BAB III
METODE PENELITIAN

3.1 Objek Penelitian

Objek penelitian ini adalah sistem klasifikasi pose yoga berbasis visi komputer yang mengintegrasikan model estimasi pose MoveNet dengan model Convolutional Neural Network (CNN). Sistem ini dirancang untuk mengenali delapan kategori pose yoga Warrior, Tree, Triangle, Dog, Cobra, Chair, Shoulder Stand, dan No Pose berdasarkan koordinat *keypoint* tubuh yang diekstraksi dari citra.

Dataset yang digunakan dalam penelitian ini adalah Yoga Poses Dataset yang dipublikasikan oleh Aman Upadhyay melalui platform Kaggle. Dataset ini terdiri dari 3.149 citra yang terdistribusi ke dalam dua direktori utama, yaitu *train* dan *test*. Setiap kelas pose memiliki keragaman dalam bentuk variasi pose, perbedaan latar belakang, sudut pengambilan gambar, serta kondisi pencahayaan. Keragaman ini memungkinkan evaluasi kemampuan generalisasi model dalam menghadapi kondisi lingkungan yang bervariasi.

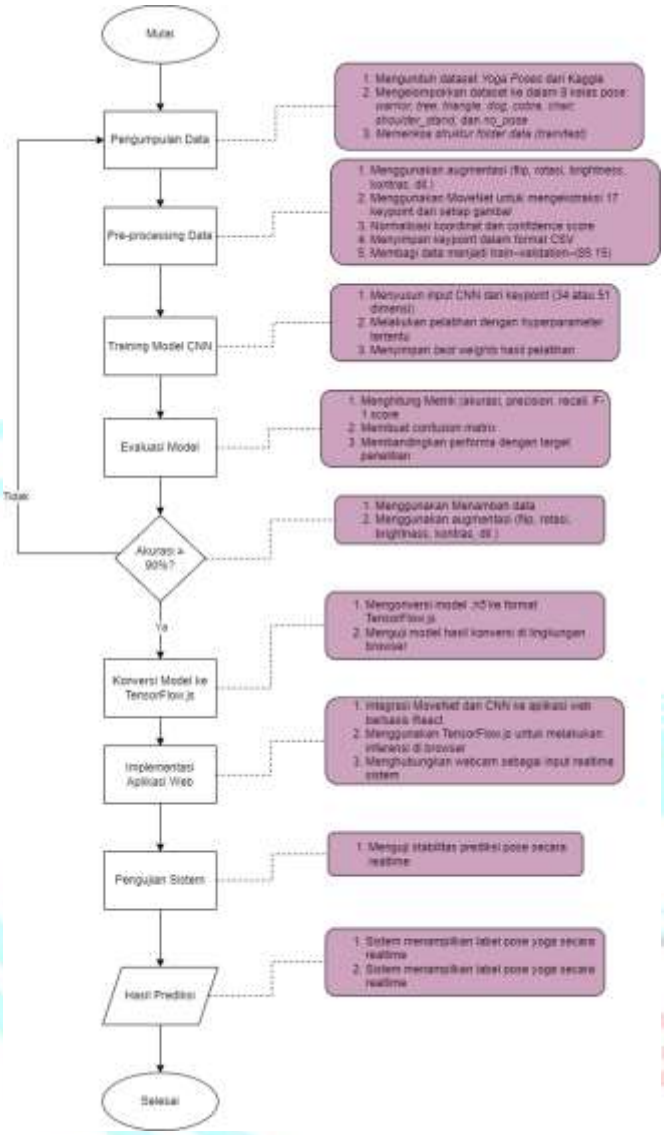
Seluruh citra dalam dataset diproses menggunakan model MoveNet untuk mengekstraksi 17 titik keypoint tubuh sebagai representasi pose. Koordinat keypoint tersebut kemudian digunakan sebagai fitur input dalam proses pelatihan model CNN, sehingga sistem mampu melakukan klasifikasi pose yoga secara efektif.

3.2 Prosedur Penelitian

Prosedur penelitian ini terdiri dari beberapa tahapan yang disusun secara sistematis dan saling berkaitan. Tahap awal dimulai dengan pengumpulan dataset *Yoga Poses* yang dipublikasikan oleh Aman Upadhyay melalui platform Kaggle, yang kemudian dianalisis untuk memastikan kualitas dan representativitas setiap kelas pose yoga. Setelah data terkumpul, dilakukan proses *pre-processing* berupa ekstraksi titik-titik kunci tubuh (*keypoints*) menggunakan model estimasi pose MoveNet. Tahap ini bertujuan menghasilkan data koordinat yang konsisten sebelum digunakan dalam proses pelatihan model.

Pada tahap berikutnya, model Convolutional Neural Network (CNN) dibangun dan dilatih menggunakan data koordinat hasil ekstraksi keypoint. Proses pelatihan dilakukan secara bertahap disertai pemantauan metrik akurasi, *loss*, precision, recall, dan F1-score untuk memastikan model memenuhi kriteria performa yang diharapkan. Jika model belum mencapai performa optimal, dilakukan penyesuaian parameter atau pelatihan ulang hingga hasil yang diperoleh stabil.

Setelah model mencapai performa terbaik, langkah selanjutnya adalah melakukan konversi model ke TensorFlow.js agar dapat diimplementasikan dalam aplikasi berbasis web. Model yang telah dikonversi kemudian diintegrasikan ke dalam sistem web menggunakan React dan TensorFlow.js sehingga dapat bekerja secara realtime dalam mengklasifikasikan pose yoga berdasarkan input dari kamera perangkat. Tahap terakhir adalah pengujian sistem secara langsung untuk memastikan kemampuan model dalam mendeteksi pose dengan akurat pada berbagai kondisi pencahayaan, latar belakang, maupun variasi posisi tubuh. Alur lengkap prosedur penelitian ditunjukkan pada Gambar 3.1.



Gambar 3. 1 Flowchart Prosedur Penelitian

Sumber: (Dokumen Pribadi,2026)

Gambar 3.1 menjelaskan rangkaian proses penelitian mulai dari pengumpulan data, pra-pemrosesan keypoint, pelatihan model CNN, konversi model ke TensorFlow.js, hingga implementasi dan pengujian aplikasi berbasis web. Setiap tahap saling terhubung dan membentuk alur kerja yang sistematis untuk memastikan model mampu beroperasi secara efektif dalam skenario penggunaan nyata.

1.2.1 Tahap Pengumpulan Data

Tahap pengumpulan data merupakan fondasi utama dalam penelitian ini karena kualitas model klasifikasi sangat dipengaruhi oleh kelengkapan, keragaman,

dan representativitas data citra yang digunakan. Pada penelitian ini, sumber data berasal dari dataset Yoga Poses yang dipublikasikan oleh Aman Upadhyay melalui platform Kaggle. Dataset ini berisi kumpulan citra pose yoga yang diperoleh dari berbagai individu, lingkungan, sudut pandang kamera, serta kondisi pencahayaan yang berbeda. Variasi tersebut memberikan keragaman visual yang memadai sehingga model dapat belajar mengenali pose yoga secara lebih robust.

Dataset Yoga Poses terdiri dari delapan kelas pose, yaitu *warrior, tree, triangle, dog, cobra, chair, shoulder_stand,* dan *no_pose.* Total keseluruhan dataset berjumlah 3.149 citra, yang telah dikelompokkan oleh penyusun dataset ke dalam dua kategori utama, yaitu train dan test. Pembagian awal yang disediakan oleh pemilik dataset mempermudah proses verifikasi struktur data sebelum memasuki tahap pra-pemrosesan.

Untuk memberikan gambaran mengenai bentuk data yang digunakan dalam penelitian ini, Gambar 3.2 menampilkan beberapa contoh citra dari dataset. Setiap kelas memiliki karakter visual yang khas yang merepresentasikan struktur pose yoga tertentu.



Gambar 3. 2 Contoh Citra Dataset

Sumber: (www.kaggle.com/datasets/amanupadhyay/yoga-poses, Kaggle 2026)

a. Target Dataset dan Struktur Data

Dataset yang digunakan pada penelitian ini mengikuti struktur folder yang telah ditetapkan oleh pembuat dataset, yaitu pemisahan data ke dalam folder *train* dan *test*, masing-masing terdiri dari delapan kelas pose. Untuk keperluan pelatihan model, folder *train* dibagi kembali menjadi dua bagian, yaitu 85% sebagai data latih

dan 15% sebagai data validasi dengan metode *stratified sampling* agar distribusi data antar kelas tetap seimbang. Sementara itu, folder *test* dipertahankan utuh sebagai data uji pada tahap evaluasi akhir.

Pembagian ini bertujuan memastikan bahwa model dapat diuji menggunakan data yang benar-benar belum pernah dilihat selama proses pelatihan, sehingga performa evaluasi lebih objektif dan menggambarkan kemampuan generalisasi model secara nyata.

b. Proses Verifikasi dan Seleksi Citra

Meskipun dataset telah disusun dalam struktur yang rapi, peneliti tetap melakukan proses verifikasi manual untuk memastikan kualitas dan konsistensi citra. Proses verifikasi ini mencakup:

1) Memastikan format file konsisten dan dapat dibaca oleh sistem.
2) Menghilangkan citra yang rusak (*corrupt*) atau tidak dapat diproses.
3) Memvalidasi kesesuaian antara citra dan label kelas pada folder penyimpanan.
4) Meninjau kualitas visual citra, seperti ketajaman, pencahayaan, dan kesesuaian objek pose.

Tahap verifikasi ini memastikan bahwa seluruh citra yang digunakan pada proses pelatihan telah memenuhi standar kualitas dan konsistensi sehingga dapat mendukung proses pembentukan model klasifikasi pose yoga berbasis CNN secara optimal.

1.2.2 Tahap Pre-Processing

Tahap pre-processing bertujuan menyiapkan dataset agar memenuhi standar input yang dibutuhkan oleh model klasifikasi berbasis Convolutional Neural Network (CNN). Pada penelitian ini, pre-processing dilakukan melalui pendekatan *pose estimation* dengan memanfaatkan model MoveNet untuk mengekstraksi 17 titik tubuh (*keypoints*) dari setiap citra pose yoga. Proses ini bertujuan menyederhanakan citra menjadi representasi numerik yang lebih stabil sehingga model dapat mempelajari pola pose secara efisien.

Seluruh tahapan pre-processing ini berpengaruh langsung terhadap kualitas fitur yang diberikan kepada model CNN, sehingga menentukan stabilitas dan akurasi model pada tahap pelatihan.

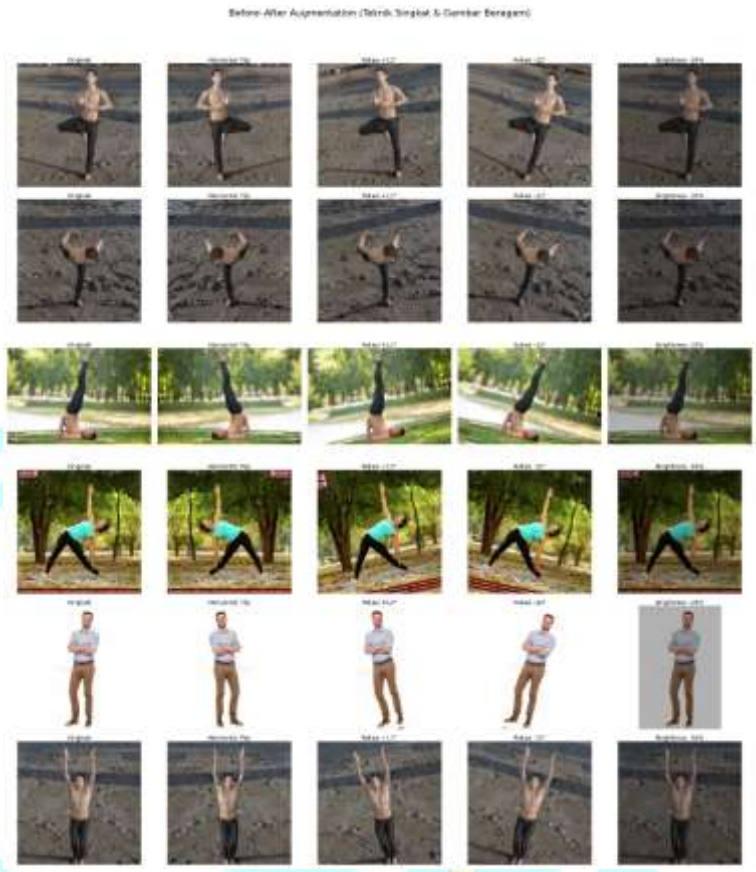
a. Augmentasi Dataset

Tahap pertama augmentasi data dilakukan untuk meningkatkan keragaman sampel dalam dataset serta memperkuat kemampuan generalisasi model CNN. Variasi augmentasi diperlukan karena dataset asli memiliki kemiripan pose, pencahayaan, dan sudut pengambilan gambar yang relatif seragam, sehingga model berpotensi mengalami overfitting apabila hanya dilatih menggunakan citra tanpa variasi tambahan.

Pada penelitian ini, augmentasi dilakukan secara manual menggunakan pemrograman Python, kemudian diterapkan ke seluruh kelas pose yoga. Teknik augmentasi yang digunakan selaras dengan contoh visual yang ditampilkan dan dijelaskan sebagai berikut:

1) *Horizontal Flip* : Teknik ini membalik citra secara horizontal untuk menghasilkan pose yang setara dengan orientasi tubuh pada sisi berlawanan. Variasi ini membantu model mengenali pose yang sama meskipun arah tubuh atau kaki bertumpu berubah.
2) Rotasi Minor (±10–15°) : Rotasi ringan digunakan untuk menyimulasikan perbedaan sudut kamera. Teknik ini bermanfaat untuk meningkatkan kemampuan model dalam mengenali pose yoga yang tidak selalu berada dalam orientasi tegak lurus terhadap kamera.
3) Penyesuaian Brightness : Variasi pencahayaan diterapkan untuk mensimulasikan kondisi lingkungan yang berbeda, seperti cahaya terang, redup, atau sedikit gelap. Teknik ini penting agar model tetap stabil meskipun citra memiliki intensitas cahaya yang bervariasi.

Seluruh augmentasi diterapkan secara terkontrol sehingga tidak mengubah struktur utama pose yoga. Contoh hasil augmentasi dapat dilihat pada gambar berikut.



Gambar 3. 3 Hasil Augmentasi Citra

Sumber: (Dokumen Pribadi,2026)

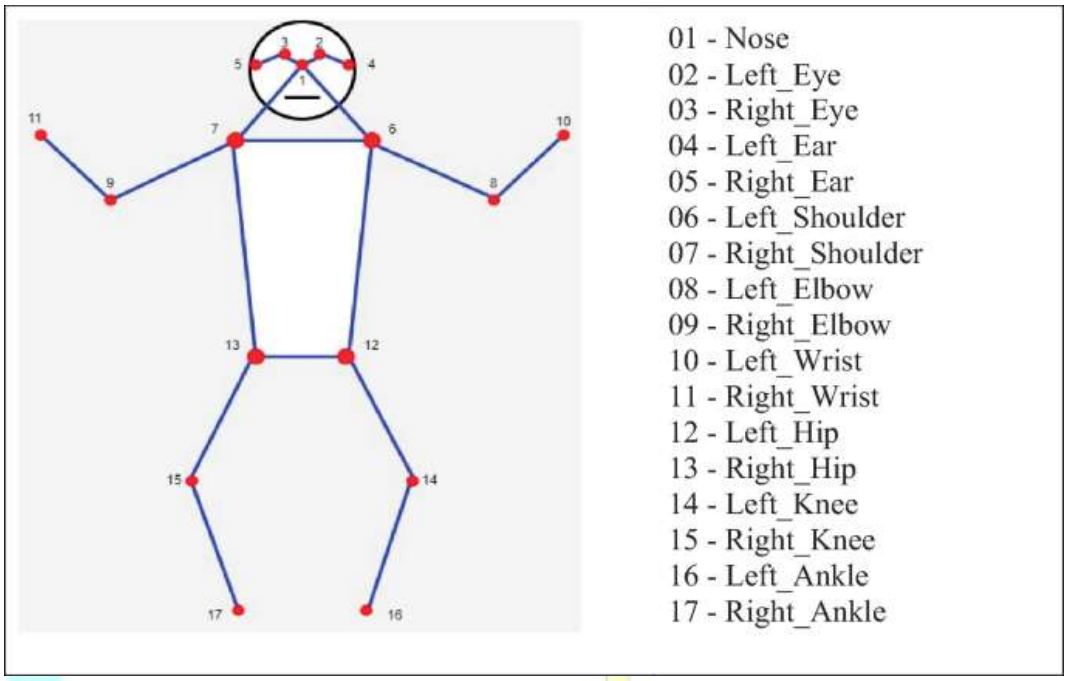
KARAWANG

b. Target Dataset dan Struktur Data

Tahap selanjutnya pre-processing dilakukan dengan menggunakan model MoveNet SinglePose (Thunder) untuk mendeteksi 17 titik tubuh manusia pada setiap citra dalam dataset. Setiap titik tubuh memiliki tiga atribut, yaitu:

1) Koordinat **x**
2) Koordinat **y**
3) Nilai confidence score

Hasil ekstraksi keypoint dari setiap citra berjumlah 51 nilai (17 × 3) dan disimpan dalam berkas CSV (train_data.csv dan test_data.csv). Representasi skeleton dari MoveNet digunakan sebagai dasar transformasi pose sebelum masuk ke model CNN.



Gambar 3. 4 Movenet Skeleton

Sumber: (https://link.springer.com,2026)

Gambar tersebut menggambarkan bagaimana MoveNet membentuk kerangka tubuh (*skeleton*) berdasarkan posisi keypoint, yang menjadi dasar proses klasifikasi pose yoga pada tahap berikutnya.

c. Normalisasi Koordinat Keypoint

Setelah keypoint diekstraksi, langkah berikutnya adalah melakukan normalisasi untuk menyeragamkan skala dan posisi pose antar citra. Normalisasi dilakukan dengan:

1) Menentukan pusat pose (*pose center*) melalui midpoint pinggul kiri dan kanan.
2) Menggeser seluruh koordinat agar berpusat pada pose center.
3) Menskalakan seluruh koordinat berdasarkan ukuran pose.

Normalisasi ini mengurangi pengaruh jarak kamera, tinggi subjek, dan posisi objek di citra, sehingga model hanya fokus pada struktur pose.

d. Transformasi Keypoint Menjadi Pose Embedding

Keypoint yang telah dinormalisasi kemudian ditransformasikan menjadi *pose embedding*. Proses ini dilakukan dengan cara:

1) Mengorganisasi kembali keypoint ke dalam bentuk matriks 17×2
2) Meratakannya menjadi vektor 34 fitur (*flatten*)
3) Mengubahnya menjadi representasi numerik satu dimensi

Vektor pose embedding ini menjadi input utama bagi model CNN dalam proses pelatihan.

e. Pembagian Data Train–Validation

Dataset yang sudah melalui tahap ekstraksi dan embedding kemudian dibagi menjadi tiga bagian:

1. Data latih (train)

   Diambil dari folder *train* dataset Kaggle dan dibagi ulang menjadi:

   a) 85% sebagai data latih
   b) 15% sebagai data validasi

Pembagian ini memastikan bahwa performa model dinilai menggunakan data yang benar-benar belum pernah dilihat sebelumnya.

2. Data validasi (validation)

   Digunakan untuk mengevaluasi performa model selama pelatihan.

f. Filtering dan Pembersihan Data

Tahap berikutnya adalah pemeriksaan dan pembersihan data untuk memastikan kualitas setiap sampel. Proses ini mencakup:

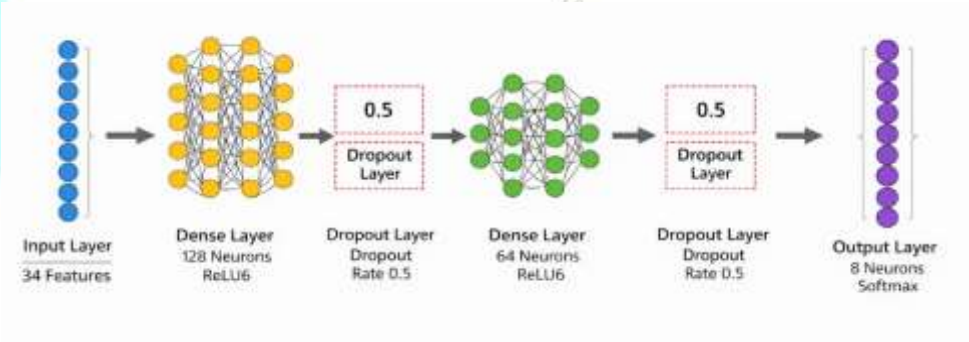
1) Memastikan seluruh keypoint terdeteksi dengan *confidence score* yang layak.
2) Menghilangkan data yang rusak atau tidak dapat diproses.
3) Memeriksa kesesuaian label kelas pada setiap baris data.
4) Menghapus baris yang mengandung nilai kosong atau tidak valid.

Proses ini memastikan bahwa model menerima input yang konsisten dan bebas dari kesalahan structural.

1.2.3 Tahap Pelatihan Model CNN

Tahap pelatihan model merupakan inti dari proses penelitian ini, di mana model Convolutional Neural Network (CNN) dibangun dan dilatih menggunakan data hasil *pose embedding* yang telah diperoleh pada tahap pre-processing. Pada penelitian ini, arsitektur CNN dirancang untuk mempelajari pola koordinat pose yoga yang telah dinormalisasi, sehingga mampu mengenali perbedaan struktur tubuh pada masing-masing kelas pose.

Model CNN dilatih untuk melakukan klasifikasi terhadap delapan kelas pose yoga, yaitu *warrior*, *tree*, *triangle*, *dog*, *cobra*, *chair*, *shoulder_stand*, dan *no_pose*. Proses pelatihan dilakukan secara bertahap dan terukur untuk memastikan model mencapai performa optimal tanpa mengalami overfitting.



Gambar 3. 5 Arsitektur Model CNN

Gambar 3.5 memperlihatkan alur pemrosesan data mulai dari input berupa vektor pose embedding hingga keluaran berupa prediksi kelas pose yoga. Model CNN menerima data berbasis koordinat keypoint tubuh hasil ekstraksi MoveNet dan memprosesnya melalui beberapa lapisan *fully connected* untuk menghasilkan probabilitas setiap kelas pose. Pelatihan model dilakukan melalui beberapa langkah berikut:

a. Penyusunan Arsitektur Model

Model CNN menerima input berupa vektor *pose embedding* berukuran 34 fitur (17 keypoints × 2 koordinat setelah normalisasi). Arsitektur CNN dalam penelitian ini terdiri dari beberapa lapisan utama, yaitu:

1) Input Layer berukuran 34 neuron, yang menerima data embedding dari proses pre-processing.

2) Dense Layer dengan sejumlah neuron (128 dan 64 neuron) yang diaktifkan menggunakan fungsi aktivasi *ReLU6*, berfungsi untuk mengekstraksi pola dari kombinasi koordinat keypoint.
3) Dropout Layer dengan nilai 0.5 untuk mencegah overfitting dan meningkatkan kemampuan generalisasi model.
4) Output Layer dengan delapan neuron dan aktivasi *softmax*, digunakan untuk menghasilkan probabilitas kelas pose yoga.

Pemilihan arsitektur ini disesuaikan dengan karakteristik data yang berbasis koordinat, sehingga tidak memerlukan konvolusi berbasis citra seperti CNN pada objek visual.

b. Konfigurasi Hyperparameter Pelatihan

Beberapa hyperparameter yang digunakan pada proses pelatihan meliputi:

1) Optimizer : Adam, dipilih karena stabil dalam menangani data kompleks dan perubahan gradien.
2) Loss Function : Categorical Crossentropy, sesuai untuk masalah klasifikasi multikelas.
3) Batch Size : 16, memberikan keseimbangan antara kecepatan pelatihan dan stabilitas pembaruan bobot.
4) Epochs : 200, namun dikendalikan oleh mekanisme *early stopping*.
5) Learning Rate standar dari optimizer Adam untuk menjaga kestabilan proses pembelajaran.

Konfigurasi ini dipilih berdasarkan eksperimen awal dan rekomendasi umum untuk model klasifikasi berbasis fitur pose.

c. Mekanisme Pelatihan Bertahap (Training Loop)

Selama proses pelatihan, model menerima data latih dan data validasi. Data digunakan untuk memperbarui bobot jaringan agar model dapat mengenali pola pose yoga secara efektif. Performa model dievaluasi setiap epoch menggunakan metrik akurasi dan nilai loss.

Untuk meningkatkan stabilitas pelatihan, penelitian ini menerapkan dua mekanisme tambahan:

1) ModelCheckpoint : Menyimpan bobot terbaik (*best weights*) berdasarkan performa tertinggi pada data validasi.
2) Early Stopping : Menghentikan pelatihan lebih awal jika model tidak menunjukkan perbaikan signifikan dalam beberapa epoch berturut-turut. Hal ini dilakukan untuk mencegah overfitting dan menghemat waktu komputasi.

Dengan kedua mekanisme tersebut, proses pelatihan berlangsung secara adaptif dan menghasilkan bobot model yang paling optimal.

d. Pemilihan Model Terbaik

Setelah epoch pelatihan selesai, model dengan performa validasi terbaik dipilih berdasarkan hasil *ModelCheckpoint*. Model terbaik ini digunakan untuk:

1) Evaluasi akhir menggunakan data uji.
2) Proses konversi ke TensorFlow.js
3) Implementasi pada aplikasi web sebagai sistem klasifikasi pose yoga secara realtime

Pemilihan model terbaik ini memastikan bahwa implementasi yang digunakan pada aplikasi tidak hanya akurat tetapi juga stabil saat digunakan dalam kondisi nyata.

1.2.4 Tahap Evaluasi Model

Tahap evaluasi model bertujuan untuk menilai kemampuan model CNN dalam melakukan klasifikasi pose yoga secara akurat, konsisten, dan dapat digeneralisasi pada data baru yang tidak digunakan selama proses pelatihan. Evaluasi dilakukan menggunakan *test set* yang berasal dari dataset Kaggle dan tidak terlibat dalam proses pelatihan maupun validasi, sehingga mampu memberikan penilaian objektif terhadap performa akhir model. Evaluasi dilakukan melalui dua pendekatan:

1) Evaluasi kuantitatif menggunakan metrik performa, dan
2) Analisis kesalahan model melalui confusion matrix.

Hasil evaluasi ini menjadi dasar dalam menentukan apakah model layak digunakan pada tahap implementasi sistem berbasis web.

a. Metrik Evaluasi yang Digunakan

Beberapa metrik evaluasi digunakan untuk menilai performa model CNN secara menyeluruh, yaitu:

1) *Accuracy* : *Accuracy* merupakan metrik yang menghitung proporsi prediksi benar terhadap seluruh sampel data uji dan pada penelitian ini model CNN berhasil mencapai akurasi sangat tinggi, menunjukkan bahwa model memiliki kemampuan prediksi yang sangat tepat dalam mengenali berbagai pose yoga secara konsisten.

2) *Precision* : Precision mengukur ketepatan model dalam memprediksi suatu kelas dengan meminimalkan terjadinya kesalahan *false positive*, sehingga nilai precision per kelas memberikan gambaran mengenai kelas mana yang paling presisi dikenali dan menunjukkan bahwa model mampu membedakan pose yoga yang mirip tanpa memberikan prediksi yang salah pada kelas lain.

3) *Recall* : *Recall* merupakan ukuran kemampuan model dalam menangkap seluruh sampel yang benar-benar termasuk ke dalam suatu kelas tanpa terlewat (*false negative*), sehingga sangat penting untuk mengidentifikasi seberapa baik model mengenali pose yang secara visual mirip—misalnya antara pose *warrior* dan *triangle*—yang menuntut sensitivitas tinggi terhadap struktur keypoint tubuh.

4) *F1-Score* : *F1-score* adalah rata-rata harmonis antara precision dan recall yang menggambarkan keseimbangan kinerja model dalam menghindari *false positive* maupun *false negative*, dan nilai *F1-score* yang tinggi pada seluruh kelas menunjukkan bahwa model CNN tidak hanya akurat tetapi juga stabil dan konsisten dalam mengenali berbagai pose yoga meskipun terdapat variasi pose maupun individu.

5) Macro Average dan Weighted Average : Macro average menghitung rata-rata performa seluruh kelas tanpa mempertimbangkan ukuran sampel

sehingga memprioritaskan kesetaraan antar kelas, sedangkan weighted average mempertimbangkan proporsi jumlah data pada setiap kelas sehingga menghasilkan metrik evaluasi global yang lebih representatif dan tidak bias terhadap kelas dengan jumlah sampel besar maupun kecil.

6) Confusion Matrix : model juga dianalisis melalui *confusion matrix* untuk melihat distribusi prediksi benar dan salah pada setiap kelas pose yoga. Confusion matrix memberikan gambaran mengenai kemampuan model dalam membedakan pose yang secara visual memiliki kemiripan serta mengidentifikasi kemungkinan pola kesalahan antar kelas.

1.2.5 Tahap Implementasi Sistem Berbasis Web

Tahap implementasi sistem merupakan tahap akhir dari proses penelitian ini, di mana model CNN yang telah dilatih diintegrasikan ke dalam aplikasi berbasis web sehingga dapat digunakan untuk melakukan klasifikasi pose yoga secara langsung (*real-time*). Implementasi berbasis web dipilih karena memberikan kemudahan akses bagi pengguna tanpa perlu melakukan instalasi perangkat lunak tambahan, serta memungkinkan proses inferensi dilakukan langsung di browser menggunakan TensorFlow.js.

Implementasi sistem ini mencakup beberapa komponen utama, yaitu integrasi model TensorFlow.js, pemrosesan *pose estimation* menggunakan MoveNet, alur klasifikasi pose yoga, serta penyajian hasil prediksi melalui antarmuka web.

a. Integrasi Model CNN ke TensorFlow.js

Model CNN yang telah dilatih pada tahap sebelumnya disimpan dalam format model.h5 dan kemudian dikonversi menjadi format TensorFlow.js agar dapat dijalankan langsung pada browser. Konversi dilakukan menggunakan perintah:

tensorflowjs_converter --input_format=keras model.h5 model_web/

Hasil konversi berupa:

a) model.json
b) file bobot model (group1-shard*)

Model ini kemudian diimpor ke dalam aplikasi React.js untuk melakukan inferensi pose yoga secara *client-side*, sehingga proses prediksi dapat dijalankan dengan respons cepat tanpa bergantung pada server eksternal.

b. Implementasi Pose Estimation Menggunakan MoveNet

Sistem menggunakan model MoveNet SinglePose Lightning/Thunder yang diimplementasikan melalui TensorFlow.js untuk mendeteksi 17 titik tubuh pengguna secara realtime melalui kamera webcam. MoveNet menyediakan koordinat keypoint tubuh (x, y, dan confidence score) yang berfungsi sebagai input bagi model CNN. Setiap frame video dari webcam diproses melalui langkah berikut:

1) Mendapatkan citra dari webcam.
2) Mengestimasi pose menggunakan MoveNet dan menghasilkan 17 keypoints.
3) Melakukan normalisasi dan transformasi keypoint menjadi pose embedding (34 fitur), sesuai proses yang digunakan saat pelatihan.
4) Mengirim pose embedding tersebut ke model CNN untuk klasifikasi pose yoga.

Proses berjalan secara berulang pada setiap frame sehingga aplikasi mampu memberikan prediksi secara realtime.

c. Alur Klasifikasi Pose Yoga pada Aplikasi Web

Proses klasifikasi pose yoga pada aplikasi web dilakukan melalui tahapan berikut:

1) Pengguna memberikan izin akses kamera pada browser.
2) MoveNet mendeteksi pose tubuh pengguna dan menghasilkan 17 titik koordinat.
3) Pose embedding dihitung berdasarkan koordinat keypoint yang telah dinormalisasi.
4) Model CNN melakukan prediksi untuk menentukan kelas pose yoga yang paling mungkin.

5) Probabilitas prediksi diplot pada antarmuka web, menunjukkan pose yang sedang dilakukan oleh pengguna.
6) Jika probabilitas pose tertentu melebihi ambang batas (misalnya 95%), maka sistem menampilkan tanda visual berupa warna rangka tubuh berubah menjadi hijau, menandakan bahwa pose dilakukan dengan benar.

Alur ini memungkinkan aplikasi memberikan umpan balik visual secara langsung kepada pengguna.

d. Implementasi Antarmuka Pengguna (Frontend)

Bagian frontend aplikasi dibangun menggunakan React.js, dengan beberapa komponen utama yang disiapkan untuk mendukung interaksi pengguna, yaitu:

1) Komponen kamera (react-webcam) untuk menampilkan video live.
2) Komponen visualisasi skeleton untuk menggambar garis penghubung antar keypoint MoveNet.
3) Dropdown pemilihan pose yang memungkinkan pengguna memilih pose yoga yang ingin dipraktikkan.
4) Panel instruksi pose, yang menampilkan langkah-langkah melakukan pose yoga dari data *poseInstructions*.
5) Panel hasil prediksi yang menampilkan nama pose yang terdeteksi, waktu terlama, dan indikator warna.

Desain antarmuka dibuat sederhana agar mudah digunakan oleh pemula maupun pengguna umum.

e. Pengujian Fungsional Sistem

Setelah implementasi selesai, dilakukan pengujian fungsional untuk memastikan setiap bagian sistem berjalan sesuai dengan rancangan, yaitu:

1) Akses kamera dapat berjalan tanpa error.
2) MoveNet mampu mendeteksi 17 keypoints tubuh secara realtime.
3) Pose embedding yang dihitung sesuai dengan format pelatihan.
4) Model CNN memberikan prediksi yang stabil dan akurat.
5) Sistem mampu menampilkan skeleton serta warna indikator pose.

6) Waktu respons prediksi berada dalam kisaran ~20–30 ms per frame, sehingga aplikasi berjalan mulus.

Hasil pengujian menunjukkan bahwa seluruh komponen sistem dapat berfungsi dengan baik dan mampu memberikan pengalaman penggunaan yang interaktif serta akurat.

f. Kelayakan Implementasi Sistem

Berdasarkan performa prediksi yang tinggi dan hasil pengujian fungsional, sistem klasifikasi pose yoga ini dinyatakan layak diterapkan sebagai aplikasi pembelajaran yoga berbasis web. Integrasi model CNN dan MoveNet dalam TensorFlow.js terbukti mampu melakukan inferensi secara realtime tanpa bergantung pada server, sehingga efisien, ringan, dan mudah diakses oleh pengguna.

UBP
KARAWANG