

Jutisi: Jurnal Ilmiah Teknik Informatika dan Sistem Informasi

<https://ojs.stmik-banjarbaru.ac.id/index.php/jutisi/index>

Jl. Ahmad Yani, K.M. 33,5 - Kampus STMIK Banjarbaru

Loktabat – Banjarbaru (Tlp. 0511 4782881), e-mail: puslit.stmikbjb@gmail.com

e-ISSN: 2685-0893

Pengembangan Model *Long Short-Term Memory* Berbasis MediaPipe Pose untuk Klasifikasi dan Penilaian Gerakan *Push-Up*

DOI: <http://dx.doi.org/10.35889/jutisi.v14i3.3363>

Creative Commons License 4.0 (CC BY – NC)



I Kadek Bayu Danu Artha^{1*}, I Ketut Resika Arthana², Putu Hendra Suputra³

Ilmu Komputer, Universitas Pendidikan Ganesha, Singaraja, Indonesia

*e-mail Corresponding Author. hendra.suputra@undiksha.ac.id

Abstract

The counting and assessment of push-up movements are often inaccurate and subjective because they are done manually. Existing automated approaches generally only consider the top and bottom positions. This makes the assessment incomplete, so that imperfect movements can be counted as correct. This study aims to develop a Long Short-Term Memory (LSTM) model using MediaPipe Pose landmark extraction to detect and classify push-up movements based on the overall movement pattern. Data was collected in the form of videos validated by experts and processed into 3,600 video data sets with 10 movement classes. The video data was extracted to produce normalized keypoint coordinates and joint angles, and padding was added to equalize the data length. The model was trained using the K-Fold Cross Validation method with eight different architectures. The results showed that the best model achieved an average testing accuracy of 92.35% and an F1-Score of 92.53%. These findings indicate that the combination of MediaPipe Pose and LSTM can effectively recognize and classify push-up movements.

Keywords: MediaPipe Pose; Long short-term memory; Classification; Push-up; Time series

Abstrak

Penghitungan dan penilaian gerakan push-up sering tidak akurat serta bersifat subjektif akibat dilakukan secara manual. Pendekatan otomatis yang ada umumnya hanya mempertimbangkan posisi atas dan bawah. Hal ini membuat penilaian tidak secara keseluruhan sehingga gerakan yang tidak sempurna dapat terhitung benar. Penelitian ini bertujuan mengembangkan model *Long Short-Term Memory* (LSTM) dengan memanfaatkan ekstraksi landmark MediaPipe Pose untuk mendeteksi dan mengklasifikasi gerakan *push-up* berdasarkan keseluruhan pola gerakan. Data dikumpulkan dalam bentuk video yang divalidasi oleh ahli dan diolah hingga menjadi 3.600 data video dengan 10 kelas gerakan. Data video diekstraksi hingga menghasilkan koordinat *keypoint* dan sudut sendi yang dinormalisasi, serta ditambahkan padding untuk menyamakan panjang data. Model dilatih dengan metode *K-Fold Cross Validation* dengan delapan arsitektur berbeda. Hasil penelitian menunjukkan performa model terbaik memperoleh rata-rata akurasi testing 92,35% dan F1-Score 92,53%. Temuan ini menunjukkan kombinasi MediaPipe Pose dan LSTM dapat dengan baik mengenali dan mengklasifikasi gerakan push-up.

Kata kunci: MediaPipe Pose; Long short-term memory; Klasifikasi; Push-up; Time series

1. Pendahuluan

Olahraga dimasa ini telah menjadi bagian penting dalam hidup terutamanya untuk menjaga Kesehatan serta kebugaran badan. Melakukan aktivitas fisik dengan intensitas sedang secara rutin, terbukti mampu meningkatkan imunitas dan kesehatan tubuh [1]. Salah satu aktivitas fisik yang cukup populer saat ini adalah kalistenik, yang merupakan bentuk latihan fisik dengan menggunakan beban badan sebagai beban latihannya. Dengan salah satu gerakan yang paling populer adalah push up karena terbukti efektif dalam melatih otot bagian atas tubuh [2]. pada penelitian Otarsky dkk 2018 menunjukkan latihan push-up yang rutin dapat meningkatkan kekuatan otot bagian atas setara dengan latihan bench press dengan intensitas yang sama,

namun dalam peningkatan massa otot latihan bench press masih lebih unggul dibandingkan push-up [3]. Sehingga, push-up dapat menjadi latihan yang mudah, murah dan tidak perlu alat khusus untuk meningkatkan kekuatan otot bagian atas. Selain sebagai latihan kekuatan otot, push-up juga banyak digunakan dalam test kebugaran fisik, seperti pada mata pelajaran jasmani disekolah, seleksi anggota militer ataupun kepolisian. Dalam kontes ini, pencatatan jumlah repetisi push-up yang benar menjadi aspek yang penting dalam memberikan penilaian kemampuan fisik seseorang. Namun pada penelitian Artanayasa dkk. (2022) menunjukkan sebanyak 86,7% responden dalam penelitiannya yang terdiri dari dosen Fakultas Olahraga dan Kesehatan Undiksha dan atle KONI (Komite Olahraga Nasional Indonesia) mengalami kesulitan dalam menghitung repetisi push-up secara manual secara akurat [4]. Pada penelitian tersebut menunjukkan adanya kesenjangan dalam penilaian push-up yang selama ini masih bersifat subjektif dan bergantung terhadap pengamatan manusia.

Berbagai penelitian terdahulu, telah berupaya dalam memberikan solusi berupa inovasi penggunaan sensor fisik yang dapat membantu perhitungan gerakan push-up secara otomatis. Metode pendekatan dengan menggunakan sensor terbukti dapat meningkatkan akurasi perhitungan repetisi, namun masih adanya keterbatasan dalam kebebasan gerak pengguna, perlunya perangkat keras tambahan, serta penggunaanya harus sesuai dengan rancangan alatnya [5], [6]. Selain itu, penggunaan perangkat keras akan menimbulkan adanya biaya tambahan untuk pembelian alat dan juga penggunaan, serta perawatannya yang tidak mudah untuk orang awam. Di sisi lain, teknologi *computer vision* dan pengolahan citra digital semakin berkembang dan membuka peluang baru dalam melakukan monitoring dan deteksi gerak tubuh manusia secara otomatis dengan menggunakan perangkat seperti laptop atau handphone [7]. Salah satu framework yang populer digunakan adalah MediaPipe, yang dikembangkan oleh Google yang sering digunakan untuk pengolahan citra digital berbasis machine learning dengan kemampuan mendeteksi keypoint tubuh secara real-time [8].

Meskipun telah banyak penelitian yang menggunakan pengolahan citra digital untuk mendeteksi aktifitas manusia, namun dalam konteks perhitungan repetisi push-up masih terbatas. Sebagian besar penelitian hanya fokus dalam deteksi posisi atau pose atas dan bawah dalam gerakan push-up dengan pendekatan klasifikasi citra digital, atau menggunakan logika threshold pada nilai sudut tertentu dalam menghitung repetisinya [9], [10]. Pendekatan ini memiliki kelemahan, yaitu tidak mempertimbangkan bagaimana transisi posisi bawah dan atas atau sebaliknya. Hal ini dapat membuat gerakan yang tidak sempurna masih dapat terhitung benar apabila memenuhi logika posisi atas dan bawah atau posisi awal dan akhir dari gerakan push-up. Padahal dalam menilai gerakan push-up ditentukan juga oleh transisi dan kontinuitas gerakan, bukan hanya dari bagaimana posisi bawah dan atas saja. Untuk mengatasi keterbatasan ini, perlu dikembangkan metode yang mampu dalam mengenali pola gerakan dalam rentan waktu tertentu. Salah satu pendekatan yang potensial adalah *Long Short-Time Memory* (LSTM), yaitu model deep learning yang dapat memproses data sekuensial dan mampu mengingat pola data dalam jangka waktu tertentu [11], [12]. Penggunaan LSTM dalam memahami pola gerakan telah terbukti pada beberapa penelitian seperti pada penelitian yang dilakukan oleh Bakti dkk. (2025) yang menggunakan LSTM yang dikombinasikan dengan MediaPipe untuk mendeteksi jatuh pada lansia yang mencapai akurasi sebesar 91% [13]. Hal ini menunjukkan adanya potensi dapat diterapkan juga untuk mengenali gerakan push-up. Dengan memanfaatkan data sekuensial hasil ekstraksi landmark dari MediaPipe, model LSTM dapat dilatih agar mampu mengenali pola gerakan push-up secara lebih akurat.

Berdasarkan hal tersebut, pada penelitian ini mengusulkan penggunaan LSTM dengan memanfaatkan data hasil ekstraksi landmark tubuh dengan MediaPipe sebagai pendekatan untuk mendeteksi dan mengitung repetisi push-up. Kombinasi kedua metode ini diharapkan dapat menghasilkan model yang dapat menilai kualitas gerakan push-up berdasarkan pola gerakan yang dilakukan. Dengan adanya kelas gerakan salah yang dikategorikan kedalam beberapa kelas berdasarkan jenis kesalahan yang dilakukan. Pendekatan ini diharapkan agar dapat meningkatkan akurasi pengenalan pola gerakan push-up untuk alat perhitungan push-up, serta memberikan kontribusi terhadap pengembangan teknologi komputer vision khususnya dibidang olahraga dalam analisis kualitas gerakan fisik secara otomatis.

2. Tinjauan Pustaka

Dalam pemanfaatan citra digital, terdapat penelitian yang dilakukan oleh Arlin dan Munir (2021) yang mengembangkan aplikasi untuk perhitungan push-up dengan mengintegrasikan

model KNN untuk melakukan deteksi pose tubuh saat melakukan *push-up* dengan akurasi 84,7% [9]. Dengan menggunakan acuan deteksi posisi atas dan bawah saat *push-up*, dengan menggunakan data berupa gambar dari posisi atas dan bawah pada *push-up*. Hal ini membuat adanya potensi terjadi kesalahan perhitungan *push-up* yang benar karena mengabaikan transisi antara posisi bawah ke atas ataupun sebaliknya. Selain itu, terdapat penelitian dengan memanfaatkan MediaPipe untuk menghitung repetisi *push-up* yang dilakukan oleh Park dkk. (2020). Dalam penelitian ini, pemanfaatan MediaPipe digunakan untuk memperoleh nilai sudut pada beberapa sendi tubuh. Dengan menggunakan threshold untuk menentukan posisi atas dan bawah *push-up*. Hal ini memungkinkan adanya celah, yaitu dalam transisi posisi bawah ke atas atau sebaliknya yang diabaikan serta pengaruh penggunaan nilai *threshold* yang dapat mempengaruhi pengenalan *push-up* dan juga perhitungannya.

Dilansir lain, terdapat beberapa penelitian yang menggunakan LSTM untuk melakukan analisis data sekueler dari ekstraksi landmark untuk melakukan deteksi aktivitas manusia. Penelitian yang dilakukan Prayoga dkk. (2025) yang menggunakan LSTM untuk mengklasifikasikan kata bahasa isyarat dengan memanfaatkan data landmark dari data video [14]. Penelitian Tanugraha dkk. (2022) memanfaatkan LSTM untuk mengklasifikasi gerakan yoga dengan MediaPipe Pose untuk ekstraksi landmark [15]. Pada penelitian tersebut menunjukkan hasil dari model LSTM memberikan performa yang efektif dalam mengenali pola gerakan aktivitas manusia yang dilakukan. Namun belum banyak penelitian yang menggunakan kombinasi ini untuk mengenali gerakan *push-up*.

Dari tinjauan literatur, pendekatan yang berbasis pada computer vision hanya berfokus pada posisi statis *push-up*. Sehingga penilaian *push-up* yang dilakukan tidak dari keseluruhan gerakan yang dilakukan. Sedangkan LSTM memiliki potensi yang besar untuk menganalisis dan mengenali pola gerakan dari data berurutan hasil ekstraksi landmark MediaPipe. Sehingga penelitian ini berfokus pada pengembangan model LSTM untuk dapat mengenali gerakan *push-up* yang benar dan salah. Dengan memanfaatkan data sekuler hasil ekstraksi landmark. Model juga dapat mengkategorikan gerakan salah sesuai kesalahan yang dilakukan.

3. Metodologi

Dalam melakukan pengembangan model LSTM untuk pengenalan gerakan *push-up* terdapat beberapa tahapan yang dilakukan, yaitu pengumpulan data citra dan pengolahannya menjadi data siap dilakukan ekstraksi, proses ekstraksi feature, dan pembangunan model LSTM.

3.1. Pengumpulan Dan Pengolahan Data Citra

Pengumpulan data citra dilakukan dengan melakukan perekaman video dengan peraga *push-up* dengan didampingi oleh validator. Sehingga data yang digunakan pada penelitian ini dapat dikatakan telah valid dengan adanya validasi dari validator. Hasil dari perekaman data yang dilakukan, data dapat dikelompokkan menjadi 10 kelas, yaitu 1 kelas gerakan benar dan 9 kelas gerakan salah. Gerakan salah yang digunakan juga ditentukan oleh validator yang dinilai biasa dilakukan pada saat test fisik, yaitu gerakan naik atau turun yang tidak sempurna, pinggul yang naik sehingga badan tidak lurus, lutut yang menekuk, serta hanya bagian badan depan yang naik. Sedangkan untuk gerakan benar merupakan gerakan yang sesuai dengan Buku Petunjuk Teknik Tes Kesamaan Jasmani Prajurit Dan Calon Prajurit Tentara Nasional Indonesia Angkatan Darat. Data mentah yang diperoleh selanjutnya dilakukan pengolahan dengan menggunakan aplikasi CapCut Desktop.

3.2. Ekstraksi Feature

Proses ekstraksi feature dimulai dari melakukan ekstraksi landmark dengan menggunakan *MediaPipe Pose*. Dari proses ekstraksi landmark diperoleh data 33 *keypoint* dengan jumlah deret data yang sesuai dengan durasi video. Dalam tahapan ini sangat krusial dalam penelitian ini sehingga dilakukan juga tahapan untuk memastikan landmark terekstraksi secara baik setiap framenya. Hasil data ekstraksi landmark diperoleh dalam bentuk ([jumlah data], 33, 4) dengan setiap keypointnya memiliki nilai x, y, z dan visibilitas. Data ini merupakan data time series yang merupakan data hasil dari pengamatan sebuah pristiwa dalam waktu tertentu dengan pada kasus ini adalah repetisi *push-up* [16].

Data hasil ekstraksi landmark tidak dapat digunakan secara langsung karena jika menggunakan semua data dapat meningkatkan kompleksitas data yang dapat meningkatkan noise serta model sulit mencapai konvergensi dan mengalami overfitting. Sehingga dilakukan

pemilihan data untuk menghilangkan data yang kurang relevan atau dapat menimbulkan noise [17], [18]. Dalam pemilihan data landmark dilakukan dengan memperhatikan sisi tubuh kiri atau kanan yang menghadap ke kamera. Dari 33 *keypoint* hanya dipilih 7 *keypoint*, yaitu telinga, bahu, siku, pergelangan tangan, pinggul lutut dan pergelangan kaki yang dinilai sudah cukup dalam merepresentasikan pose disetiap framenya. 7 data *keypoint* ini masih memiliki noise didalamnya akibat adanya pengaruh posisi peraga dalam frame video sehingga perlu adanya normalisasi untuk menghilangkan posisi keypoiny yang absolut berdasarkan frame kamera [19]. Selain data *keypoint*, dilakukan juga pemilihan feature dengan menghitung 4 sudut sendi, yaitu bahu, siku, pinggul dan lutut dengan menggunakan data 7 *keypoint* yang telah dipilih sebelumnya. Data yang diperoleh pada tahapan ini berbentuk ([jumlah frame], 25) dengan urutan 4 sudut sendi, kemudian 7 *keypoint* dengan terdiri dari nilai x, y dan z.

Bentuk yang masih beragam berdasarkan jumlah frame per videonya, membuat data belum dapat digunakan dalam proses pelatihan model. Kondisi ini dapat diatasi dengan menggunakan metode padding, yaitu metode dengan menambahkan sebuah elemen untuk membuat dimensi atau panjang data menjadi seragam [20]. Khusus pada penelitian ini menggunakan masking padding karena agar tidak mempengaruhi dari data yang ada sehingga tetap mempertahankan bagaimana gerakan push-up yang beragam durasinya sehingga generaliasi dalam hal keberagaman durasi dapat diperoleh oleh model nantinya. Dalam penambahan padding memperhatikan dari data terpanjang dan juga nilai data yang harus tidak ada pada dataset. Sehingga dalam penelitian ini menggunakan nilai padding -1,0 dan panjang data menjadi 60 deret atau frame. Sehingga bentuk data akhir yang digunakan untuk training model adalah (60, 25).

3.3. Training Model

Sebelum melakuakn training model, data yang diperoleh perlu dibagi menjadi data untuk training dan testing dengan rasio perbandingan 80 : 20. Dalam proses training model, dilakukan dengan menggunakan metode K-Fold karena jumlah data yang tidak terlalu besar sehingga dapat meningkatkan efisiensi data dan juga melihat kestabilan model yang dibangun karena semua data akan mendapat giliran digunakan sebagai data training dan validasi. Dengan menggunakan nilai k sebesar 5 untuk tetep mempertahankan rasio perbandingan.

Manfaatkan *framework* TensorFlow untuk membangun model sequential dengan melakukan percobaan sebanyak 8 model dengan susunan layer yang berbeda-beda. 8 percobaan susunan layer dilakukan berdasarkan dari dua penelitian terdahulu yang dilakukan oleh Putra dkk. (2022), serta penelitian El Ghazi & Aknin (2023) [21], [22]. Adapun struktur 8 model yang diuji dalam penelitian ini dapat dilihat pada tabel 1

Tabel 1 List Struktur Model

Model	Layer Structure
Model 1	Input Layer
	Masking Layer
	LSTM (64)
	LSTM (32)
	Densen (64)
	Densen (softmax)
Model 2	Input Layer
	Masking Layer
	LSTM (128)
	Dropout (0.3)
	LSTM (64)
	Dropout (0.3)
Model 3	LSTM (32)
	Dropout (0.3)
	Densen (64)
	Dropout (0.3)
	Densen (softmax)
	Input Layer
Model 3	Masking Layer
	LSTM (128)

Model	Layer Structure
Model 4	Dropout (0.3)
	LSTM (64)
	Dropout (0.3)
	LSTM (32)
	Dropout (0.3)
	Densen (64)
	Densen (softmax)
	Input Layer
	Masking Layer
	LSTM (64)
	Dropout (0.3)
	LSTM (32)
Model 5	Dropout (0.3)
	Densen (64)
	Densen (softmax)
	Input Layer
	Masking Layer
	LSTM (32)
	Dropout (0.3)
	Densen (64)
	Densen (softmax)
	Input Layer
	Masking Layer
	LSTM (64)
Model 6	Dropout (0.3)
	LSTM (32)
	Dropout (0.3)
	Densen (64)
	Dropout (0.3)
	Densen (softmax)
	Input Layer
	Masking Layer
	LSTM (64)
	Dropout (0.3)
	LSTM (32)
	Dropout (0.3)
Model 7	Densen (64)
	Dropout (0.3)
	Densen (softmax)
	Input Layer
	Masking Layer
	LSTM (64)
	Dropout (0.3)
	LSTM (32)
	Densen (64)
	Dropout (0.3)
	Densen (softmax)
	Input Layer
Model 8	Masking Layer
	LSTM (128)
	LSTM (64)
	LSTM (32)
	Densen (64)
	Densen (softmax)
	Input Layer
	Masking Layer
	LSTM (64)
	LSTM (32)
	Densen (64)
	Densen (softmax)

Model dilatih dengan pengaturan *hyperparameter* yang sama. Menggunakan epoch sebanyak 1000, *optimizer* Adam, serta menerapkan mekanisme *Callback Early Stopping* untuk mencegah terjadinya *overfitting* selama training. Mekanisme ini memantau *validation loss* disetiap epochnya. Jika secara berturut-turut tidak terjadi penurunan nilai *validation loss* dalam 40 epoch maka proses pelatihan model akan dihentikan. Nilai 40 dipilih sebagai batas toleransi agar model memiliki kesempatan untuk mencapai konvergensi tanpa mengalami pelatihan yang berlebihan. Dalam training juga menerapkan penyimpanan model pada epoch terbaik dengan nilai *validation loss* terendah.

3.4. Evaluasi Model

Model yang telah dilatih dengan pengaturan *hyperparameter* yang sama diuji performanya dengan menggunakan data tes yang telah disiapkan sebelumnya. Evaluasi model menggunakan

beberapa metrik evaluasi yang umum digunakan dalam model klasifikasi yang dihitung dari matriks kebingungan untuk menganalisis performa model[23]. Akurasi, mengukur kemampuan model dalam membuat prediksi yang benar secara keseluruhan dalam proses klasifikasi. Akurasi dapat dihitung dengan menggunakan persamaan (1).

$$Akurasi = \frac{TP+TN}{TP+FP+TN+FN} \quad \dots \dots \dots (1)$$

Presisi umum digunakan dalam kasus klasifikasi untuk mengukur keakuratan prediksi positif model, yaitu dari semua prediksi positif (TP dan FP) seberapa yang seberapa yang sebenarnya positif. Presisi dapat dihitung dengan menggunakan persamaan (2).

$$Presisi = \frac{TP}{TP+FP} \quad \dots \dots \dots \quad (2)$$

Recall digunakan untuk seberapa baik model dalam melakukan prediksi positif yang sebenarnya dari semua data positif dalam data tes. Recall dihitung dengan menggunakan persamaan (3).

$$Recall = \frac{TP}{TP+FN} \quad \dots \dots \dots \quad (3)$$

Selain itu, digunakan juga *F1-Score* yang mengukur rata-rata dari nilai presisi dan *recall* model. *F1-Score* dihitung dengan menggunakan rumus pada persamaan (4).

$$F1Score = 2 * \frac{Precision * Recall}{Precision + Recall} \quad \dots \dots \dots \quad (4)$$

4. Hasil dan Pembahasan

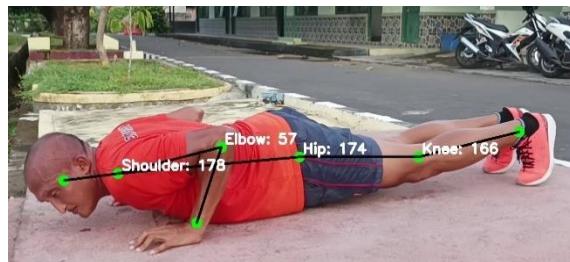
4.1. Hasil

Penelitian ini melakukan pengambilan data video secara langsung dengan menggunakan 6 sudut kamera dari sudut kiri tubuh. Dengan peraga push-up yang terdiri dari anggota TNI dan juga peneliti. Hasil yang diperoleh data video sebanyak 358 video mentahan yang selanjutnya dilakukan proses editing dengan melakukan pemotongan video membuatnya menjadi 1 video hanya mengandung 1 kali repetisi push-up. Proses augmentasi data dilakukan secara manual dalam proses editing video dengan melakukan flip pada semua video untuk mendapatkan data badan sebelah kanan yang menghadap kamera. Jumlah data video yang diperoleh dari tahapan ini adalah sebanyak 3600 video dengan durasi yang beragam. Data terbagi merata sebanyak 360 perkelasnya dengan persebaran 12 sudut kamera sesuai tabel 2. Kelas yang ada adalah benar, salah turun setengah, salah pinggul naik, salah pinggul naik turun setengah, salah pinggul naik naik setengah, salah lutut menekuk, salah lutut menekuk turun setengah, salah lutut menekuk naik setengah, salah hanya badan yang naik, serta salah naik setengah.

Tabel 2 Persebaran Data Setiap Kelas

Sudut Kamera (Nama Video)	Total Video
kiri	30
atas-kiri	30
bawah-kiri	30
depan-kiri	30
atas-depan-kiri	30
bawah-depan-kiri	30
kanan	30
atas-kanan	30
bawah-kanan	30
depan-kanan	30
atas-depan-kanan	30
bawah-depan-kanan	30

Dari data video dilakukan ekstraksi secara frame per frame dengan video yang memiliki kecepatan 30fps. Hasil dari proses ekstrasi fitur diambil 7 keypoint dan 4 sudut yang dapat merepresentasikan pergerakan push-up dilakukan seperti pada gambar 1.



Gambar 1 Visualisasi Hasil Ekstraksi Fitur Pada Gambar

Setelah semua data siap digunakan dilanjutkan dengan proses training dengan hasil disajikan pada tabel 3.

Tabel 3 Nilai Akurasi Validation dan Loss Validation Pelatihan Model

Model	Fold	Akurasi Validation	Validation Loss	Rata-Rata Akurasi Validasi	Rata-Rata Validation Loss
1	1	0,874783	0,405587		
	2	0,845217	0,497659		
	3	0,881739	0,360967	0,880000	0,370442
	4	0,888696	0,331291		
	5	0,909565	0,256707		
2	1	0,886957	0,378673		
	2	0,916522	0,307348		
	3	0,902609	0,241184	0,908870	0,286179
	4	0,921739	0,260761		
	5	0,916522	0,242927		
3	1	0,904348	0,290762		
	2	0,937391	0,202601		
	3	0,925217	0,247739	0,924174	0,233105
	4	0,935652	0,183240		
	5	0,918261	0,241185		
4	1	0,902609	0,311947		
	2	0,935652	0,217394		
	3	0,937391	0,184991	0,926957	0,223948
	4	0,918261	0,234649		
	5	0,940870	0,170758		
5	1	0,813913	0,499335		
	2	0,853913	0,441750		
	3	0,836522	0,501982	0,825739	0,491470
	4	0,782609	0,574217		
	5	0,841739	0,440064		
6	1	0,907826	0,327771		
	2	0,916522	0,354281		
	3	0,904348	0,287581	0,926609	0,252315
	4	0,960000	0,132608		
	5	0,944348	0,159335		
7	1	0,859130	0,430401		
	2	0,885217	0,387560		
	3	0,904348	0,328593	0,888000	0,351620
	4	0,874783	0,379664		
	5	0,916522	0,231884		
8	1	0,890435	0,436441		
	2	0,900870	0,381226		
	3	0,906087	0,304249	0,898783	0,360586
	4	0,885217	0,347918		
	5	0,911304	0,333095		

Dari hasil proses training ini dengan menggunakan metode k-fold menunjukkan model 4 performa yang terbaik jika dinilai dari segi nilai akurasi *validation* dan *validation loss*. Kemudian dilakukan evaluasi lebih lanjut pada model dengan melakukan pengujian dengan data tes untuk melihat performa dari nilai presisi, *recall*, *F1-Score*, dan akurasi model dalam memprediksi data baru yang ditampilkan pada tabel 4 dan 5.

Tabel 4 Nilai Akurasi dan Presisi Model

Model	Fold	Akurasi	Presisi	Rata-Rata Akurasi	Rata-Rata Presisi
1	1	87,20%	87,34%		
	2	86,37%	86,52%		
	3	89,15%	89,17%	88,59%	88,71%
	4	90,96%	91,12%		
	5	89,29%	89,39%		
2	1	90,68%	91,00%		
	2	95,13%	95,27%		
	3	93,18%	93,58%	91,82%	92,10%
	4	90,68%	90,92%		
	5	89,43%	89,74%		
3	1	93,88%	93,51%		
	2	92,46%	93,52%		
	3	91,52%	91,76%	92,24%	92,42%
	4	93,07%	92,85%		
	5	90,26%	90,46%		
4	1	91,79%	92,25%		
	2	92,35%	92,65%		
	3	92,90%	92,97%	92,35%	92,79%
	4	90,68%	91,91%		
	5	94,02%	94,18%		
5	1	82,01%	82,71%		
	2	85,95%	86,82%		
	3	84,56%	84,76%	82,88%	83,46%
	4	79,69%	79,91%		
	5	82,20%	83,10%		
6	1	89,57%	89,69%		
	2	91,52%	91,69%		
	3	88,18%	88,81%	91,77%	91,98%
	4	95,97%	96,00%		
	5	93,60%	93,69%		
7	1	84,56%	84,88%		
	2	90,40%	90,62%		
	3	91,79%	91,89%	89,26%	89,48%
	4	89,57%	89,67%		
	5	89,99%	90,34%		
8	1	88,46%	88,68%		
	2	91,52%	91,61%		
	3	92,07%	92,29%	90,29%	90,45%
	4	89,29%	89,36%		
	5	90,13%	90,29%		

Tabel 5 Nilai *Recall* dan *F1-Score* Model

Model	Fold	<i>Recall</i>	<i>F1-Score</i>	Rata-Rata <i>Recall</i>	Rata-Rata <i>F1-Score</i>
1	1	87,22%	87,23%		
	2	86,38%	86,38%		
	3	89,16%	89,14%	88,60%	88,59%
	4	90,96%	90,95%		
	5	89,30%	89,24%		

Model	Fold	Recall	F1-Score	Rata-Rata Recall	Rata-Rata F1-Score
2	1	90,69%	90,60%	91,82%	91,80%
	2	95,13%	95,15%		
	3	93,18%	93,18%		
	4	90,69%	90,68%		
	5	89,43%	89,38%		
3	1	93,89%	93,89%	92,44%	92,42%
	2	93,47%	93,46%		
	3	91,52%	91,51%		
	4	93,06%	93,07%		
	5	90,26%	90,18%		
4	1	90,58%	90,47%	92,57%	92,53%
	2	91,53%	91,53%		
	3	91,18%	91,09%		
	4	95,96%	95,96%		
	5	93,61%	93,58%		
5	1	82,08%	82,10%	82,90%	82,90%
	2	85,95%	85,96%		
	3	84,56%	84,60%		
	4	79,71%	79,67%		
	5	82,21%	82,17%		
6	1	89,58%	89,47%	91,77%	91,72%
	2	91,53%	91,52%		
	3	88,18%	88,09%		
	4	95,96%	95,96%		
	5	93,61%	93,58%		
7	1	84,57%	84,55%	89,27%	89,26%
	2	90,41%	90,44%		
	3	91,80%	91,76%		
	4	89,58%	89,56%		
	5	89,99%	89,97%		
8	1	88,47%	88,42%	90,30%	90,27%
	2	91,52%	91,52%		
	3	92,07%	92,08%		
	4	89,30%	89,27%		
	5	90,13%	90,06%		

Dari proses training dan evaluasi model, menunjukkan model 4 memiliki performa yang terbaik dibandingkan dengan model dengan struktur yang lain. Sehingga dari pengujian yang dilakukan model dengan struktur 2 *layer* LSTM, 2 *layer* Dropout dan 2 *layer* Dense merupakan arsitektur yang terbaik.

4.2. Pembahasan

Perbandingan hasil training dan evaluasi delapan arsitektur model menunjukkan variasi jumlah lapisan LSTM, Dropout dan kedalaman jaringan berpengaruh pada performa model dalam mengenali pola gerakan push-up. Pada model 3 yang memiliki struktur 3 lapisan LSTM, 3 Dropout, dan 2 Dense memiliki performa yang mendekati model 4. Nilai akurasi dan F1-Score selisih 0,11% yang menunjukkan penambahan lapisan LSTM dan Dropout tidak memberikan manfaat baik pada performa model. Meningkatnya kompleksitas model menyebabkan model menangkap noise pada data sehingga mengurangi stabilitas generalisasi model. Hal ini juga terlihat pada model 2 dengan satu lapisan Dropout lebih banyak dari model 3. Performa model menurun dan lebih rentang mengalami overfitting karena terlalu banyak neuron yang dinoaktifkan. Sedangkan model 5 dengan satu lapisan LSTM dan Dropout serta 2 lapisan Dense menunjukkan performa yang paling rendah dibandingkan yang lain. Hal ini menunjukkan kedalaman jaringan yang terlalu dangkal membuat model tidak mampu menangkap perbedaan antar kelas yang ada. Terdapat model 1 dan 8 yang tidak menggunakan lapisan Dropout, menunjukkan performa tidak seoptimal model yang menggunakan Dropout pada jumlah lapisan

LSTM yang sama. Sehingga penggunaan Dropout dapat membantu model menghindari overfitting dan mendukung pembelajaran pola temporal yang umum. Dengan fungsi Dropout menonaktifkan secara acak unit neuron dan koneksinya pada sebuah jaringan saraf tiruan [24].

Temuan penelitian ini memberikan kontribusi terhadap penelitian berbasis computer vision. Pada penelitian yang membahas perhitungan push-up cenderung menggunakan pendekatan statis seperti pada penelitian Arlin dan Munir (2021) yang hanya mendeteksi posisi atas dan bawah dengan klasifikasi gambar [9]. Sedangkan penelitian Park dkk. (2020) menggunakan sudut tubuh dengan threshold untuk menentukan posisi atas dan bawah push-up [10]. Kedua penelitian tersebut mengabaikan gerakan transisi antar posisi selama push-up, sehingga menyisakan celah pada gerakan salah yang terhitung benar. Pada penelitian memperkuat sekaligus memperluas temuan dengan membuktikan bahwa penilaian kualitas push-up perlu analisis temporal tidak hanya analisis pose statis. Model LSTM mampu untuk memahami pola gerakan dengan baik dan membedakan sepuluh kelas push-up. Dengan sembilan kategori kesalahan gerakan yang tidak dapat di tangkap dengan pendekatan statis.

Selain itu, penelitian ini juga berkontribusi pada penelitian yang memanfaatkan kombinasi MediaPipe dan LSTM pada deteksi aktivitas manusia. Penelitian oleh Prayoga dkk. (2025) menggunakan kombinasi MediaPipe dan LSTM untuk mengklasifikasikan kata bahasa isyarat [14]. Sedangkan penelitian Tanugraha dkk. (2022) menggunakananya untuk mengklasifikasi gerakan yoga [15]. Dari kedua penelitian tersebut menunjukkan LSTM dapat bekerja dengan baik dalam mengenali gerakan yang terjadi. Penelitian ini mengintegrasikan temuan tersebut pada domain baru, yaitu penilaian kualitas gerakan fisik. Hal ini membuktikan LSTM efektif untuk mengenali gerakan pendek namun kompleks, serta variatif. Sekaligus menunjukkan kombinasi MediaPipe dan LSTM dapat menggantikan pendekatan menggunakan sensor dengan tingkat akurasi yang baik. Penelitian ini berfokus pada pengembangan model dengan melakukan percobaan pada delapan arsitektur yang berbeda dengan hasil yang sudah cukup baik. Namun masih adanya peluang untuk meningkatkan performa model dengan melakukan pengujian perubahan *hyperparameter*. Selain itu, implementasi model pada dunianya nyata masih belum dilakukan pengujianya untuk mengetahui apakah model dapat dengan baik digunakan secara real time dalam menilai dan menghitung repetisi push-up.

5. Simpulan

Penelitian ini berhasil dalam mengembangkan model LSTM untuk medeteksi dan mengklasifikasi gerakan push-up berdasarkan keseluruhan gerakan dengan memanfaatkan ekstraksi landmark dengan MediaPipe Pose. Menggunakan metode K-Fold Cross Validation dalam pelatihan model dengan percobaan 8 arsitektur yang berbeda. Hasil pengujian menunjukkan model 4 dengan arsitektur dua lapisan LSTM (64 dan 32 unit), dua lapisan Dropout dan dua lapisan Dense memberikan performa terbaik dengan rata-rata akurasi validasi 92,69% dan *loss* validasi diangka 0,22. Rata-rata akurasi pengujian 92,35% dan F1-Score 92,53%. Penggunaan LSTM yang dikombinasikan MediaPipe Pose secara efektif mengenali variasi pola gerakan push-up pada 10 kelas yang ada. Kombinasi ini dapat menjawab keterbatasan penelitian-penelitian sebelumnya yang mengabaikan transisi antar posisi atas dan bawah selama gerakan push-up. Berdasarkan hasil tersebut, model tidak hanya dapat digunakan untuk repetisi push-up namun juga menilai kualitas gerakan push-up berdasarkan pola gerakan yang dilakukan.

Pada penelitian selanjutnya disarankan untuk melakukan pengujian pada penerapan penggunaan *hyperparameter* yang berbeda untuk meningkatkan performa model. Selain itu dapat juga melakukan percobaan menggunakan varian LSTM lain seperti BiLSTM atau GRU. Pada penelitian berikutnya dapat juga melakukan integrasi model pada aplikasi real-time dan juga melakukan pengujian dengan metode multi-subject cross-validation untuk memastikan model dapat mengenali gerakan push-up dengan variasi tubuh dan gaya gerakan.

Daftar Referensi

- [1] S. F. M. Chastin *et al.*, "Effects of Regular Physical Activity on the Immune System, Vaccination and Risk of Community-Acquired Infectious Disease in the General Population: Systematic Review and Meta-Analysis," *Sports Medicine*, vol. 51, no. 8, pp. 1673–1686, 2021, doi: 10.1007/s40279-021-01466-1.
- [2] F. M. Lulie, "Effects Of Calisthenics Exercise On Selected Health Related Physical Fitnes Qualities In The Case Of Tagel Secondary And Preparatory School Male Student", Tesis, Department of Sport Science, Bahir Dar University, Ethiopia, 2024.

- [3] C. H. J. K. Otarsky, B. R. K. C. Hristensen, and J. A. S. M. Iller, "Effect Of Progressive Calisthenic Push-Up Training On Muscle Strength And Thickness," *J Strength Cond Res*, vol. 32, no. 3, pp. 651–659, 2018, doi: 10.1519/JSC.0000000000002345.
- [4] I. W. Artanayasa, K. C. A. Kusuma, and K. U. Ariawan, "Need analysis of digital technology-based push up test: Instrument of arm muscle strength," *Journal Sport Area*, vol. 7, no. 3, pp. 361–368, Dec. 2022, doi: 10.25299/sportarea.2022.vol7(3).10641.
- [5] I. W. Artanayasa, K. C. A. Kusuma, and K. U. Ariawan, "Push-Up Counter (PUC) as an instrument of arm muscle strength: Validity and reliability," *Journal Sport Area*, vol. 8, no. 3, pp. 350–359, Sep. 2023, doi: 10.25299/sportarea.2023.vol8(3).13245.
- [6] R. Mardela, R. Irawan, A. Agung, N. L. Putra, Y. Marlina, and R. Ependi, "Development Of A Digital-Based Push Up And Sit Up Test Counter," *Jurnal Laman Olahraga Nusantara*, pp. 287–298, 2023, doi: 10.31851/hon.v6i1.10723.
- [7] J. Zhang *et al.*, "MobiPose: Real-time multi-person pose estimation on mobile devices," in *Sensys 2020 - Proceedings of the 2020 18th ACM Conference on Embedded Networked Sensor Systems*, Association for Computing Machinery, Inc, Nov. 2020, pp. 136–149. doi: 10.1145/3384419.3430726.
- [8] S. P. Raj, S. V Raam, S. B. Gopi, and D. Kumar, "Real-Time Body Pose Estimation Using OpenCV And Mediapipe," *International Journal of Creative Research Thoughts*, vol. 12, pp. 2320–2882, 2024, doi: <https://ijcrt.org/papers/IJCRT24A5046.pdf>.
- [9] R. Arlin and R. Munir, "The Development of Push Up Counter Android Application with Computer Vision," Bandung, 2021.
- [10] H.-J. Park, J.-W. Baek, and J.-H. Kim, *Imagery based Parametric Classification of Correct and Incorrect Motion for Push-up Counter Using OpenPose*. 2020. doi: 10.0/Linux-x86_64.
- [11] A. Y. Setiawan, I. G. M. Darmawiguna, and G. Pradnyana, "Sentiment Summarization Learning Evaluation Using LSTM (Long Short Term Memory) Algorithm," *European Alliance for Innovation (EAI)*, Singaraja, 1 – 17, 27 November, 2021, doi: 10.4108/eai.27-11-2021.2315533.
- [12] B. A. Wisesa, W. Andriyani, and B. Purnomosidi, "Usage of LSTM Method On Hand Gesture Recognition For Easy Learning of Sign Language Based On Desktop Via Webcam," *International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, pp. 148–153, Mar. 2023, doi: 10.1109/ISRITI56927.2022.10053076.
- [13] G. B. Pratama Putra, P. H. Suputra, N. W. Marti, and K. F. Sugiantari, "Deteksi Jatuh Lansia Berbasis Landmark Sendi Pada Model LSTM," *Jurnal Informatika Polinema*, vol. 11, no. 3, pp. 261–266, 2025, doi: 10.33795/jip.v11i3.6823.
- [14] I. M. P. Prayoga, I. G. Ayu, A. Diatri, P. H. Suputra, and P. August, "Pengenalan Kata Kolok Secara Real Time Menggunakan Mediapipe dan Algoritma Long Short-Term Memory (LSTM)," *Science and Information Technology Journal*, vol. 8, no. 2, pp. 178–187, 2025, doi: <https://doi.org/10.31598/sintechjournal.v8i2.1945>.
- [15] F. D. Tanugraha, H. Pratikno, M. Musayanah, and W. I. Kusumawati, "Pengenalan Gerakan Olahraga Berbasis (Long Short- Term Memory) Menggunakan Mediapipe," *Journal of Advances in Information and Industrial Technology*, vol. 4, no. 1, pp. 37–45, May 2022, doi: 10.52435/jait.v4i1.182.
- [16] N. P. N. P. Dewi, N. K. Kertiasih, and N. L. D. Sintiari, "Modifikasi Fruit Fly Optimiziation Algorithm untuk Optimasi General Regression Neural Network pada Kasus Prediksi Time-Series," *Jurnal Nasional Pendidikan Teknik Informatika (JANAPATI)*, vol. 11, no. 3, pp. 192–204, 2022, doi: 10.23887/janapati.v11i3.54521.
- [17] N. K. T. A. Saputri, I. G. A. Gunadi, and I. M. G. Sunarya, "Sentiment Analysis of Online Services at the Engineering and Vocational Faculty of Ganesha Education University Using Naïve Bayes and LSTM Algorithms," *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, vol. 4, no. 3, pp. 1120–1129, 2024, doi: <https://doi.org/10.57152/malcom.v4i3.1336>.
- [18] I. G. A. P. Mahendra, I. M. A. Wirawan, and I. G. A. Gunadi, "Enhancement performance of the Naïve Bayes method using AdaBoost for classification of diabetes mellitus dataset type II," *International Journal of Advances in Applied Sciences*, vol. 13, no. 3, pp. 733–742, 2024, doi: 10.11591/ijaas.v13.i3.pp733-742.
- [19] I. M. A. Wirawan, I. G. M. Darmawiguna, and I. B. N. Pascima, "Improving baseline reduction for emotion recognition based on electroencephalogram signals," *IAES*

- International Journal of Artificial Intelligence*, vol. 13, no. 4, pp. 4263–4272, 2024, doi: 10.11591/ijai.v13.i4.pp4263-4272.
- [20] F. Alrasheedi, X. Zhong, and P.-C. Huang, “Padding Module: Learning the Padding in Deep Neural Networks,” *IEEE Access*, vol. 11, pp. 7348-7357 2023, doi: 10.1109/ACCESS.2023.3238315.
- [21] I. A. Putra, O. D. Nurhayati, and D. Eridani, “Human Action Recognition (HAR) Classification Using MediaPipe and Long Short-Term Memory (LSTM),” *Teknik*, vol. 43, no. 2, pp. 190–201, 2022, doi: 10.14710/teknik.v43i2.46439.
- [22] M. El Ghazi and N. Aknin, “Optimizing Deep LSTM Model through Hyperparameter Tuning for Sensor-Based Human Activity Recognition in Smart Home,” *Informatica (Slovenia)*, vol. 47, no. 10, pp. 109–122, 2023, doi: 10.31449/inf.v47i10.5268.
- [23] I. G. M. W. K. Widiantara, K. Y. E. Aryanto, and I. M. G. Sunarya, “Application of the Learning Vector Quantization Algorithm for Classification of Students with the Potential to Drop Out,” *Brilliance: Research of Artificial Intelligence*, vol. 3, no. 2, pp. 262–269, 2023, doi: 10.47709/brilliance.v3i2.3155.
- [24] K. F. Sugiantari, P. H. Suputra, L. J. E. Dewi, and G. B. P. Putra, “Pendekatan MLP Dalam Klasifikasi Bahasa Isyarat: Analisis Jarak Euclidean Landmark Tangan,” *JIKA (Jurnal Informatika)*, vol. 9, no. 2, p. 209, 2025, doi: 10.31000/jika.v9i2.13368.