

Pengenalan Gerakan Olahraga Berbasis (Long Short- Term Memory) Menggunakan Mediapipe

Filbert Daniel Tanugraha¹⁾, Heri Pratikno²⁾, Musayyanah^{*3)}, dan Weny Indah Kusumawati⁴⁾

¹⁾Teknik Komputer, Fakultas Teknologi dan Informatika, Universitas Dinamika

Jalan Raya Kedung Baruk No.98, Surabaya, 60298, Indonesia

Email: filbertdaniel21@gmail.com¹⁾, heri@dinamika.ac.id²⁾, musayyanah@dinamika.ac.id³⁾,

weny@dinamika.ac.id⁴⁾

Abstrak

Olahraga Yoga menjadi gaya hidup yang sangat trend di kalangan masyarakat. Untuk memastikan gerakan Yoga yang dilakukan benar atau tidak, maka perlu adanya sistem deteksi/pengenalan dan klasifikasi yang terkenal dengan istilah HAR (Human Activity Recognition). Sistem pengenalan gerakan olahraga aktivitas manusia, seperti Olahraga Yoga yang dikenal dengan Tree Pose, T-Pose, Warrior II Pose. Gerakan ini dapat dikenali secara real time dan akurat menggunakan MediaPipe. Sedangkan untuk klasifikasi olahraganya manusia menggunakan Long Short-Term Memory (LSTM) dengan memanfaatkan keypoints dari urutan video tersebut. Klasifikasi dan deteksi gerakan menggunakan kamera handphone dalam pengambilan video menggunakan aplikasi DroidCam untuk mendeteksi keypoint. Pada penelitian ini, hasil akurasi dari training sebesar 91% dan loss sebesar 29%. Hasil pengenalan gerakan T-Pose, Warrior II Pose, dan Tree Pose berturut-turut adalah 100%, 85% dan 80%.

Kata kunci: Long Short – Term Memory, Human Activity Recognition, Machine Learning, MediaPipe

1. Pendahuluan (Introduction)

Pengenalan gerakan manusia sudah pernah diidentifikasi menggunakan CCTV, Robot, dan IMK(Interaksi Manusia Komputer). Seiring berjalannya kemajuan teknologi, pengenalan tersebut diterapkan menggunakan keilmuan bidang Computer Vision dan Machine Learning. Computer Vision dan Machine Learning dapat mendeteksi gerakan manusia dengan akurat yang biasanya dikenal dengan istilah HAR (Human Activity Recognition). HAR dapat mendeteksi gerakan yang dilakukan manusia di dalam ruangan seperti berjalan, berdiri, duduk dan berbicara. Namun, HAR belum dapat mendeteksi gerakan spesifik dari Olahraga tertentu.

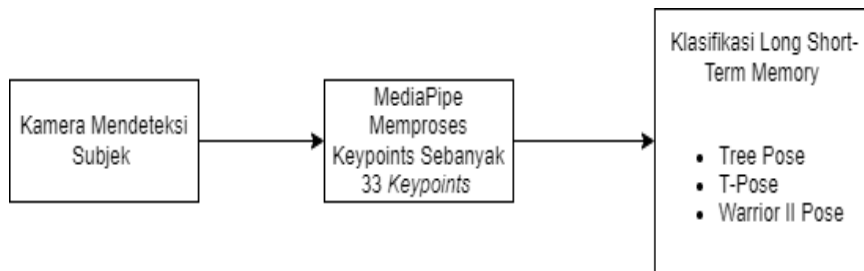
Yoga merupakan salah satu gerakan olahraga yang banyak diikuti kaum wanita. Yoga memiliki gerakan yang unik (Yulinda, Purwaningsih and Sudarta, 2017). Jenis gerakan yoga dianggap dapat memberikan keseimbangan dan kesehatan bagi tubuh serta mental (Arisman *et al.*, 2021). Dengan adanya teknologi HAR pada Computer Vision, gerakan Yoga dapat dipantau agar tidak salah gerakannya, sehingga tidak perlu didampingi oleh tutor Yoga. Selama pandemi, sanggar Olahraga ditutup dan dihimbau untuk olahraga di rumah. Karena gerakan Yoga dianggap lumayan sulit bagi pemula (Firdausi, 2018), sehingga dibutuhkan adanya teknologi HAR untuk membantu mengenalkan gerakan tersebut. Agar gerakan Yoga yang dilakukan benar.

Pada penelitian sebelumnya yang dilakukan oleh (Sun *et al.*, 2018) melakukan pengenalan gerakan manusia dengan CNN, dengan cara melakukan *tracking* kemudian ditampilkan secara visual dengan kamera. Namun pengenalan ini dianggap tingkat klasifikasinya kurang akurat. CNN dianggap bekerja dengan baik pada deteksi gambar dan ucapan saja yang tidak real time (Li *et al.*, 2019). Oleh sebab itu, pada artikel ini diangkat metode lain yaitu, LSTM (Long Short-Term Memory) yang berbasis RNN yang dapat mendeteksi suatu gerakan secara *riel time* (Talita and Wiguna, 2019).

Pada artikel ini membahas mengenai sistem pengenalan dan klasifikasi gerakan Yoga yaitu T-Pose, Warrior II Pose, dan Tree Pose menggunakan Mediapipe dan LSTM. Gerakan Yoga yang benar dapat dengan mudah dideteksi pada sistem ini. Sistem ini sangat tepat diterapkan pada saat pandemi Covid-19, menghindari keramaian dan seseorang tidak perlu keluar rumah untuk berolahraga Yoga.

2. Metode Penelitian (Methods)

Secara prosedur teori pada HAR, aktifitas gerakan tubuh manusia di setiap Frame dapat diklasifikasikan kemudian dilanjutkan dengan analisis bagian tubuh dari gerakan tersebut berdasarkan pengaturan waktu dari klasifikasiya. Prosedur pertama gerakan direkam menggunakan kamera, kemudian dilanjutkan dengan deteksi 33 Keypoints dari postur badan manusia. Setelah itu, video dengan 33 keypoint pada frame video diamati apakah ada yang tidak terdeteksi. Prosedur kedua menganalisis gerakan tubuh dengan 33 keypoints secara *riel time* dan membuat klasifikasi gerakan dengan LSTM secara *riel time*. Prosedur penelitian ini digambarkan pada Gambar 1.

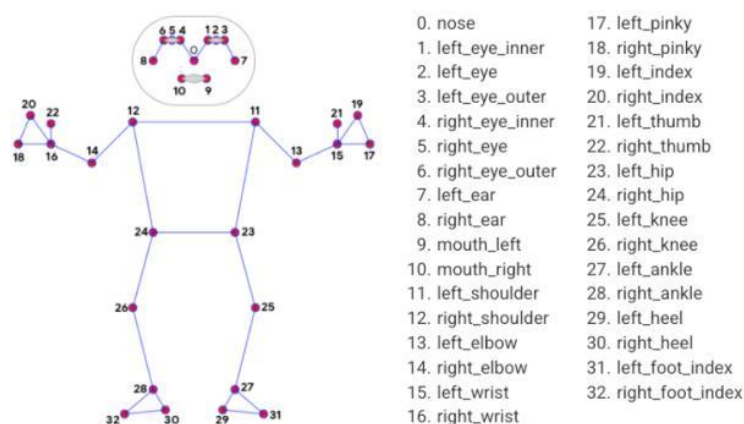


Gambar 1. Prosedur deteksi dan klasifikasi gerakan

Deteksi Keypoints

MediaPipe membangun sebuah “pipeline” dan akan mengolah data yang masuk sembarangan. *Mediapipe* merupakan sebuah framework yang dirancang untuk kecerdasan buatan dalam sebuah aplikasi. *Mediapipe* juga mengandung *Tensor Flow* yang mendukung akselerasi GPU (*Grapihcal Processing Unit*) dan CPU (*Central Processing Unit Flow*) dari sebuah perangkat (Zhang *et al.*, 2020).

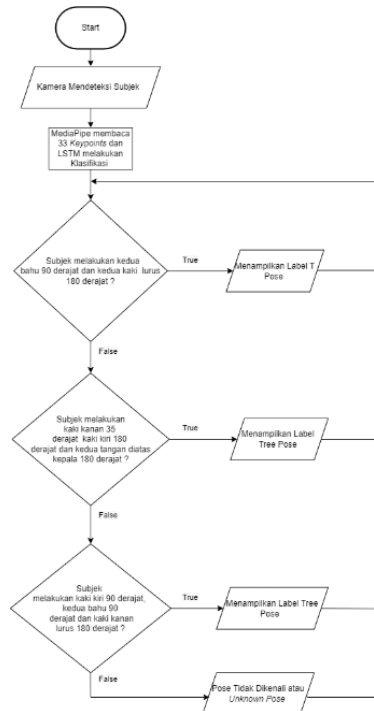
Langkah pertama dicapai dengan menggunakan kamera atau webcam yang digunakan, lalu *MediaPipe* menampilkan kerangka mewakili dtubuh badan manusia. postur tubuh atau *body skeletons* sebanyak (33 *keypoints*). Setelah itu mengamati satu *frame* dalam sebuah video atau gambar. Kemudian menganalisis gerakan dari tubuh secara berubah dan membuat prediksi itu. dilakukan dengan menggunakan *Long Short-Term Memory* (LSTM) (Sanalohit and Katanyukul, 2022). *Keypoints* dari urutan *frame* dikirim ke LSTM untuk klasifikasi memprediksi aktivitas yang sesuai. Pada penelitian ini, digunakan CPU dengan prosessor yang tinggi agar komputasi dari proses deteksi dan klasifikasi lebih efisien serta lebih cepat. Khususnya untuk deteksi 33 Keypoints pada inputan vidio, yang ditunjukkan pada Gambar 2.



Gambar 2 Index Keypoints pada *Mediapipe*
 (MediaPipe, 2020)

Pada Gambar 3 menjelaskan algoritma sistem mendeteksi dan mengkalsifikasi gerakan menggunakan kamera. Deteksi pertama adalah *MediaPipe* memproses pembacaan 33 *keypoints* kemudian dilanjutkan dengan training dataset gerakan menggunakan LSTM. Kemudian terjadi pengecekan kondisi untuk klasifikasi gerakan. Kondisi pertama untuk gerakan T pose adalah kedua baru

searah 90 derajat dan kedua kaki lurus 180 derajat, jika benar maka akan tercetak label “T Pose”. Jika Flase, maka dilanjutkan dengan pengecekan gerakan Tree Pose yaitu dengan kaki kanan 35 derajat, kaki kiri 180 derajat, kedua tangan di atas kepala 180 derajat. Jika kondisi untuk Tree Pose benar maka akan tercetak label “Tree Pose”. Jika salah, maka dilanjutkan untuk pengecekan gerakan Warrior I Pose. Gerakan Warior II Pose kaki kiri ditekuk 90serajat, kedua bahu lurus 90 derajat dan kaki kanan lurus 180 derajat. Jika gerakan benar, maka tercetak label “Warrior II Pose”. Jika ketiga kondisi tersebut tidak terdeteksi maka akan tercetak “Unknown Pose” atau kumpulan gerakan yang tidak dapat diklasifikasi. Hal ini dilakukan pengulangan sebanyak 33 point pada Mediapipe.

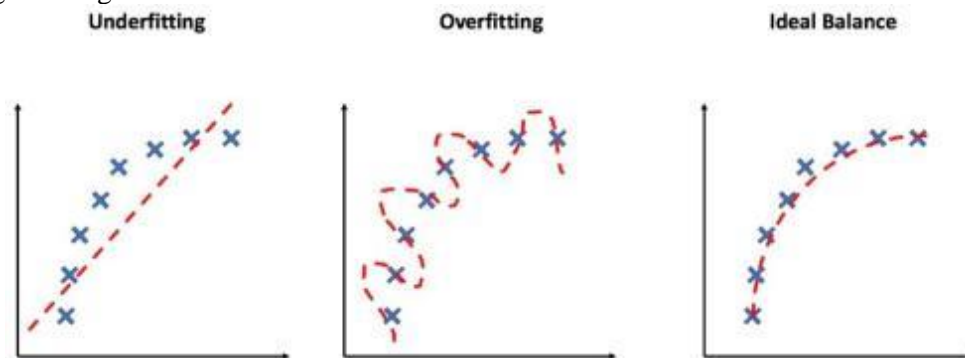


Gambar 3. Flowchart keseluruhan sistem

3. Hasil dan Pembahasan (Results and Discussions)

Pengujian Hasil Training Long Short-Term Memory

LSTM digunakan untuk training dataset. Poses training menggunakan 400 epocs, setiap epocs memiliki ukuran batch 512 dan iterasinya 57 kali untuk satu epocs. Proses Training menerapkan optimizer “Adam” dengan *learning rate* 0.001. Optimizer “Adam” dianggap mudah, efisien, komputasi ringan, dan ampuh untuk *noise* tinggi. Proses training hanya ada pada grafik *train_loss*, *rain_acc*, *val_loss*, dan *val_acc*. Durasi training rata-rata 10 menit sampai 12 menit untuk 4 kali percobaan dengan epocs 390-399. Untuk hasil perbandingan proses training dimulai dari waktu 10 menit, hal ini disebabkan oleh terdapat *transfer learning* pada LSTM untuk menghasilkan grafik yang lebih halus dibandingkan dengan metode lain.



Gambar 3. Grafik (a) tampilan underfitting, (b) tampilan overfitting, (c) tampilan ideal balance (Karthik Ramasubramanian, 2020)

Pada Gambar 3 menjelaskan bahwa *Overfitting* merupakan hasil dengan loss rendah dan akurasi rendah. *Underfitting* merupakan kondisi dengan loss yang tinggi dan akurasi tinggi. Dan *Ideal Balance* menunjukkan loss yang rendah dan akurasi yang tinggi.

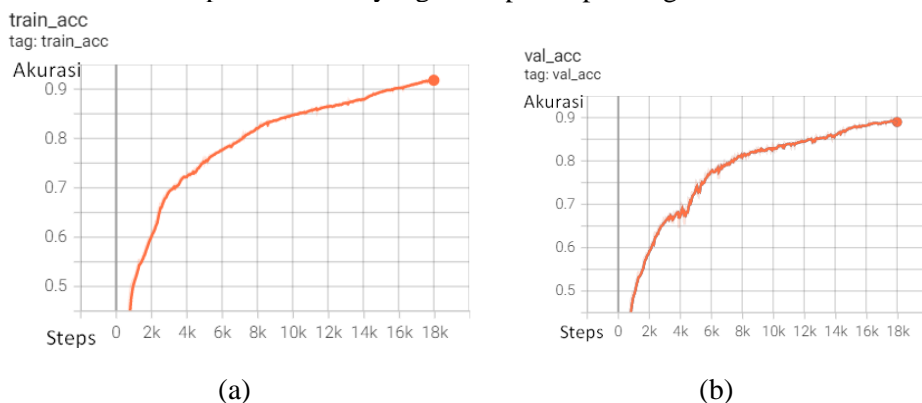
Perhitungan Training Dan Step

Sebelum proses training dataset, ada beberapa prosedur harus yang dilakukan penentuan Epochs. Epochs digunakan untuk step semua data training, yang nantinya dihitung untuk mendapatkan gradien dan optimisasi untuk kebutuhan training model. Sedangkan penggunaan step suatu ukuran Epochs training untuk mendapatkan sebuah model. Jadi jumlah *batch* sama dengan jumlah iterasi untuk satu epoch (Jalil, 2022). Berdasarkan pengujian data dengan menggunakan 400 Epochs untuk 7200000 data, maka didapatkan persamaan (1). Berdasarkan persamaan (1) didapatkan 18.000 steps.

$$\begin{aligned} \text{Steps} &= \frac{\text{Banyaknya data training examples}}{\text{Epochs}} & (1) \\ &= \frac{7.200.000}{400} = 18.000 \text{ Steps} \end{aligned}$$

Pengujian Parameter Accuracy Dan Validation Accuracy

Pada pengujian ini merupakan hasil perhitungan dari parameter akurasi deteksi dan validasi akurasi. Perhitungan ini intinya digunakan untuk mengontrol parameter *error* dalam proses training. Seperti pada Gambar 5 sampai Gambar 6 yang merupakan plotan grafik dari hasil training.



Gambar 5 (a) Hasil parameter *train_accuracy* (b) Hasil parameter *val_acc*

Gambar 5 (a) merupakan akurasi training dari *function Training Dataset* dengan hasil 29% dapat dilihat pada titik akhir dari grafik. Sedangkan pada Gambar 5 (b) adalah hasil dari parameter validasi akurasi sebesar 91%, yang menunjukkan kinerja model dari prediksi Mediapipe.



Gambar 6 (a) Hasil plot *batch_train_acc* (b) Hasil plot *batch_val_acc*

Gambar 6 (a) menentukan banyaknya contoh training yang diproses secara paralel untuk interferensi training. Parameter *Batch_Train_Acc* dijadikan sebagai ukuran batch akurasi dari intereferensi training. Ukuran batch pada saat proses Training dapat mempengaruhi seberapa cepat dan seberapa baik training yang dilakukan. Loss dari parameter *Batch_Train_Acc* adalah 93%. Sedangkan pada Gambar 6 (b) menunjukkan parameter *Batch_Val_Acc*, yang digunakan untuk mengontrol ukuran *batch* penentu akurasi melalui *dataset Training*. Loss yang dihasilkan dari parameter *Batch Val Acc* sekitar 88%.

Tabel 1 Perbandingan Parameter Akurasi dan Validasi dari Training

Epochs	Batch_Train Acc (%)	Batch_Val Acc (%)	Train_Acc (%)	Val_Acc (%)
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 1 menunjukkan hasil klasifikasi keberhasilan setiap gerakan pada saat program dijalankan. Parameternya adalah *Batch_Train_Acc*, *Batch_Val_Acc*, *Train_Acc*, *Val_Acc*. Proses klasifikasi gerakan dilakukan pada Epochs 390-399 dengan tujuan menghindari terjadinya *Overfitting* maupun *Underfitting* selama proses training.

Pengujian Keypoint atau Mediapipe

Pengujian *Keypoints* menggunakan kamera handphone dengan aplikasi *DroidCam*. Pengujian ini melibatkan 4 sampel dengan 20 kali percobaan. Pengujian ini bertujuan untuk mengetahui hasil *Real-Time Keypoints 33 Keypoint*. Bentuk dari 33 Keypoint disajikan pada Gambar 7 sampai Gambar 9. Tabel 2 merupakan hasil pengujian 33 Keypoints yang terdiri dari *nose* (hidung) sampai pangkal titik kaki kanan.

Tabel 2. Hasil Deteksi 33 Keypoints

Index	Nama Keypoint	Keberhasilan Keypoint	Keterangan
0	Hidung	√	✓: Berhasil
1	Mata kiri bagian dalam	√	- : Tidak berhasil
2	Mata kiri	√	
3	Mata kiri bagian luar	√	

4	Mata kanan bagian dalam	√
5	Mata kanan	√
6	Mata kanan bagian luar	√
7	Telinga kiri	√
8	Telinga kanan	√
9	Mulut kiri	√
10	Mulut kanan	√
11	Bahu kiri	√
12	Bahu kanan	√
13	Siku kiri	√
14	Siku kanan	√
15	Pergelangan tangan kiri	√
16	Pergelangan tangan kanan	√
17	Jari keliling kiri	√
18	Jari keliling kanan	√
19	Index kiri	√
20	Index kanan	√
21	Jari jempol kiri	√
22	Jari jempol kanan	√
23	Pinggul kiri	√
24	Pinggul kanan	√
25	Lutut kiri	√
26	Lutut kanan	√
27	Pergelangan kaki kiri	√
28	Pergelangan kaki kanan	√
29	Tumit kiri	√
30	Tumit kanan	√
31	Index kaki kiri	√
32	Index kaki kanan	√

Pengujian Klasifikasi Gerakan

Pengujian klasifikasi gerakan melibatkan 4 subjek dengan postur tubuh yang berbeda-beda. Pengujian ini dilakukan sebanyak 5 kali pada lokasi dan waktu yang sama. Setiap gerakan subjek tersebut direkam oleh kamera handphone dengan DroidCam untuk mendapatkan titik keypoints postur tubuh manusia secara maksimal. Gerakan tersebut tersimpan sebagai frame. Kemudian frame tersebut diproses oleh Long Short-Term Memory (LSTM) untuk klasifikasi dan memprediksi gerakan yang sesuai yaitu: T-Pose, Warrior II Pose, Tree Pose. Terdapat beberapa faktor yang dapat mempengaruhi hasil uji coba ini yaitu peletakan kamera, pemilihan lokasi, peniruan gerakan yang tepat. Kamera diusahakan menggunakan tripod agar kamera dapat mengambil frame gerakan tidak bergetar. Pemilihan lokasi yang dimaksud adalah selama proses pengujian hilangkan benda sekitar yang dapat mempengaruhi proses deteksi serta luas yang tidak dihalangi benda. Peniruan gerakan harus dilakukan dengan benar. Gerakan T-Pose dan Warrior II Pose kedua lengan lurus dan bahu harus pada sudut tertentu. Untuk gerakan Warrior II Pose, satu kaki lurus dan kaki lainnya ditekuk pada sudut yang diperlukan, sedangkan T-Pose gerakan kedua kaki lurus harus lurus dan kedua tangan lurus 180° derajat.

Pengujian pada Tabel 3 merupakan percobaan untuk klasifikasi gerakan, dimana nama klasifikasi gerakan ini muncul di bagian atas frame. Kegagalan dalam mengklasifikasi gerakan dapat dilihat pada Gambar 9, yaitu gerakan Warrior dengan keterangan *Unknown Pose* (tanda kotak merah). Kegagalan tersebut disebabkan oleh tidak tepatnya gerakan yang dilakukan. Gerakan Warrior II Pose seharusnya posisi kaki kiri menekuk pada sudut 90 derajat dan kaki kanan lurus 180 derajat. Kemudian

Gerakan T-Pose seharusnya kedua tangan dan kedua kaki yang harus lurus 180 derajat dan Gerakan Tree Pose seharusnya posisi kaki bertumpu 180 derajat di satu lutut dan 35 derajat di kanan lutut atau 335 derajat kiri lutut. Berdasarkan pengujian, pada Tabel 3 menunjukkan bahwa pengenalan gerakan mendapatkan nilai akurasi sebesar 85% untuk Warrior II Pose, 100% untuk T-Pose, dan 80% untuk Tree Pose.

Tabel 3 Keberhasilan klasifikasi gerakan

Subject	Warrior II Pose	T-Pose	Tree Pose	Keterangan
1	✓	✓	✓	✓: dikenali -: tidak dikenali
	✓	✓	✓	
	✓	✓	✓	
	✓	✓	-	
	-	✓	✓	
2	-	✓	✓	✓: dikenali -: tidak dikenali
	✓	✓	✓	
	✓	✓	✓	
	✓	✓	✓	
	-	✓	✓	
3	✓	✓	-	✓: dikenali -: tidak dikenali
	✓	✓	-	
	✓	✓	✓	
	✓	✓	✓	
	✓	✓	✓	
4	✓	✓	-	✓: dikenali -: tidak dikenali
	✓	✓	✓	
	✓	✓	✓	
	✓	✓	✓	
	✓	✓	✓	
Rata-Rata	85%	100%	80%	

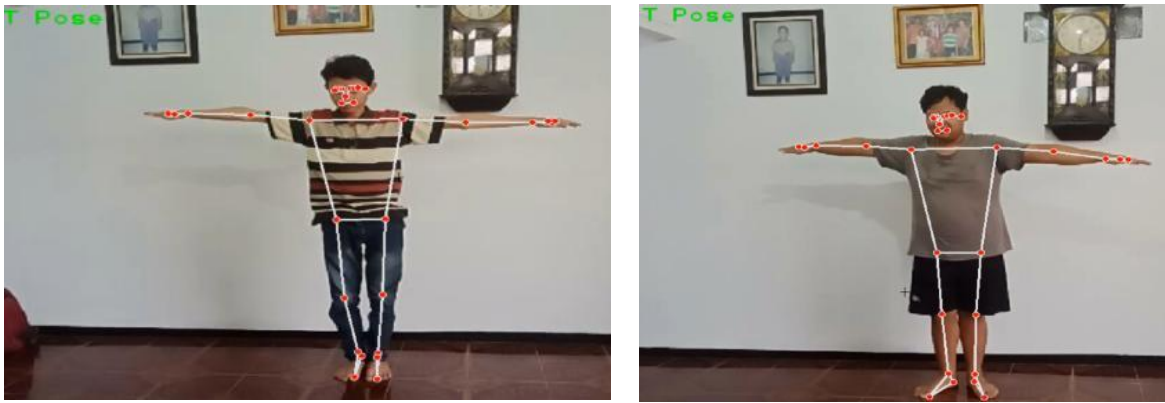
$$\text{Rata-rata Keberhasilan gerakan} = \frac{\text{Gerakan Terklasifikasi} \times \text{Total Percobaan}}{\text{Banyaknya Subjek}} * 100\%$$

$$\text{Gerakan Warrior II Pose} = \left(\frac{17 \times 20}{4}\right) * 100\% = 85\%$$

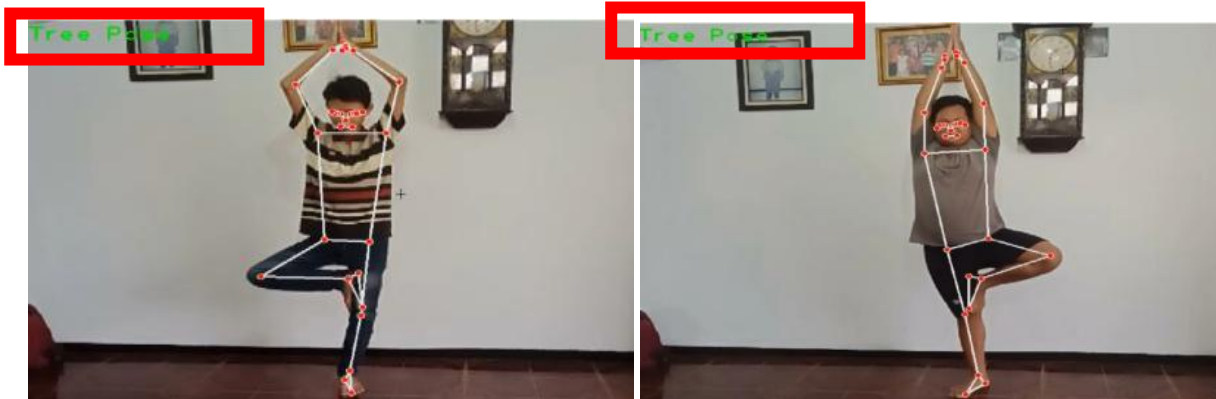
$$\text{Gerakan T-Pose} = \left(\frac{20 \times 20}{4}\right) * 100\% = 100\%$$

$$\text{Gerakan Tree Pose} = \left(\frac{16 \times 20}{4}\right) * 100\% = 80\%$$

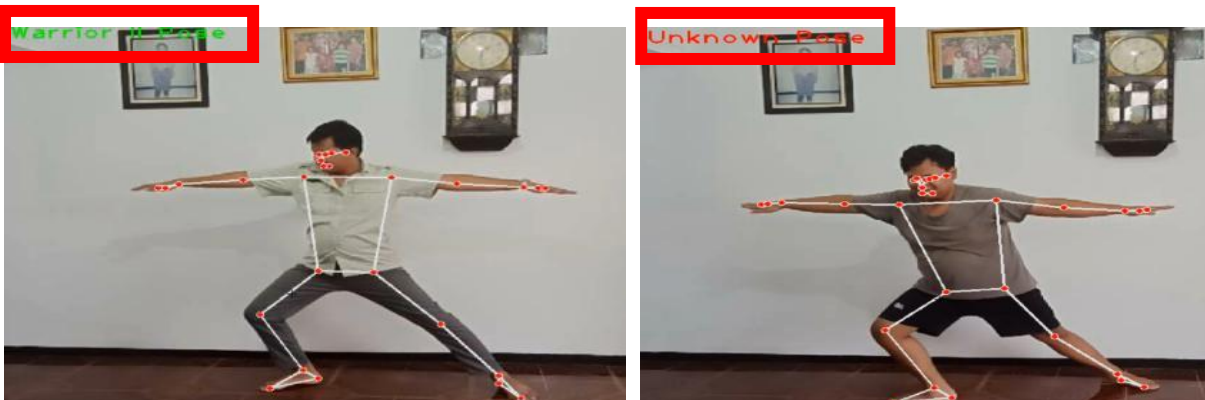




Gambar 7. Pengenalan Gerakan T-Pose



Gambar 8. Pengenalan Gerakan Tree-Pose



Gambar 9. Pengenalan Gerakan Warrior

3.1. Kesimpulan (Conclusion)

Akurasi Training Dataset LSTM dari gerakan yang telah dilakukan adalah 91% . Parameter validasi menghasilkan nilai akurasi sebesar 89%, dengan nilai loss sebesar 0.29 dan validasi loss sebesar 0.35 dengan 400 Epocs. Klasifikasi gerakan “T-Pose” sudah berhasil terklasifikasi 100%. Klasifikasi gerakan “Warrior II Pose berhasil terklasifikasi 85%. Penyebab dari gerakan yang tidak berhasil dikenali diakrenakan pencahayaan pada arena sekitar gerakan kurang merata, posisi kedua tangan tidak sejajar dengan tepat dan belum searah sebesar 180° derajat. Sedangkan klasifikasi “Tree Pose” didapatkan nilai akurasi sebesar 80%.

Daftar Pustaka

Arisman, A. *et al.* (2021) “Resistensi Yoga dalam Meningkatkan Konsentrasi Ketepatan Memanah,” *Jurnal Patriot*, 3, pp. 71–81. doi: 10.24036/patriot.v.

- Firdausi, I. A. (2018) “Komunikasi Instruksional Di Kelas Yoga Club Health Fatimah Kota Serang,” *LONTAR: Jurnal Ilmu Komunikasi*, 6(2), p. 139. doi: 10.30656/lontar.v6i2.952.
- Jalil (2022) *Penentuan Epocs dan Batch Size pada Klasifikasi CNN*.
- Karthik Ramasubramanian (2020) *Applied Survised Learning With R. In J.M.*
- Li, T. H. S. *et al.* (2019) “CNN and LSTM Based Facial Expression Analysis Model for a Humanoid Robot,” *IEEE Access*, 7, pp. 93998–94011. doi: 10.1109/ACCESS.2019.2928364.
- Sanalohit, J. and Katanyukul, T. (2022) “TFS Recognition: Investigating MPH] {Thai Finger Spelling Recognition: Investigating MediaPipe Hands Potentials,” pp. 1–19. Available at: <http://arxiv.org/abs/2201.03170>.
- Sun, J. *et al.* (2018) “Sequential human activity recognition based on deep convolutional network and extreme learning machine using wearable sensors,” *Journal of Sensors*, 2018(1). doi: 10.1155/2018/8580959.
- Talita, A. S. and Wiguna, A. (2019) “Implementasi Algoritma Long Short-Term Memory (LSTM) Untuk Mendeteksi Ujaran Kebencian (Hate Speech) Pada Kasus Pilpres 2019,” *MATRIK : Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer*, 19(1), pp. 37–44. doi: 10.30812/matrik.v19i1.495.
- Yulinda, Y., Purwaningsih, D. and Sudarta, C. M. (2017) “Latihan Yoga Dapat Menurunkan Tingkat Kecemasan pada Siklus Menstruasi Remaja Puteri,” *Jurnal Ners dan Kebidanan Indonesia*, 5(1), p. 20. doi: 10.21927/jnki.2017.5(1).20-26.
- Zhang, F. *et al.* (2020) “MediaPipe Hands: On-device Real-time Hand Tracking.” Available at: <http://arxiv.org/abs/2006.10214>.

Halaman ini sengaja dikosongkan