

Implementasi Algoritma Support Vector Machine Terhadap Klasifikasi Pose Balet

Romindo¹, Okky Putra Barus^{2,*}, Jefri Junifer Pangaribuan³, Yudhistira Adhitya Pratama⁴, Evelyn Wiliem⁵

Fakultas Ilmu Komputer, Sistem Informasi, Universitas Pelita Harapan, Jakarta, Indonesia

Email: ¹romindo@uph.edu, ^{2,*}okky.barus@uph.edu, ³jefri.pangaribuan@uph.edu, ⁴yudhistira.adhitya@lecturer.uph.edu, ⁵ew80020@student.uph.edu

Email Penulis Korespondensi: okky.barus@uph.edu

Submitted: 08/12/2022; Accepted: 27/12/2022; Published: 30/12/2022

Abstrak—Balet telah dianggap sebagai salah satu tarian yang paling sulit dikarenakan postur teknis yang dituntut. Jika dilakukan tanpa bimbingan dapat menyebabkan postur yang buruk pada penari balet dan beberapa cedera serius. Sebuah model dalam mengidentifikasi berbagai pose balet telah dikembangkan dengan kecerdasan buatan untuk mengatasi masalah ini. Tujuan utama dari penelitian ini adalah untuk menunjukkan sebuah metodologi yang menyederhanakan pengenalan pose balet menggunakan framework opensource MediaPipe dan algoritma machine learning Support Vector Machine. Cara kerja model ini akan melewati dua tahap: pertama, mengekstrak titik data dari kumpulan data gambar menggunakan library MediaPipe Pose Estimation, lalu memproses, melatih, memvalidasi, dan menguji data tersebut menggunakan algoritme Support Vector Machine untuk melakukan beberapa klasifikasi pose. Model ini dilatih dalam tujuh pose balet yang berbeda, termasuk First Position, Second Position, Third Position, Fourth Position, Fifth Position, Tendu Devant, and Tendu Derrière. Hal ini dilakukan untuk menilai kompetensi model klasifikasi yang dibuat. Skor akurasi setinggi 87% dicapai dari model klasifikasi pose balet yang dibuat dan telah dikembangkan untuk bekerja input berupa gambar dan video langsung.

Kata Kunci: Machine Learning; Pose Recognition; MediaPipe; OpenCV

Abstract— Ballet is considered as one of the most difficult dance due to its technical posture demanded. If performed without guidance it may cause bad posture to ballerina and some serious injuries. A model in identifying different ballet poses is developed with artificial intelligence in order to tear down this barrier. The main purpose of this paper is to demonstrate a methodology that simplified Ballet Pose Recognition using an opensource framework called MediaPipe and a machine learning algorithm called Support Vector Machine. How the model work is it will pass through two stages: first, it extracts data points from an image dataset using the MediaPipe Pose Estimation library, and then it preprocesses the data, trains, validates, and tests it using the Support Vector Machine algorithm to do some pose classification. The model is trained in seven distinct ballet poses, including First Position, Second Position, Third Position, Fourth Position, Fifth Position, Tendu Devant, and Tendu Derrière. This is purposely done in order to assess the competence of the classification model. An accuracy score of 87% is achieved from the ballet pose classification model and is developed to work on images and live videos.

Keywords: Machine Learning, Pose Recognition, MediaPipe, OpenCV

1. PENDAHULUAN

Untuk seorang penari balet mencapai langkah, gerakan, dan lompatan yang indah dan sempurna, dibutuhkan latihan yang benar dengan memperhatikan teknik setiap postur yang dilakukan. Sangat wajar bagi seorang penari berlatih selama berbulan-bulan hingga bertahun. Hal ini dikarenakan tarian balet sendiri sudah dikenal akan sebuah bentuk seni yang sangat sulit untuk dicapai [1]. Untuk mengatasi hal tersebut, sangat penting adanya bimbingan dan perhatian satu per satu oleh guru saat dikelas dan melakukan latihan tambahan dengan sendirinya [2]. Tetapi tanpa bimbingan guru, penari memiliki kemungkinan besar tidak dapat melakukan pose dengan benar walaupun sudah melihat video gerakan balet. Jika tidak ditangani benar, ini dapat menyebabkan kebiasaan penempatan postur tubuh penari balet yang buruk dan juga risiko cedera yang lebih tinggi yang di mana didapati sebesar 11.2% penari balet baik pria maupun wanita mengalami cedera yang kerap terjadi pada pergelangan kaki [3].

Seiring berkembangnya teknologi, berbagai inovasi juga telah dilakukan untuk membantu mendeteksi sebuah postur dengan benar. Metode yang pernah dilakukan sebelumnya adalah menggunakan *sensor based*. Akan tetapi, metode ini sangatlah mahal dan juga kompleks untuk digunakan. Dilain hal, pendeteksian postur tanpa harus menggunakan alat yang mahal dan hanya bermodal sebuah kamera juga dapat digunakan, yaitu dengan *pose estimation*. *Pose estimation* adalah sebuah teknik pada *computer vision* yang digunakan untuk mendeteksi gerakan dari seseorang ataupun sebuah objek dengan mengidentifikasikan lokasi *landmarks* atau titik tubuh seorang. Implementasi *pose estimation* sendiri juga sudah sering digunakan pada dunia *Augmented Reality*, *robotics*, animasi, bahkan sampai dunia olahraga [4]. Salah satu peneliti oleh Y. Agrawal, Y. Shah, and A. Sharma pada tahun 2022 yang telah mengimplementasikan *pose estimation* pada yoga mengungkapkan bahwa *pose estimation* bukan hanya dapat digunakan untuk mendeteksi postur tetapi dapat digunakan juga sebagai sebuah sarana untuk membantu seorang melakukan postur yoga yang lebih akurat [5]. Akan tetapi penentuan postur tubuh tidak bisa berlangsung hanya dengan *pose estimation* saja. Dengan itu, dibutuhkan sebuah sarana untuk melakukan penentuan atau pendeteksian jenis pose daripada data *pose estimation* yang sudah didapati. Untuk mengatasi hal ini dapat digunakan sebuah metode yang sudah lama digunakan dalam dunia klasifikasi yaitu, *machine learning*. *Machine learning* sendiri sudah dikenal dengan kemampuannya dalam melakukan prediksi ataupun klasifikasi hanya dari data yang diberikan untuk menghasilkan prediksi ataupun mengategorikan data ke kelas tertentu [6]. Implementasi *machine learning* juga

sudah beredar pada berbagai industri seperti kesehatan, *e-commerce*, transportasi, dan lain lagi. Hal ini menunjukkan seberapa efektif dan efisien penggunaan *machine learning* dalam membantu melakukan prediksi atau klasifikasi [7]. Beberapa penelitian sebelumnya pada dunia *pose detection* atau penentuan sebuah pose yakni, mendeteksi dan mengklasifikasi isyarat tangan yang menggunakan metode *MediaPipe* dengan *Support Vector Machine (SVM)* memiliki akurasi sebesar 99,15 % [8], dan mendeteksi pose balet menggunakan metode *OpenPose* yang dibantu dengan *Support Vector Machine (SVM)* untuk klasifikasi. Penelitian ini menunjukkan akurasi dalam pendeteksian dan klasifikasi senilai 99,09% [9].

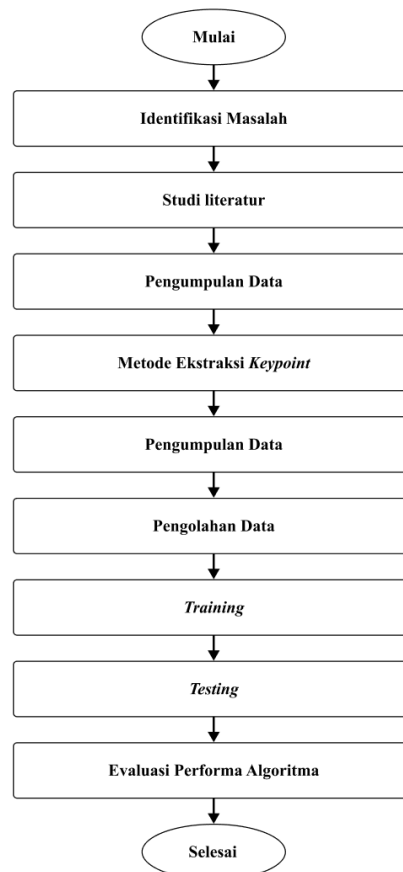
Berdasarkan permasalahan yang ada dan keefektifan penggunaan *pose detection* dari penelitian – penelitian sebelumnya, peneliti mengimplementasikan *pose detection* pada pose balet dengan menggunakan *library MediaPipe* sebagai metode untuk melakukan deteksi titik tubuh penari balet dan juga akan dibantu dengan *Support Vector Machine (SVM)* untuk melakukan klasifikasi pose balet dengan menggunakan bahasa pemrograman *Python* dan *library OpenCV*. Dengan itu, peneliti mengangkat judul penelitian “Implementasi Algoritma Support Vector Machine Terhadap Klasifikasi Pose Balet”.

Penelitian sebelumnya membahas tentang tanggapan atau opini masyarakat luas, khususnya pada media sosial Twitter tentang perusahaan mereka [10][11][12]. Algoritma Support Vector Machine (SVM) yang telah dioptimisasi oleh algoritma Particle Swarm Optimization (PSO) diharapkan menghasilkan tingkat akurasi yang lebih baik dalam analisis sentiment dibandingkan dengan algoritma Support Vector Machine (SVM) yang digunakan secara konvensional [13].

2. METODOLOGI PENELITIAN

2.1 Tahapan Penelitian

Untuk tahapan penelitian yang akan digunakan dijelaskan alur penelitian secara diagram pada gambar 1.



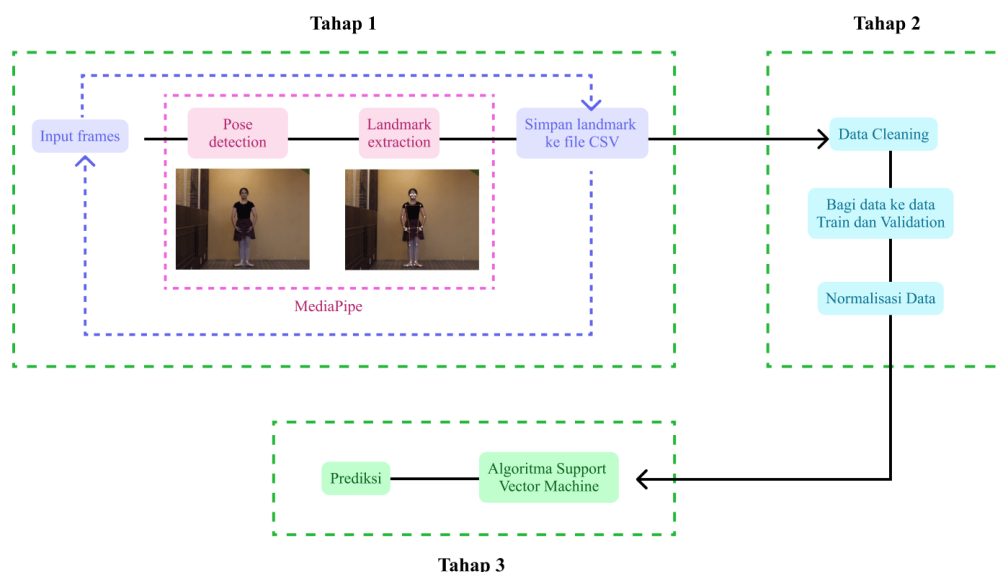
Gambar 1. Tahapan penelitian

Tahapan penelitian berawal dari persiapan penelitian seperti pada gambar 1 pada tahap ini, dilakukan penentuan latar belakang dari suatu masalah yang didapati dan juga batasan masalah untuk dibahas pada penelitian ini. Selain itu untuk mengetahui fungsional dibuatnya penelitian ini, maka dicantumkan juga tujuan dan manfaat dari penelitian ini. Selanjutnya dilanjutkan dengan tahap persiapan data. Tahap ini terdiri dari empat tahapan, yaitu: pengambilan *image dataset* yang diambil dengan sendiri, menentukan *human pose estimation landmarks* menggunakan *MediaPipe*, melakukan *data cleaning*, *data normalisation*, pembagian data ke data pelatihan (*training*) dan data pengujian (*testing*).

Tahap selanjutnya adalah tahap *machine learning*. Membuat model klasifikasi menggunakan algoritme *Support Vector Machine* yang dibantu dengan *Python* beserta *library Python* lainnya. Model *Support Vector Machine* yang dibuat dilakukan pelatihan terhadap modelnya. Lalu hasil model yang sudah dilakukan pelatihan akan diuji dengan data pengujian. Tahapan terakhir adalah melakukan analisis menggunakan akurasi, *precision*, *recall*, *f1 score* terhadap model klasifikasi *Support Vector Machine* dan mengambil kesimpulan serta memberikan saran dari penelitian yang telah dibuat.

2.2 Metode Penyelesaian

Pada gambar 2 menjelaskan cara mengimplementasikan model *machine learning* dengan *Support Vector Machine* pada klasifikasi pose balet dan menjelaskan arsitektur yang menjadi panduan dan tahapan yang akan dirancang untuk penelitian ini.



Gambar 2. Arsitektur

2.3 Preprocessing Input Data Gambar Untuk Mendapatkan *Human Pose Estimation Landmarks* Menggunakan *MediaPipe*

Penelitian ini berawal dari mendapatkan *landmark point* dari *dataset* gambar pose balet yang diberikan. Untuk melakukan ini akan dibantu dengan *MediaPipe* yang menawarkan model *Person/Pose Detection* dan model *Pose Landmark* sebagai solusi.

Pertama-tama akan diambil sebuah gambar dari folder *dataset* sebagai data *input*. Selanjutnya model *Person/Pose Detection* akan digunakan untuk mendeteksi wilayah yang memiliki manusia dari sebuah gambar dengan membuat dua *key points* secara *virtual*. Setelah itu, model *Pose Landmark* akan mendeteksi 33 koordinat sendi tubuh 3D (yaitu, sumbu x, y, dan z) pada wilayah yang terdeteksi pose manusia. *Landmark* yang diperoleh akan disimpan dalam *file* berformat CSV. Proses deteksi *landmark* ini akan berlangsung sampai seluruh *dataset* telah diperoleh nilai *landmark*nya pada *file* CSV tersebut. Biasanya proses ini membutuhkan waktu kisa ran 10 menit sampai 15 menit tergantung ukuran *dataset* yang ada.

2.4 Pembersihan Data

Setelah mengumpulkan semua *landmark* dan disimpan ke dalam *file* CSV, pembersihan data (*data cleaning*) dan normalisasi data datang selanjutnya. Untuk melakukan pembersihan dan normalisasi data, *library Pandas* akan digunakan untuk membantu proses ini.

Dalam proses pembersihan data, *Pandas library* berfungsi untuk memeriksa entri *null* yang ditemukan di *file* CSV. Biasanya nilai *null* diperoleh karena gambar buram dari *dataset* yang diberikan sehingga menyebabkan kegagalan dalam mendeteksi pose karena *library MediaPipe* karena tidak dapat mengidentifikasi setiap titik *landmark* dengan benar. Untuk meminimalisir bias pada saat proses klasifikasi nantinya, maka perlu dilakukan pembersihan titik-titik *landmark* tersebut pada *file* CSV. Ini dapat dilakukan dengan melakukan pembersihan data dengan *Pandas library*. *Pandas library* akan melakukan beberapa pencarian untuk nilai *null* dan akan menghapus setiap baris dari tabel ketika nilai *null* ditemukan menggunakan indeksinya. Selain itu, disini juga akan dibuang kolom z dan v karena tidak digunakan pada penelitian ini .

2.5 Data Pelatihan dan Validasi

Selanjutnya, data yang dibersihkan dari *file* CSV akan digunakan sebagai *dataset* untuk melatih model *machine learning*. *Dataset* akan dibagi menjadi set pelatihan (*training*) dan set uji (*testing*). Untuk set pelatihan akan menggunakan 70% dari *dataset* yang ada. 30% data lainnya digunakan untuk menguji model.

2.6 Normalisasi Data

Pada tahap ini akan dilakukan normalisasi data koordinat X dan Y dengan *library Pandas*. Proses ini akan membantu dalam menyediakan data dengan kualitas yang lebih tinggi dengan menghasilkan data dengan *range* nilai 0 hingga 1. Rumus yang akan digunakan untuk normalisasi adalah rumus min-max sebagai berikut [14]:

$$\text{Normalisasi} = \frac{\text{data}_x - \text{data}_{\min}}{\text{data}_{\max} - \text{data}_{\min}} \quad (1)$$

2.7 Melakukan Prediksi dengan Support Vector Machine

Setelah dilakukan proses sebelumnya, prediksi pose akan dilakukan selanjutnya. Untuk prediksi akan digunakan algoritme *Support Vector Machine*. *Support Vector Machine* akan menggunakan rumus sebagai berikut untuk menyelesaikan masalah penelitian ini [15] [16]:

$$\min_{\frac{1}{2}} |w|^2 + C(\sum_{i=1}^n \xi_i) \quad (2)$$

$$y_i(x_i \cdot w + b) \geq 1 - \xi_i ; \xi_i \geq 0 \quad (3)$$

2.8 Metode Analisis Data

Terakhir, hasil klasifikasi akan dianalisis untuk menentukan validitas model *machine learning* yang dibuat dengan *dataset* pengujian. Ini akan diselesaikan dengan menggunakan matriks kinerja seperti akurasi, *presicion*, *recall*, dan *f1 score*. Lebih jelasnya akan dijelaskan di bawah ini [17] [18]:

a. Akurasi

Akurasi akan menentukan berapa banyak titik data yang dihitung dengan benar dari semua titik data yang ditemukan. Hal ini dapat dihitung dengan persamaan berikut:

$$\text{Akurasi} = \frac{\text{Jumlah data klasifikasi benar}}{\text{Jumlah seluruh data}} \quad (4)$$

b. Presisi

Presisi akan membantu meminimalkan nilai *false positive (FP)* dengan menentukan mana dari semua prediksi positif yang merupakan positif secara kebenaran. Hal ini dapat dihitung dengan persamaan berikut:

$$\text{Precision} = \frac{TP}{TP+FP} \quad (5)$$

c. Recall

Recall akan mengidentifikasi berapa banyak nilai positif secara kebenaran yang benar-benar dikategorikan oleh model dengan melabelinya sebagai positif. Hal ini dapat dilakukan seperti yang terlihat pada persamaan di bawah ini:

$$\text{Recall} = \frac{TP}{TP+FN} \quad (6)$$

d. F1 Score

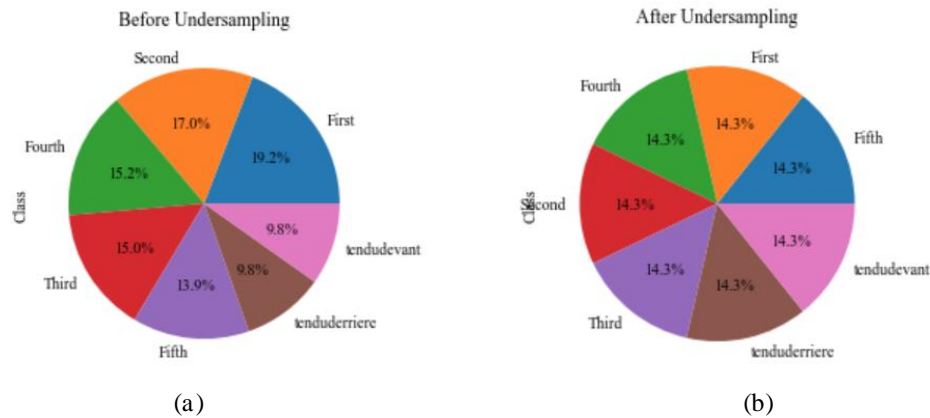
Terakhir, *f1 score* akan digunakan untuk menangani ketidakseimbangan dalam klasifikasi yang dibuat dari model *Support Vector Machine*. Ini dapat dilakukan dengan menggabungkan nilai presisi dan *recall* seperti yang terlihat pada persamaan di bawah ini:

$$f1 - \text{score} = \frac{2(\text{Precision} * \text{Recall})}{\text{Precision} + \text{Recall}} \quad (7)$$

3. HASIL DAN PEMBAHASAN

3.1 Hasil Pembersihan Data

Penelitian ini berawal dari proses pembersihan data untuk memastikan data yang digunakan sudah layak. Disini akan dihapus kolom data *z* dan *visibility* beserta data-data yang duplikat dan *null*. Hasilnya, *dataset* yang awalnya berjumlah 1488 baris dan 133 kolom pun berakhir dengan 1090 baris dan 67 kolom. *Undersampling* juga dilakukan untuk mengatasi data yang tidak seimbang. Pada gambar 3 adalah rangkuman hasil *dataset* sebelum dan sesudah *undersampling*.

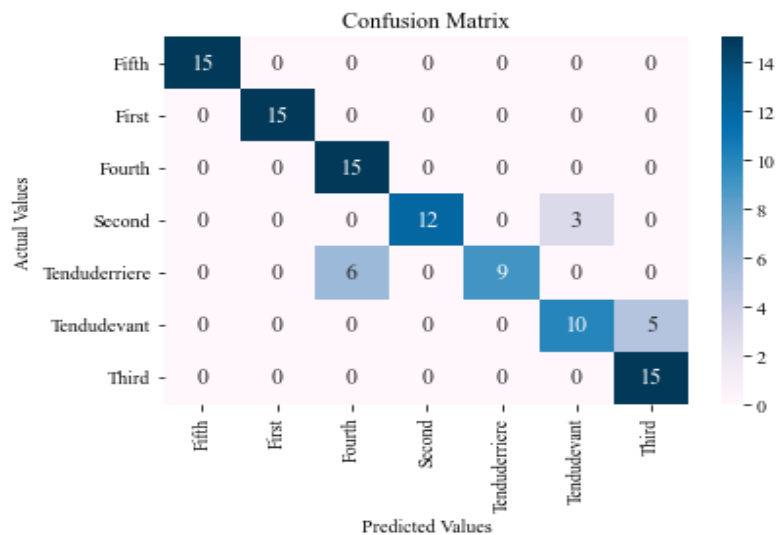


Gambar 3. (a) Gambar sebelum dan (b) gambar sesudah dilakukan undersampling

Dataset yang telah dilakukan undersampling dibagi menjadi 70% data pelatihan (training) dengan jumlah data pelatihan yang berjumlah 524 data dan 30% data validasi (validation) dengan jumlah data validasi berjumlah 225 data yang akan dilanjutkan dengan normalisasi agar setiap data pada dataset ada dalam rentang yang sama.

3.2 Hasil Klasifikasi Algoritma

Hasil validasi model klasifikasi menghasilkan sebanyak 205 data dari 225 data yang berhasil diprediksi dengan benar. Pada tahap pengujian diuji langsung menggunakan webcam laptop dengan kondisi per pose diuji dengan kriteria hadap depan, kanan, dan kiri. 5 gambar per pose dan kriteria akan disimpan untuk dilihat performanya. Untuk deteksi pose dengan kondisi hadap, algoritme Support Vector Machine dapat melakukan klasifikasi dengan benar untuk 35 sampel data. Akan tetapi dari 35 sampel data yang didapati untuk melakukan deteksi pose dengan kondisi hadap depan, 1 diantaranya tidak dapat melakukan prediksi pose dengan baik. Selain itu dari 35 sampel data yang didapati untuk melakukan deteksi pose dengan kondisi hadap kanan, 13 diantaranya tidak dapat melakukan prediksi pose dengan baik. Pada pengujian kelas Second, model mengenal pose ini sebagai kelas Tendu Devant. Gambar 5 mendemonstrasikan klasifikasi pose menggunakan webcam laptop dan confusion matrix diilustrasikan pada gambar 4 untuk menjelaskan hasil klasifikasi lebih detail [19][20].



Gambar 4. Confusion Matrix

Dari confusion matrix dapat dilihat bahwa algoritme Support Vector Machine mampu melakukan klasifikasi dengan benar pada kelas Fifth, First, Fourth dan Third dengan data yang diberikan tiap kelas sebanyak 15 data. Akan tetapi pada kelas Second, Support Vector Machine hanya mampu mengklasifikasi sebanyak 12 data dengan benar. Sisa 3 data yang seharusnya tergolong pada kelas Second diklasifikasikan sebagai kelas Tendu Devant. Terdapat juga pada kelas Tendu Derrière, Support Vector Machine juga hanya mampu mengklasifikasi sebanyak 9 data dengan benar. 6 data lainnya yang seharusnya tergolong pada kelas Tendu Derrière diklasifikasikan sebagai kelas Fourth. Untuk Tendu Devant, Support Vector Machine juga hanya mampu mengklasifikasi sebanyak 10 data dengan benar. 5 data lainnya yang seharusnya tergolong pada kelas Tendu Devant diklasifikasikan sebagai kelas Third. Nilai True Positive, False Negative, False Positive, dan True Negative dari tiap kelas juga dapat dihasilkan untuk tiap kelasnya dari confusion matrix. Nilai-nilai tersebut dirangkum pada tabel 1.



Tabel 1. Tabel perhitungan manual *confusion matrix*

Kelas	True Positive	False Negative	False Positive	True Negative
First	15	0	17	90
Second	12	3	0	90
Third	15	0	5	85
Fourth	15	0	6	84
Fifth	15	0	17	90
Tendu Devant	10	5	3	87
Tendu Derrière	9	6	0	90

Berdasarkan nilai-nilai yang didapati dari tiap kelas, dapat dihitung pula nilai *accuracy*, *precision*, *recall*, dan *f1 score*. Berikut jabaran hasil perhitungannya:

a. Akurasi

Performa akurasi secara keseluruhan pada hasil penelitian ini menghasilkan akurasi setinggi 87%. Persentase nilai akurasi didapati dari hitungan berikut:

$$\text{Akurasi} = \frac{\text{Jumlah data klasifikasi benar}}{\text{Jumlah seluruh data}} = \frac{91}{105} \approx 0.87 * 100 = 87\%$$

b. Performa untuk kelas *Fifth*

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} = \frac{15}{15 + 0} = 1 * 100 = 100\%$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} = \frac{15}{15 + 0} = 1 * 100\% = 100\%$$

$$f1 \text{ score} = \frac{2(\text{Precision} * \text{Recall})}{\text{Precision} + \text{Recall}} = \frac{2(1 * 1)}{1 + 1} = 1 * 100 = 100\%$$

Dari hasil performa untuk kelas *Fifth*, upaya *Support Vector Machine* dalam mengkalsifikasi data yang diberikan label *Fifth* saat prediksi yang sebenarnya adalah label *Fifth* setinggi 100%. Secara keseluruhan kelas *Fifth*, *Support Vector Machine* juga berhasil mengklasifikasikan ke dalam label *Fifth* dengan benar setinggi 100%. Keseimbangan *precision* dan *recall* adalah 100%.

c. Performa untuk kelas *First*

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} = \frac{15}{15 + 0} = 1 * 100 = 100\%$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} = \frac{15}{15 + 0} = 1 * 100\% = 100\%$$

$$f1 \text{ score} = \frac{2(\text{Precision} * \text{Recall})}{\text{Precision} + \text{Recall}} = \frac{2(1 * 1)}{1 + 1} = 1 * 100 = 100\%$$

Dari hasil performa untuk kelas *First*, upaya *Support Vector Machine* dalam mengkalsifikasi data yang diberikan label *First* saat prediksi yang sebenarnya adalah label *First* setinggi 100%. Secara keseluruhan kelas *First*, *Support Vector Machine* juga berhasil mengklasifikasikan ke dalam label *first* dengan benar setinggi 100%. Keseimbangan *precision* dan *recall* adalah 100%.

d. Performa untuk kelas *Fourth*

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} = \frac{15}{15 + 6} \approx 0.71 * 100 = 71\%$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} = \frac{15}{15 + 0} = 1 * 100\% = 100\%$$

$$f1 \text{ score} = \frac{2(\text{Precision} * \text{Recall})}{\text{Precision} + \text{Recall}} = \frac{2(0.71 * 1)}{0.71 + 1} \approx 0.83 * 100 = 83\%$$

Dari hasil performa untuk kelas *Fourth*, upaya *Support Vector Machine* dalam mengkalsifikasi data yang diberikan label *Fourth* saat prediksi yang sebenarnya adalah label *Fourth* setinggi 71%. Secara keseluruhan kelas *Fourth*, *Support Vector Machine* juga berhasil mengklasifikasikan ke dalam label *fourth* dengan benar setinggi 100%. Keseimbangan *precision* dan *recall* adalah 83%.

e. Performa untuk kelas *Second*



$$Precision = \frac{TP}{TP + FP} = \frac{12}{12 + 0} = 1 * 100 = 100\%$$

$$Recall = \frac{TP}{TP + FN} = \frac{12}{12 + 3} \approx 0.8 * 100\% = 80\%$$

$$f1\ score = \frac{2(Precision * Recall)}{Precision + Recall} = \frac{2(1 * 0.8)}{1 + 0.8} \approx 0.89 * 100 = 89\%$$

Dari hasil performa untuk kelas *Second*, upaya *Support Vector Machine* dalam mengklasifikasi data yang diberikan label *Second* saat prediksi yang sebenarnya adalah label *Second* setinggi 100%. Secara keseluruhan kelas *Second*, *Support Vector Machine* juga berhasil mengklasifikasikan ke dalam label *second* dengan benar setinggi 80%. Keseimbangan *precision* dan *recall* adalah 89%.

f. Performa untuk kelas *Third*

$$Precision = \frac{TP}{TP + FP} = \frac{15}{15 + 5} \approx 0.75 * 100 = 75\%$$

$$Recall = \frac{TP}{TP + FN} = \frac{15}{15 + 0} = 1 * 100\% = 100\%$$

$$f1\ score = \frac{2(Precision * Recall)}{Precision + Recall} = \frac{2(0.75 * 1)}{0.75 + 1} \approx 0.86 * 100 = 86\%$$

Dari hasil performa untuk kelas *Third*, upaya *Support Vector Machine* dalam mengklasifikasi data yang diberikan label *Third* saat prediksi yang sebenarnya adalah label *Third* setinggi 75%. Secara keseluruhan kelas *third*, *Support Vector Machine* juga berhasil mengklasifikasikan ke dalam label *Third* dengan benar setinggi 100%. Keseimbangan *precision* dan *recall* adalah 86%.

g. Performa untuk kelas *Tendu Derrière*

$$Precision = \frac{TP}{TP + FP} = \frac{9}{9 + 0} = 1 * 100 = 100\%$$

$$Recall = \frac{TP}{TP + FN} = \frac{9}{9 + 6} \approx 0.6 * 100\% = 60\%$$

$$f1\ score = \frac{2(Precision * Recall)}{Precision + Recall} = \frac{2(1 * 0.6)}{1 + 0.6} \approx 0.75 * 100 = 75\%$$

Dari hasil performa untuk kelas *Tendu Derrière*, upaya *Support Vector Machine* dalam mengklasifikasi data yang diberikan label *Tendu Derrière* saat prediksi yang sebenarnya adalah label *Tendu Derrière* setinggi 100%. Secara keseluruhan kelas *Tendu Derrière*, *Support Vector Machine* juga berhasil mengklasifikasikan ke dalam label *Tendu Derrière* dengan benar setinggi 60%. Keseimbangan *precision* dan *recall* adalah 75%.

h. Performa untuk kelas *Tendu Devant*

$$Precision = \frac{TP}{TP + FP} = \frac{10}{10 + 3} \approx 0.77 * 100 = 77\%$$

$$Recall = \frac{TP}{TP + FN} = \frac{10}{10 + 5} \approx 0.67 * 100\% = 67\%$$

$$f1\ score = \frac{2(Precision * Recall)}{Precision + Recall} = \frac{2(0.77 * 0.67)}{0.77 + 0.67} \approx 0.71 * 100 = 71\%$$

Dari hasil performa untuk kelas *Tendu Devant*, upaya *Support Vector Machine* dalam mengklasifikasi data yang diberikan label *Tendu Devant* saat prediksi yang sebenarnya adalah label *Tendu Devant* setinggi 77%. Secara keseluruhan kelas *Tendu Devant*, *Support Vector Machine* juga berhasil mengklasifikasikan ke dalam label *Tendu Devant* dengan benar setinggi 67%. Keseimbangan *precision* dan *recall* adalah 71%.

Selanjutnya nilai *True Positive*, *False Negative*, *False Positive*, dan *True Negative* digunakan untuk menghitung performa model klasifikasi dengan algoritma *Support Vector Machine*. Nilai tersebut dapat menghitung performa akurasi model secara keseluruhan, nilai *precision*, *recall*, dan juga *F1 score*. Hasil performa yang didapati dirangkum pada tabel 2.

Tabel 2. Rangkuman hasil perhitungan *confusion matrix*

Kelas	Overall Accuracy	Precision	Recall	F1-score
<i>Fifth</i>	87%	100%	100%	100%

<i>First</i>	100%	100%	100%
<i>Fourth</i>	71%	100%	83%
<i>Second</i>	100%	80%	89%
<i>Tendu Derrière</i>	100%	60%	75%
<i>Tendu Devant</i>	77%	67%	71%
<i>Third</i>	75%	100%	86%

Berdasarkan hasil performa yang didapati, secara keseluruhan akurasi model yang dihasilkan mencapai 87% dengan nilai *precision* berkisar 71% sampai dengan 100%, nilai *recall* berkisar 60% sampai dengan 100% dan nilai *F1 score* berkisar 71% sampai dengan 100%.

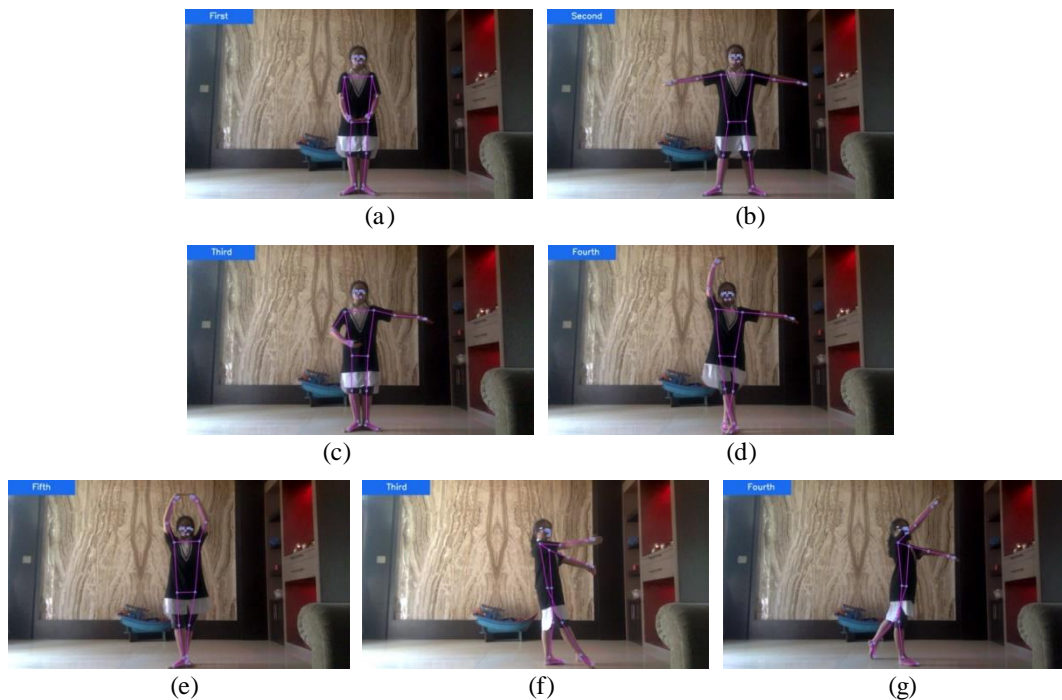
Dari hasil penelitian yang dilakukan, model *machine learning* yang dibuat secara keseluruhan berhasil melakukan klasifikasi pose dengan benar yang telah diuji ke tiga kondisi yaitu hadap depan, kanan, dan kiri. Dalam pengujian ke tiga kondisi ini, model *machine learning* yang dibuat mengalami beberapa kesulitan. Berikut penjelasannya yang terlihat pada gambar 5:

a. Hadap depan

Dari 35 sampel data yang didapati untuk melakukan deteksi pose dengan kondisi hadap depan, 1 diantaranya tidak dapat melakukan prediksi pose dengan baik. Pada pengujian kelas *Tendu Derrière*, model mengenal pose ini sebagai kelas *Fourth Position*.

b. Hadap kanan

Dari 35 sampel data yang didapati untuk melakukan deteksi pose dengan kondisi hadap kanan, 13 diantaranya tidak dapat melakukan prediksi pose dengan baik. Pada pengujian kelas *Second*, model mengenal pose ini sebagai kelas *Tendu Devant*.



Gambar 5. (a-g) Simulasi klasifikasi pose balet menggunakan webcam secara *real time*

4. KESIMPULAN

Penerapan algoritma *Support Vector Machine* untuk mengklasifikasi pose balet dilakukan dengan menggunakan bahasa pemrograman *Python* dengan *library* dan perangkat lunak pendukung *Jupyter Notebook* untuk merancang model *machine learning* dengan bahasa pemrograman *Python*, *OpenCV* sebagai *library* yang diperlukan untuk mengambil citra/gambar serta mengidentifikasi dan mengenali wajah, *Pandas* yang digunakan untuk pembersihan dan normalisasi data dan *Sickitlearn* yang digunakan untuk menyediakan algoritma *support vector machine* dan digunakan untuk *imbalanced-learned* untuk mengatasi data yang tidak seimbang. Performa yang dihasilkan dalam mengklasifikasi pose balet memiliki akurasi secara keseluruhan setinggi 87% dengan memanfaatkan *library MediaPipe* untuk penentuan pose dan algoritme *Support Vector Machine* untuk klasifikasi pose. Kedepannya penelitian ini dapat dikembangkan dengan menghitung sudut tubuh dari sebuah pose balet agar dapat digunakan sebagai sarana untuk membantu penari balet mendapatkan postur balet yang lebih akurat juga dapat diimplementasikan pada website, Android maupun iOS agar pengklasifikasian pose dapat digunakan kapanpun dan dimanapun dan mudah diakses.

REFERENCES

- [1] MoveDance, “The Most Difficult Moves in Ballet - Move Dance,” Jun. 2019. https://www.movedancewear.com/blog/the_most_difficult_moves_in_ballet/486/ (accessed Dec. 08, 2021).
- [2] M. Fourie and D. van der Haar, “Computer vision for the ballet industry: A comparative study of methods for pose recognition,” *Lecture Notes in Business Information Processing*, vol. 389 LNBIP, pp. 118–129, 2020, doi: 10.1007/978-3-030-53337-3_9.
- [3] B. Novosel, D. Sekulic, M. Peric, M. Kondric, and P. Zaletel, “Injury Occurrence and Return to Dance in Professional Ballet : Prospective Analysis of Specific Correlates”, doi: 10.3390/ijerph16050765.
- [4] A. Gupta, “Human Pose Estimation Using Machine Learning in Python,” 2021. <https://www.analyticsvidhya.com/blog/2021/10/human-pose-estimation-using-machine-learning-in-python/> (accessed Dec. 09, 2021).
- [5] Y. Agrawal, Y. Shah, and A. Sharma, “Implementation of machine learning technique for identification of yoga poses,” *Proceedings - 2020 IEEE 9th International Conference on Communication Systems and Network Technologies, CSNT 2020*, pp. 40–43, 2020, doi: 10.1109/CSNT48778.2020.9115758.
- [6] E. Burns, “What Is Machine Learning and Why Is It Important?,” 2019. <https://searchenterpriseai.techtarget.com/definition/machine-learning-ML> (accessed Dec. 09, 2021).
- [7] Q. Innovation, “Why is Machine Learning Important and How Will It Impact Business - Quantilus,” *Quantilus Innovation*, 2020. <https://quantilus.com/why-is-machine-learning-important-and-how-will-it-impact-business/> (accessed Dec. 09, 2021).
- [8] A. Halder and A. Tayade, “Real-time Vernacular Sign Language Recognition using MediaPipe and Machine Learning,” *International Journal of Research Publication and Reviews*, vol. 2, no. 5, 2021.
- [9] M. Fourie and D. van der Haar, “A feature importance study in ballet pose recognition with openpose,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 12217 LNCS, pp. 243–254, 2020, doi: 10.1007/978-3-030-50334-5_16.
- [10] F. Rahutomo, P. Y. Saputra, and ..., “Implementasi Twitter Sentiment Analysis Untuk Review Film Menggunakan Algoritma Support Vector Machine,” *Jurnal Informatika ...*, 2018, [Online]. Available: <http://jip.polinema.ac.id/ojs3/index.php/jip/article/view/152>
- [11] D. T. Lukmana, S. Subanti, and Y. Susanti, “Analisis Sentimen Terhadap Calon Presiden 2019 Dengan Support Vector Machine Di Twitter,” *Seminar & Conference ...*, 2019, [Online]. Available: <http://jurnal.umt.ac.id/index.php/cpu/article/view/1693>
- [12] A. Nabillah, S. Alam, and M. G. Resmi, “Twitter User Sentiment Analysis Of TIX ID Applications Using Support Vector Machine Algorithm,” *RISTEC ...*, 2022, [Online]. Available: <https://journal.institutpendidikan.ac.id/index.php/ristec/article/view/1898>
- [13] R. Darmawan, I. Indra, and A. Surahmat, “Optimalisasi Support Vector Machine (SVM) Berbasis Particle Swarm Optimization (PSO) Pada Analisis Sentimen Terhadap Official Account Ruang Guru di Twitter,” *Jurnal Kajian Ilmiah*, 2022, [Online]. Available: <http://ejournal.uharajaya.ac.id/index.php/JKI/article/view/1130>
- [14] I. N. F. Astuti, I. Darmawan, and ..., “... Sentimen Pada Data Kuesioner Evaluasi Dosen Oleh Mahasiswa (edom) Prodi Sistem Informasi Telkom University Menggunakan Algoritma Support Vector Machine,” *eProceedings ...*, 2020, [Online]. Available: <https://openlibrarypublications.telkomuniversity.ac.id/index.php/engineering/article/view/12676>
- [15] A. P. Sembiring and S. Faza, “... SUPPORT VECTOR MACHINE (SVM): PREDIKSI TINGKAT KEJAHATAN BERDASARKAN ARTIKEL BERITA NASIONAL MENGGUNAKAN METODE SUPPORT ...,” *ABEC Indonesia*, 2021, [Online]. Available: <https://abecindonesia.org/proceeding/index.php/abec/article/view/197>
- [16] L. Hakim, “Penentuan Tes Kepribadian Calon Mahasiswa Berdasarkan Sidik Jari Menggunakan Minutiae dan Support Vector Machine,” *Journal of Applied Informatics and Computing*, 2020, [Online]. Available: <https://103.209.1.42/index.php/JAIC/article/view/1883>
- [17] N. W. Utami and N. N. Saptiari, “Penerapan Data Mining Untuk Klasifikasi Penyebab Kematian Menggunakan Algoritma Support Vector Machine,” *Jurnal Ilmiah Ilmu Terapan ...*, 2020, [Online]. Available: <https://online-journal.unja.ac.id/JIITUJ/article/view/13268>
- [18] E. ERYKSON, ... SENTIMEN PUBLIK PENANGANAN COVID-19 DI KABUPATEN PONOROGO MENGGUNAKAN ALGORITMA SUPPORT VECTOR MACHINE (SVM) DAN RANDOM ... *eprints.umpo.ac.id*, 2022. [Online]. Available: <http://eprints.umpo.ac.id/8924/>
- [19] A. Setiyono and H. F. Pardede, “Klasifikasi Sms Spam Menggunakan Support Vector Machine,” *Jurnal Pilar Nusa Mandiri*, 2019, [Online]. Available: <http://ejournal.nusamandiri.ac.id/index.php/pilar/article/view/693>
- [20] A. S. Handayani, S. Soim, T. E. Agusdi, and ..., “Klasifikasi Kualitas Udara Dengan Metode Support Vector Machine,” *Jurnal Informatika ...*, 2020, [Online]. Available: <http://e-journal.stmiklombok.ac.id/index.php/jire/article/view/303>