

Analisis Performa Metode CNN Dan LSTM Untuk Deteksi Gerakan Angkat Beban

¹Gea Vista Yulia Candra, ²Danar Putra Pamungkas, ³Patmi Kasih

^{1,2,3}Teknik Informatika, Universitas Nusantara PGRI Kediri

E-mail: ¹geavistayuliacandra@gmail.com, ²danar@unpkediri.ac.id, ³fatkasi@gmail.com

Penulis Korespondens : Gea Vista Yulia Candra

Abstrak— Kesalahan dalam teknik angkat beban dapat meningkatkan risiko cedera dan mengurangi efektivitas latihan. Oleh karena itu, diperlukan sistem otomatis yang mampu mengidentifikasi kualitas gerakan secara akurat. Penelitian ini bertujuan untuk membangun dataset terlabeli dua kelas, yaitu "Benar" dan "Salah", serta mengimplementasikan model *deep learning* berbasis *Convolutional Neural Network* (CNN) dan *Long Short-Term Memory* (LSTM) untuk klasifikasi gerakan angkat beban. Data diperoleh dari video latihan, kemudian diekstrak menjadi *frame* dan dilabeli berdasarkan kategori folder. CNN digunakan untuk mengekstraksi fitur spasial dari setiap *frame*, sedangkan LSTM mempelajari pola temporal antar *frame*. Hasil pelatihan menunjukkan bahwa model mampu mencapai akurasi sebesar 80% dalam membedakan gerakan yang dilakukan dengan benar dan salah. Temuan ini menunjukkan bahwa pelabelan dua kelas yang sistematis dan pemanfaatan *arsitektur* CNN-LSTM dapat digunakan secara efektif untuk mendeteksi kualitas gerakan dalam aktivitas angkat beban. Sistem ini berpotensi diterapkan sebagai alat bantu pelatihan yang cerdas dan adaptif di bidang kebugaran.

Kata Kunci— angkat beban, klasifikasi gerakan, ekstraksi *frame*, CNN, LSTM, deteksi gerakan

Abstract— *Errors in weightlifting techniques can increase the risk of injury and reduce the effectiveness of training. Therefore, an automated system is needed that is able to identify movement quality accurately. This study aims to build a two-class labeled dataset, namely "Correct" and "Incorrect", and implement a deep learning model based on Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) for weightlifting movement classification. Data is obtained from training videos, then extracted into frames and labeled based on folder categories. CNN is used to extract spatial features from each frame, while LSTM learns temporal patterns between frames. The training results show that the model is able to achieve an accuracy of 80% in distinguishing between correct and incorrect movements. These findings indicate that systematic two-class labeling and the utilization of CNN-LSTM architecture can be used effectively to detect movement quality in weightlifting activities. This system has the potential to be applied as an intelligent and adaptive training aid in the field of fitness.* a brief conclusion. Do not use Symbols, Special characters, or Maths in the abstract.

Keywords— *weight lifting, motion classification, frame extraction, CNN, LSTM, motion detection*

This is an open access article under the CC BY-SA License.



I. PENDAHULUAN

Memiliki tubuh ideal sangat penting untuk kesehatan tubuh manusia, karena jika terjadi penggemukan badan akan berdampak pada kesehatan manusia yang dapat terjadinya resiko berbagai penyakit. Untuk memiliki tubuh ideal memiliki banyak cara, mulai dari mengatur pola makan, melakukan olahraga seperti zumba, aerobik, senam dan latihan angkat beban (*weight training*) [1]. Latihan angkat beban juga dapat membentuk massa otot seperti otot perut dan

lengan juga dikenal sebagai *biceps* dan *triceps*, ini merupakan pilihan yang umum bagi orang yang ingin membentuk otot untuk menjadi lebih percaya diri [2].

Dalam melakukan latihan angkat beban membutuhkan teknik dan pengetahuan dasar untuk memahami penggunaan alat dan efek yang dihasilkan dari penggunaan alat kebugaran [3]. Personal trainer sangat penting di sini karena dapat membantu anggota pusat kebugaran untuk memaksimalkan jumlah latihan yang mereka lakukan. Akan tetapi, banyak orang tidak menggunakan personal trainer karena berbagai alasan, seperti mahal atau tidak nyaman. Namun, tanpa personal trainer, risiko cedera meningkat [4]. Perkembangan teknologi kecerdasan buatan telah mengubah banyak hal, termasuk olahraga dan kebugaran [5]. Penggunaan teknologi *deep learning* untuk mengidentifikasi dan menganalisis gerakan olahraga, seperti olahraga angkat beban secara otomatis adalah salah satu implementasi dari perkembangan teknologi kecerdasan buatan [6].

Penelitian sebelumnya memanfaatkan sensor akselerometer dan giroskop hingga penerapan algoritma machine learning seperti *K-Nearest Neighbor* (KNN) dan *Support Vector Machine* (SVM) menunjukkan bahwa algoritma KNN dapat memberikan hasil akurasi tinggi dalam mengklasifikasikan enam jenis aktivitas fisik berbasis data sensor inersia [7]. Sementara itu, penelitian lain membuktikan bahwa sistem berbasis Raspberry Pi Zero W dan SVM mampu mendeteksi gerakan *dumbbell* secara efektif dengan akurasi di atas 90% [8].

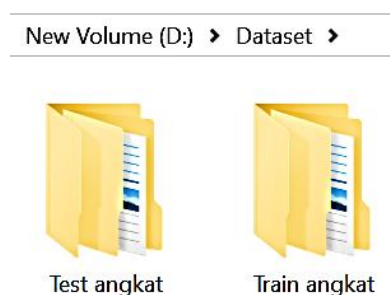
II. METODE

Penelitian ini menggunakan pendekatan kuantitatif dengan mengimplementasikan metode *Convolutional Neural Network* (CNN) dan *Long Short Term-Memory* (LSTM) untuk mengklasifikasikan gerakan angkat beban.

2.1 Analisis Data

Pengumpulan data dikumpulkan dalam bentuk video yang merekam aktivitas angkat beban dari beberapa partisipan yang ada di *Master Gym* yang berlokasi di Kecamatan Kepung, Kabupaten Kediri. Proses perekaman dilakukan dengan variasi intensitas dan jenis beban, seperti *dumbbell* dan *barbell*. Setiap video berdurasi antara 1-20 detik dan direkam dari sudut samping, depan, dan belakang untuk mendapatkan visualisasi gerakan secara menyeluruh. Video disimpan dalam bentuk format MP4 dan MOV dengan resolusi 13MP.

2.2 Pengolahan Data



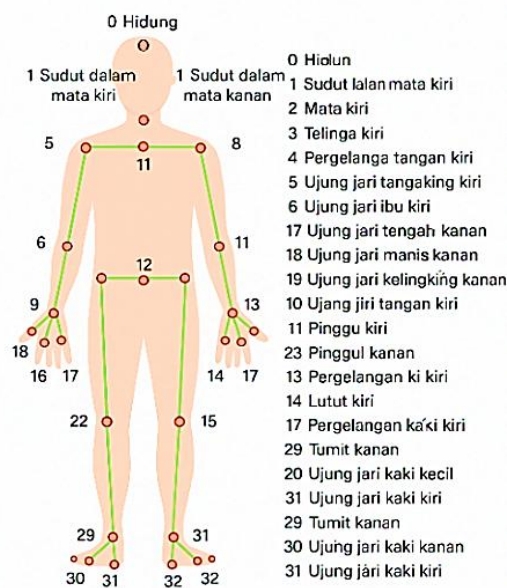
Gambar 1. Pembagian Data

Setelah proses pengambilan data yang dilakukan secara langsung, data dibagi menjadi dua kelompok utama yaitu data latihan (data pelatihan) dan data uji (data pengujian). Tujuan dari pembagian ini adalah untuk memastikan bahwa model yang dikembangkan dapat mempelajari

data yang tersedia melalui data latih dan untuk menguji kemampuan pola dari data uji yang belum pernah dilihat sebelumnya.

2.3 Preprocessing Data

Pada tahap ini, setiap video diubah menjadi serangkaian gambar (*frame*) dengan interval tertentu. Proses ini memanfaatkan *pose landmark* dari *MediaPipe Pose* yang menghasilkan 33 *keypoints* dari titik tubuh manusia per *frame*. Setiap titik memiliki koordinat x, y, z, yang menunjukkan posisi spesial titik tersebut dalam setiap *frame*. Titik-titik ini meliputi bagian tubuh seperti hidung (titik 0), bahu kiri dan kanan (titik 11 dan 12), siku (titik 13 dan 14), pergelangan tangan (titik 15 dan 16), pinggul (titik 23 dan 24). Hasil dari *pose landmark* digambarkan secara *visual* seperti berikut.



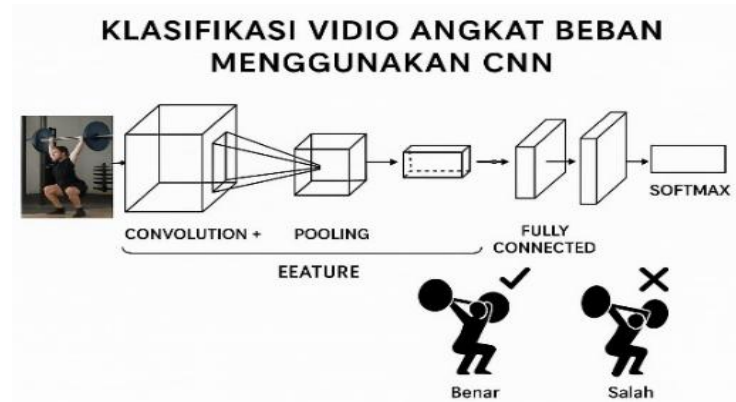
Gambar 2. *Pose Landmark*

2.4 Memuat dan Membagi Data

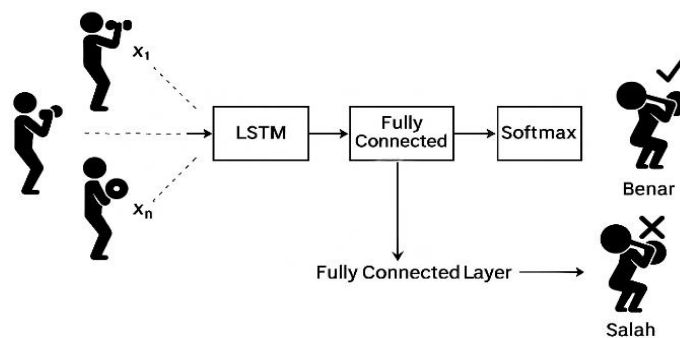
Setelah melakukan *preprocessing* data, data dari fitur yang ada di x.npy dan label yang ada di y.npy dimuat kembali ke dalam memori untuk dilakukan proses pelatihan model. File x.npy berisi data pose hasil ekstraksi dari video, sedangkan y.npy berisi label "Benar" atau "Salah" untuk setiap sampel. Data dibagi menjadi dua bagian, yaitu data latih dan data validasi menggunakan *train_test_split* dari *scikit-learn*. Pembagian data ini dilakukan untuk menguji kinerja model pada data yang tidak dilatih sehingga dapat mengevaluasi kemampuan model.

2.5 Arsitektur Model

Convolutional Neural Network (CNN) merupakan jenis jaringan saraf tiruan yang dibuat khusus untuk mengenali pola visual dalam gambar. CNN melakukan ini dengan menggunakan proses konvolusi dan pooling untuk mengekstrak fitur penting seperti tepi, bentuk, dan struktur objek dari gambar [9]. CNN digunakan untuk mengekstraksi fitur dari koordinat *keypoints pose* tubuh manusia di setiap *frame* video. Ini memungkinkan sistem mengklasifikasikan gerakan menjadi "Benar" atau "Salah" berdasarkan citra tubuh yang diambil kamera.

Gambar 3. *Arsitektur CNN*

LSTM merupakan jenis Recurrent Neural Network (RNN) yang dirancang khusus untuk menangani data dengan dependensi jangka panjang, seperti urutan gerakan dalam video, model ini terdiri dari beberapa layer, termasuk layer convolutional, layer LSTM, layer dense, dan layer output yang diaktifkan sigmoid [10]. LSTM digunakan untuk menangani urutan gerakan antar *frame* serta memproses data sekuensial yang berasal dari video gerakan angkat beban, dengan output akhir yang akan diteruskan ke lapisan *fully connected* dan dilanjutkan ke lapisan *softmax* untuk menghasilkan klasifikasi “Benar” dan “Salah”.

Gambar 4. *Arsitektur LSTM*

2.5 Klasifikasi

Model dilatih dengan data latih menggunakan epoch. Dalam proses ini, model mempelajari pola pose tubuh dan hubungannya dengan label "Benar" atau "Salah". Selama pelatihan, data divalidasi untuk memastikan apakah model masih umum atau mengalami *overfitting*. Untuk keperluan analisis, nilai loss dan akurasi pada training dan validasi dicatat.

2.6 Evaluasi

Peforma model dievaluasi berdasarkan data validasi dalam mengklafisikasikan video angkat beban dalam kategori “Benar” dan “Salah”. Beberapa *confusion matrix* seperti akurasi, *precision*, *f1-score*, dan *recall* yang dapat memberikan gambaran tentang kemampuan model dalam mengenali masing-masing kelas secara akurat. Nilai-nilai yang terdapat pada metrik dihitung menggunakan rumus sebagai berikut:

Akurasi

(1)

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN}$$

Precision (2)

$$Precision = \frac{TP}{TP + FP}$$

Recall (3)

$$Recall = \frac{TP}{TP + FN}$$

F1-score (4)

$$F1 - Score = 2x \frac{Precision \times Recall}{Precision + Recall}$$

III. HASIL DAN PEMBAHASAN

1. Ekstraksi Frame dan Ekstraksi Fitur Pose

Penelitian ini menggunakan ekstraksi frame dan metode CNN yang berperan sebagai ekstraksi fitur spasial dari tiap frame. dan metode LSTM untuk menangkap dinamika temporal, yaitu perubahan pose dari waktu ke waktu sehingga cocok untuk mendeteksi pola gerakan yang terjadi selama durasi video. frame-frame ini dibagi menjadi test dan train dengan label “benar” dan “salah” yang dimana masing-masing memiliki koordinat (x, y, z) dengan jumlah frame yang dihasilkan video menyesuaikan pada durasi dan frame rate (fps) pada video. Maka total frame yang diekstraksi adalah sekitar 150 frame. Pada gambar 3.2 ekstraksi frame menunjukkan pembagian data train dan test setelah melakukan ekstraksi yang menghasilkan file berupa .csv.

```
df = pd.read_csv('/content/drive/MyDrive/skripsi/1111/train_dataset.csv')
df.head()
```

	x0	y0	z0	x1	y1	z1	x2	y2	z2	x3	...	x30	y30	z30	x31	y31	z31	x32	y32	z32	label
0	0.303484	0.622519	-0.554874	0.260489	0.619346	-0.506867	0.259782	0.619785	-0.507024	0.259242	...	1.444013	1.123935	0.791547	1.558589	0.983131	1.193727	1.539672	1.078319	0.515433	salah
1	0.563589	0.104498	-0.085013	0.576954	0.083861	-0.099104	0.579250	0.085166	-0.099262	0.581683	...	0.568528	1.230410	0.281689	0.545234	1.262104	0.122670	0.513304	1.248600	0.245358	benar
2	0.452055	0.413059	-0.483956	0.464189	0.403332	-0.457359	0.475289	0.404605	-0.457317	0.485239	...	0.435104	0.945838	0.480559	0.610473	0.954442	0.494834	0.447701	0.885853	0.317736	benar
3	0.479795	0.069286	-0.908139	0.500041	0.011163	-0.973396	0.509181	0.008717	-0.973364	0.520013	...	0.384895	2.023842	0.747806	0.524817	2.119080	0.471569	0.391605	2.105241	0.407901	benar
4	0.547049	0.222353	0.238843	0.537844	0.211407	0.199486	0.533018	0.210447	0.199475	0.528150	...	0.547907	0.491111	0.130738	0.484066	0.603499	0.296786	0.625008	0.498044	0.217512	salah

5 rows x 100 columns

```
[ ] df = pd.read_csv('/content/drive/MyDrive/skripsi/1111/test_dataset.csv')
df.head()
```

	x0	y0	z0	x1	y1	z1	x2	y2	z2	x3	...	x30	y30	z30	x31	y31	z31	x32	y32	z32	label
0	0.501317	0.392137	-0.339099	0.512582	0.399288	-0.393409	0.520046	0.388597	-0.393607	0.527808	...	0.718342	1.067429	0.455592	0.610052	1.078987	0.105521	0.629146	1.069510	0.374784	benar
1	0.556517	0.805314	-0.374559	0.563587	0.810484	-0.400334	0.569071	0.807201	-0.400413	0.574023	...	0.526446	0.500750	0.674931	0.557452	0.564458	0.730197	0.529894	0.577234	0.715648	salah
2	0.625656	0.429284	-0.670460	0.635229	0.416078	-0.668754	0.642827	0.414457	-0.668829	0.650819	...	0.614307	0.778296	0.744434	0.755497	0.831105	0.391043	0.555389	0.810650	0.584504	salah
3	0.370905	-0.005488	-0.019217	0.366290	-0.033123	-0.012943	0.365383	-0.032107	-0.013111	0.364808	...	0.694035	0.597337	-0.332960	0.734590	0.417816	-0.067899	0.753182	0.424669	-0.457321	salah
4	0.310463	0.306234	-0.278089	0.348063	0.283079	-0.360476	0.356846	0.283246	-0.360821	0.365349	...	0.381636	1.387410	0.913199	0.211745	1.420365	0.323514	0.246501	1.395566	0.691416	salah

5 rows x 100 columns

Gambar 5. Ekstraksi frame

Selanjutnya adalah melakukan ekstraksi fitur *pose* menggunakan *framework MediaPipe Pose*. *MediaPipe Pose* mampu mendeteksi 33 *keypoint* tubuh manusia pada setiap *frame* gambar, termasuk titik-titik seperti kepala, bahu, siku, pergelangan tangan, pinggul, lutut, dan pergelangan kaki. Tiga jenis data dihasilkan oleh setiap *keypoint* dengan koordinat x, y, z yang menunjukkan posisi relatif titik tersebut terhadap gambar. Setiap *frame* menghasilkan array berukuran 33 dan 3, yang kemudian diratakan menjadi vektor berukuran 99. Hasil ekstraksi digambarkan secara visual seperti berikut:



Gambar 6. Ekstraksi Fitur Pose

3. Arsitektur Model

Pada tabel 1 merupakan model yang terdiri dari beberapa layer, dimulai dari Conv1D yang menghasilkan output berukuran (28, 64) dan mengekstraksi fitur dari data urutan. Setelah itu, MaxPooling1D mengurangi dimensi menjadi (14, 64), diikuti oleh Dropout untuk mencegah overfitting. Kemudian, LSTM menangkap pola berurutan dan menghasilkan output (64). Selanjutnya, Dense layer mereduksi ke 32 neuron dan terakhir Dense output layer menghasilkan 1 output untuk klasifikasi biner (benar atau salah). Total parameter yang dilatih adalah sekitar 54 ribu. Selanjutnya proses pelatihan model, khususnya tentang penggunaan epoch sebagai salah satu parameter penting dalam performa CNN+LSTM.

Table 1. Arsitektur Model

No	Arsitektur Model		
	Layer	Output Shape	Jumlah Parameter
1	conv1d_1 (Conv1D)	(None, 28, 64)	19,072
2	max_pooling1d_1 (MaxPooling1D)	(None, 14, 64)	0
3	dropout_2 (Dropout)	(None, 14, 64)	0
4	lstm_1 (LSTM)	(None, 64)	33,024
5	dropout_3 (Dropout)	(None, 64)	0
6	dense_2 (Dense)	(None, 32)	2,080
7	dense_3 (Dense)	(None, 1)	33
Total Params			54,209

Berdasarkan hasil pelatihan model yang telah dilakukan, diketahui bahwa akurasi pelatihan (training accuracy) meningkat secara bertahap dari nilai awal 0.54 hingga mencapai sekitar 0.75, dan akurasi validasi mencapai sekitar 0.78 pada epoch ke-16. Hal ini menunjukkan bahwa model mampu mempelajari pola dari data, namun peningkatan performa cenderung stag setelah beberapa epoch. Selain itu, selisih antara akurasi pelatihan dan validasi relatif kecil, yang mengindikasikan bahwa model belum mengalami overfitting secara signifikan. Model yang dilatih menunjukkan akurasi 80%, yang belum mencapai target >90%.

Hal ini kemungkinan disebabkan oleh kualitas data pose yang kurang bersih, arsitektur model yang kurang kompleks, atau data yang tidak seimbang. Untuk meningkatkan akurasi, perlu dilakukan pembersihan dan augmentasi data, penyesuaian arsitektur model (seperti menambahkan LSTM atau Bidirectional LSTM), serta penggunaan teknik regularisasi dan callback. Dengan optimasi tersebut, akurasi di atas 90% lebih memungkinkan untuk dicapai.

Epoch	Step	Time	Accuracy	Loss	Val Accuracy	Val Loss
Epoch 1/100	1378/1378	30s	0.5495	0.6827	0.6484	0.6217
Epoch 2/100	1378/1378	40s	0.6549	0.6215	0.6968	0.5726
Epoch 3/100	1378/1378	41s	0.6802	0.5940	0.7194	0.5531
Epoch 4/100	1378/1378	40s	0.6972	0.5755	0.7304	0.5400
Epoch 5/100	1378/1378	43s	0.7062	0.5673	0.7155	0.5504
Epoch 6/100	1378/1378	38s	0.7115	0.5587	0.7381	0.5205
Epoch 7/100	1378/1378	44s	0.7196	0.5522	0.7480	0.5181
Epoch 8/100	1378/1378	30s	0.7220	0.5468	0.7511	0.5201
Epoch 9/100	1378/1378	25s	0.7294	0.5397	0.7576	0.4970
Epoch 10/100	1378/1378	41s	0.7287	0.5433	0.7575	0.5027
Epoch 11/100	1378/1378	43s	0.7355	0.5327	0.7599	0.4909
Epoch 12/100	1378/1378	38s	0.7362	0.5290	0.7623	0.4913
Epoch 13/100	1378/1378	41s	0.7378	0.5267	0.7684	0.4894
Epoch 14/100	1378/1378	42s	0.7437	0.5242	0.7675	0.4901
Epoch 15/100	1378/1378	44s	0.7431	0.5223	0.7673	0.4824
Epoch 16/100						

Gambar 7. Training Model

4. Klasifikasi

Hasil klasifikasi menggunakan model CNN+LSTM menunjukkan bahwa model berhasil membedakan dua kelas pada data latihan angkat beban dengan akurasi yang tinggi. Proses pelatihan data memperlihatkan kinerja yang baik, ditandai dengan peningkatan akurasi pelatihan serta validasi, dan penurunan nilai loss seiring bertambahnya epoch. Hal ini menandakan bahwa model mampu belajar dengan efektif dari data yang digunakan. Ringkasan hasil training selama 100 epoch disajikan pada tabel berikut.

Table 2. Ringkasan Hasil Training

Epoch	Ringkasan Hasil Training			
	Akurasi Pelatihan	Loss Pelatihan	Akurasi Validasi	Loss Validasi
11	0.7355	0.5327	0.7599	0.4909
83	0.7857	0.4646	0.7933	0.4430
94	0.7924	0.4507	0.7976	0.4348
100	0.8058	0.4547	0.7977	0.4354

5. Hasil Pelatihan Model dan Evaluasi

Pada Penelitian ini, dilakukan dengan memproses data menjadi bentuk urutan (sequence) yang sesuai untuk pelatihan dan pengujian model CNN-LSTM. Masing-masing data sequence terdiri dari 30 timestep, yang merepresentasikan 30 titik waktu berurutan dari gerakan yang terekstraksi. Jumlah 55096 pada data train dan 13752 pada data test menunjukkan banyaknya sequence yang digunakan dalam proses training dan evaluasi. Sedangkan label (55096,) dan (13752,) menunjukkan bahwa setiap sequence memiliki satu label, yaitu 0 untuk gerakan benar dan 1 untuk gerakan salah. Bentuk data seperti ini memungkinkan model untuk mempelajari

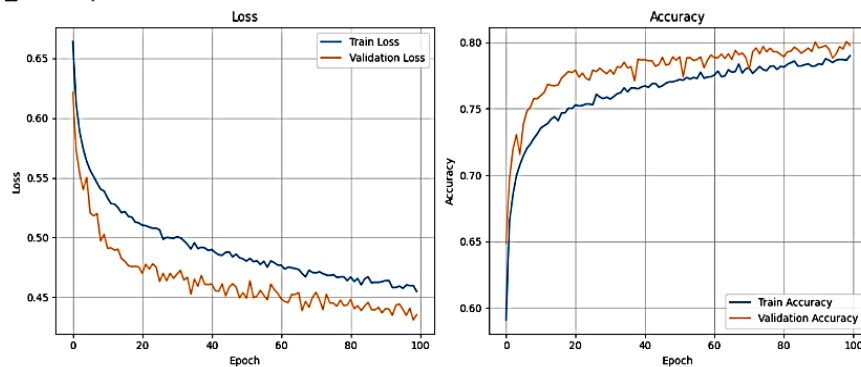
pola gerakan dan untuk mendeteksi apakah suatu gerakan angkat beban dilakukan dengan teknik yang tepat atau tidak.

```
print("Test:", X_test.shape, y_test.shape)
```

Train: (55096, 30, 99) (55096,)
Test: (13752, 30, 99) (13752,)

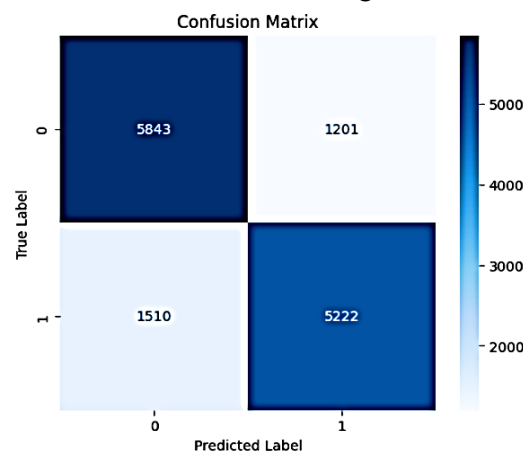
Gambar 8. Data *Train* dan *Test*

Selanjutnya, hasil dari proses pelatihan dan evaluasi ini ditampilkan dalam grafik loss dan akurasi. Validation loss cenderung lebih rendah dibandingkan train loss, yang mengindikasikan bahwa model tidak mengalami overfitting dan mampu melakukan generalisasi dengan baik. Ini menunjukkan bahwa model dapat mempelajari pola data dengan stabil selama proses pelatihan. Baik akurasi pelatihan maupun validasi meningkat secara bertahap pada grafik akurasi. Akurasi validasi hampir 90%, lebih tinggi dari akurasi pelatihan, yang menunjukkan bahwa model CNN+LSTM dapat mengklasifikasikan gerakan dengan cukup akurat. Selain itu, akurasi pengujian 80,32% menunjukkan bahwa model bekerja dengan baik pada data yang belum pernah dilihat.



Gambar 9. *Loss* dan Akurasi

Setelah model dievaluasi berdasarkan akurasi dan loss, selanjutnya adalah hasil dari confusion matrix yang menunjukkan bahwa model berhasil mengklasifikasikan data dengan cukup baik.



Gambar 10. *Confusion Matrix*

Dari confusion matrix yang ditampilkan, dapat dihitung berbagai metrik evaluasi berdasarkan kelas Benar(0) dan kelas Salah(1) untuk mengevaluasi kinerja model klasifikasi. Berikut adalah rincian metrik evaluasi yang diperoleh:

Akurasi (1)

$$Akurasi = \frac{5843 + 5222}{5843 + 1201 + 1510 + 5222} = \frac{11065}{13776} = 80.25\%$$

Precision (2)

$$Precision^0 = \frac{5843}{5843 + 1510} = \frac{5843}{7353} = 0.7949$$

$$Precision^1 = \frac{5222}{5222 + 1201} = \frac{5222}{6423} = 0.8132$$

Recall (3)

$$Recall^0 = \frac{5843}{5843 + 1201} = \frac{5843}{7044} = 0.8294$$

$$Recall^1 = \frac{5222}{5222 + 1510} = \frac{5222}{6732} = 0.7755$$

F1-score (4)

$$F1 - Score_0 = 2x \frac{0.7949 \cdot 0.8294}{0.7949 + 0.8294} = 0.8118$$

$$F1 - Score^0 = 2x \frac{0.8132 \cdot 0.7755}{0.8132 + 0.7755} = 0.7939$$

Secara keseluruhan, model ini menunjukkan performa yang cukup baik, dengan akurasi total 81.93% dan keseimbangan yang baik antara presisi dan recall di kedua kelas. Meskipun recall untuk kelas 1 sedikit lebih rendah, F1-Score menunjukkan bahwa model ini efektif dalam kedua kelas tersebut.

IV. KESIMPULAN

Penelitian ini membahas analisis performa untuk mendeteksi gerakan angkat beban menggunakan arsitektur *Convolutional Neural Network* (CNN) dan *Long Short Term Memory* (LSTM). Penerapan kombinasi arsitektur *Convolutional Neural Network* (CNN) dan *Long Short-Term Memory* (LSTM) dalam sistem deteksi gerakan angkat beban berhasil mengklasifikasikan gerakan dengan akurasi mencapai sekitar 80%. Meskipun belum mencapai target akurasi di atas 90%, hasil ini tetap menunjukkan bahwa pendekatan deep learning memiliki potensi kuat dalam mengenali pola gerakan tubuh dari data pose yang diekstrak. Model mampu membedakan antara gerakan yang dilakukan dengan benar dan salah sehingga dapat digunakan sebagai dasar dalam sistem pembinaan olahraga atau aplikasi pelatihan otomatis. Untuk meningkatkan performa, disarankan untuk memperkaya jumlah dan kualitas data pelatihan, menyesuaikan parameter model secara lebih optimal, serta mempertimbangkan teknik augmentasi data dan regularisasi yang lebih lanjut. Dengan pengembangan lanjutan, sistem ini diharapkan dapat memberikan

kontribusi nyata dalam membantu evaluasi teknik angkat beban secara lebih akurat, efisien, dan otomatis.

DAFTAR PUSTAKA

- [1] D. N. Afgenesya, I. Hariadi, and N. R. Fadhli, "Pengaruh Latihan Beban Terhadap Penurunan Berat Badan Dan Persentase Lemak," *J. Master Penjas Olahraga*, vol. 5, no. 1, pp. 385–394, 2024, doi: [10.37742/jmpo.v5i1.100](https://doi.org/10.37742/jmpo.v5i1.100).
- [2] M. Al Hafiz Sy and Edwarsyah, "Pengaruh Latihan Barbell Curl dan Dumbbell Crul terhadap Pembentukan Otot Lengan Bagian Depan (Biceps)," *J. Pendidik. dan Olahraga*, vol. 2, no. 3, pp. 9–11, 2019 <http://jpdo.ppj.unp.ac.id/index.php/jpdo/search>.
- [3] M. F. Akbar, D. Syauqy, and R. R. M. Putri, "Sistem Bantu Wearable Pada Training Angkat Beban Untuk Biceps Dengan Algoritma Jaringan Saraf Tiruan," *J. Pengemb. Teknol. Inf. dan Ilmu Komputer*, vol. 1, no. 1, 2024, [Online]. Available: <https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/13619/6055>
- [4] M. M. Gharasuie, N. Jennings, and S. Jain, "Performance Monitoring for Exercise Movements using Mobile Cameras," *BodySys 2021 - Proc. 2021 ACM Work. Body Centric Comput. Syst.*, pp. 1–6, 2021, doi: [10.1145/3469260.3469665](https://doi.org/10.1145/3469260.3469665).
- [5] J. C. Núñez, R. Cabido, J. J. Pantrigo, A. S. Montemayor, and J. F. Vélez, "Pengenalan Pola Jaringan Syaraf Konvolusional dan Memori Jangka Pendek Panjang," vol. 76, pp. 2017–2019, 2018 <https://doi.org/10.1016/j.patcog.2017.10.033>.
- [6] V. M. Putra Salawazo, D. P. Jaya Gea, R. F. Gea, and F. Azmi, "Implementasi Metode Convolutional Neural Network (CNN) Pada Pengenalan Objek Video CCTV," *Indones. Jalan Sekip Sikambing*, vol. 3, no. 1, pp. 74–79, 2019.
- [7] S. Mohsen, A. Elkaseer, and S. G. Scholz, *Human Activity Recognition Using K-Nearest Neighbor Machine Learning Algorithm*, vol. 262 SIST, no. November. Springer Singapore, 2022. doi: [10.1007/978-981-16-6128-0_29](https://doi.org/10.1007/978-981-16-6128-0_29).
- [8] E. M. SARTIKA, A. DARMAWAN, W. E. JAYA, and E. Wianto, "Deteksi Gerakan Tangan menggunakan Support Vector Machine pada Dumbbell Berbasis Raspberry Pi Zero," *ELKOMIKA J. Tek. Energi Elektr. Tek. Telekomun. Tek. Elektron.*, vol. 10, no. 1, p. 105, 2022, doi: [10.26760/elkomika.v10i1.105](https://doi.org/10.26760/elkomika.v10i1.105).
- [9] K. Y. Chen, J. Shin, M. A. M. Hasan, J. J. Liaw, O. Yuichi, and Y. Tomioka, "Fitness Movement Types and Completeness Detection Using a Transfer-Learning-Based Deep Neural Network," *Sensors*, vol. 22, no. 15, 2022, doi: [10.3390/s22155700](https://doi.org/10.3390/s22155700).
- [10] R. Dwi, W. Santosa, M. Arif Bijaksana, and A. Romadhony, "Implementasi Algoritma Long Short-Term Memory (LSTM) untuk Mendeteksi Penggunaan Kalimat Abusive Pada Teks Bahasa Indonesia," *J. Tugas Akhir Fak. Inform.*, vol. 12, no. 2020, p. 621, 2020, doi: <https://openlibrarypublications.telkomuniversity.ac.id/index.php/engineering/article/view/14318>.