, “Human Activity Recognition Using Support Vector Machine for Automatic Security System,” in *Journal of Physics: Conference Series*, May 2019, vol. 1192, no. 1. doi: 10.1088/1742-6596/1192/1/012017.

[17] A. Halder and A. Tayade, “Real-time Vernacular Sign Language Recognition using MediaPipe and Machine Learning,” *International Journal of Research Publication and Reviews*, vol. 2, no. 5, pp. 9–17, 2021, [Online]. Available: www.ijrpr.com

[18] J. Sanalohit and T. Katanyukul, “Thai Finger Spelling Recognition: Investigating MediaPipe Hands Potentials,” Jan. 2022, [Online]. Available: http://arxiv.org/abs/2201.03170

[19] F. D. Tanugraha, “TA : Sistem Pengenalan Aktivitas Manusia Menggunakan Long Short-Term Memory dan Mediapipe - Repositori Universitas Dinamika,” Feb. 2022. Accessed: Apr. 05, 2022. [Online]. Available: https://repository.dinamika.ac.id/id/eprint/6190/

[20] Y. Kong and Y. Fu, “Human Action Recognition and Prediction: A Survey,” *Computing Research Repository (CoRR)*, Jun. 2018, [Online]. Available: http://arxiv.org/abs/1806.11230

[21] J. M. Chaquet, E. J. Carmona, and A. Fernández-Caballero, “A Survey of Video Datasets for Human Action and Activity Recognition,” *Computer Vision and Image Understanding*, vol. 117, no. 6, pp. 633–659, 2013, doi: 10.1016/j.cviu.2013.01.013.

[22] T. N. Phyu, “Survey of Classification Techniques in Data Mining,” in *International MultiConference of Engineers and Computer Scientists*, 2009, p. 2210. Accessed: Jan. 24, 2022. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary;jsessionid=1E3676C1DEF43A85F8E8E4D26F2C90E7?doi=10.1.1.148.6474

[23] A. P. Wibawa, M. Guntur, A. Purnama, M. Fathony Akbar, and F. A. Dwiyanto, “Metode-Metode Klasifikasi,” *Prosiding Seminar Ilmu Komputer dan Teknologi Informasi*, vol. 3, no. 1, 2018, Accessed: Jan. 24, 2022. [Online]. Available: http://e-journals.unmul.ac.id/index.php/SAKTI/article/view/2101

[24] R. Y. Dillak, D. M. Pangestuty, and M. G. Bintiri, “Klasifikasi Jenis Musik Berdasarkan File Audio Menggunakan Jaringan Syaraf Tiruan Learning Vector Quantization,” *Seminar Nasional Informatika (SEMNASIF)*, vol. 1, no. 3, 2012, Accessed: Jan. 24, 2022. [Online]. Available: http://jurnal.upnyk.ac.id/index.php/semnasif/article/view/1064

[25] M. Ekman, *Learning Deep Learning : Theory and Practice of Neural Networks, Computer Vision, Natural Language Processing, and Transformers Using TensorFlow*. Addison-Wesley, 2021. Accessed: Apr. 05, 2022. [Online]. Available: https://openlibrary.telkomuniversity.ac.id/home/catalog/id/177699/slug/learning-deep-learning-theory-and-practice-of-neural-networks-computer-vision-natural-language-processing-and-transformers-using-tensorflow.html

[26] J. W. G. Putra, *Pengenalan Konsep Pembelajaran Mesin dan Deep Learning*, 1.4. Tokyo: self-published work, 2020. Accessed: Apr. 05, 2022. [Online]. Available: https://wiragotama.github.io/ebook\_machine\_learning.html

[27] C. Lugaresi *et al.*, “MediaPipe: A Framework for Building Perception Pipelines,” *Computing Research Repository (CoRR)*, Jun. 2019, [Online]. Available: http://arxiv.org/abs/1906.08172

[28] H. Xu, E. G. Bazavan, A. Zanfir, W. T. Freeman, R. Sukthankar, and C. Sminchisescu, “GHUM & GHUML: Generative 3D Human Shape and Articulated Pose Models,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 6184–6193. Accessed: Jan. 27, 2022. [Online]. Available: http://openaccess.thecvf.com/content\_CVPR\_2020/papers/Xu\_GHUM\_\_GHUML\_Generative\_3D\_Human\_Shape\_and\_Articulated\_Pose\_CVPR\_2020\_paper.pdf

[29] Google Developers, “Pose Detection,” Aug. 12, 2021. https://developers.google.com/ml-kit/vision/pose-detection (accessed Jan. 27, 2022).

[30] H. Sak, A. Senior, and F. Beaufays, “Long Short-Term Memory Based Recurrent Neural Network Architectures for Large Vocabulary Speech Recognition,” *Computing Research Repository (CoRR)*, Feb. 2014, [Online]. Available: http://arxiv.org/abs/1402.1128

[31] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

[32] C. Olah, “Understanding LSTM Networks,” Aug. 27, 2015. https://colah.github.io/posts/2015-08-Understanding-LSTMs/ (accessed Jan. 25, 2022).

[33] I. Habibie, “Identifikasi Judul Berita Clickbait Berbahasa Indonesia dengan Algoritma Long Short Term Memory (LSTM) Recurrent Neural Network,” Medan, May 2018. Accessed: Jan. 26, 2022. [Online]. Available: https://repositori.usu.ac.id/handle/123456789/8874

[34] T. N. Minh, M. Sinn, H. T. Lam, and M. Wistuba, “Automated Image Data Preprocessing with Deep Reinforcement Learning,” *Computing Research Repository (CoRR)*, Jun. 2018, doi: 10.48550/ARXIV.1806.05886.

[35] S. Beniwal, G. Jambheshwar, and J. Arora, “Classification and Feature Selection Techniques in Data Mining,” *International Journal of Engineering Research & Technology (IJERT)*, vol. 1, no. 6, Aug. 2012, [Online]. Available: https://www.researchgate.net/publication/263662705

[36] S. Amershi *et al.*, “Software Engineering for Machine Learning: A Case Study,” in *Proceedings - 2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice, ICSE-SEIP 2019*, May 2019, pp. 291–300. doi: 10.1109/ICSE-SEIP.2019.00042.

[37] H. Mason and C. Wiggins, “A Taxonomy of Data Science,” 2010. https://sites.google.com/a/isim.net.in/datascience\_isim/taxonomy (accessed Jan. 27, 2022).

[38] J. Janssens, “Data Science at the Command Line, 2e: Obtain, Scrub, Explore, and Model Data with Unix Power Tools,” Dec. 17, 2021. https://datascienceatthecommandline.com/2e/ (accessed Jan. 27, 2022).

[39] J. Xu, Y. Zhang, and D. Miao, “Three-Way Confusion Matrix for Classification: A Measure Driven View,” *Information Sciences*, vol. 507, pp. 772–794, Jan. 2020, doi: 10.1016/j.ins.2019.06.064.

[40] M. Sokolova and G. Lapalme, “A Systematic Analysis of Performance Measures for Classification Tasks,” *Information Processing and Management*, vol. 45, no. 4, pp. 427–437, Jul. 2009, doi: 10.1016/j.ipm.2009.03.002.

[41] N. Chinchor, “MUC-4 Evaluation Metrics,” in *Proceedings of the 4th conference on Message understanding  - MUC4 ’92*, 1992, p. 22. doi: 10.3115/1072064.1072067.

[42] Y. Sasaki, “The Truth of The F-measure,” Jan. 2007. Accessed: Apr. 12, 2022. [Online]. Available: https://www.researchgate.net/publication/268185911\_The\_truth\_of\_the\_F-measure

[43] T. Wang, Y. Chen, M. Zhang, J. Chen, and H. Snoussi, “Internal Transfer Learning for Improving Performance in Human Action Recognition for Small Datasets,” *IEEE Access*, vol. 5, pp. 17627–17633, Sep. 2017, doi: 10.1109/ACCESS.2017.2746095.

[44] N. Reimers and I. Gurevych, “Reporting Score Distributions Makes a Difference: Performance Study of LSTM-networks for Sequence Tagging,” *EMNLP 2017 - Conference on Empirical Methods in Natural Language Processing, Proceedings*, pp. 338–348, Jul. 2017, doi: 10.48550/arxiv.1707.09861.

[45] Keras, “ModelCheckpoint.” https://keras.io/api/callbacks/model\_checkpoint/ (accessed Apr. 05, 2022).

[46] N. McCullum, “The Ultimate Guide to Recurrent Neural Networks in Python,” Jul. 13, 2020. https://www.freecodecamp.org/news/the-ultimate-guide-to-recurrent-neural-networks-in-python/ (accessed Apr. 05, 2022).

[47] C. Schuldt, I. Laptev, and B. Caputo, “Recognizing Human Actions: A Local SVM Approach,” in *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, 2004, pp. 32-36 Vol.3. doi: 10.1109/ICPR.2004.1334462.

[48] “Recognition of Human Actions,” Jan. 18, 2005. https://www.csc.kth.se/cvap/actions/ (accessed Apr. 05, 2022).

# BIODATA MAHASISWA

|  |  |
| --- | --- |
|  | Nama Mahasiswa : Ichsan Arsyi Putra  NIM : 21120118120029  Konsentrasi : Multimedia  Tempat, Tgl. Lahir : Kab. Semarang, 16 April 2000  Alamat Sekarang : Langensari, RT 12 RW 6, Kec.  Ungaran Barat, Kab. Semarang  No. Telepon/HP : 085643413814  Alamat *e-mail* : ichsan.x.an@outlook.co.id  Nama Orang Tua : Samiran  Alamat Orang Tua : Langensari, RT 12 RW 6, Kec.  Ungaran Barat, Kab. Semarang  Telp./HP: 085729144748  IP Kumulatif : 3,98  Tahun Lulus : 2022  Masa Studi : 4 tahun |

**Pengalaman dan Prestasi yang pernah diraih:**

|  |  |  |
| --- | --- | --- |
| No. | Pengalaman atau Prestasi | Tahun |
| 1. | Staf Bidang Ekonomi dan Bisnis, Himpunan Mahasiswa Teknik Komputer | 2019 |
| 2. | Staf Muda Bidang Ekonomi, Al-Muharrik | 2019 |
| 3. | Staf Muda Bidang Media, Forum Studi Teknik | 2019 |
| 4. | Kepala Bidang Ekonomi dan Bisnis, Himpunan Mahasiswa Teknik Komputer | 2020 |
| 5. | Sekretaris Bidang Ekonomi, Al-Muharrik | 2020 |
| 6. | Staf Ahli Bidang Media, Forum Studi Teknik | 2020 |
| 7. | Anggota Divisi Multimedia, *Computer Engineering Research Club* | 2020-2021 |
| 8. | Anggota Bidang Desain, Ruang Karya | 2021 |
| 9. | Staf Unit Pengembangan dan *Monitoring* Organisasi, Himpunan Mahasiswa Teknik Komputer | 2021 |
| 10. | Asisten Praktikum Multimedia | 2021 |

Semarang, XX Juni 2022

Ichsan Arsyi Putra