

**Representasi Tahapan Gerakan dengan Ontologi dan Klasifikasi
Gerakan Tarian menggunakan *K-Nearest Neighbors* pada Tari
Tradisional Jaipong Oray Welang**



TUGAS AKHIR

**Diajukan untuk Memenuhi Persyaratan
Sarjana Strata Satu (S-1)**

Disusun oleh:

Dwita Wulandari (0102516009)

**PROGRAM STUDI INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS AL AZHAR INDONESIA
JAKARTA**

2021

LEMBAR PENYATAAN KEASLIAN

Yang bertanda tangan dibawah ini:

Nama : Dwita Wulandari
NIM : 0102516009
Fakultas : Sains dan Teknologi
Program Studi : Informatika
Judul Tugas Akhir : Representasi Tahapan Gerakan dengan Ontologi
dan Klasifikasi Gerakan Tarian menggunakan K-
Nearest Neighbors pada Tari Tradisional Jaipong
Oray Welang

Menyatakan bahwa sesungguhnya tugas akhir yang akan dibuat ini belum pernah diajukan pada Perguruan Tinggi atau Instansi manapun untuk tujuan memperoleh gelar akademik tertentu. Saya juga menyatakan bahwa laporan Tugas Akhir ini benar-benar hasil karya saya sendiri dan tidak mengandung bahan-bahan yang pernah ditulis atau diterbitkan oleh pihak lain kecuali sebagai bahan rujukan yang dinyatakan dalam naskah.

Jakarta, 18 Juni 2021



Dwita Wulandari
NIM. 0102516009

LEMBAR PENGESAHAN

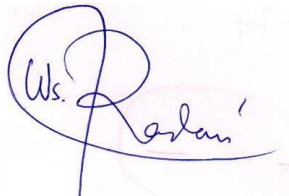
Yang bertanda tangan dibawah ini menyatakan bahwa Tugas Akhir dari mahasiswa:

Nama : Dwita Wulandari
NIM : 0102516009
Fakultas : Sains dan Teknologi
Program Studi : Informatika
Judul Tugas Akhir : Representasi Tahapan Gerakan dengan Ontologi dan Klasifikasi Gerakan Tarian menggunakan K-Nearest Neighbors pada Tari Tradisional Jaipong Oray Welang

Telah diperiksa dan disetujui oleh:

Jakarta, 18 Juni 2021

Menyetujui,
Pembimbing I

A handwritten signature in blue ink, appearing to read 'Ws. Pradani', is written over a faint, light-colored rectangular stamp.

Ir. Winangsari Pradani, M.T

Mengetahui,
Ketua Program Studi Informatika
Fakultas Sains dan Teknologi

A handwritten signature in blue ink, appearing to read 'Riri Safitri', is written over a faint, light-colored rectangular stamp.

Riri Safitri, S.Si, M.T

LEMBAR PERSETUJUAN PENGUJI TUGAS AKHIR

	Tanda Tangan	Tanggal
Ketua Sidang: Riri Safitri, S.Si., M.T		2 Juli 2021
Penguji I: Prof. Dr. Rahmat Budiarto	<i>Rahmat Budiarto</i>	1 Juli 2021
Penguji II: Dr. Toto Haryanto, S.Kom, M.Si		2 Juli 2021
Pembimbing I: Ir. Winangsari Pradani, M.T		1 Juli 2021
Dinyatakan lulus pada tanggal:	24 Juni 2021	

KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarakatuh.

Puji syukur kehadiran Allah Subhanahu Wa Ta'ala karena hanya dengan rahmat dan hidayah-Nya penulis dapat mencapai dan menyelesaikan Tugas Akhir yang berjudul “Representasi Tahapan Gerakan dengan Ontologi dan Klasifikasi Gerakan Tarian menggunakan *K-Nearest Neighbors* pada Tari Tradisional Jaipong Oray Welang”. Penyusunan laporan ini dibuat untuk memenuhi salah satu syarat kelulusan dan mendapatkan gelar kesarjanaan Strata Satu (S1) pada Program Studi Informatika Fakultas Sains dan Teknologi Universitas Al Azhar Indonesia. Kelancaran dan keberhasilan dalam penyusunan Tugas Akhir ini tidak lepas dari bantuan, dukungan, dan do'a dari berbagai pihak dengan tulus dan ikhlas memberikan semangat, serta masukan yang sangat berarti untuk penulis. Oleh karena itu, dalam kesempatan ini dengan kerendahan hati penulis mengucapkan terima kasih kepada:

1. Allah Subhanahu Wa Ta'ala atas segala limpahan rahmat, kemudahan, kelancaran dalam penyusunan Tugas Akhir ini.
2. Kedua Orang Tua, kakak, dan keponakan penulis. Terima kasih atas kasih sayang, dukungan, do'a, penyemangat dikala penulis merasa jenuh dan selalu ada untuk penulis.
3. Ibu Ir. Winangsari Pradani, M.T. selaku dosen pembimbing Tugas Akhir yang selalu memberikan ilmu, masukan, semangat dan dukungan demi terciptanya Tugas Akhir ini, dan sudah mengorbankan waktu, pikiran, dan tenaga untuk membimbing penulis hingga akhir. Mohon maaf apabila penulis sebagai mahasiswa bimbingan masih banyak kekurangan, namun penulis sangat bersyukur dapat memiliki pembimbing seperti Ibu.
4. Ibu Deviana, Ibu Desi, Valerie, dan pihak LKP Sanggar Tari Indra Kusuma yang sudah mengizinkan dan membantu mengumpulkan data Tugas Akhir.
5. Ibu dan Bapak Dosen Informatika yang sudah memberikan penulis banyak sekali ilmu, pengalaman yang sangat berharga bagi masa depan penulis.

6. Adit dan Tyas yang sudah menemani, memberikan dukungan, mendengar keluh dan kesah, dan masukan-masukan untuk penulis.
7. Okta, Desi, Dita, Dhean, Tasya, Mamih (Jelita), Dennisa, Olip, Zidna, yang sudah memberikan dukungan, do'a, dan kebersamaan menjalani masa-masa kuliah penulis.
8. Teman-teman Informatika 2016 dan seluruh keluarga besar Informatika, baik senior maupun junior yang sudah melewati masa-masa kuliah bersama-sama, terutama pada saat non-akademis atau organisasi.
9. Teman-teman penulis DINER, keluarga cemara, keluarga ROH24, yang sudah memberikan dukungan penulis.

Semoga Allah Subhanahu Wa Ta'ala membalas semua do'a, kebaikan dan ilmu, serta semangat yang telah diberikan kepada penulis. Penulis menyadari bahwa penulisan laporan ini masih memiliki banyak kekurangan. Oleh karena itu, penulis mengharapkan masukan, baik saran maupun kritik yang dapat membangun penulis dari semua pihak. Semoga laporan Tugas Akhir ini bermanfaat bagi penulis maupun pembaca.

Wassalamu'alaikum Warahmatullahi Wabarakatuh.

Jakarta, 18 Juni 2021

Dwita Wulandari

RIWAYAT HIDUP



Nama : Dwita Wulandari
NIM : 0102516009
Agama : Islam
Alamat : Jl. Cendana 19, Gg. Mushola Al-Hidayah,
Jakasampurna, Bekasi Barat.
Email : dwitawulandari@if.uai.ac.id

Penulis dilahirkan di Bekasi, pada tanggal 25 Agustus 1998 dan anak kedua dari ayah bernama Abad Syahbudin dan ibu Mar'atus Sholikhah. Penulis melanjutkan Pendidikan di Universitas Al Azhar Indonesia dengan pilihan Program Studi Informatika melalui jalur tes regular. Selama perkuliahan penulis aktif pada organisasi intra universitas, pengurus Himpunan Mahasiswa Informatika (HMIF) selama dua periode 2017/2019 dan 2019/2020.

ABSTRAK

DWITA WULANDARI. **Representasi Tahapan Gerakan dengan Ontologi dan Klasifikasi Gerakan Tarian menggunakan K-Nearest Neighbor pada Tari Tradisional Jaipong Oray Welang**

Indonesia memiliki beragam warisan budaya dari yang benda dan takbenda, termasuk tari Sunda yang memiliki banyak tarian dengan gerakan dan latar belakang yang berbeda-beda pula, termasuk Tari Jaipong yang merupakan tarian paling populer dan memiliki berbagai macam tarian, seperti jaipong Keser Bojong, jaipong Oray Welang, jaipong Banda Urang, dan masih banyak lagi. Penelitian ini akan difokuskan ke satu macam tari jaipongan, yaitu jaipong Oray Welang yang merupakan tarian pola dasar dari jaipongan, dan melakukan pencatatan isi dari gerakan-gerakan tarian tersebut dan direpresentasikan dalam sebuah bentuk ontologi, lalu *frame-frame* gambar dari data video tarian akan dilatih untuk klasifikasi gambar dengan menggunakan pembelajaran mesin atau *machine learning*. Untuk membangun ontologi dilakukan menggunakan *library owlready2*, aplikasi *protege*, dan *SPARQL query* untuk menguji *Ontology Web Language (OWL)*. Lalu klasifikasi gambar akan dilakukan perbandingan dengan data tanpa ekstraksi fitur *Histogram of Oriented Gradient (HOG)* dan data dengan ekstraksi fitur HOG, dan menggunakan model KNN. Hasil evaluasi tertinggi pada data tanpa ekstraksi fitur HOG dengan parameter *Manhattan metric*, *n_neighbors=3*, yaitu nilai *precision* 84%, *recall* 84%, *F1-score* 84%, dan akurasi 85%. Sedangkan pada data dengan ekstraksi fitur HOG dengan parameter *Euclidean metric*, *n_neighbors=3*, yaitu nilai *precision* 91%, *recall* 90%, *F1-score* 90%, dan akurasi 90%.

Kata Kunci: Tari Jaipong Oray Welang, *Ontology Web Language*, Klasifikasi Gambar, Ekstraksi Fitur HOG, *K-Nearest Neighbors*

ABSTRACT

DWITA WULANDARI. *Representation of Dance Movements with Ontology and Dance Movements Classification using K-Nearest Neighbor in Jaipong Oray Welang Traditional Dance.*

Indonesia has a variety of cultural heritages from tangible and intangible, including Sundanese dance which has a lot of dances with different movements and backgrounds. Sundanese dance has Jaipong dance which is the most popular of tradisional dance which has even been modernization and has various types, such as Jaipong Keser Bojong, Jaipong Oray Welang, Jaipong Banda Urang, and many more. This research will focus on one kind of Jaipong dance, namely Jaipong Oray Welang which is a basic dance of Jaipong, and will be records the contents of these dance movements and represented in an ontology. Then the pictures from videos data will be trained for image classification using machine learning. To create the ontology is done using owlready2 library, protégé application, and SPARQL queries for the Ontology Web Language testing. This research then compares thr proposed ontology with and without HOG feature extraction, and uses k-NN model for dance movemens classification. Highest evaluation results with data without HOG feature extraction is using Manhattan metric and n_neighbors=3 for parameters with precision 84%, recall 84%, F1-score 84%, and accuracy 85%, then data with HOG feature extraction is using Euclidean metric and n_neighbors=3 for paramters with precision 91%, recall 90%, F1-score 90%, and accuracy 90%.

Keywords: *Jaipong Oray Welang Dance, Ontology Web Language, Image Classification, HOG Extraction Feature, K-Nearest Neighbors*

DAFTAR ISI

LEMBAR PENYATAAN KEASLIAN	ii
LEMBAR PENGESAHAN	iii
LEMBAR PERSETUJUAN PENGUJI TUGAS AKHIR	iv
KATA PENGANTAR	v
RIWAYAT HIDUP	vii
ABSTRAK.....	viii
ABSTRACT.....	ix
DAFTAR ISI.....	x
DAFTAR GAMBAR	xiii
DAFTAR TABEL.....	xv
BAB I PENDAHULUAN.....	16
1.1 Latar Belakang	16
1.2 Rumusan Masalah	18
1.3 Tujuan.....	19
1.4 Manfaat.....	19
1.5 Batasan Masalah.....	19
1.6 Metode Penelitian.....	19
1.7 Sistematika Pembahasan	21
BAB II LANDASAN TEORI.....	22
2.1 Tari Tradisional	22
2.1.1 Tari Jaipongan.....	22
2.3 Ontologi.....	27
2.4 OWL (<i>Ontology Web Language</i>)	29

2.5	<i>Owlready2</i>	31
2.6	RDF (<i>Resource Description Framework</i>)	31
2.7	Ekstraksi Fitur	32
2.8.	<i>K-Nearest Neighbor</i>	33
BAB III METODE PENELITIAN		39
3.1.	Pengumpulan Data	40
3.2.	Deskripsi Tarian	41
3.3.	Membangun Ontologi	41
3.3.1	Menentukan domain dan ruang lingkup ontologi	41
3.3.2	Mempertimbangkan ontologi yang sudah ada	42
3.3.3	Enumerisasi kata-kata penting dalam ontologi	42
3.3.4	Menentukan <i>class</i> dan hirarki dari sebuah <i>class</i>	42
3.3.5	Mendefinisikan <i>property</i> dari <i>class</i>	43
3.3.6	Mendefinisikan <i>facet</i> (batasan) dari <i>property</i>	43
3.3.7	Membuat <i>instance</i>	44
3.4.	Pengujian Ontologi.....	44
3.5.	Pembuatan Dataset	45
3.6.	<i>Data Preprocessing</i>	45
3.6.1.	<i>Extracting Frames</i>	45
3.6.2.	<i>Cropping</i>	45
3.6.3.	<i>Resizing</i>	46
3.7.	Ekstraksi Fitur	46
3.7.1.	<i>Histogram of Oriented Gradients</i>	46
3.8.	KNN (<i>K-Nearest Neighbor</i>) <i>Classification</i>	51
3.9.	Evaluasi	52
BAB IV IMPLEMENTASI DAN ANALISIS		55

4.1.	Deskripsi Tarian	55
4.2.	Pembangunan Ontologi	55
4.4.	<i>Preprocessing Data</i>	62
4.5.	Ekstraksi Fitur HOG.....	65
4.6.	Analisis Pengujian.....	66
4.7.	Evaluasi	71
BAB V KESIMPULAN DAN SARAN		75
5.1.	Kesimpulan.....	75
5.2.	Saran.....	75
DAFTAR PUSTAKA		77

DAFTAR GAMBAR

Gambar 2. 1. Gerakan Tari Jaipong Oray Welang (Sumber: Sanggar Tari Indra Kusuma).....	23
Gambar 2. 2. Contoh Penerapan Ontologi pada Indian Classical Dance [4]	29
Gambar 2. 3. Hubungan antar subkelas [18].....	30
Gambar 2. 4. Penulisan RDF	32
Gambar 2. 5. Contoh menentukan k dengan kasus sederhana (analyticsvidhya.com)	33
Gambar 2. 6. Menandakan k dengan berpusat pada data uji atau bintang ungu (analyticsvidhya.com)	34
Gambar 2. 7. Contoh penerapan k terhadap real data (analyticsvidhya.com)	35
Gambar 2. 8. Kurva Training Error Rate (analyticsvidhya.com)	35
Gambar 2. 9. Kurva Validation Error Rate (analyticsvidhya.com)	36
Gambar 2. 10. Contoh Dataset dan Hasil Prediksi Gambar Uji (cs231n.github.io)	36
Gambar 2. 11. Perhitungan L1 distance (cs231n.github.io)	37
Gambar 3. 1. Gambaran Besar Penelitian	39
Gambar 3. 2. Metode Penelitian	40
Gambar 3. 3. Preprocessing gambar (Cropping dan Resize)	47
Gambar 3. 4. Contoh block cell yang bergeser di setiap piksel gambar [26]	48
Gambar 3. 5. (a) Grid yang menunjukkan posisi blok yang ditempatkan pada rata-rata gambar yang dihasilkan. (b) blok yang teridentifikasi gambar yang diinginkan [26].....	49
Gambar 3. 6. Pengelompokan blok untuk melakukan normalisasi [26]	51
Gambar 4. 1. Class Ontologi pada Tahapan Gerakan Tarian.....	56
Gambar 4. 2. Object Property Ontology	57
Gambar 4. 3. Object Property hasGerakan pada domain Tarian	57
Gambar 4. 4. Graph Sub-Object property hasNextStep1 dan hasNextStep2.....	58
Gambar 4. 5. Data Property pada Ontologi.....	59
Gambar 4. 6. (a)Instance kelas Tarian, (b) Instance kelas Gerakan.....	60
Gambar 4. 7. Hasil query SPARQL gerakan-gerakan tari jaipong oray welang ...	61

Gambar 4. 8. Hasil query SPARQL Gerakan A hasNextStep Gerakan B	61
Gambar 4. 9. Hasil query SPARQL data setiap Gerakan	62
Gambar 4. 10. Hasil Video Setelah di Extract	63
Gambar 4. 11. Contoh cropping gambar.....	63
Gambar 4. 12. Gambar setelah dilakukan cropping dan resize ke seluruh gambar/data.....	64
Gambar 4. 13. Contoh gambar yang sudah di Ekstraksi Fitur menggunakan HOG	65
Gambar 4. 14. Grafik (a) menunjukkan perbedaan nilai akurasi dan f1 metric manhattan dan euclidean pada data HOG dan data tanpa HOG, dan Grafik (b) menunjukkan perbedaan nilai akurasi data latih dan data validasi metric Manhattan dan Euclidean pada data HOG dan Tanpa HOG	69
Gambar 4. 15. Nilai error pada data latih dan validasi manhattan dan Euclidean dengan data HOG dan Tanpa HOG	70

DAFTAR TABEL

Tabel 2. 1. Nama-Nama Tari Jaipong dan Contoh Cuplikan Gerakan Berdasarkan Tingkatan	24
Tabel 2. 2. Definisi Ontologi dari Berbagai Komunitas dan Filsafat	28
Tabel 3. 1. Confusion Matrix	52
Tabel 4. 1. Rancangan Object Property	57
Tabel 4. 2. Rancangan Data Property Class Gerakan	59
Tabel 4. 3. Jumlah Data Setiap Kelas	64
Tabel 4. 4. Pembagian Data Latih, Validasi, dan Uji.....	65
Tabel 4. 5. Nilai pengukuran menggunakan parameter HOG yang berbeda-beda	67
Tabel 4. 6. Nilai pengukuran pada data HOG dan tanpa HOG dengan $n = 3$ s/d 13	68
Tabel 4. 7. Nilai pengukuran pada data tanpa HOG	71
Tabel 4. 8. Nilai pengukuran pada data HOG.....	71
Tabel 4. 9. Confusion Matrix Data Tanpa HOG.....	72
Tabel 4. 10. Confusion Matrix Data HOG.....	72

BAB I

PENDAHULUAN

1.1 Latar Belakang

Seni dan budaya adalah hal yang sangat melekat dengan Indonesia serta harus dijaga dan dilestarikan. Melakukan pencatatan isi gerakan tarian tradisional Indonesia adalah termasuk menghargai, menjaga, sekaligus melestarikan seni dan budaya Indonesia. Saat ini informasi di internet tentang isi dari gerakan tari tradisional masih sangat sedikit sehingga untuk mendapatkan informasi penuh tentang gerakan suatu tarian adalah dengan melakukan wawancara dengan seseorang yang profesional di bidangnya, yaitu melalui sanggar tari. Tari jaipong adalah salah satu tari tradisional Jawa Barat yang sangat terkenal di kalangan masyarakat, namun untuk informasi tentang isi dari gerakan tersebut masih sangat sedikit.

Pelestarian warisan budaya dan memastikan aksesibilitasnya yang lama sudah mendapatkan dorongan atau bantuan dari multimedia digital. Seperti penelitian [1] yang menjelaskan bahwa melestarikan sumber daya warisan budaya benda seperti monumen, patung, dan lainnya dengan teknik digitalisasi dan pemodelan 2D dan 3D. Sedangkan warisan budaya takbenda biasanya dilestarikan oleh orang dengan cara turun-temurun oleh keluarga dari seniman tersebut dan biasanya tidak berdokumen. Warisan budaya tersebut seperti musik, tari, syair, sejarah, dan warisan budaya takbenda lainnya. Menurut UNESCO warisan budaya takbenda mencakup tradisi yang diwarisi dari nenek moyang dan diwariskan kepada keturunannya. Tarian rakyat atau tari tradisional adalah bagian penting dari warisan budaya takbenda, sehingga ada kebutuhan untuk mempromosikan dan melindunginya untuk diwariskan kepada generasi berikutnya [2]. Indonesia memiliki beragam warisan budaya dari yang benda hingga takbenda, termasuk tari Sunda yang memiliki berbagai jenis tarian dengan gerakan dan latar belakang yang berbeda-beda pula.

Tari Sunda merupakan tradisi tarian berupa ritual, ekspresi seni, serta hiburan di kalangan masyarakat Sunda, Jawa Barat. Tari Sunda juga memiliki ciri khas dalam gerakannya, yaitu ekspresif, ceria, dan dinamis. Tari Sunda memiliki

berbagai jenis tarian, termasuk Tari Jaipong yang merupakan jenis tarian paling populer di kalangan masyarakat luas. Jaipongan memiliki berbagai macam gerakan didalamnya, seperti gerak bukaan, gerak pencugan, gerak nibakeun, dan gerak mincid [3]. Dengan banyaknya macam dan jenis tari jaipong, untuk penelitian kali ini akan mengambil satu jenis tari jaipong, yaitu tari jaipong Oray Welang yang akan dilakukan pencatatan gerakan-gerakan tersebut sebagai bentuk pelestarian warisan budaya takbenda khususnya tari tradisional. Setelah itu *frame-frame* gambar yang terdapat pada video-video tarian yang ada akan lebih baik jika di *retrieve* dengan melatih data-data tersebut menggunakan algoritma pembelajaran mesin atau *machine learning* sebagai bentuk proses pengenalan gambar. Ide tersebut muncul dari pembuatan *semantic web* yang dijelaskan pada penelitian Malik, Ghosh, dan Chaudhury [4] yang membahas tentang pengembangan aplikasi *browsing* untuk akses semantik ke koleksi warisan video tarian India.

Semantic Web diperkenalkan pertama kali oleh Tim Berners Lee pada tahun 1996. Tujuan dari *semantic web* adalah untuk mengakses data dari sebuah *Web* yang terstruktur dengan baik [5]. *Semantic web* menyediakan hubungan bagi mesin (*search engine*) dan suatu pengkodean (SPARQL, yang menghubungkan RDF/XML dengan *user interface* atau *search engine*) untuk menginterpretasikan sebuah pengetahuan menjadi proses otomatisasi. Domain representasi pengetahuan menggunakan ontologi adalah untuk membuat kosakata umum sebagai kata kunci yang menggambarkan tentang pengetahuan domain tertentu secara rinci. Proses otomatisasi yang dilakukan dapat berupa aplikasi *browsing* ataupun *search engine*. *Semantic Web* yang dijelaskan pada penelitian [4] adalah untuk menghubungkan pengetahuan warisan budaya tarian klasik India dan rekaman video tarian klasik India. Dilakukan dengan cara menyediakan data konseptual (ontologi) di tingkat pengetahuan (gerakan tarian) dan data multimedia/rekaman sebuah tarian di tingkat fitur dengan melakukan pengklasifikasian *frame* dari video.

Penelitian ini merupakan sebagian pekerjaan dari target yang lebih besar, yaitu aplikasi *search engine* atau *information retrieval* untuk Tari Tradisional Indonesia. Di sini, akan dibangun sebuah ontologi tarian Jaipong sebagai struktur data pengetahuan Tarian. Ontologi ini akan menyimpan data tarian dalam bentuk RDF/XML. Selain membangun ontologi, pada penelitian ini juga akan dibangun

model klasifikasi gambar gerakan tarian. Hasil dari pembangunan model klasifikasi berguna dalam dua hal, yang pertama, data gambar gerakan yang telah melalui tahap *preprocessing* akan tersimpan untuk melengkapi data ontologi. Hal yang kedua, *query* atau pertanyaan mengenai sebuah gambar dari *user* akan terjawab dengan acuan gambar yang telah tersimpan. Di masa mendatang, diharapkan data ontologi maupun data gambar dapat ditambahkan dengan tarian lainnya sehingga pencarian data tarian menjadi lebih lengkap dan luas lagi.

Proses ontologi akan dilakukan dengan menggunakan kode ontologi berbasis *python*, yaitu *owlready2* dan memvisualkannya menggunakan aplikasi *protegé* [4], lalu melakukan klasifikasi pada gambar gerakan suatu tarian untuk mempermudah mengenali dan mengetahui nama gerakan setiap gambar yang dilatih. Terdapat beberapa penelitian klasifikasi gambar menggunakan algoritma *machine learning* ataupun model *neural network*. Penelitian [5] melakukan *Dance Action Categorization* dan *Dance Posture Recognition* menggunakan *spatio-temporal features* untuk *Dance Action Categorization* dan *SIFT Features* untuk *Dance Posture Recognition*, dan juga SVM sebagai *classifier*. Penelitian lain yang dilakukan oleh Priya dan Arulselvi [7] yang membahas tentang klasifikasi gerak *Bharathnatyam* dengan 10 kelas dan menghasilkan akurasi sebesar 65% menggunakan model *Convolutional Neural Network* dan VGG16. Adapula penelitian Kumar dan Kishore [8] dengan mengklasifikasi tarian *Mudra* menggunakan *Histogram of Oriented Gradient (HOG) feature extraction* dan SVM *classifier* yang menghasilkan akurasi sebesar 90%.

Berdasarkan latar belakang tersebut, penulis mengangkat tema dan judul **“Representasi Tahapan Gerakan dengan Ontologi dan Klasifikasi Gerakan Tarian menggunakan *K-Nearest Neighbor* pada Tari Tradisional Jaipong Oray Welang”**. Representasi tahapan Gerakan menggunakan model ontologi, dan klasifikasi menggunakan ekstraksi fitur *Histogram of Oriented Gradient* dan *KNNclassifiers*. Dataset yang akan digunakan yaitu data berupa *frame-frame* gambar dari video tari jaipong Oray Welang.

1.2 Rumusan Masalah

Berdasarkan uraian latar belakang, maka didapatkan rumusan masalah seperti:

1. Bagaimana teknologi digital dapat mendukung upaya pelestarian budaya tarian klasik?
2. Bagaimana menerapkan klasifikasi gerakan tari jaipong Oray Welang menggunakan metode KNN?

1.3 Tujuan

Penelitian yang dilakukan memiliki beberapa tujuan. Tujuan dari penelitian yang ingin dicapai sebagai berikut:

1. Membangun model ontologi untuk tahapan gerakan-gerakan tarian pola dasar jaipong Oray Welang.
2. Melakukan investigasi keberhasilan algoritma KNN dalam pengklasifikasian gerakan tarian pola dasar jaipong Oray Welang.

1.4 Manfaat

Manfaat dari penelitian ini adalah untuk pelestarian warisan budaya dengan memanfaatkan teknologi digital, dan metode yang sudah dilakukan pada penelitian ini dapat dipakai untuk tarian lainnya. Manfaat bagi penulis adalah untuk menerapkan ilmu yang sudah penulis pelajari selama masa perkuliahan dan pengembangan pengetahuan penulis pada mata kuliah terkait.

1.5 Batasan Masalah

Batasan masalah pada penelitian ini, yaitu:

1. Data yang digunakan adalah informasi tari jaipongan dari sanggar tari, serta pengumpulan data video dan gambar yang didapat dari youtube dan sanggar tari.
2. Berfokus pada satu tarian pola dasar jaipong Oray Welang.
3. Hanya menerapkan algoritma KNN dan akurasi, *precision*, *recall*, *F1-Score* sebagai metrik evaluasi.

1.6 Metode Penelitian

Pada penelitian ini digunakan beberapa metode penelitian, yaitu:

1. Studi Literatur

Studi literatur dilakukan untuk mengakses, melampirkan serta media untuk pembelajaran yang berkaitan dengan tugas akhir melalui buku, jurnal dan media *online*.

2. Wawancara

Penelitian ini akan dilakukan wawancara dengan pihak sanggar untuk mencari informasi tentang tari jaipong Oray Welang, yang meliputi sejarah, jenis tari jaipongan, nama gerakan, serta gerakan-gerakan tari jaipong Oray Welang dari awal sampai akhir dengan acuan dari salah satu data video.

3. Pengumpulan Data

Tahap pengumpulan data untuk pembuatan ontologi dilakukan validasi informasi tentang tahapan-tahapan gerakan tarian di Sanggar Tari Indra Kusuma yang berada di Jatiwaringin, Bekasi dan dilakukan dengan narasumber ahli tari jaipongan Ibu Deviana Nur dan Ibu Dewi Nur. Sedangkan data yang dijadikan untuk penelitian pada tahap klasifikasi adalah dengan mengumpulkan data video dari Youtube dan pengambilan video dan gambar di sanggar.

4. *Preprocessing* Data

Data video akan dilakukan *extracting frames* dari data video menjadi potongan-potongan gambar menggunakan *OpenCV* (cv2.VideoCapture), lalu pelabelan data. Setelah itu dilakukan *cropping* dan *resizing* gambar ke seluruh data dengan ukuran 280x450 piksel.

5. Perancangan Ontologi

Pada tahap ini akan dilakukan perancangan dari data yang ada sehingga mendapat suatu deskripsi pada setiap gerakan untuk dapat direpresentasikan dengan ontologi.

6. Implementasi

Tahap implementasi adalah membangun ontologi dan melakukan klasifikasi pada data yang sudah di *preprocessing* sebelumnya menggunakan algoritma *machine learning*, yaitu KNN.

7. Evaluasi hasil

Tahap evaluasi hasil ini adalah memeriksa hasil model ontologi apakah model yang dibangun sudah sesuai dengan tahapan pada informasi yang

sudah didapat sebelumnya, dan untuk klasifikasi memeriksa dan menyesuaikan nilai-nilai metrik evaluasi yang didapat sudah sesuai dengan ekspektasi dan juga lebih baik atau tidak dibandingkan dengan penelitian-penelitian acuan pada tugas akhir ini.

1.7 Sistematika Pembahasan

Untuk memudahkan dalam penyusunan tugas akhir, maka penyusun membuat sistematika dalam 5 bab yaitu:

1. BAB I : Pendahuluan

Bab ini menggambarkan tentang garis besar pembuatan tugas akhir, yang berisi latar belakang masalah, rumusan masalah, tujuan penulisan, batasan masalah, metodologi penelitian, serta sistematika penulisan.

2. BAB II : Landasan Teori

Bab ini menerangkan teori-teori yang berkaitan dengan topik utama dan menjadi acuan untuk penelitian tugas akhir.

3. BAB III : Metodologi Penelitian

Bab ini menerangkan sudut pandang permasalahan dan analisis tahapan-tahapan dalam melakukan penelitian tugas akhir.

4. BAB IV : Pembahasan

Bab ini akan membahas analisis proses hasil implementasi dan pengujian terhadap permasalahan yang ada.

5. BAB V : Penutup

Bab ini memuat kesimpulan serta saran-saran selama penulis melakukan penelitian tugas akhir dan juga acuan untuk pengembangan selanjutnya.

BAB II

LANDASAN TEORI

2.1 Tari Tradisional

Tari merupakan gerak tubuh yang dilakukan berirama di tempat dan waktu tertentu dengan maksud untuk mengungkapkan perasaan, rasa syukur, atau menceritakan sebuah kehidupan pada acara tertentu. Tari tradisional adalah suatu tarian yang tumbuh, tercipta, dan berkembang di suatu daerah yang dilakukan secara turun temurun oleh masyarakatnya. Tari tradisional memiliki makna cerita yang bergantung dengan adat istiadat di daerah terciptanya dan memiliki pedoman yang luas [10]. Tari Tradisional Indonesia mencerminkan kekayaan budaya Indonesia yang beraneka ragam. Banyak sekali tradisi seni tari, seperti tarian Bali, tarian Melayu, tarian Jawa, tarian Sunda, tarian Sumatera, tarian Aceh, tarian Palembang, dan masih banyak tradisi seni tari lainnya. Dari setiap daerah memiliki banyak lagi macam dan jenis tarian yang di setiap tarian memiliki fungsi dan tujuan yang berbeda-beda. Terutama daerah Jawa Barat memiliki banyak sekali tarian, dan tarian yang sangat terkenal di kalangan masyarakat Indonesia yaitu tari jaipongan.

2.1.1 Tari Jaipongan

Tari Jaipong atau Jaipongan adalah salah satu tari daerah yang berasal dari Bandung, Jawa Barat. Tari Jaipong merupakan tarian yang lahir dari budaya Sunda. Tarian ini diciptakan oleh Gugum Gumbira dan H. Suanda [11]. Gerakan-gerakan tari Jaipong adalah gabungan dari beberapa kesenian tradisional, seperti Wayang Golek, Pencak Silat dan Ketuk Tilu. Karakteristik tarian ini sangat enerjik dan unik, dan dalam setiap pementasannya akan diiringi oleh musik tradisional bernama degung. Selain enerjik, tarian ini juga dilakukan dalam suasana ceria dan humoris sehingga sangat menghibur penonton. Tari Jaipong awalnya hanya tari hiburan bagi rakyat biasa, seiring berjalannya waktu tari Jaipong saat ini disebut sebagai jenis kesenian tari tersendiri di Jawa Barat dan juga menjadi tarian yang sering ditampilkan dalam acara-acara penting, seperti menjadi tarian untuk menyambut tamu negara yang berkunjung dan juga menjalankan misi kesenian ke mancanegara.







Gambar 2. 1. Gerakan Tari Jaipong Oray Welang (Sumber: Sanggar Tari Indra Kusuma)



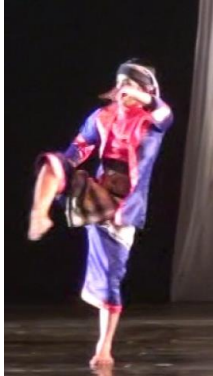

Sejarah singkat dari Jaipongan ini berawal pada tahun 1976, yaitu Haji Suanda melakukan inovasi dengan mencampurkan beberapa gerakan kesenian yang beliau kuasai, terutama Pencak Silat, Wayang Golek dan Ketuk Tilu. Hasilnya dari penggabungan tersebut maka lahirlah karya seni unik yang disukai masyarakat, dimana pada saat itu pertunjukan yang beliau gelar diberi nama Ketuk Tilu. Kesenian itu disebut memiliki sifat hero, demokratis, erotis, dan akrobatik. Ketika Gugum Gumbira telah menguasai tarian Ketuk Tilu tersebut, beliau kemudian mengemas ulang gerakan-gerakan pada tarian tersebut sehingga tercipta tari jaipong yang awalnya bernama Ketuk Tilu Perkembangan [12].


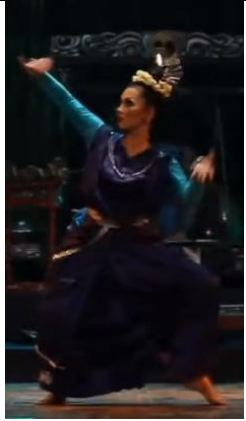

Jaipong tercipta pada saat seorang kendang menyebutkan jaipong dari kata “*blaktingpong*”. Inti dari tari jaipong adalah gerak, yaitu bagian kaki, tubuh, tangan, dan kepala bergerak harmonis, karena tarian ini diciptakan sebagai tarian pergaulan dan pertunjukan maka mata penari harus fokus dan selalu memandang ke depan atau penonton sehingga tercipta komunikasi secara harmonis antara penari dan penontonnya [13]. Menurut guru tari Sanggar Tari Indra Kusuma, jaipong atau jaipongan memiliki tingkatan tarian yang berbeda-beda dari mulai tingkatan dasar, madya, terampil, hingga mahir. Perbedaan tingkatan ini memiliki isi dan ciri tarian yang berbeda-beda pula, jika pada tarian tingkat dasar dan madya memiliki isi dan ciri gerakannya tidak berubah karena mengandung gerakan-gerakan dasar pada tari jaipong. Berbeda dengan tingkat terampil dan mahir yang memiliki isi dan ciri gerakannya sudah sangat variatif, karena setiap jenis tariannya memiliki variasi


gerakan yang berbeda-beda, bahkan untuk satu jenis tarian di setiap sanggar atau penari ahli bisa berbeda gerakan-gerakan dan koreografinya.

Tabel 2. 1. Nama-Nama Tari Jaipong dan Contoh Cuplikan Gerakan Berdasarkan Tingkatan

Nama Tarian	Tingkatan	Contoh Gerakan
Oray Welang	Dasar	
Keser Bojong	Dasar	
Banda urang	Dasar	
Toka-Toka	Dasar	

Nama Tarian	Tingkatan	Contoh Gerakan
Sonteng	Dasar	
Rendeng Bojong	Madya	
Pencug	Madya	
Kawung Anten	Madya	

Nama Tarian	Tingkatan	Contoh Gerakan
Setrasari	Madya	
Rawayan	Madya	
Mojang Priangan	Terampil	

Nama Tarian	Tingkatan	Contoh Gerakan
Genjring Party	Mahir	
Aduh Manis	Mahir	
Ronggeng Nyentrik	Mahir	

Dari berbagai tingkatan dan tarian-tarian yang terdapat pada Tabel 2.1., penelitian ini akan difokuskan dengan satu jenis tarian pada tingkatan dasar tarian, yaitu tari jaipong Oray Welang, yang merupakan lagu dan tarian dasar atau tarian pertama yang ditemukan oleh Gugum Gumbira karena Ketuk Tilu adalah tarian yang masih hidup saat itu.

2.3 Ontologi

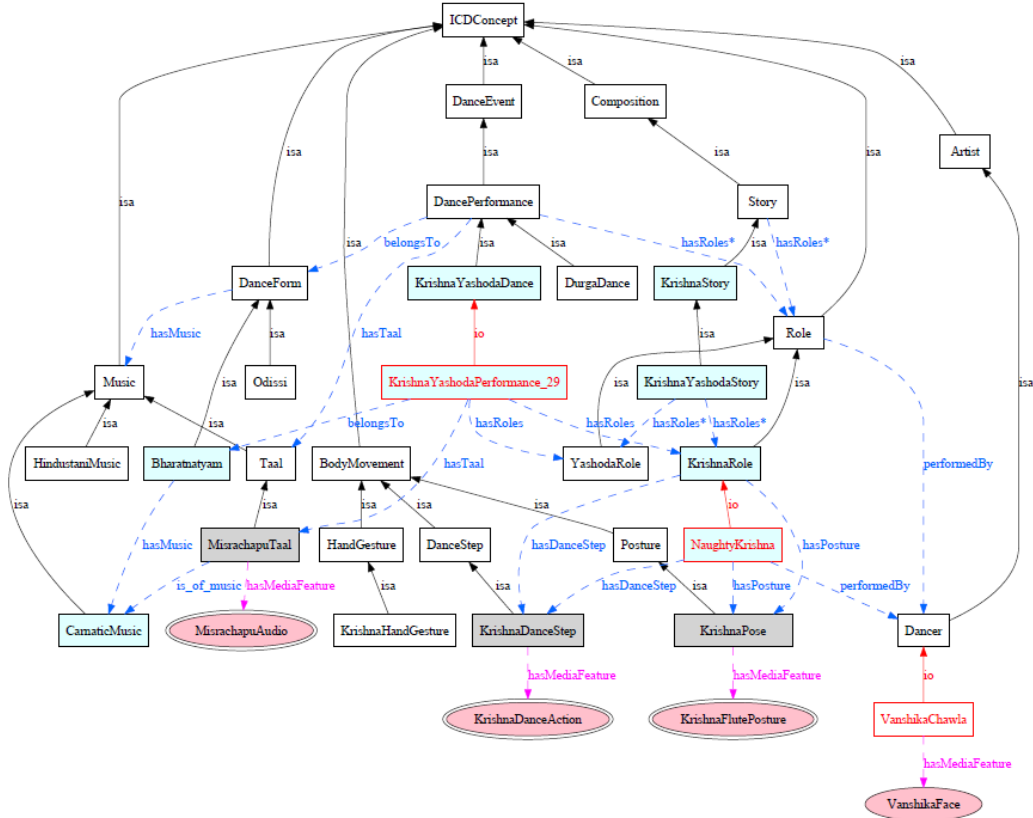
Ontologi memiliki arti yang berbeda-beda oleh beberapa komunitas dan filsafat, seperti berikut:

Tabel 2. 2. Definisi Ontologi dari Berbagai Komunitas dan Filsafat

Nama dan Penelitian	Istilah Penting	Deskripsi dan Definisi
S. Studer, V.R. Benjamins, dan D. Fensel [14]	Konseptualisasi, formal, dan eksplisit	“Konseptualisasi” mengacu pada model abstrak dari beberapa fenomena di dunia yang mengidentifikasi konsep yang relevan dari fenomena tersebut. “Formal” berarti ontologi yang harus bisa dibaca oleh mesin. “Eksplisit” yang mengacu pada fakta bahwa jenis konsep yang digunakan dan batasan penggunaannya harus didefinisikan secara eksplisit.
		Ontologi merupakan konsep abstrak dari fenomena yang didapat harus bisa dibaca secara umum dan menjadi konsep yang relevan yang mengacu pada fakta dengan jenis konsep dan batasan penggunaannya didefinisikan secara eksplisit.
S. Abburu, G.S. Babu [15]	-	Ontologi adalah suatu teori yang menjelaskan tentang makna, <i>property</i> , dan relasi dari suatu objek yang mungkin terjadi pada suatu domain.
H.Li, W.Li, Q.Cai, dan H.Li [16]	-	Ontologi merupakan sebuah teknologi yang efektif yang memungkinkan integrasi dengan sumber daya terkait dan menghindari informasi yang tidak relevan.
D.L. McGuinness dan F.Van Harmelen [17]	-	Ontologi adalah grafik khusus yang menggambarkan entitas domain dan sifatnya, serta hubungan antara keduanya.

Secara umum ontologi merupakan suatu teori yang disusun dalam serangkaian konsep yang menjelaskan tentang makna dan relasi dalam bentuk formal dan bukan kata-kata. Ontologi dinyatakan dalam bentuk bahasa pemrosesan, yaitu RDF (*Resource Description Framework*) dan OWL (*Ontology Web Language*) yang menggunakan Bahasa XML (*Extensible Markup Language*) sebagai dasar sintaksnya. Adapun tahapan-tahapan dalam membangun sebuah ontologi [18] : (i) Menentukan domain dan ruang lingkup ontologi, (ii) Mempertimbangkan ontologi yang sudah ada, (iii) Enumerisasi kata-kata penting dalam ontologi, (iv) Menentukan *class* dan hirarki dari sebuah *class*, (v)

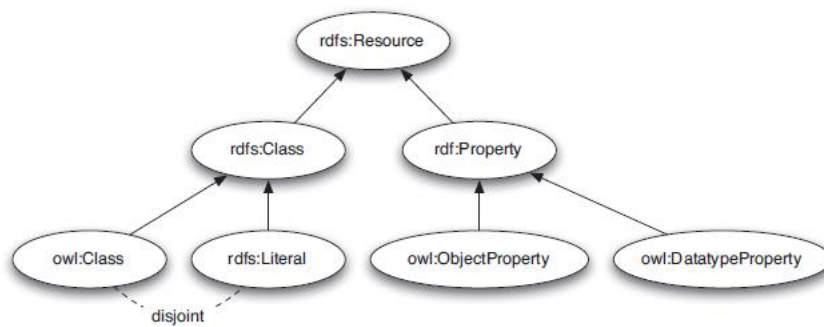
Mendefinisikan *property* dari *class*, (vi) Mendefinisikan *facet* (batasan) dari *property*, (vii) Membuat *instances*. Berikut adalah contoh ontologi pada *Indian Classical Dance* [5]:



Gambar 2. 2. Contoh Penerapan Ontologi pada Indian Classical Dance [4]

2.4 OWL (Ontology Web Language)

Ontology Web Language (OWL) adalah bahasa formal yang digunakan untuk ontologi pada *Semantic Web* yang dikembangkan oleh kelompok kerja Ontologi Web W3C [19] [20]. OWL didesain untuk merepresentasikan informasi tentang kategori suatu objek dan bagaimana objek tersebut saling berhubungan. Bahasa ini juga memberikan fasilitas yang sangat besar untuk mesin dari konten web yang didukung oleh XML, RDF, dan RDFS dengan memberikan kosakata tambahan untuk menggambarkan properti dan kelas, seperti hubungan antar kelas (*disjointness*), kardinalitas, kesetaraan, pengetikan properti yang lebih kaya, karakteristik properti (misalnya. Simetri), dan kelas yang disebutkan [19].



Gambar 2. 3. Hubungan antar subkelas [18]

Terdapat beberapa elemen dalam bahasa ontologi, yaitu (i) *Class*, merupakan kumpulan dari individual dengan karakteristik yang sama. (ii) *Instance* (Individual), merupakan komponen ontologi yang menggambarkan objek dalam sebuah domain dan setiap individualnya harus berada dalam kelas yang memiliki karakteristik yang sama. (iii) *Properties*, merupakan *binary relationship* yang menghubungkan individual satu dengan individual yang lain. *Properties* memiliki beberapa karakteristik [19] :

- (i) *Object Properties*, yaitu menghubungkan individu satu dengan individu lainnya. *Ontology Web Language* memungkinkan seseorang untuk menggunakan Skema XML *datatypes* untuk menunjukkan tipe literal atau menentukan rentang *datatype property*. Tipe data yang ditentukan pengguna dapat ditentukan dalam skema XML dan kemudian digunakan dalam ontologi OWL. Karena batasan dari semantik langsung, hanya tipe *functional property* yang dapat ditetapkan ke *datatype property* di OWL.
- (ii) *Functional Properties*, yaitu relasi dimana satu individual hanya bisa berhubungan dengan paling banyak satu individual saja.
- (iii) *Inverse Functional Properties*, yaitu relasi invers dari *functional property*.
- (iv) *Transitive Properties*, yaitu relasi antar individual, seperti jika individual a berhubungan dengan individual b, lalu individual b berhubungan dengan individual c, maka individual a juga memiliki hubungan dengan individual c.
- (v) *Symmetric Properties*, yaitu relasi timbal balik antar individual.
- (vi) *Asymmetric Properties*, yaitu relasi yang tidak bisa timbal balik antara dua individual.

- (vii) *Irreflexive Properties*, yaitu relasi yang tidak dapat digunakan dengan individual itu sendiri.

2.5 *Owlready2*

Owlready2 adalah *package* untuk memanipulasi ontologi OWL 2.0 dengan *Python* dan dapat memuat, memodifikasi, menyimpan ontologi, dan mendukung penalaran melalui *HermiT* (termasuk). *Owlready* memungkinkan akses transparan ke ontologi OWL. *Owlready2* dapat melakukan:

- Impor ontologi dalam format RDF / XML, OWL / XML atau *NTriples*.
- Memanipulasi kelas ontologi, contoh dan anotasi seolah-olah mereka adalah objek *Python*.
- Menambahkan metode *Python* ke kelas ontologi.
- Klasifikasi ulang instans secara otomatis, menggunakan alasan *HermiT*.
- Impor terminologi medis dari UMLS (melihat *PyMedTermio2*).

Arsitektur pada *owlready2*, yaitu memelihara *quadstore* RDF dalam *database* yang dioptimalkan (SQLite3), baik di memori atau di disk. *Quadstore* yaitu menyediakan akses tingkat tinggi ke kelas dan objek dalam ontologi (alias. Pemrograman berorientasi ontologi). Kelas dan Individual dimuat secara dinamis dari *quadstore* sesuai kebutuhan, disimpan dalam *cache* dalam memori dan dimusnahkan saat tidak lagi diperlukan [21].

2.6 **RDF (Resource Description Framework)**

RDF atau *Resource Description Framework* adalah model sederhana yang menjelaskan relasi antara *property* dan *value*. Atribut dari sebuah *resource* disebut sebagai *property*. Atribut tersebut berfungsi untuk merepresentasikan hubungan antar *resource* [16]. RDF memiliki struktur yang terdiri dari subyek, predikat, dan objek. Satu struktur tersebut dapat disebut sebagai pernyataan atau kalimat. Subyek adalah sumber daya yang dideskripsikan sebagai URI. Predikat biasanya disebut sebagai properti yang berfungsi untuk menjelaskan keterkaitan subyek dengan objek. RDF memiliki skema yang disebut sebagai skema RDF. *Property* dan *class* dari sumber skema tersebut dilengkapi dengan sebuah hierarki semantik yang menyamaratakan *property* dan *class*. Skema yang dimiliki RDF berfungsi untuk

mengendalikan sekumpulan terminologi pada dokumen atau bagian kode yang lain. Skema RDF diibaratkan seperti sebuah *master checklist* atau definisi tata bahasa.

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-
  rdf-syntax-ns#"
  xmlns:berita="http://berita.org/elements/">
  <rdf:Description about="
  http://www.detik.com/news/100107">
    <berita:judul>Adam Air</berita:judul>
  </rdf:Description>
</rdf:RDF>
```

Gambar 2. 4. Penulisan RDF

2.7 Ekstraksi Fitur

Ekstraksi fitur adalah proses pengambilan ciri dari sebuah citra yang menggambarkan karakteristik dari citra tersebut dalam sebuah informasi yang dijadikan sebagai acuan untuk membedakan citra satu dengan yang lainnya. Tujuan utama dari ekstraksi fitur adalah untuk mendapatkan informasi yang paling relevan dari data asli dan merepresentasikan informasi tersebut dalam ruang dimensi yang lebih rendah. Ketika data masukan ke suatu algoritma terlalu besar untuk diproses dan diduga redundan, maka data masukan akan diubah menjadi kumpulan fitur representasi yang dikurangi. Mengubah data masukan menjadi sekumpulan fitur disebut ekstraksi fitur [22]. Ekstraksi fitur adalah salah satu bidang terpenting dalam kecerdasan buatan, yang terdiri dari mengekstrak fitur yang paling relevan dari suatu gambar dan menetapkan ke dalam label.

Dalam klasifikasi citra, langkah terpenting adalah menganalisis properti fitur citra dan mengatur fitur numerik ke dalam kelas. Dengan kata lain, suatu gambar dikelompokkan menurut isinya. Fitur adalah atribut individual atau properti dari sebuah objek yang relevan dalam mendeskripsikan dan mengenali objek dan bisa membedakannya dari objek lain. Fitur suatu objek dapat dikenali berdasarkan tekstur, warna, maupun bentuk. Beberapa metode yang digunakan untuk ekstraksi fitur berbasis tekstur, seperti *Wavelet Correlogram*, *Gabor Wavelet Correlogram*, *Gray Level Cooccurrence Matrices*, *Local Binary Patterns* (LBP), dan lain sebagainya. Metode yang digunakan untuk ekstraksi fitur berbasis warna, seperti *Fuzzy Colour Histogram* (FCH). Lalu metode yang digunakan untuk ekstraksi fitur berbasis bentuk misalnya adalah *Histogram of Oriented Gradients* (HOG).

2.8. *K-Nearest Neighbor*

K-Nearest Neighbor atau KNN adalah algoritma *supervised learning* yang melakukan klasifikasi sesuai dengan banyaknya kedekatan jarak yang mayoritas terhadap data yang diklasifikasi. Tujuan dari algoritma *k-Nearest Neighbours* (KNN) adalah untuk menggunakan data latih di mana titik data dipisahkan menjadi beberapa kelas terpisah untuk memprediksi klasifikasi titik sampel baru [23]. *Instance k* ditentukan dengan menghitung jarak tertentu seperti: *Euclidian Distance*, *City Blocks Distance* atau *Manhattan Distance*. Label dari *instance input* akan menjadi label yang paling banyak direpresentasikan di antara *k-nearest neighbor*. Berikut adalah algoritma yang digunakan dalam proses klasifikasi menggunakan *k-nearest neighbor*:

Diketahui:

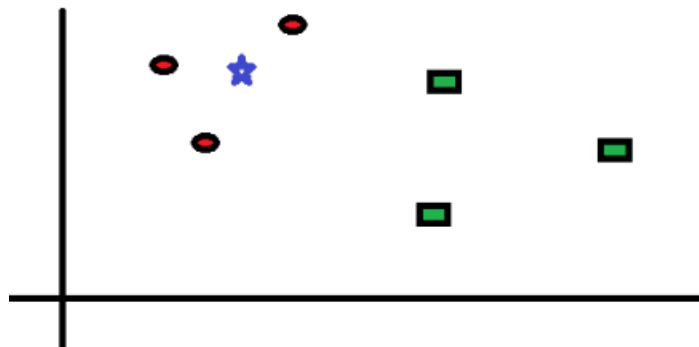
Data latih $\{x_i, y_i\}$; x_i : nilai atribut, dan y_i : label kelas

Instance uji x

Algoritma:

1. Hitung jarak $D(x, x_i)$ untuk semua x_i
2. Pilih k tetangga terdekat dengan labelnya
3. \hat{y} = mayoritas dari label tetangga terdekat

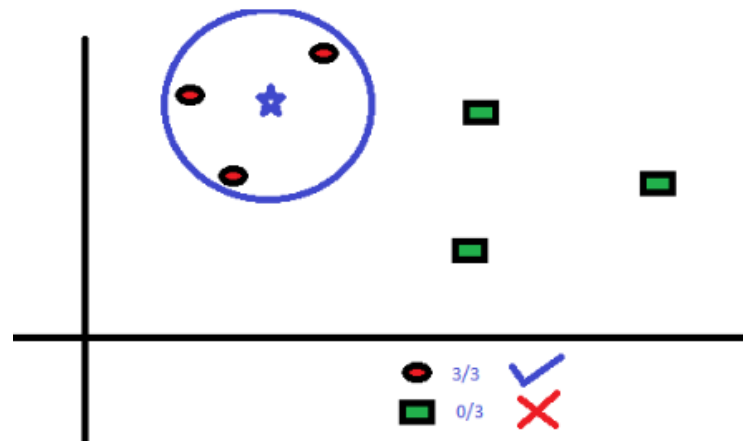
Memilih Faktor k



Gambar 2. 5. Contoh menentukan k dengan kasus sederhana (analyticsvidhya.com)

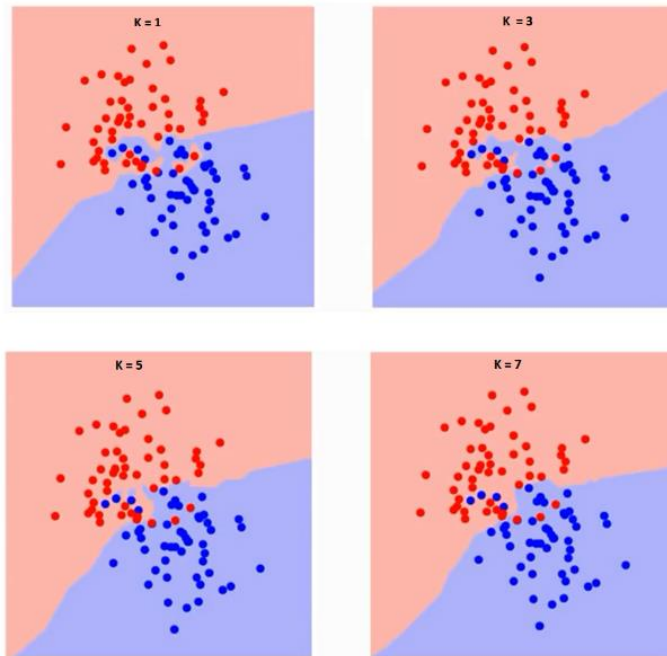
Sebagai contoh seperti Gambar 2.5. bahwa terdapat kelas **lingkaran merah**, **kotak hijau**, dan **bintang ungu**. Lalu kesimpulannya adalah ingin menentukan apakah data **bintang ungu** akan masuk ke dalam kelas **lingkaran merah** atau **kotak hijau**. Algoritma " k " adalah kNN atau tetangga terdekat tempat untuk mengambil suara, misalnya $k = 3$. Oleh karena itu, pada gambar selanjutnya akan dibuat lingkaran

dengan **bintang ungu** sebagai pusatnya hanya sebesar dengan menyertakan tiga titik data pada bidang tersebut.



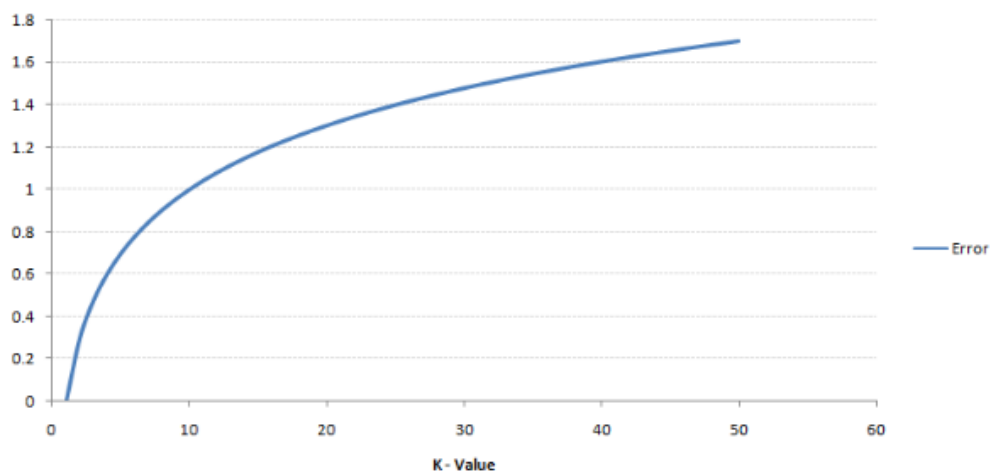
Gambar 2. 6. Menandakan k dengan berpusat pada data uji atau bintang ungu
(analyticsvidhya.com)

Tiga titik terdekat dengan **bintang ungu** semuanya adalah **lingkaran merah**. Maka, dengan tingkat kepercayaan yang baik dapat dikatakan bahwa **bintang ungu** dimiliki oleh kelas **lingkaran merah**. Dalam kasus ini pilihan menjadi sangat jelas karena ketiga suara dari tetangga terdekat diberikan kepada kelas **lingkaran merah**. Pemilihan parameter k menjadi sangat penting dalam algoritma ini. Selanjutnya adalah pemahaman faktor-faktor apa saja yang harus dipertimbangkan untuk menyimpulkan k terbaik. Seperti kasus di atas, berikut ini adalah pemberian batasan kepada dua kelas terhadap *real data* dengan parameter k yang berbeda-beda.



Gambar 2. 7. Contoh penerapan k terhadap real data (analyticsvidhya.com)

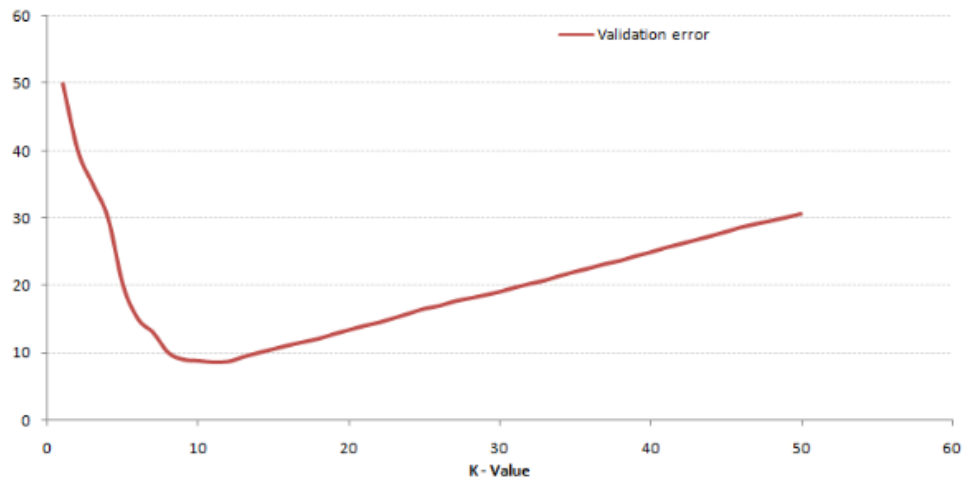
Jika dilihat dari Gambar 2.7. *decision boundaries* menjadi lebih halus dengan meningkatnya nilai k . Dengan k yang meningkat hingga tak terbatas, akhirnya batas di bagian biru menjadi biru seluruhnya atau sebaliknya bergantung pada mayoritas total. *Training error rate* dan *Validation error rate* adalah dua parameter yang dibutuhkan untuk mengakses nilai k yang berbeda. Berikut adalah kurva *error rate training* dengan nilai k yang bervariasi:



Gambar 2. 8. Kurva Training Error Rate (analyticsvidhya.com)

Pada Gambar 2.8. menunjukkan tingkat kesalahan pada $k = 1$ selalu nol untuk sampel data pelatihan. Hal ini karena titik terdekat ke titik data latih adalah data itu

sendiri, sehingga prediksi selalu akurat kepada $k = 1$. Jika pada kurva *validation error* serupa dengan *training error*, maka pilihannya adalah $k = 1$. Berikut adalah kurva *validation error* dengan nilai k yang bervariasi:



Gambar 2. 9. Kurva Validation Error Rate (analyticsvidhya.com)

Untuk tingkat kesalahan $k = 1$ pada kurva *validation error* nilainya melewati batas. Setelah itu, tingkat kesalahan pada k selanjutnya menurun dan mencapai nilai minimum, lalu titik minimum kemudian meningkat dengan bertambahnya k . Untuk mendapatkan nilai k yang optimal dapat memisahkan data latih dan data validasi dari seluruh dataset. Pada plot kurva *validation error* mendapatkan nilai k yang optimal dan nilai k tersebut digunakan untuk melakukan prediksi pada data uji.

Menentukan *Distance Metric* atau *Similarity Function*



Gambar 2. 10. Contoh Dataset dan Hasil Prediksi Gambar Uji (cs231n.github.io)

Misalkan terdapat dataset yang terdiri dari 50.000 gambar (5.000 gambar untuk setiap label), dan kita ingin memberi label 10.000 sisanya. Pengklasifikasi *k-nearest neighbor* akan mengambil gambar uji, lalu membandingkannya dengan

setiap gambar pelatihan, dan memprediksi label gambar pelatihan terdekat. Gambar 2.10. sebelah kanan terdapat contoh hasil dari prosedur seperti itu untuk 10 contoh gambar uji. Perhatikan bahwa hanya sekitar 3 dari 10 contoh gambar dari kelas yang sama diambil, sedangkan pada 7 contoh lainnya tidak demikian (pada baris ke 4, 5 dan 6 saja yang diduga benar dengan kelas yang dimaksud). Sebagai contoh, pada baris ke-8 gambar latihan yang terdekat dengan kepala kuda adalah mobil merah. Mungkin karena latar belakang hitam yang kuat, akibatnya gambar kepala kuda tersebut salah diberi label sebagai mobil. Untuk data pada Gambar 2.10., yaitu sebuah gambar yang dibiarkan detailnya tentang bagaimana membandingkan dua gambar, dan data tersebut hanya terdiri dari dua blok berukuran 32 x 32 x 3. Salah satu cara yang memungkinkan dengan cara paling sederhana adalah membandingkan gambar piksel dengan piksel dan menjumlahkan semua perbedaannya. Dengan kata lain, diberikan dua gambar dan mewakili mereka sebagai vektor ***I1*** dan ***I2***, pilihan lain untuk membandingkannya mungkin dengan menggunakan *metric distance*, yaitu ***L1 distance/manhattan/city blocks distance*** dan ***L2 distance/Euclidean distance***.

a) ***Manhattan Distance***

$$d1(I1, I2) = \sum_p |I_1^p - I_2^p| \quad (2.1)$$

Dimana penjumlahan diambil alih oleh semua piksel, dan berikut contoh pengaplikasian ***L1 distance*** pada perhitungan piksel gambar dengan *pixel-wise absolute value differences*.

test image					training image					pixel-wise absolute value differences			
56	32	10	18		10	20	24	17		46	12	14	1
90	23	128	133		8	10	89	100		82	13	39	33
24	26	178	200	-	12	16	178	170	=	12	10	0	30
2	0	255	220		4	32	233	112		2	32	22	108
										→ 456			

Gambar 2. 11. Perhitungan L1 distance (cs231n.github.io)

Gambar 2.11. merupakan *pixel-wise absolute value differences* untuk membandingkan dua gambar pada *channel* satu warna, yaitu dua gambar dikurangi elemen dan kemudian semua *differences* ditambahkan ke satu

nomor. Jika dua gambar identik maka hasilnya akan nol, tetapi jika gambarnya sangat berbeda maka hasilnya akan bernilai besar.

b) *Euclidean Distance* :

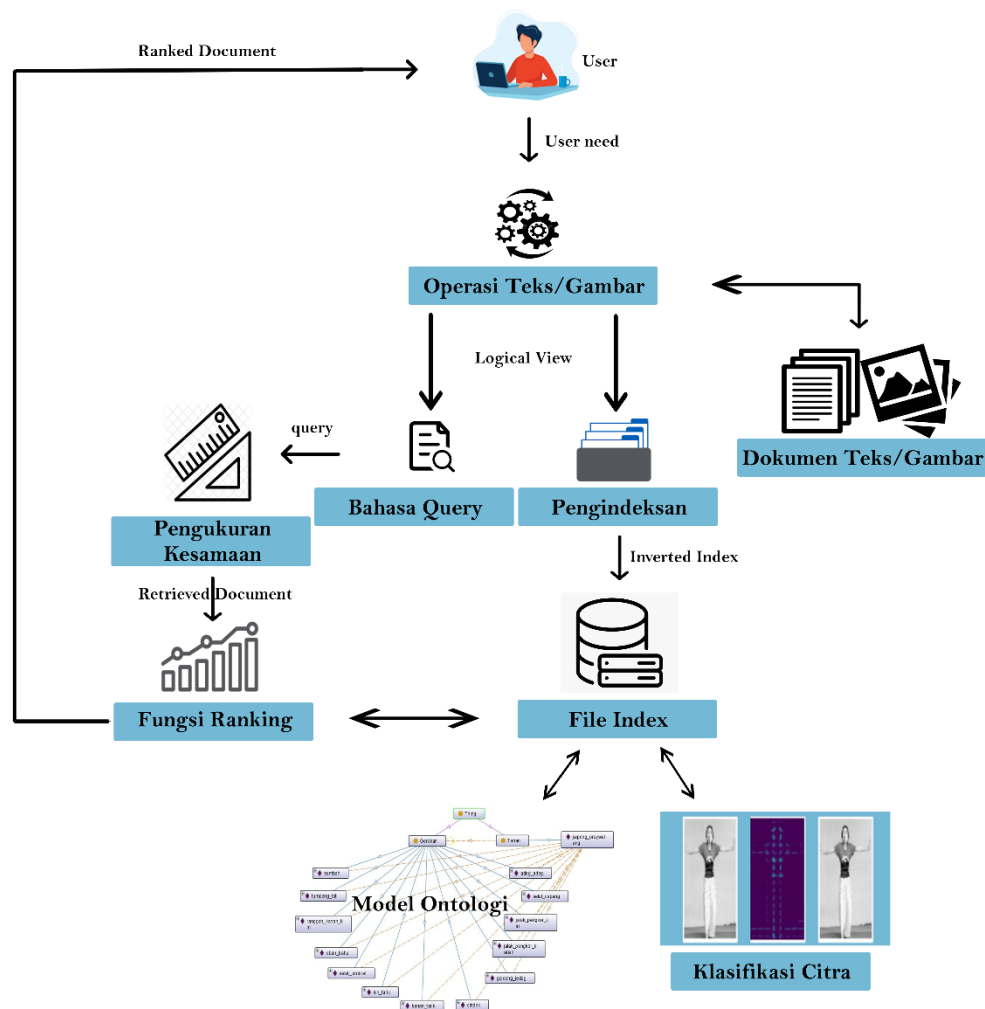
$$d2(I1, I2) = \sqrt{\sum_p (I_1^p - I_2^p)^2} \quad (2.2)$$

Jika terdapat lebih dari 2 *independent variables* atau multi-kelas untuk menghitung jaraknya bisa menggunakan rumus *Euclidean Distance*. Mirip dengan *Pythagoras*, hanya saja *Euclidean Distance* memiliki dimensi lebih dari dua. *Euclidean distance* akan menghitung *pixel-wise difference* seperti pada *L1 distance* atau *Manhattan distance*, tetapi *metric* tersebut mengkuadratkan semua nilainya, yaitu dengan menjumlahkannya lalu diakar kuadratkan.

BAB III

METODE PENELITIAN

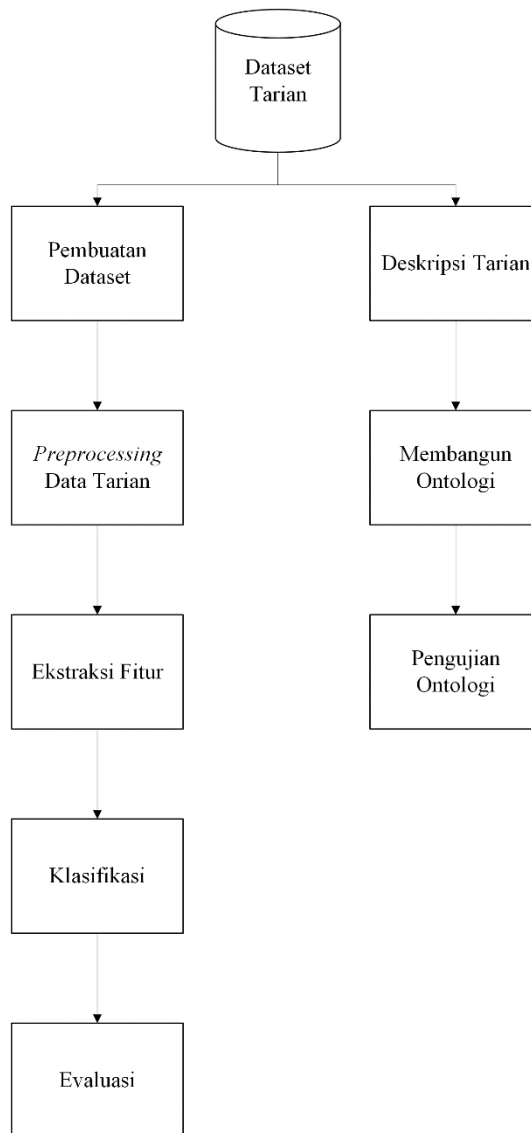
Bab ini akan menjelaskan tentang metode-metode yang akan digunakan. Sebelumnya, Gambar 3.1. merupakan gambaran secara umum dan besar pada penelitian ini, yaitu *information retrieval system* untuk warisan budaya tarian. Penelitian ini akan melakukan sebagian proses dari gambaran yang ada, yaitu membangun model ontologi dan melakukan klasifikasi citra.



Gambar 3. 1. Gambaran Besar Penelitian *Information Retrieval System* Warisan Budaya Tarian

Penelitian ini dibagi menjadi beberapa proses, yaitu pembuatan dataset, *preprocessing data*, ekstraksi fitur, klasifikasi dan evaluasi untuk proses klasifikasi. Lalu deskripsi tarian, membangun ontologi, dan pengujian ontologi merupakan proses pembangunan ontologi. Selanjutnya deskripsi tarian, membangun ontologi, dan pengujian ontology

merupakan proses dari pembangunan model ontologi. Berikut adalah gambaran umum dari penelitian tugas akhir ini:



Gambar 3. 2. Metode Penelitian

3.1. Pengumpulan Data

Penelitian ini menggunakan jenis penelitian kualitatif deskriptif, yang merupakan prosedur penelitian yang menghasilkan data deskriptif berupa kata-kata tertulis atau lisan dari orang-orang atau perilaku yang diamati [23]. Metode tersebut menyesuaikan pendapat antara peneliti dengan informan. Pemilihan metode ini dilakukan karena analisisnya bukan dalam bentuk angka dan peneliti dapat lebih mendeskripsikan segala informasi tentang tarian secara jelas dari informan.

Penelitian ini dilakukan secara bertahap sesuai dengan jadwal yang telah ditentukan, yaitu pengumpulan data informasi pembuatan ontologi yang dilakukan proses wawancara secara lisan untuk validasi tahapan-tahapan gerakan tarian di LKP Sanggar Tari Indra Kusuma yang berada di Jatiwaringin, Bekasi dan dilakukan oleh narasumber ahli tari jaipongan Ibu Deviana Nur dan Ibu Desi. Selanjutnya melakukan observasi berupa pengambilan video dan foto di Sanggar Tari Indra Kusuma oleh Valerie Febriana Putri sebagai penari dan pencarian tari jaipong Oray Welang di *youtube*, lalu diunduh dan terkumpul 6 video dari seluruh proses tersebut. Lalu keenam data video tersebut di *extract* menjadi potongan-potongan gambar atau disebut dengan *extracting frames* yang akan menghasilkan sekitar 11.000 *frame* atau gambar dan akan dipilah ke dalam 13 kelas (gerakan) yang ada.

3.2. Deskripsi Tarian

Deskripsi tarian adalah bagian dari pembangunan ontologi yang merupakan domain pada ontologi yang akan dibangun. Hal ini dilakukan untuk mengulas seputar tarian/suatu domain yang akan dipakai agar mudah untuk mengetahui isi dari domain tersebut, lalu diaplikasikan ke dalam ontologi.

3.3. Membangun Ontologi

Pembangunan ontologi dilakukan untuk mengintegrasikan dari berbagai sumber daya yang terkait, serta berbagi pengetahuan yang benar agar terhindar dari informasi yang tidak relevan [17]. Modul yang digunakan untuk membangun sebuah model ontologi yaitu *OWLReady2*, dalam bentuk bahasa pemrograman *python*. Sebelum melakukan pembangunan ontologi harus mendeskripsikan tarian yang akan dibangun, seperti setiap tahap gerakan memiliki unsur apa saja, misal memiliki nama gerakan, durasi, jenis struktur dan deskripsi dari gerakan itu sendiri. Selanjutnya untuk melakukan pembangunan ontologi yang sudah dideskripsikan sebelumnya dengan menggunakan model ontologi harus melalui tahapan-tahapan berikut [18]:

3.3.1 Menentukan domain dan ruang lingkup ontologi

Ontologi dibangun agar dapat digunakan dengan mendefinisikan satu set data dan strukturnya. Membuat sebuah ontologi dari domain bukanlah suatu tujuannya dan tahap ini merupakan tahapan awal proses pembangunan ontologi,

yaitu menentukan domain dan ruang lingkup dengan cara menjawab 3 pertanyaan dasar yang dapat dijawab oleh model ontologi [18], yaitu:

1. Domain apa yang akan dikembangkan oleh ontologi?
2. Untuk apa kita menggunakan ontologi?
3. Siapa yang akan menggunakan dan mengelola ontologi?

Jawaban dari pertanyaan di atas dapat berubah-ubah selama proses desain ontologi, tetapi dengan bantuan pertanyaan tersebut dapat membuat batasan atau *scope* dari ontologi yang akan dikembangkan.

3.3.2 Mempertimbangkan ontologi yang sudah ada

Tahap ini melakukan pengecekan apakah ontologi yang sudah ada dapat digunakan kembali atau perlu mengembangkan ontologi dari awal. Jika menggunakan ontologi yang sudah ada dapat dilakukan perbaikan dan perluasan. Banyak penyebaran ontologi terutama domain seperti obat, geografis, jejaring sosial yang dapat digunakan. Dengan demikian tidak diharuskan mengembangkan ontologi dari awal [19].

3.3.3 Enumerisasi kata-kata penting dalam ontologi

Enumerasi kata adalah tahap menuliskan semua kata yang diperlukan dalam pembentukan ontologi. Jenis kata yang diperlukan adalah kata benda untuk membentuk nama kelas, dan kata kerja untuk membentuk nama properti. Contohnya seperti kata-kata penting yang berhubungan dengan tarian adalah nama tarian, nama gerakan. Lalu tarian memiliki sub didalamnya, seperti nama penari, musik tarian, gerakan, kostum, dan lain-lain. Gerakan juga memiliki sub didalamnya, seperti jenis gerakan, deskripsi, rentang waktu, dan contoh gerakannya.

3.3.4 Menentukan *class* dan hirarki dari sebuah *class*

Tahap ini dilakukan setelah mengidentifikasi istilah-istilah yang relevan, dan ketentuan-ketentuan tersebut harus disusun dalam taksonomi hirarki berdasarkan kemiripan. Ada tiga cara yang dapat dipilih untuk membentuk tingkatan kelas. Ketiga cara tersebut dapat dilakukan berdasarkan kemampuan masing-masing personal dalam memandang suatu domain. Bagi beberapa pengembang ontologi, kombinasi *top down* dan *bottom up* adalah proses yang

paling mudah karena konsep “di tengah” cenderung lebih deskriptif dalam domain. Misalnya, jika A adalah *subclass* dari B, maka setiap *instance* dari A juga harus merupakan *instance* dari B atau *instances* yang berkaitan harus saling berkomplemen [19]. Berikut adalah cara yang dapat dipilih untuk membentuk tingkatan kelas:

- **Proses pengembangan *top-down***, yaitu memulai pendefinisian dari konsep yang umum, lalu dilanjutkan konsep yang lebih khusus. Contohnya, pengembangan ontologi diawali dengan pembentukan *class* Tarian dan Gerakan. Kemudian Tarian diperinci menjadi beberapa *subclass*, seperti musik, penari, kostum, gerakan. Lalu Gerakan dapat diperinci lagi, seperti jenis gerakan, deskripsi, rentang waktu, dan contoh gerakan.
- **Proses pengembangan *bottom-up***, yaitu memulai pendefinisian dari konsep khusus menuju ke konsep yang lebih umum. Sebagai contoh, pembentukan ontologi dimulai dari pembuatan *class* jaipong Oray Welang, kemudian dibentuk *superclass* untuk *class* tersebut, yaitu Tarian.
- **Kombinasi kedua proses**, yaitu memulai pendefinisian dari konsep yang lebih penting dan melakukan generalisasi atau spesifikasi secara tepat. Proses dapat dimulai dari konsep *top level* seperti Tarian dan beberapa konsep spesifik seperti Oray Welang. Kemudian keduanya dihubungkan dengan konsep *middle level* seperti Musik.

3.3.5 Mendefinisikan *property* dari *class*

Tahap ini adalah mengatur properti yang menghubungkan kelas-kelas sementara dan mengorganisir *class* dalam hirarki. Perlu diingat bahwa semantik dari hubungan *subClassOf* menuntut bahwa *instances* yang berkaitan harus saling berkomplemen [19]. Dengan begitu tahapan ini akan mendefinisikan properti dari masing-masing kelas, seperti kelas Tarian *hasGerakan* Gerakan atau Gerakan *isPartOf* Tarian.

3.3.6 Mendefinisikan *facet* (batasan) dari *property*

Setelah tahapan-tahapan sebelumnya dilakukan, selanjutnya ontologi hanya akan memerlukan ekspresivitas yang disediakan oleh skema RDF dan tidak menggunakan salah satu primitif tambahan di OWL. Tahap ini akan berubah

dengan memperkaya properti yang telah didefinisikan sebelumnya dengan aspek [19]:

- *Cardinality*, yaitu ketentuan untuk properti sebanyak mungkin apakah diperbolehkan atau diperlukan untuk memiliki sejumlah nilai yang berbeda. Sering kali, kasus yang terjadi adalah “*at least one value*”, yaitu properti yang diperlukan dan “*at most one value*”, yaitu properti bernilai tunggal.
- *Required values*, biasanya kelas didefinisikan berdasarkan properti tertentu dengan memiliki nilai tertentu dan nilai tersebut dapat ditentukan menggunakan *owl:hasValue*. Terkadang properti juga bisa diperlukan untuk memiliki beberapa nilai dari kelas tertentu, juga belum tentu dengan menggunakan *owl:someValuesFrom*.
- *Relational characteristics*, yaitu *facets* terakhir yang menyangkut properti *relational characteristics*, yaitu: *symmetry*, *transitivity*, *inverse properties*, dan *functional values*.

3.3.7 Membuat *instance*

Tahap pembuatan ontologi ada langkah untuk mengatur *set instance* dan itu adalah langkah terpisah untuk mengisi ontologi dengan *instances*. Ontologi memiliki ukuran yang bervariasi dari beberapa ratus kelas untuk puluhan ribu kelas. Karena jumlah besar ini, memopulasikan ontologi dengan *instance* biasanya tidak dilakukan secara manual. Biasanya *instance* diambil dari sumber data *legacy* seperti *database*. Teknik lain yang sering digunakan adalah ekstraksi otomatis *instance* dari teks Corpus [19]. Dengan begitu, tahap ini merupakan tahap mendefinisikan sebuah member atau anggota dari suatu kelas dengan melakukan pemilihan kelas, pembuatan individu dari kelas, dan pengisian nilai slot.

3.4. Pengujian Ontologi

Tahap ini adalah tahapan terakhir dari bagian pembangunan ontologi, yaitu menguji apakah ontologi yang dibangun sudah sesuai dengan deskripsi dan langkah-langkah sebelumnya, dan apakah dapat berjalan dengan baik dengan cara melakukan *query* menggunakan SPARQL dengan beberapa pertanyaan yang sudah disediakan.

3.5. Pembuatan Dataset

Tahap ini dilakukan untuk menyediakan data yang akan digunakan pada proses klasifikasi. Data yang digunakan adalah 6 data video dan 500 gambar. Untuk data video akan melalui *extracting frames* untuk mengambil potongan-potongan gambar di dalamnya. Selanjutnya seluruh gambar yang dihasilkan dari *extracting frames* dan 500 gambar yang ada akan dilakukan *cropping dan resizing*. Penjelasan tersebut ada di dalam proses *preprocessing data*.

3.6. Data Preprocessing

Preprocessing dilakukan untuk mempersiapkan data-data sebelum data tersebut diproses ke tahap selanjutnya dalam klasifikasi. Tahap ini akan melakukan proses *extracting frames*, *cropping*, dan *resizing*.

3.6.1. Extracting Frames

Tahap ini dilakukan untuk mengumpulkan dataset dari data video yang sudah dikumpulkan, yaitu dengan cara memotong video menjadi gambar-gambar yang terdapat dalam video tersebut. Dengan *frame rate 1fps (frame per second)* misal video dengan durasi 5 menit maka akan mendapatkan 300 *frame* per video, dan setiap video akan menghasilkan jumlah *frame* yang berbeda-beda. Setelah itu *frame* tersebut akan dipilah kembali sesuai dengan gambar gerakan yang akan dipakai dari beberapa gerakan transisi dari setiap gerakan dan disimpan didalam satu folder sesuai dengan kelas gerakan yang ada. Pada penelitian ini *extracting frames* akan dilakukan dengan *frame rate 5 fps* untuk mendapatkan lebih banyak detail gambar setiap video.

3.6.2. Cropping

Cropping adalah proses memotong gambar, yang tujuannya adalah untuk memfokuskan objek pada gambar tersebut. Proses *cropping* ini juga dapat membantu pada saat pemrosesan ekstraksi fitur untuk penelitian ini, karena pada proses ekstraksi fitur hanya akan fokus pada objek tariannya saja, maka dari itu proses *cropping* sangat membantu untuk menghilangkan informasi yang tidak penting pada gambar dan memfokuskan di bagian objek tertentu.

3.6.3. *Resizing*

Resizing adalah mengubah ukuran gambar yang mengacu pada skala gambar. Tujuan dari *resizing* pada tahap pra-pemrosesan gambar atau *preprocessing* adalah untuk mengubah piksel sebuah gambar dan hanya memiliki beberapa keuntungan, misalnya dalam penyelesaian pengenalan atau pengklasifikasian sebuah gambar akan lebih membantu sebuah model *machine learning* atau *neural network* yang dapat mengurangi waktu pemrosesannya karena semakin banyak jumlah piksel dalam sebuah gambar akan semakin banyak pula jumlah *node input* yang pada gilirannya meningkatkan kompleksitas pada model tersebut.

3.7. Ekstraksi Fitur

Ekstraksi fitur merupakan bagian dari proses sebelum melakukan klasifikasi. Ekstraksi fitur yang digunakan adalah *Histogram of Oriented Gradients* (HOG) yang merupakan ekstraksi fitur berbasis bentuk dengan cara melakukan ekstraksi dan karakterisasi komponen pada citra. Ekstraksi fitur dilakukan menggunakan HOG akan menghasilkan vektor fitur dan disimpan di dalam *database* fitur.

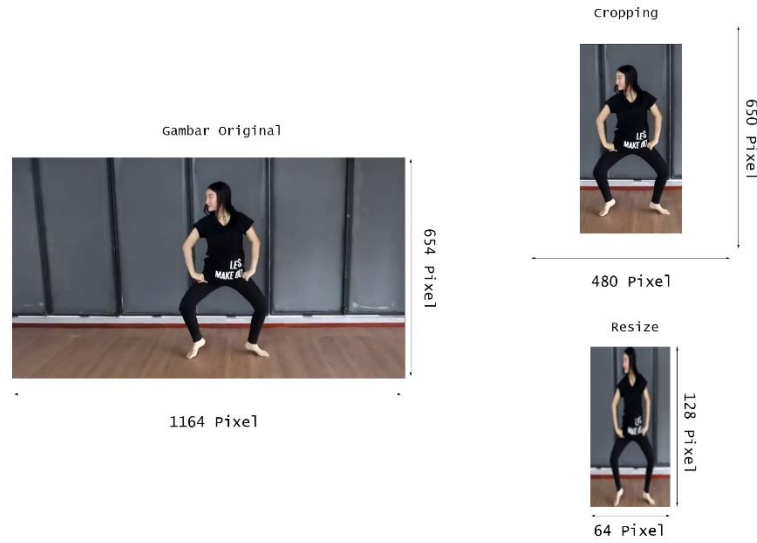
3.7.1. *Histogram of Oriented Gradients*

HOG atau *Histogram of Oriented Gradients* adalah salah satu teknik dari ekstraksi fitur. Tujuan dari HOG adalah untuk mendeskripsikan gambar dengan histogram gradien berorientasi lokal. Histogram ini mewakili kemunculan orientasi gradien tertentu di bagian lokal gambar. HOG memberikan kinerja yang sangat baik dibandingkan dengan fitur lain yang ada. Hipotesis dasarnya adalah bahwa penampakan dan bentuk objek lokal sering dapat dikarakterisasi dengan cukup baik oleh distribusi intensitas gradien lokal atau *edge direction*, bahkan tanpa pengetahuan yang tepat tentang posisi gradien atau *edge* yang sesuai [9]. Ekstraksi fitur HOG memiliki beberapa langkah penting, yaitu:

3.7.1.1. *Preprocessing*

Langkah pertama, yaitu perlu memproses gambar terlebih dahulu dan menurunkan rasio lebar ke tinggi menjadi 1:2. Ukuran gambar sebaiknya 64x128

piksel untuk mengekstrak fitur. Karena dengan ukuran yang lebih kecil akan membuat semua perhitungan menjadi sangat sederhana [24].



Gambar 3. 3. Preprocessing gambar (Cropping dan Resize)

3.7.1.2. Calculate the Gradient Image

Deskriptor HOG dihitung dengan gradien horizontal dan vertikal. Metode ini membutuhkan satu fitur kernel, dan fitur kernel dapat menggunakan operasi *Sobel* untuk menghitung gradien x dari gambar (G_x) dan (G_y). Operator *Sobel* adalah salah satu operator yang menghindari adanya perhitungan gradient di titik interpolasi. Operator ini menggunakan kernel ukuran 3×3 piksel untuk perhitungan gradient, sehingga perkiran gradien berada tepat di tengah jendela. [9].

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} \quad (3.1)$$

$$G_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & 0 & +1 \end{bmatrix} \quad (3.2)$$

G_x = Sobel Horizontal matrices operation

G_y = Sobel Vertical matrices operation

Arah gradien menunjukkan arah perubahan gambar yang paling signifikan. Selanjutnya menghitung dari jumlah akar kuadrat dari Gradien x (G_x) dan Gradien y (G_y). Nilai arah gradien dapat dihitung dengan persamaan berikut [9]:

$$G = \sqrt{Gx^2 + Gy^2} \quad (3.3)$$

G = Gradient Magnitude

Gx = Sobel Horizontal matrices operation

Gy = Sobel Vertical matrices operation

Selanjutnya adalah menghitung orientasi (atau arah) untuk piksel yang sama yaitu dengan menggunakan rumus tan untuk sudut:

$$\tan(\Phi) = \frac{Gy}{Gx} \quad (3.4)$$

Jadi, nilai sudut atau besaran gradien untuk mengetahui informasi tentang seberapa besar perubahan yang terjadi menggunakan persamaan [9]:

$$R_0 = \arctan \frac{Gy}{Gx} \quad (3.5)$$

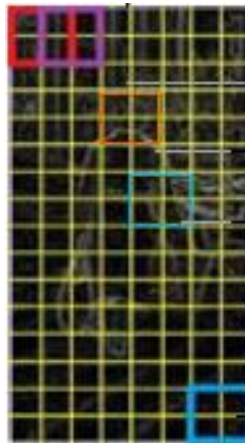
R_0 = Gradient Orientation

Gx = Sobel Horizontal matrices operation

Gy = Sobel Vertical matrices operation

3.7.1.3. Membagi input image menjadi cells dan blocks

Citra gradien yang dihasilkan dibagi menjadi sel berukuran 8x8 piksel. Window yang bergeser dengan ukuran 16x16 piksel melalui *cell*, dengan setiap langkah mencakup empat *cell* tetangga. Setiap kelompok dari empat *cell* yang bertetangga membentuk satu blok. Blok tetangga tumpang tindih satu sama lain seperti yang ditunjukkan pada Gambar. 3.4. Melalui proses ini, total 105 blok terbentuk pada gambar 64x128 piksel. Pembagian gambar menjadi blok mirip dengan HOG asli. Tahap ini diperlukan untuk memfasilitasi ekstraksi fitur pada langkah selanjutnya [25].

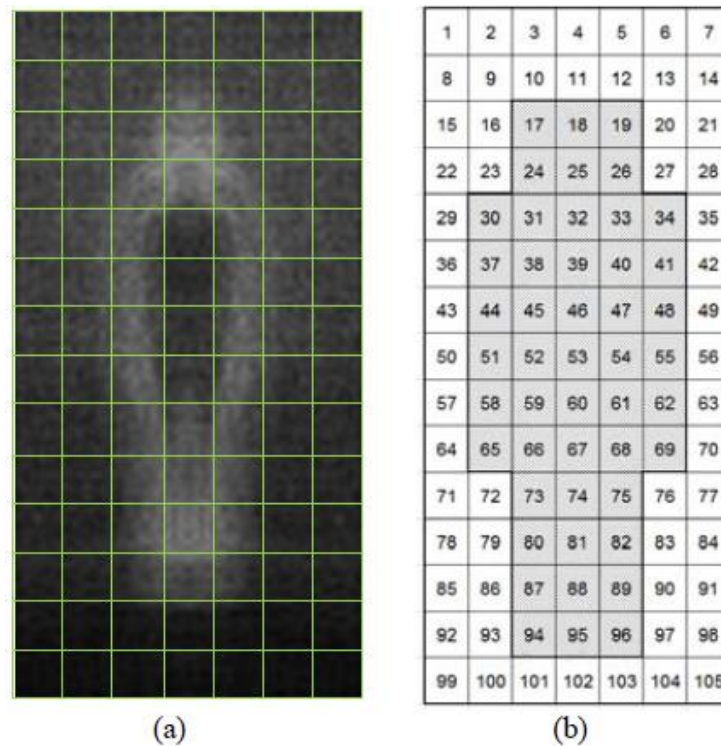


Gambar 3. 4. Contoh block cell yang bergeser di setiap piksel gambar [26]

3.7.1.4. Calculate Histogram of Gradient in 8x8 cells

Untuk setiap blok, histogram untuk orientasi gradien dibuat dengan memilih sudut orientasi setiap piksel ke dalam jumlah *bin* histogram yang telah ditentukan. Menggunakan jumlah *bin* yang lebih tinggi akan mengekstrak informasi orientasi yang lebih detail dari gambar, tetapi akan menghasilkan jumlah fitur yang lebih banyak. Untuk memperkecil ukuran fitur dan tetap mempertahankan detail penting dalam fitur, jumlah *bin* histogram yang berbeda digunakan untuk wilayah berbeda pada gambar. Jumlah *bin* histogram yang lebih tinggi digunakan untuk mengekstraksi fitur-fitur pada kawasan yang mungkin merupakan bagian dari orang yang sedang melakukan gerakan tarian, sementara jumlah *bin* yang lebih rendah digunakan untuk kawasan lainnya [25].

Untuk mengidentifikasi daerah yang mungkin merupakan bagian dari objek yang sedang melakukan gerakan tarian, gambar rata-rata dibangun dari beberapa sampel pelatihan positif. Blok yang menunjukkan posisi blok tersebut, kemudian ditempatkan pada gambar rata-rata seperti yang ditunjukkan pada Gambar. 3.5. (a). Dari gambar ini, blok yang mungkin berisi orang yang sedang melakukan gerakan tarian diidentifikasi. Blok-blok ini ditunjukkan pada Gambar. 3.5. (b).



Gambar 3. 5. (a) Grid yang menunjukkan posisi blok yang ditempatkan pada rata-rata gambar yang dihasilkan. (b) blok yang teridentifikasi gambar yang diinginkan [26]

Jumlah *bin* histogram yang lebih tinggi digunakan untuk mengekstrak fitur pada blok yang diarsir, sementara jumlah *bin* histogram yang lebih rendah digunakan untuk blok yang tersisa. Nilai optimum untuk jumlah *bin* tinggi dan rendah yang akan digunakan ditentukan secara empiris.

3.7.1.5.16x16 Block Normalization

Langkah sebelumnya menjelaskan tentang membuat histogram berdasarkan gradien gambar yang diketahui bahwa gradien gambar sensitif terhadap keseluruhan pencahayaan. Tahap ini tujuannya adalah ingin "menormalkan" histogram dengan mengambil 16x16 blok, sehingga tidak terpengaruh oleh variasi pencahayaan. *L2-norm* adalah skema normalisasi terbaik untuk ekstraksi ciri HOG dan oleh karena itu digunakan dalam metode yang diusulkan [25]. Skema *L2-norm* didefinisikan sebagai:

$$v_n = \frac{v}{\sqrt{\|v\|_2^2 + \varepsilon^2}} \quad (3.6)$$

v = vektor *unnormalized*

v_n = vektor yang dinormalisasi

$\|v\|_2$ = nilai *L2-norm*

ε = *small normalization* untuk menghindari pembagian dengan nol

Nilai *L2-norm* dihitung menggunakan persamaan berikut:

$$\|v\|_2 = \sqrt{\sum_{i=1}^n v_i^2} \quad (3.7)$$

Normalisasi multi blok dilakukan dengan mengelompokkan beberapa blok yang bertetangga sesuai dengan jumlah *bin* orientasi yang digunakan dalam pembuatan histogram. Sebanyak 30 kelompok dibentuk seperti yang ditunjukkan pada Gambar. 3.6.

1		2		3
4		5		6
7	8	9	10	11
12	13	14	15	16
17	18	19	20	21
22		23		24
25		26		27
28		29		30

Gambar 3. 6. Pengelompokan blok untuk melakukan normalisasi [26]

Skema *L2-norm* kemudian digunakan untuk menormalkan semua nilai histogram pada setiap kelompok. Histogram yang dinormalisasi dari semua grup akan digabungkan untuk membentuk vektor fitur yang diusulkan.

3.8. KNN (*K-Nearest Neighbor Classification*)

Dalam penelitian ini algoritma klasifikasi yang digunakan adalah KNN atau *k-nearest neighbor*, yang telah ditingkatkan kinerja kecepatannya dengan alat *sklearn* dengan *Python*. Klasifikasi KNN menggunakan *tools sklearn* hanya membutuhkan waktu 0.4 detik untuk pengerjaannya. Selain itu, untuk meningkatkan performa kecepatan, pada penelitian ini mengubah ukuran data dari 280x450 piksel menjadi 64x128 piksel. Jadi, dengan mengubah ukuran piksel gambar menjadi lebih kecil akan meningkatkan kecepatan kinerja KNN.

Dalam pengerjaan penelitian ini, algoritma *k-Nearest Neighbors* (KNN) digunakan untuk klasifikasi tentang gerakan-gerakan yang terdapat pada tarian. *K-Nearest Neighbours* (KNN) adalah algoritma yang sangat sederhana, dan mudah untuk menerapkan algoritma *supervised machine learning* yang dapat digunakan untuk menyelesaikan beberapa masalah klasifikasi. Karena KNN ditentukan dengan perhitungan *Euclidean Distance*, berikut persamaan yang digunakan:

$$ED = \sqrt{\sum_{i=1}^k (x_i - y_i)^2} \quad (3.8)$$

$ED = Euclidean Distance$

$x_i = \text{Data Uji}$

$y_i = \text{Data Latih}$

$k = \text{Number of Neighbors}$

Nilai k harus dipilih dengan hati-hati. Jika nilai k terlalu kecil, maka akan terlalu sensitif terhadap *noise*. Jika nilai k terlalu besar, lingkungan tersebut juga dapat menyertakan kelas lain yang masuk ke kelas yang tidak seharusnya [9]. ED adalah nilai fitur berbobot, ketika nilai ED besar akan menjadi tetangga yang sangat jauh dari nilai ED yang kecil. Hasil klasifikasi adalah nilai yang besar maka tidak akan menjadi satu kelas dengan nilai yang kecil, karena jaraknya yang sangat jauh.

3.9. Evaluasi

Evaluasi adalah tahap untuk mengetahui seberapa berhasil pekerjaan-pekerjaan (klasifikasi) pada tahap-tahap sebelumnya. Penelitian ini menggunakan *confusion matrix* sebagai alat untuk menghitung *evaluation metrics* klasifikasi yang membandingkan kelas yang asli dari masukan atau dengan kata lain berisi informasi nilai aktual dan prediksi pada proses klasifikasi. Dari model *confusion matrix* terdapat empat informasi, yaitu *True Positive*, *False Positive*, *True Negative*, dan *False Negative*.

Tabel 3. 1. *Confusion Matrix*

		<i>Prediction</i>	
		<i>Positive</i>	<i>Negative</i>
	<i>Positive</i>	TP	FN
	<i>Negative</i>	FP	TN

1. *True Positive* (TP), didefinisikan oleh jumlah total output yang akurat ketika kelas sebenarnya dari data latih adalah *True* dan prediksinya juga merupakan nilai *True*.
2. *True Negative* (TN), didefinisikan oleh jumlah total output yang akurat ketika kelas sebenarnya dari data latih adalah *False* dan prediksi juga merupakan nilai *False*.
3. *False Positive* (FP), ketika kelas sebenarnya dari data latih adalah *False* dan nilai keluaran dari prediksi adalah nilai *True*.
4. *False Negative* (FN), ketika kelas sebenarnya dari data latih adalah *True* dan nilai keluaran dari prediksi adalah nilai *False*.

Confusion matrix memberikan informasi yang berguna tentang seberapa baik model tersebut. Namun, elemennya dapat digunakan untuk menghitung banyak *performance metric* untuk mendapatkan lebih banyak informasi, seperti *Accuracy*, *Precision*, *Recall*, *F-Score* [26].

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (3.9)$$

Persamaan 3.9 merupakan perhitungan untuk data *binary class*

$$Accuracy = \frac{\sum_{i=1}^n \frac{TP_i + TN_i}{TP_i + FP_i + TN_i + FN_i}}{n} \quad (3.10)$$

Persamaan 3.10 merupakan perhitungan untuk data *multiclass*

Accuracy merupakan ukuran kinerja yang paling intuitif, dan didefinisikan sebagai rasio jumlah objek yang diklasifikasikan dengan benar terhadap jumlah total objek yang dievaluasi.

$$Precision = \frac{TP}{TP + FP} \quad (3.11)$$

Persamaan 3.11 merupakan perhitungan untuk data *binary class*

$$Precision = \frac{\sum_{i=1}^n \frac{TP_i}{TP_i + FP_i}}{n} \quad (3.12)$$

Persamaan 3.12 merupakan perhitungan untuk data *multiclass*

Precision itu hanyalah rasio objek data positif yang diprediksi dengan benar terhadap total objek data positif yang diprediksi.

$$Recall = \frac{TP}{TP + FN} \quad (3.13)$$

Persamaan 3.13 merupakan perhitungan untuk data *binary class*

$$Recall = \frac{\sum_{i=1}^n \frac{TP_i}{TP_i + FN_i}}{n} \quad (3.14)$$

Persamaan 3.14 merupakan perhitungan untuk data *multiclass*

Recall ditentukan oleh jumlah hasil positif yang benar dibagi dengan jumlah sampel yang relevan (semua sampel yang seharusnya diidentifikasi sebagai positif).

$$F1 - Score = \frac{2 (Recall \times Precision)}{(Recall + Precision)} \quad (3.15)$$

Persamaan 3.15 merupakan perhitungan untuk data *binary class*

$$F1 - Score = 2 \times \frac{(Recall \times Precision)}{(Recall + Precision)} \quad (3.16)$$

Persamaan 3.16 merupakan perhitungan untuk data *multiclass*

F1-score dapat didefinisikan sebagai *weighted average* dari *precision* dan *recall*. *F1-score* dianggap sempurna saat mencapai nilai terbaiknya adalah 1, sedangkan saat model *total failure*-nya mencapai nilai 0.

BAB IV

IMPLEMENTASI DAN ANALISIS

Bab ini akan menjelaskan tentang implementasi dan analisis hasil dari dua kelompok proses, yaitu Pembangunan Ontologi yang dimulai dari subbab 4.1. sampai 4.3. dan Klasifikasi yang dimulai dari subbab 4.4. sampai 4.8.

4.1. Deskripsi Tarian

Tari jaipong Oray Welang memiliki 13 gerakan secara umum, yaitu adeg-adeg, cindek, galeong jedag, jalak pengkor kanan, jalak pengkor kiri, kanan tarik, kiri tarik, obah bahu, ranggah kanan-kiri, selut capang, selut jambret, sembah, tumpang tali. Terdapat pula beberapa jenis gerakan, yaitu gerakan kaki, tangan, bahu, dan kombinasi. Adeg-adeg dan cindek adalah jenis gerakan kaki, lalu selut capang, selut jambret, ranggah kanan-kiri, tumpang tali, dan sembah termasuk jenis gerakan tangan. Obah bahu adalah jenis gerakan bahu, terakhir yaitu galeong jedag, jalak pengkor kanan, jalak pengkor kiri, kanan tarik, dan kiri tarik adalah jenis gerakan kombinasi; kepala dan bahu, kaki dan tangan. Tari jaipong Oray Welang memiliki durasi selama 5-6 menit untuk satu tarian penuh dan terdapat 70 langkah gerakan dalam satu tarian penuh.

4.2. Pembangunan Ontologi

Tahap ini seperti yang sudah dijelaskan di dalam Bab III, yaitu membangun ontologi untuk merepresentasikan sebuah tarian, tari jaipong Oray Welang yang akan menjelaskan tahapan-tahapan tarian tersebut dari awal sampai akhir. Tarian ini berisi tari gerak dasar yang menggambarkan sebuah ciri khas dari tari jaipong itu sendiri yang sebagian gerakannya juga dipakai ke seluruh tari jaipong. *Library* yang dipakai untuk membangun ontologi adalah *owlready2*, yaitu modul ontologi yang menggunakan bahasa pemrograman *python* dan *protégé* untuk melihat *graph* dari ontologi yang sudah dibuat menggunakan *owlready2*.

4.2.1. Menentukan domain dan cakupan ontologi

Untuk membangun sebuah ontologi harus diawali dengan mendefinisikan domain dan menentukan *scope* atau cakupan pada ontologi. Untuk domain ontologi yang akan dikembangkan pada penelitian ini adalah Tahapan Tarian Jaipong Oray Welang yang seluruh informasinya didapatkan dari ahli atau guru tari di LKP Sanggar Tari Indra Kusuma. Tujuannya adalah untuk merepresentasikan tahapan-tahapan tarian dan akan berguna untuk mengetahui struktur dari sebuah tarian.

4.2.2. Mempertimbangkan ontologi yang sudah ada

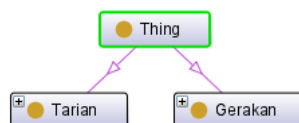
Tahap ini adalah menentukan apakah mengembangkan ontologi yang sudah ada atau akan mengembangkan ontologi dari awal. Karena tema representasi tarian untuk saat ini masih sangat sedikit, tetapi ada salah satu penelitian yang menjelaskan *Indian Classical Dance* [5], namun penelitian tersebut mencakup keseluruhan tarian dan sangat kompleks. Maka dari itu untuk penelitian ini akan mengembangkan ontologi dari awal, yang hanya merepresentasikan tahapan dari sebuah tarian.

4.2.3. Enumerasi kata-kata penting dalam ontologi

Enumerasi kata-kata penting yang digunakan untuk membangun ontologi tahapan-tahapan gerakan pada tarian, yaitu memiliki kategori tarian dan gerakan, lalu tarian memiliki gerakan, dan gerakan memiliki langkah-langkah gerakan, nama gerakan, jenis gerakan, *time range*, deskripsi dan contoh gambar.

4.2.4. Menentukan *class* dan hirarki dari sebuah *class*

Sebuah *class* merupakan sekumpulan objek-objek yang menjelaskan konsep dari sebuah domain. *Class* juga dapat memiliki *subclass*, yaitu *class* yang merepresentasikan konsep lebih spesifik dari *superclass*-nya. Penelitian ini terdapat dua *class*, yaitu *class* Tarian dan Gerakan.



Gambar 4. 1. Class Ontologi pada Tahapan Gerakan Tarian

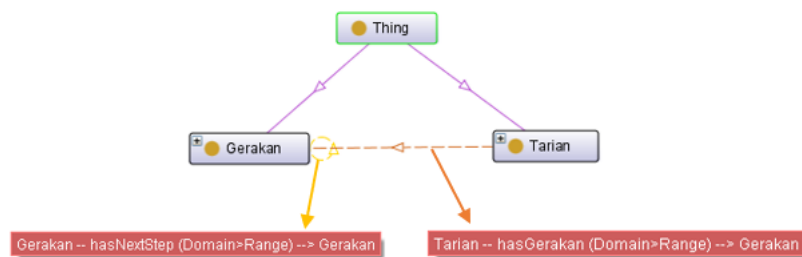
4.2.5. Mendefinisikan *property* dari *class*

Setelah menentukan *class*, selanjutnya adalah tahap menentukan hubungan antar kelas atau *property* yang akan memperjelas hubungan setiap kelasnya di dalam sebuah ontologi. Penelitian ini memiliki *object property* dan *data property* seperti berikut:

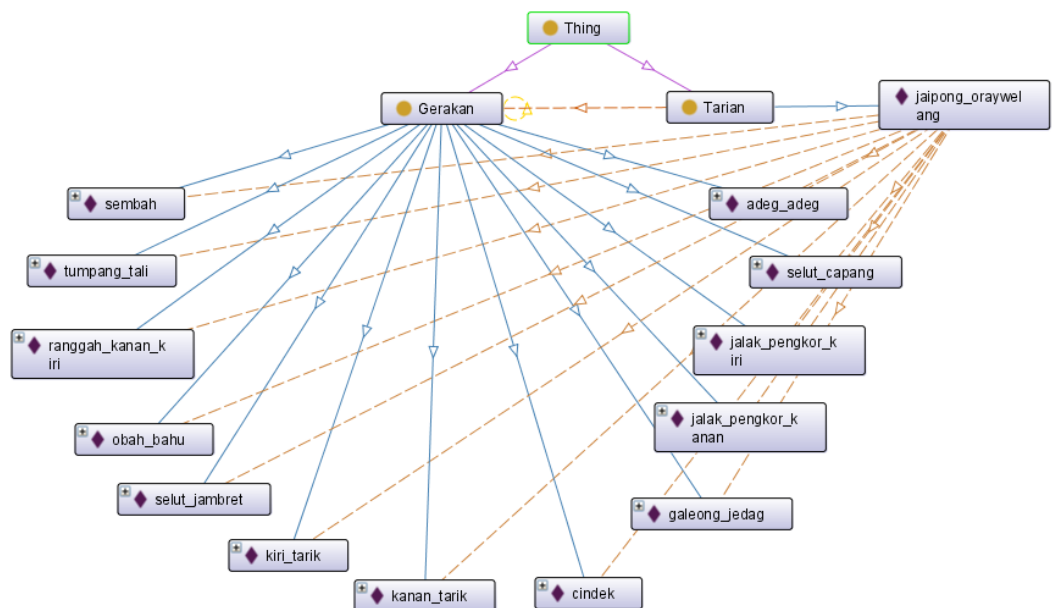
- a) *Object Property* yang menjelaskan hubungan antar kelas dan untuk penelitian ini adalah setiap tarian memiliki gerakan, lalu setiap gerakan memiliki langkah-langkah gerakan selanjutnya.

Tabel 4. 1. Rancangan Object Property

<i>Object Property</i>	<i>Domain</i>	<i>Range</i>
hasGerakan	Tarian	Gerakan
hasNextStep	Gerakan	Gerakan

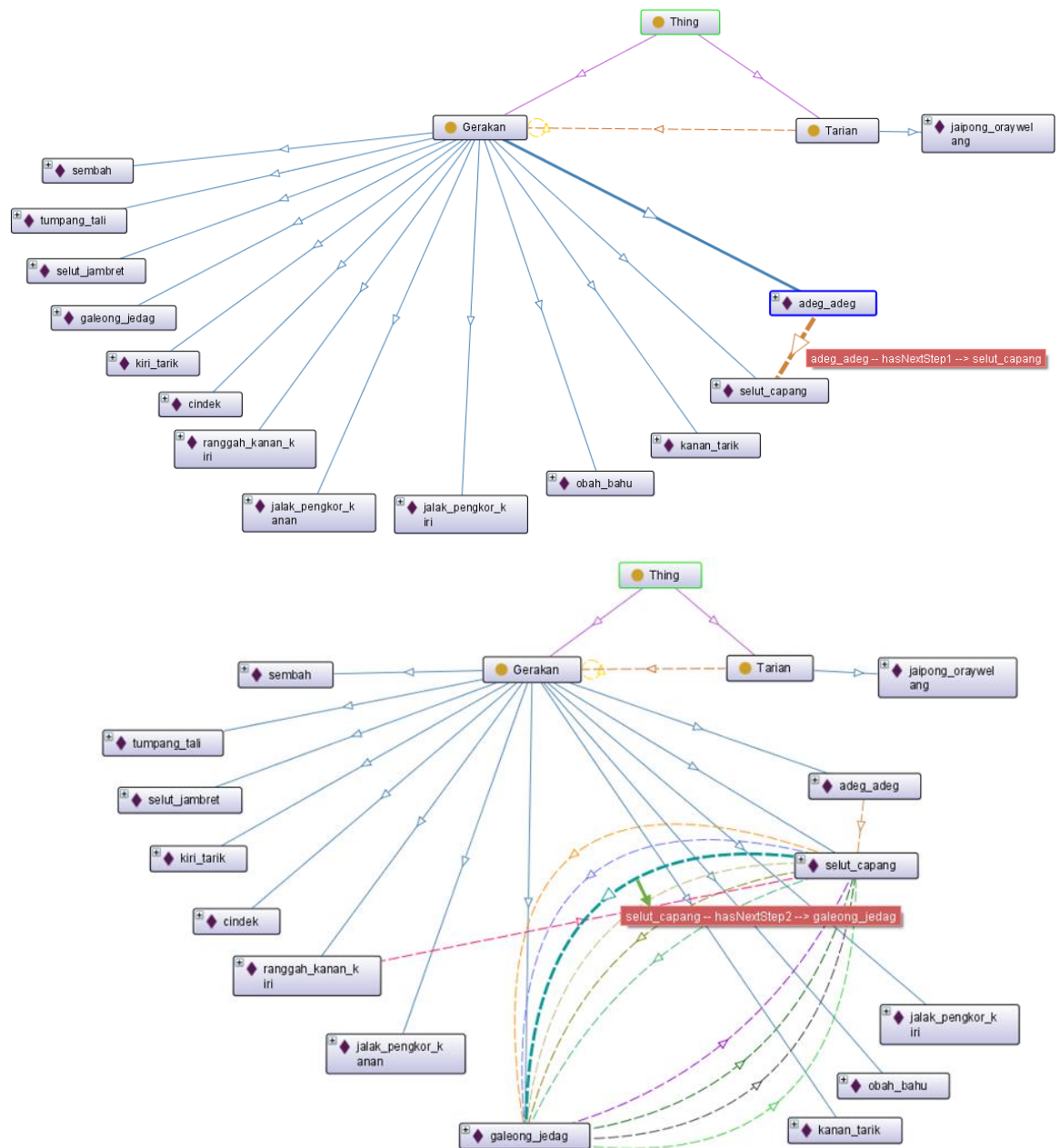


Gambar 4. 2. Object Property Ontology



Gambar 4. 3. Object Property hasGerakan pada domain Tarian

- b) *Sub-Object Property* merupakan *object property* yang menjelaskan *super-object property* lebih detail lagi. Seperti pada penelitian ini terdapat *sub-object property* dari *hasNextStep* dimana *object property* tersebut merepresentasikan Gerakan A memiliki langkah gerakan selanjutnya, yaitu Gerakan B, begitu seterusnya sampai proses tarian berakhir. *Sub-object property* disini adalah *hasNextStep1* sampai *hasNextStep69* karena terdapat 70 tahap gerakan dengan durasi tarian selama kurang lebih 6 menit.



Gambar 4. 4. Graph Sub-Object property hasNextStep1 dan hasNextStep2

- c) *Data Property* adalah atribut-atribut dari sebuah *class*. Kelas Tarian memiliki *data property* nama tarian dan untuk kelas Gerakan memiliki nama gerakan, jenis gerakan, *time range*, deskripsi, dan contoh gerakan.

Tabel 4. 2. Rancangan Data Property Class Gerakan

<i>Class</i>	<i>Data Property</i>	<i>Domain</i>	<i>Tipe</i>
Gerakan	jenis_gerakan	Gerakan	<i>String</i>
Gerakan	time_range	Gerakan	<i>String</i>
Gerakan	deskripsi	Gerakan	<i>String</i>
Gerakan	contoh_gerakan	Gerakan	<i>String</i>

```
tumpang_tali
URI: http://test.org/ontologi_tariJaipong.owl#tumpang_tali
Object property assertions:
  tumpang_tali hasNextStep32 obah_bahu
  tumpang_tali hasNextStep20 jalak pengkor kanan
Data property assertions:
  tumpang_tali contoh_gambar
    "https://drive.google.com/file/d/1IzPjJoSeMD0jwJHx75dPwlyQ5MIs_4lh/view"^^s
    tring
  tumpang_tali jenis_gerakan "Tangan"^^string
  tumpang_tali deskripsi "gerakan kedua tangan ke depan, menyilang dengan
    posisi tangan kanan tertumpang/berada di atas tangan kanan dengan jari-jari
    ke arah bawah, lalu tangan kiri lurus ke atas dengan posisi ibu jari diteku...
  tumpang_tali dari_sampai_waktu_ke "0:01:10-0:01:11,
    0:02:09-0:02:10"^^string
```

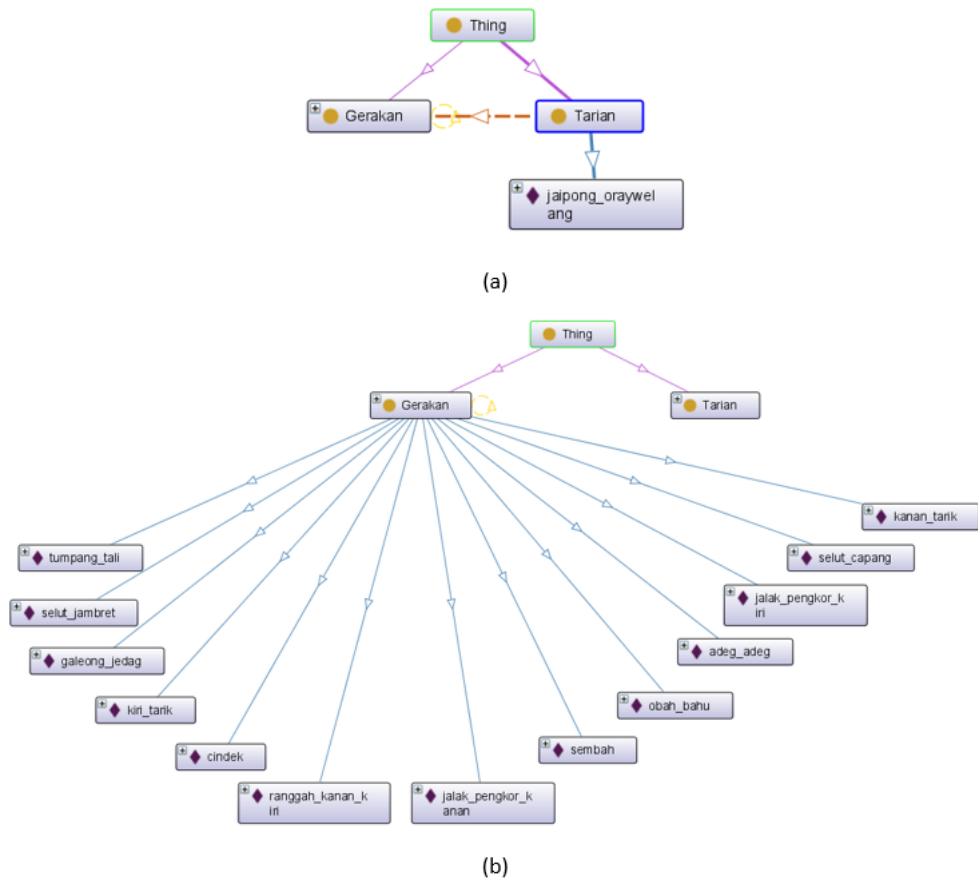
Gambar 4. 5. Data Property pada Ontologi

4.2.6. Mendefinisikan *facet* (batasan) dari *property*

Facet atau batasan adalah tipe data atau kardinalitas pada *property*. Untuk tipe data yang digunakan pada ontologi tarian adalah *string* seperti pada Tabel 4.2.

4.2.7. Membuat *instance*

Instance disebut juga *individual* yang merupakan anggota atau member dari setiap *class* ontologi. Gambar 4.6. adalah hasil dari pembentukan *instance* untuk setiap kelas pada ontologi tarian, pada bagian (a) menunjukkan *instance* dari kelas Tarian dan bagian (b) adalah *instances* dari kelas Gerakan.



Gambar 4. 6. (a) Instance kelas Tarian, (b) Instance kelas Gerakan

4.3. Pengujian Ontologi

Tahap ini adalah melakukan pengujian dari ontologi yang sudah dibuat dengan harapan sesuai dengan apa yang sudah dirancang pada tahap-tahap sebelumnya, yaitu merepresentasikan tahapan-tahapan gerakan pada tari jaipong oray welang. Tahap pengujian ontologi ini dilakukan dengan menggunakan *query SPARQL*.

1. Menguji apakah gerakan-gerakan pada tari jaipong Oray Welang sudah sesuai dengan gerakan yang ada (13 gerakan):

SPARQL query:	
<pre> PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> PREFIX owl: <http://www.w3.org/2002/07/owl#> PREFIX xsd: <http://www.w3.org/2001/XMLSchema#> PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#> PREFIX : <http://test.org/ontologi_tariJaipong.owl#> SELECT ?Tarian ?Gerakan WHERE { ?Tarian :hasGerakan ?Gerakan } </pre>	
Tarian	Gerakan
jaipong_oraywelang	adeg_adeg
jaipong_oraywelang	ranggah_kanan_kiri
jaipong_oraywelang	selut_jambret
jaipong_oraywelang	sembah
jaipong_oraywelang	kiri_tarik
jaipong_oraywelang	kanan_tarik
jaipong_oraywelang	selut_capang
jaipong_oraywelang	jalak_pengkor_kanan
jaipong_oraywelang	tumpang_tali
jaipong_oraywelang	jalak_pengkor_kiri
jaipong_oraywelang	cindek
jaipong_oraywelang	galeong_jedag
jaipong_oraywelang	obah_bahu

Gambar 4. 7. Hasil query SPARQL gerakan-gerakan tari jaipong oray welang

Hasil pengujian pada Gambar 4.7. adalah sudah sesuai dengan pertanyaan yang ada, yaitu bahwa jaipong Oray Welang memiliki 13 gerakan dan di dalam ontologi tersebut *jaipong_oraywelang* memiliki ke-13 gerakan.

2. Menguji apakah Gerakan A *hasNextStep* Gerakan B teranotasi dengan benar atau tidak:

SPARQL query:	
<pre> PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> PREFIX owl: <http://www.w3.org/2002/07/owl#> PREFIX xsd: <http://www.w3.org/2001/XMLSchema#> PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#> PREFIX : <http://test.org/ontologi_tariJaipong.owl#> SELECT ?Gerakan ?hasNextStep WHERE { ?Gerakan :hasNextStep1 ?hasNextStep } </pre>	
Gerakan	hasNextStep
adeg_adeg	selut_capang

SPARQL query:	
<pre> PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> PREFIX owl: <http://www.w3.org/2002/07/owl#> PREFIX xsd: <http://www.w3.org/2001/XMLSchema#> PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#> PREFIX : <http://test.org/ontologi_tariJaipong.owl#> SELECT ?Gerakan ?hasNextStep WHERE { ?Gerakan :hasNextStep2 ?hasNextStep } </pre>	
Gerakan	hasNextStep
selut_capang	galeong_jedag

Gambar 4. 8. Hasil query SPARQL Gerakan A *hasNextStep* Gerakan B

Hasil pengujian pada Gambar 4.8. menyatakan bahwa sudah benar dengan pengujian Gerakan *hasNextStep1* Gerakan adalah *adeg_adeg* *hasNextStep* *selut_capang* dan Gerakan *hasNextStep2* Gerakan adalah *selut_capang*

hasNextStep *galeong_jedag* yang artinya pengujian tersebut menjawab pertanyaan yang ada.

3. Menguji apakah *data property* dari setiap gerakan teranotasi dengan baik atau tidak:

```

SPARQL query:
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX : <http://test.org/ontologi_tariJaipong.owl#>
SELECT ?Gerakan ?link_gambar ?time ?deskripsi ?jenis_gerakan
WHERE {
    ?Gerakan contoh_gambar ?link_gambar;
           dari_sampai_waktu_ke ?time;
           deskripsi ?deskripsi;
           jenis_gerakan ?jenis_gerakan.
}

```

Gerakan	link_gambar	time	deskripsi	jenis_gerakan
cindek	"https://drive.google.com/file/d/17YPAz..."	"0:00:31-0:00:33, 0:00:60-0:01:00, 0:"	"sikap kaki adeg-adeg kembar, gerak r..."	"Kaki"^^<http://www.w3.org/2001/XM...
tumpang_tali	"https://drive.google.com/file/d/1IzPjJo..."	"0:01:10-0:01:11, 0:02:09-0:02:10"^^	"gerakan kedua tangan ke depan, men..."	"Tangan"^^<http://www.w3.org/2001/XM...
sembah	"https://drive.google.com/file/d/1P-q5Ij..."	"0:06:20-0:06:24"^^<http://www.w3.org/2001/XM...	"kedua telapak tangan dirapatkan sep..."	"Tangan"^^<http://www.w3.org/2001/XM...
ranggang_kanan_kiri	"https://drive.google.com/file/d/1GqYZ..."	"0:03:25-0:03:28, 0:04:20-0:04:23"^^	"kedua tangan diangkat ke atas dan ki..."	"Tangan"^^<http://www.w3.org/2001/XM...
galeong_jedag	"https://drive.google.com/file/d/1PqUSF..."	"0:00:16-0:00:19, 0:00:24-0:00:28, 0:"	"gerakan memutarakan kepala dan bahu..."	"Kombinasi(kepala dan bahu)"^^<http://www.w3.org/2001/XM...
kin_tarik	"https://drive.google.com/file/d/1yW0X..."	"0:00:31-0:00:33, 0:00:60-0:01:00, 0:"	"gerakan kaki yang ditarik dari arah ki..."	"Kombinasi(kaki dan tangan)"^^<http://www.w3.org/2001/XM...
jalak_pengkor_kiri	"https://drive.google.com/file/d/1n-Ycq..."	"0:00:50-0:00:53, 0:01:49-0:01:51, 0:"	"gerakan kaki terpatang-patang ke arah l..."	"Kombinasi(kaki dan tangan)"^^<http://www.w3.org/2001/XM...
selut_capang	"https://drive.google.com/file/d/1n-Ycq..."	"0:00:13-0:00:15, 0:00:43-0:00:44, 0:"	"gerakan membuka kedua tangan kem..."	"Tangan"^^<http://www.w3.org/2001/XM...
selut_jambret	"https://drive.google.com/file/d/1n-Ycq..."	"0:00:29-0:00:30, 0:00:58-0:00:59, 0:"	"gerakan tangan kanan dan kiri yang d..."	"Tangan"^^<http://www.w3.org/2001/XM...
jalak_pengkor_kanan	"https://drive.google.com/file/d/17qN2..."	"0:00:20-0:00:23, 0:01:34-0:01:37"^^	"gerakan kaki terpatang-patang ke arah l..."	"Kombinasi(kaki dan tangan)"^^<http://www.w3.org/2001/XM...
kanan_tarik	"https://drive.google.com/file/d/16daU..."	"0:00:30-0:00:31, 0:01:00-0:01:02, 0:"	"gerakan kaki yang ditarik dari arah ka..."	"Kombinasi(kaki dan tangan)"^^<http://www.w3.org/2001/XM...
adeg_adeg	"https://drive.google.com/file/d/1c1zF..."	"0:00:05-0:00:12"^^<http://www.w3.org/2001/XM...	"sikap kaki diregangkan merendah, t..."	"Kaki"^^<http://www.w3.org/2001/XM...
obah_bahu	"https://drive.google.com/file/d/1H2Gz..."	"0:02:03-0:02:05, 0:02:13-0:02:14, 0:"	"menggerakkan bahu ke atas dan ke b..."	"Bahu"^^<http://www.w3.org/2001/XM...

Gambar 4. 9. Hasil query SPARQL data setiap Gerakan

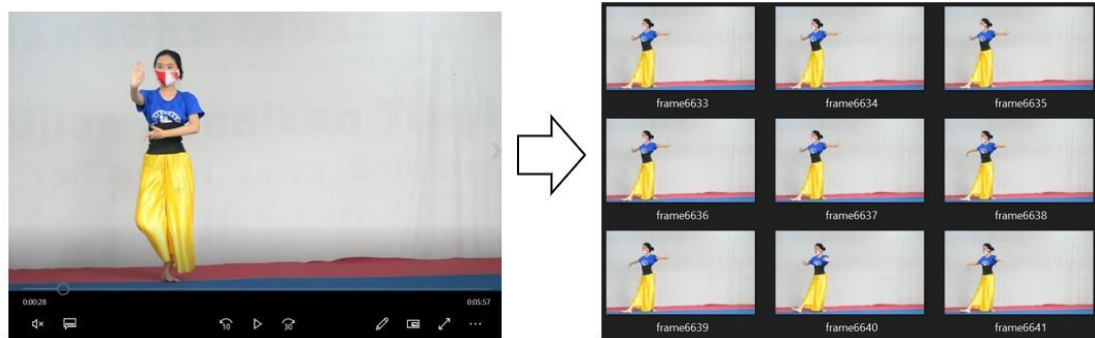
Hasil pengujian Gambar 4.9. menunjukkan *data property* dari keseluruhan individual *kelas* Gerakan, yang artinya pengujian *data property* tersebut menghasilkan sesuai dengan pertanyaan yang ada.

4.4. Preprocessing Data

Bagian ini adalah bagian awal untuk memproses klasifikasi sebuah gerakan tari dengan membagi dataset yang digunakan, *extracting frames*, dan proses *cropping* dan *resizing*.

4.4.1. Extracting Frames

Proses *extracting frames* adalah proses yang paling utama, yaitu memotong-motong video menjadi bagian-bagian gambar atau *frames* untuk dijadikan dataset pada klasifikasi gambar. Gambar 4.10. menunjukkan bagian dari video berdurasi 6:14 menit dan menghasilkan 1.870 gambar, karena *frame rate* pada penelitian ini menggunakan 5 *fps*. Dataset keseluruhan memiliki 6 video, lalu total keseluruhan data yang di *extract* sekitar 11.220 gambar. Dari keseluruhan data tersebut data dipilah sesuai dengan gerakan/kelas yang ada, dan data yang akan dipakai untuk klasifikasi terdapat 2.750 data dengan 13 kelas.



Gambar 4. 10. Hasil Video Setelah di Extract

4.4.2. *Cropping*

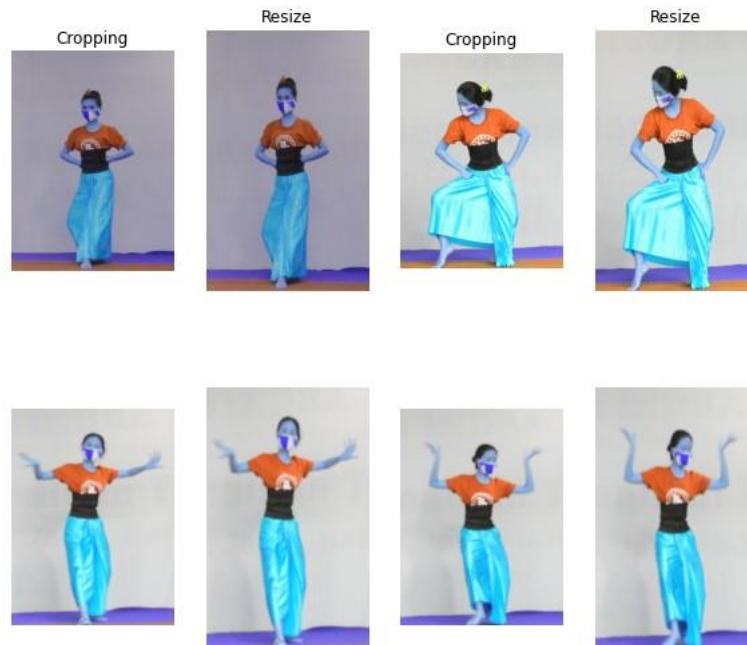
Setelah dataset terkumpul, langkah selanjutnya adalah *cropping* keseluruhan data. *Cropping* dilakukan secara manual dengan ukuran yang berbeda-beda karena piksel dan posisi objek yang berbeda-beda. Untuk piksel yang berukuran 1920x1080 dipotong menjadi 480x650 piksel dan 640x360 piksel dipotong menjadi 280x450 piksel.



Gambar 4. 11. Contoh cropping gambar

4.4.3. *Resizing*

Resizing dilakukan ke keseluruhan data dengan ukuran 280x450 piksel. Tujuannya agar data yang digunakan tidak memiliki ukuran yang berbeda-beda. Gambar 4.12. menunjukkan hasil *resizing* dari data yang sudah di *cropping* sebelumnya.



Gambar 4. 12. Gambar setelah dilakukan cropping dan resize ke seluruh gambar/data

4.4.4. Dataset

Setelah melakukan proses *extracting frames*, *cropping*, dan *resizing* data yang dihasilkan adalah 2.750 data dengan 13 kelas. Selanjutnya adalah pembagian dataset untuk data latih, data validasi, dan data uji yang dijelaskan secara detail pada Tabel 4.3. dan Tabel 4.4.

Tabel 4. 3. Jumlah Data Setiap Kelas

Kelas	Jenis Data	
	Data Latih dan Validasi	Data Uji
Adeg-Adeg	175	45
Cindek	175	45
Galeong Jedag	169	40
Jalak Pengkor Kanan	169	40
Jalak Pengkor Kiri	172	40
Kanan Tarik	171	44
Kiri Tarik	175	45
Obah Bahu	173	44
Ranggah Kanan Kiri	160	40
Selut Capang	160	40
Selut Jambret	160	40
Sembah	173	45

Tumpang Tali	168	42
Total	2200	550

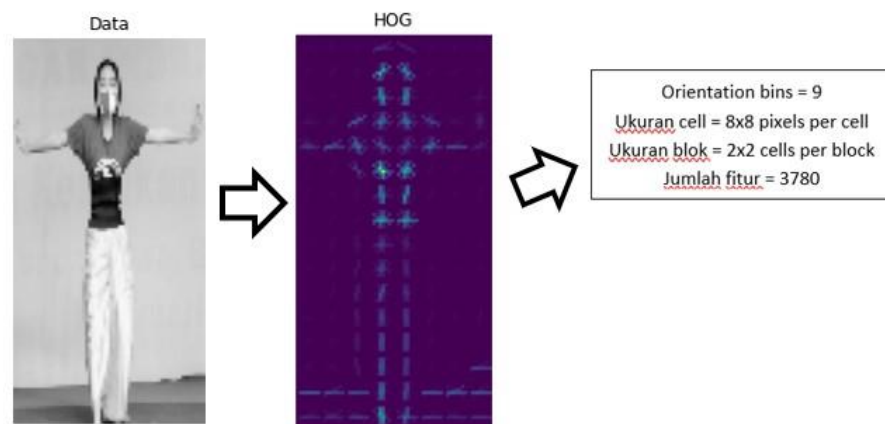
Untuk pembagian data latih dan data validasi akan dibagi menggunakan **train_test_split** dengan perbandingan data latih 80% dan data validasi 20% seperti pada Tabel 4.4.

Tabel 4. 4. Pembagian Data Latih, Validasi, dan Uji.

Jenis Data	Jumlah Data
Data Latih	1760 data
Data Validasi	440 data
Data Uji	550 data

4.5. Ekstraksi Fitur HOG

HOG atau *Histogram of Oriented Gradient* adalah ekstraksi fitur yang mendeskripsikan sebuah gambar menjadi histogram gradien dan kemudian menggunakan histogram tersebut untuk membuat matriks 1 dimensi yang akan digunakan untuk melatih model. Berikut adalah hasil gambar yang di ekstraksi fitur menggunakan HOG:



Gambar 4. 13. Contoh gambar yang sudah di Ekstraksi Fitur menggunakan HOG

Gambar 4.13. adalah contoh gambar sebelum dan setelah di ekstraksi fitur. Penelitian ini akan membandingkan beberapa parameter, yaitu *pixels per cell*: 5x5, 8x8, 10x10, dan 16x16, lalu *cells per block*: 1x1 dan 2x2, dan *orientation bins*: 9 dan akan dipilih parameter dengan nilai metrik evaluasi terbaik.

4.6. Analisis Pengujian

Data yang digunakan untuk klasifikasi, yaitu data tanpa dan dengan ekstraksi fitur HOG dan juga membandingkan nilai F1 dan akurasi dengan *metric Manhattan* dan *Euclidean distance* pada proses tersebut. Untuk pengujian pertama adalah membandingkan jumlah parameter *pixels-per-cell* dan *cells-per-block* yang berbeda-beda dan terdiri dari 13 kelas yang bertujuan untuk memilih parameter terbaik yang akan digunakan. Untuk k yang dipakai pada proses ini adalah 5 dengan *metric euclidean distance* atau pada *classifier* dituliskan *metric=minkowski* dan $p=2$ yang merupakan p untuk *Euclidean distance*, dan parameter tersebut sudah otomatis terpasang, maka proses ini menggunakan nilai k dan *metric* tidak ditentukan oleh penulis. Hasil pengujian tersebut akan ditunjukkan pada Tabel 4.5.

Tabel 4. 5. Nilai pengukuran menggunakan parameter HOG yang berbeda-beda

<i>Pixels per Cell</i>	<i>Cells per Block</i>	<i>Precision</i>	<i>Recall</i>	<i>F1</i>	<i>Acc</i>	<i>Runtime</i>
4x4	2x2	0.9097	0.9010	0.9003	0.9023	27.6s
5x5	2x2	0.9117	0.9049	0.9038	0.9068	15.7s
8x8	2x2	0.9309	0.9260	0.9253	0.9273	6.39s
10x10	2x2	0.9220	0.9168	0.9148	0.9182	3.33s
16x16	2x2	0.9226	0.9189	0.9180	0.9205	1.23s
4x4	1x1	0.8792	0.8752	0.8735	0.8773	7.72s
5x5	1x1	0.8723	0.8656	0.8636	0.8682	4.46s
8x8	1x1	0.8964	0.8912	0.8898	0.8932	1.85s
10x10	1x1	0.8763	0.8701	0.8677	0.8727	1.06s
16x16	1x1	0.9070	0.9025	0.9023	0.9045	472ms

Hasil pengujian pertama dengan variasi *pixels per cell* dan *cells per block* yang berbeda-beda menghasilkan nilai F1 dan akurasi yang berbeda-beda dengan waktu *running model* yang berbeda pula. Metrik evaluasi yang digunakan adalah *precision*, *recall*, F1 dan akurasi karena *precision* adalah melihat rasio objek positif dari keseluruhan data terprediksi positif, lalu *recall* adalah melihat rasio objek positif dari keseluruhan data yang benar-benar terprediksi positif, nilai F1 adalah *weighted average* dari nilai *precision* dan *recall*. Nilai F1 dikatakan memiliki nilai yang cukup baik atau sangat baik, ketika *precision* dan *recall*nya memiliki nilai yang seimbang. Begitupun jika nilai F1 dan akurasinya seimbang dan nilai keduanya sama-sama memiliki nilai tertinggi, maka model tersebut adalah yang terbaik dan yang akan dipakai penulis untuk tahap selanjutnya.

Tabel 4.5. menunjukkan untuk *pixels per cell* yang bernilai kecil, waktu *running model*nya mencapai puluhan detik, karena jumlah fitur yang dihasilkan sangat besar, maka membutuhkan waktu *running model* yang lebih lama dibandingkan dengan nilai *pixels per cell* yang besar karena menghasilkan nilai fitur yang lebih kecil. Hal tersebut dikarenakan bergantung pada perhitungan *pixels per cell*, semakin besar *pixels per cell* yang digunakan, maka akan menghasilkan nilai

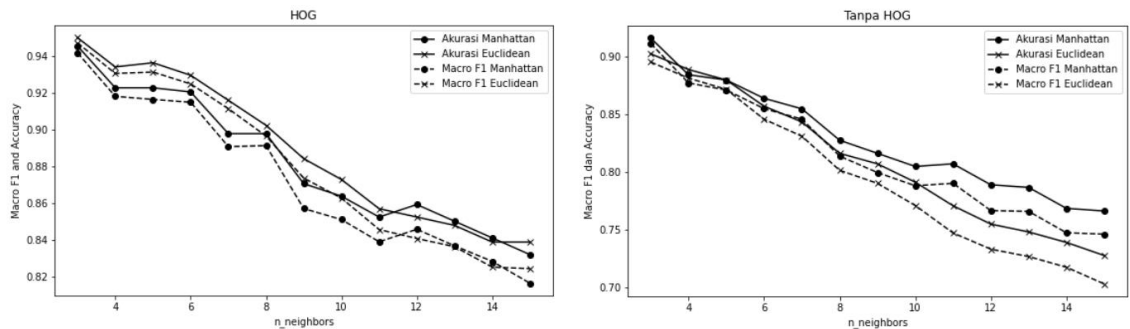
fitur yang lebih kecil dan waktu yang dibutuhkan untuk *running model* juga lebih cepat.

Pengujian kedua adalah memilih k atau nilai tetangga terbaik dengan menggunakan data tanpa HOG dan data HOG dengan parameter HOG yang sudah ditentukan dari tahap sebelumnya, yaitu ukuran *pixels per cell* 8x8 dan *cells per block* 2x2. Nilai k yang akan digunakan yaitu 3 sampai dengan 13 karena untuk melihat bagaimana hasil dari metrik evaluasi jika k nya hingga mendekati jumlah kelas data dan menggunakan dua *distance metric* yang berbeda, yaitu *Manhattan* dan *Euclidean*. Hasil pengujian tersebut akan ditunjukkan pada Tabel 4.6.

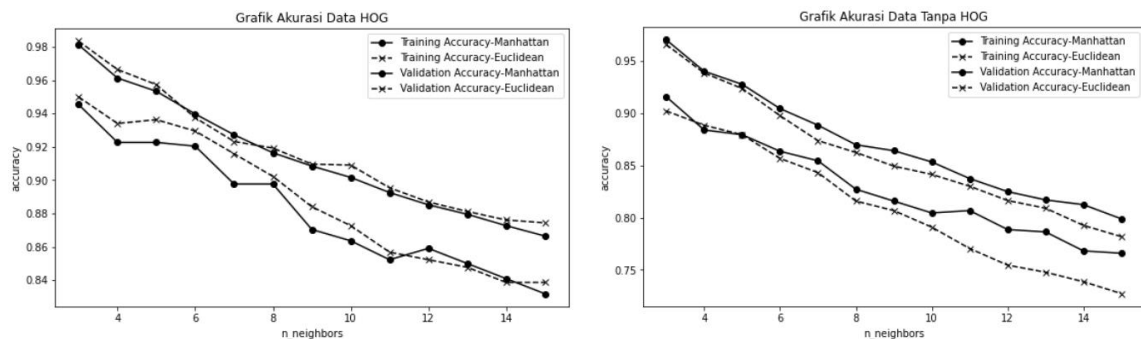
Tabel 4. 6. Nilai pengukuran pada data HOG dan tanpa HOG dengan $n = 3$ s/d 13

<i>Metric</i>	k	Data Tanpa HOG				Data HOG			
		<i>Precision</i>	<i>Recall</i>	F1	<i>Acc</i>	<i>Precision</i>	<i>Recall</i>	F1	<i>Acc</i>
<i>Manhattan</i>	3	0.9241	0.9198	0.9193	0.9205	0.9486	0.9471	0.9469	0.9477
	4	0.8990	0.8913	0.8901	0.8932	0.9334	0.9309	0.9307	0.9318
	5	0.8816	0.8754	0.8754	0.8773	0.9253	0.9216	0.9213	0.9227
	6	0.8815	0.8705	0.8711	0.8727	0.9179	0.9144	0.9136	0.9159
	7	0.8558	0.8476	0.8463	0.8500	0.9098	0.9074	0.9054	0.9091
	8	0.8441	0.8334	0.8316	0.8364	0.9067	0.9025	0.8894	0.9045
	9	0.8129	0.8008	0.7960	0.8045	0.8981	0.8928	0.8900	0.8955
	10	0.8080	0.7964	0.7898	0.8000	0.8964	0.8885	0.8857	0.8909
	11	0.7908	0.7830	0.7725	0.7864	0.8810	0.8719	0.8687	0.8750
	12	0.7701	0.7643	0.7546	0.7682	0.8727	0.8656	0.8620	0.8682
	13	0.7502	0.7437	0.7354	0.7477	0.8666	0.8587	0.8548	0.8614
<i>Euclidean</i>	3	0.9312	0.9257	0.9264	0.9273	0.9599	0.9587	0.9587	0.9591
	4	0.9165	0.9091	0.9096	0.9114	0.9390	0.9354	0.9354	0.9364
	5	0.8843	0.8818	0.8807	0.8841	0.9257	0.9240	0.9231	0.9250
	6	0.8725	0.8649	0.8645	0.8682	0.9221	0.9188	0.9181	0.9205
	7	0.8497	0.8414	0.8388	0.8455	0.9095	0.9072	0.9051	0.9091
	8	0.8326	0.8231	0.8194	0.8273	0.9105	0.9073	0.9049	0.9091

	9	0.8182	0.8019	0.7969	0.8068	0.8988	0.8954	0.8912	0.8977
	10	0.8070	0.7902	0.7832	0.7955	0.8831	0.8794	0.8765	0.8818
	11	0.7874	0.7697	0.7612	0.7750	0.8843	0.8792	0.8753	0.8818
	12	0.7922	0.7768	0.7680	0.7818	0.8817	0.8772	0.8747	0.8795
	13	0.7864	0.7674	0.7575	0.7727	0.8726	0.8680	0.8642	0.8705



(a)

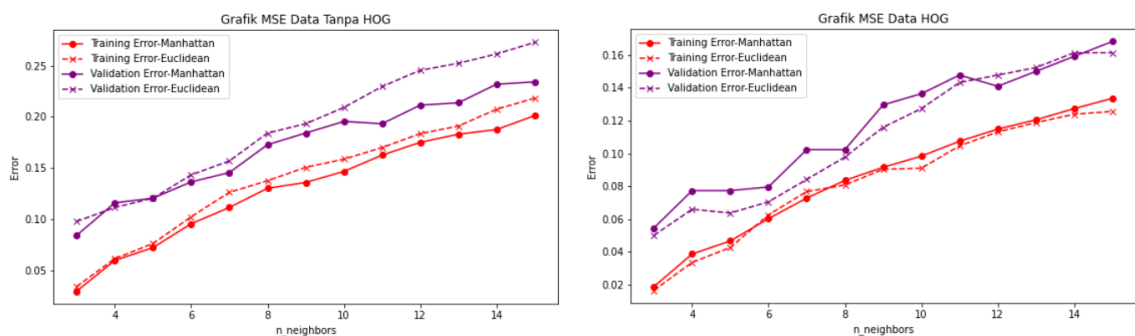


(b)

Gambar 4. 14. Grafik (a) menunjukkan perbedaan nilai akurasi dan f1 metric manhattan dan euclidean pada data HOG dan data tanpa HOG, dan Grafik (b) menunjukkan perbedaan nilai akurasi data latih dan data validasi metric Manhattan dan Euclidean pada data HOG dan Tanpa HOG

Tabel 4.6. menunjukkan perbedaan hasil F1 dan akurasi dari data tanpa HOG dan data HOG yang cukup signifikan. Seperti pada Gambar 4.14. (a) menunjukkan perbedaan nilai akurasi dan macro F1 pada metric *Manhattan dan Euclidean* memiliki perbedaan yang sangat sedikit, baik pada data yang menggunakan HOG maupun tidak. Namun jika dilihat secara detail dan melihat ke Tabel 4.6. akurasi pada metric *Euclidean* lebih baik dibandingkan metric *Manhattan* untuk kedua data. Lalu perbedaan nilai pengukuran pada data HOG pun lebih baik dibandingkan dengan yang tidak menggunakan HOG, hal ini

kemungkinan disebabkan oleh nilai fitur yang dipakai. Pada fitur HOG yang melokalkan fitur di antara beberapa gambar dan menormalisasikan warna pada sebuah citra, maka fitur yang menggunakan HOG lebih menghasilkan nilai akurasi dan F1 lebih baik. Sedangkan pada Gambar 4.14. (b) adalah nilai akurasi pada data latih dan data validasi yang menunjukkan bahwa akurasi data latih pada data HOG dengan metric *euclidean distance* lebih baik dibandingkan dengan metric *manhattan distance*, begitu pula dengan nilai akurasi data validasi. Sebaliknya pada data tanpa HOG yang nilai akurasi data latih *Manhattan distance* lebih baik dibandingkan dengan *Euclidean distance*, namun akurasi pada data validasi *Euclidean distance* lebih baik dibandingkan dengan *Manhattan distance*.



Gambar 4. 15. Nilai error pada data latih dan validasi manhattan dan Euclidean dengan data HOG dan Tanpa HOG

Jika diperhatikan nilai pengukuran-pengukuran pada Tabel 4.6., dengan seiring bertambahnya nilai k maka nilai akurasinya pun semakin menurun baik pada data latih maupun data validasi, begitu pun pada nilai error yang semakin meningkat. Hal tersebut kemungkinan karena model mengalami *overfitting* yang artinya data yang digunakan memiliki bias yang rendah tetapi memiliki varians yang tinggi. Kemungkinan hal tersebut dapat terjadi karena kelas yang digunakan banyak, lalu ada beberapa kelas memiliki ciri yang serupa, sehingga jarak data antar kelas yang memiliki ciri yang serupa akan berdekatan dan semakin banyak k yang digunakan, maka kelas yang digunakan adalah kelas yang dominan. Kemungkinan terakhir adalah karena kurangnya variasi pada data yang digunakan.

4.7. Evaluasi

Setelah melakukan analisis pengujian, nilai terbaik yang didapatkan dengan menggunakan serangkaian proses dan ditunjukkan pada Tabel 4.7. untuk data tanpa HOG dan Tabel 4.8. untuk data HOG.

Tabel 4. 7. Nilai pengukuran pada data tanpa HOG

<i>metric</i>	<i>k</i>	<i>Precision</i>	<i>Recall</i>	F1	<i>Acc</i>
<i>Manhattan</i>	3	0.8483	0.8472	0.8422	0.8527
	4	0.8213	0.8228	0.8176	0.8291
	5	0.7810	0.7838	0.7774	0.7909
<i>Euclidean</i>	3	0.8257	0.8257	0.8211	0.8327
	4	0.7845	0.7877	0.7799	0.7964
	5	0.7743	0.7756	0.7657	0.7836

Tabel 4. 8. Nilai pengukuran pada data HOG

<i>Metric</i>	<i>n</i>	<i>Precision</i>	<i>Recall</i>	F1	<i>Acc</i>
<i>Manhattan</i>	3	0.9014	0.8978	0.8954	0.9018
	4	0.8985	0.8948	0.8936	0.8982
	5	0.8933	0.8895	0.8874	0.8927
<i>Euclidean</i>	3	0.9108	0.9055	0.9039	0.9091
	4	0.9118	0.9042	0.9032	0.9073
	5	0.9030	0.8971	0.8949	0.9000

Terlihat dari Tabel 4.7. dan Tabel 4.8. bahwa *k* 3 sampai 5 menghasilkan nilai F1 dan akurasi yang selisihnya tidak berbeda jauh, tetapi metric *Euclidean distance* menghasilkan nilai yang lebih baik untuk data HOG dan metric *Manhattan distance* menghasilkan nilai yang lebih baik untuk data yang tidak menggunakan HOG. Berbeda dengan hasil dengan menggunakan data validasi yang kedua data menghasilkan nilai terbaik dengan metric *Euclidean distance*. Selanjutnya akan melihat bagaimana hasilnya jika diaplikasikan ke klasifikasi gambar dengan

menggunakan $k = 3$ dan metric *Euclidean distance* untuk data HOG, metric *Manhattan distance* untuk data tanpa HOG.

Tabel 4. 9. Confusion Matrix Data Tanpa HOG

Kelas	adeg-adeg	cindek	galeong jedag	jalak pengkor kanan	jalak pengkor kiri	kanan tarik	kiri tarik	obah bahu	rangguh kanan kiri	selut capang	selut jambret	sembah	tumpang tali
adeg-adeg	45	0	0	0	0	0	0	0	0	0	0	0	0
cindek	2	43	0	0	0	0	0	0	0	0	0	0	0
galeong jedag	0	1	25	6	4	1	0	0	0	0	3	0	0
jalak pengkor kanan	0	2	0	34	0	0	0	0	0	0	0	0	4
jalak pengkor kiri	0	0	0	2	35	0	2	1	0	0	0	0	0
kiri tarik	0	0	0	0	0	41	1	0	2	0	0	0	0
kanan tarik	0	0	0	0	0	1	43	0	0	0	1	0	0
obah bahu	0	0	0	0	0	0	0	44	0	0	0	0	0
rangguh kanan kiri	0	0	2	0	0	2	2	0	32	0	2	0	0
selut capang	0	0	0	0	4	0	0	0	0	36	0	0	0
selut jambret	1	2	3	4	0	1	3	0	4	1	17	0	4
sembah	0	0	0	0	2	0	0	0	0	0	1	42	0
tumpang tali	0	0	0	0	0	7	0	0	0	2	0	1	32

Tabel 4. 10. Confusion Matrix Data HOG

Kelas	adeg-adeg	cindek	galeong jedag	jalak pengkor kanan	jalak pengkor kiri	kanan tarik	kiri tarik	obah bahu	rangguh kanan kiri	selut capang	selut jambret	sembah	tumpang tali
adeg-adeg	45	0	0	0	0	0	0	0	0	0	0	0	0
cindek	2	43	0	0	0	0	0	0	0	0	0	0	0
galeong jedag	0	0	32	3	2	0	0	1	1	0	1	0	0
jalak pengkor kanan	0	0	2	37	0	0	0	0	0	0	0	0	1
jalak pengkor kiri	0	0	2	2	35	0	0	1	0	0	0	0	0

kanan tarik	0	0	0	0	0	43	0	0	1	0	0	0	0
kiri tarik	0	0	0	0	0	0	42	0	0	3	0	0	0
obah bahu	0	0	0	0	0	0	0	44	0	0	0	0	0
ranggha kanan kiri	0	0	3	0	0	0	1	0	36	0	0	0	0
selut capang	2	0	0	0	1	0	0	0	0	36	0	0	1
selut jambret	1	0	2	4	0	4	0	0	3	1	24	0	1
sembah	0	0	0	0	0	0	0	0	0	0	0	45	0
tumpang tali	0	0	0	0	2	1	0	0	0	1	0	0	38

Tabel 4.9. dan Tabel 4.10. merupakan *confusion matrix* yang menunjukkan hasil klasifikasi dari 13 kelas, yaitu kotak yang berwarna oranye menunjukan data yang teridentifikasi benar dan kotak yang berwarna merah menunjukan data yang teridentifikasi kelas lain. *Confusion matrix* akan menunjukan seberapa besar kesalahan klasifikasi pada kelas tersebut dan masuk ke kelas mana saja yang tidak teridentifikasi kelas sebenarnya. Seperti pada Gambar 4.15. pada label/kelas galeong jedag yang teridentifikasi adalah kelas jalak pengkor kanan, dan terbukti pada *confusion matrix* bahwa beberapa data yang salah diidentifikasi pada label/kelas galeong jedag lebih banyak di label/kelas jalak pengkor kanan, yaitu sebanyak 6 data yang salah diidentifikasi. Hal tersebut terjadi kemungkinan karena posisi kaki atau posisi tangan kedua gerakan tersebut sama, maka ada beberapa gerakan yang diidentifikasi ke gerakan atau kelas lain yang memiliki ciri yang serupa.

Jika melihat perbandingan antara Tabel 4.9. dan Tabel 4.10., nilai kesalahan pada data yang menggunakan HOG diminimumkan. Seperti pada kelas selut jambret dengan data yang tidak menggunakan ekstraksi fitur HOG, data yang diidentifikasi benar hanya 17 data dari 40 data yang artinya hanya 42% data yang teridentifikasi benar. Lalu kelas selut jambret saat menggunakan ekstraksi fitur HOG data yang diidentifikasi benar terdapat 24 data dari 40 data yang artinya 60% data teridentifikasi benar. Artinya, dengan menggunakan ekstraksi fitur dapat memperbaiki citra pada saat proses klasifikasi.

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Beberapa kesimpulan yang diperoleh dari penelitian ini sebagai berikut:

1. Ontologi tarian jaipong Oray Welang yang telah dibuat pada penelitian ini dapat digunakan sebagai cara menyimpan informasi rinci gerakan tarian. Dengan demikian, siapapun dapat menggunakan model ontologi tersebut untuk mendapatkan pengetahuan atau mempelajari tarian, sehingga upaya pelestarian tarian klasik dapat didukung.
2. Berdasarkan uji coba 3 pertanyaan kompetensi yang sudah disetujui oleh para ahli tari, model ontologi yang sudah dibangun dapat menjawab ketiga pertanyaan kompetensi.
3. Proses pemilihan parameter HOG pada KNN *classifiers* menghasilkan *pixels per cell* 8x8 dan *cells per block* 2x2 sebagai parameter terbaik dengan nilai *precision* 93%, *recall* 92%, *F1* 92% dan akurasi 92%.
4. Klasifikasi tanpa menggunakan HOG menghasilkan nilai *precision* 84%, *recall* 84%, *F1-Score* 84%, dan akurasi 85%, sedangkan klasifikasi menggunakan HOG menghasilkan nilai *precision* 91%, *recall* 90%, *F1-Score* 90%, dan akurasi 90%. Dinyatakan bahwa klasifikasi menggunakan HOG menghasilkan nilai metrik evaluasi yang lebih baik dalam proses pengklasifikasian. Dengan bertambahnya nilai *k*, nilai metrik performa model tersebut semakin menurun.

5.2. Saran

Beberapa saran dari penelitian ini untuk penelitian selanjutnya sebagai berikut:

1. Penelitian selanjutnya dapat menambahkan kelas atau entitas lain, seperti gerakan yang lebih detail (gerakan tangan, kaki, kombinasi), kostum yang dikenakan, musik, dan lain sebagainya pada representasi sebuah tarian agar memperluas informasi ontologi tarian.
2. Melakukan klasifikasi dengan menggunakan model *machine learning* atau ekstraksi fitur lainnya dapat dilakukan pada penelitian selanjutnya

untuk melihat perbandingan dari performa model yang digunakan pada data gerakan tarian.

3. Untuk meningkatkan performa sebuah model dapat dilakukan dengan menambah variasi data, melakukan segmentasi atau *masking* dengan menghilangkan bagian yang tidak diperlukan, karena dapat membantu ekstraksi fitur lebih terfokus pada objek gerakannya saja atau dapat mengurangi terjadinya *overfitting*.

DAFTAR PUSTAKA

- [1] A. E. Foni, G. Papagiannakis and N. Magnenat-Thalmann, "A taxonomy of visualization strategies for cultural heritage applications," *Journal on Computing and Cultural Heritage (JOCCH)*, vol. 3.1, pp. 1-21, 2010.
- [2] S. Giannoulakis, Tsapatsoulis, Nicolas and N. Grammalidis, "Metadata for Intangible Cultural Heritage The Case of Folk Dance," in *Proceedings of the 13th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, 2018.
- [3] T. Mulyadi, "Kajian Gerak Tari Sunda Studi Kasus Tari Jaipong," Institut Seni Indonesia (ISI) , Surakarta, 2016.
- [4] A. Mallik, H. Ghosh and S. Chaudhury, "Nrityakosha : Preserving the Intangible Heritage of Indian Classical Dance," *Journal on Computing and Cultural Heritage*, vol. 2, 2012.
- [5] A. G. Latha, C. Satyanarayana and Y. Srinivas, "Semantic Image Annotation using Ontology And SPARQL," *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, vol. 9, no. 3, pp. 2278-3075, 2020.
- [6] P. B. G. and A. D. M. , "DEEP LEARNING FOR ACTION CLASSIFICATION USING BHARATHNATYAM DATASET," *International Journal of Management, Technology And Engineering*, vol. 8, no. X, pp. 1229-1304, 2018.
- [7] K. Kumar and P. Kishore, "Indian Classical Dance Mudra Classification Using HOG Features and SVM Classifier," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 7, no. 5, pp. 2537-2546, 2017.
- [8] Yulianti, Pengantar Seni Tari., Bandung: Cipta Dea Pustaka., 2009.
- [9] E. Caturwati and L. Ramlan, Gugum Gumbira: Dari Chacha ke Jaipongan., Bandung: Sunan Ambu Press, 2007.

- [10] N. Kurniati, "Asal Usul dan Perkembangan Jaipongan Dewasa ini di Jawa Barat," Tesis Universitas Gadjah Mada, Yogyakarta, 1995.
- [11] "<https://id.wikipedia.org/>," Wikipedia, 7 July 2020. [Online]. Available: <https://id.wikipedia.org/wiki/Jaipongan>. [Accessed 1 August 2020].
- [12] S. R. , B. V. R. and F. D. , "Knowledge engineering: principles and methods," *Data & Knowledge Engineering*, vol. 25, no. 1-2, pp. 161-197, 1998.
- [13] S. Abburu and G. S. Babu, "A Framework for Ontology Based Knowledge Management.," *International Journal of Soft Computing and Engineering (IJSCE)*, pp. 2231-2307, 2013.
- [14] H. Li, W. Li, Q. Cai and H. Li, "A Framework of Ontology Based Knowledge Management System," *IEEE International Conference on Computer Science and Information Technolog*, pp. 374-377, 2009.
- [15] D. L. McGuinness and F. v. Harmelen, "OWL Web Ontology Language overview," in *W3C Recommendation*, 2004.
- [16] N. Noy and D. McGuinness, "Ontology Development 101: A Guide to Creating Your First Ontology," Stanford Knowledge Systems Laboratory Technical Report KSL-01-05, 2001.
- [17] G. Antoniou, P. Groth, F. v. Harmelen and R. Hoekstra, "Constructing Ontologies Manually," in *A Semantic Web Primer Third Edition*, London, The MIT Press, 2012, p. 194.
- [18] I. Horrocks, P. F. Patel-Schneider and F. v. Harmelen, "SHIQ and RDF to OWL: The making of a Web Ontology Language," *Journal of Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 1, no. 1, pp. 7-26, 2003.
- [19] J.-B. Lamy, Owlready: Ontology-oriented programming in Python with automatic classification and high level constructs for biomedical ontologies., 2017.
- [20] S. A. Medjahed, "A Comparative Study of Feature Extraction Methods in Images Classification," *Modern Education and Computer Science Press*, vol. 3, pp. 16-23, 2015.

- [21] O. Sutton, "Introduction to k Nearest Neighbour Classification and Condensed Nearest Neighbour Data Reduction," 2012.
- [22] L. J. Moleong, *Metodologi penelitian kualitatif*, Bandung: Remaja Rosdakarya, 2010.
- [23] F. A. I. A. Putra, F. Utaminingrum and W. F. Mahmudy, "HOG Feature Extraction and KNN Classification for Detecting Vehicle in The Highway," *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, vol. 14, no. 3, p. 231~242, 2020.
- [24] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection".
- [25] C. Q. LAI and S. S. TEOH, "An Efficient Method of HOG Feature Extraction Using Selective Histogram Bin and PCA Feature Reduction," *Advances in Electrical and Computer Engineering*, vol. 16, no. 4, 2016.
- [26] N. Ali, D. Neagu and P. Trundle, "Evaluation of k-nearest neighbour classifier performance for heterogeneous data sets," *SN Applied Sciences*, 2009.
- [27] S. Lagrue, N. Chetcuti-Sperandio, F. Delorme, D. N. Thi, C. M. Thi, K. Tabia and S. Benferhat, "An OntologyWeb Application-based Annotation Tool for Intangible Cultural Heritage Dance Videos," in *Structuring and Understanding of Multimedia heritAge Contents (SUMAC)*, Nice, 2019.
- [28] C. Shorten and T. M. Khoshgoftaar, "A survey on Image Data Augmentation," Creative Commons Attribution 4.0 International License, Florida, 2019.
- [29] D. A. Ledford, *An Introduction to Machine Learning*, MAN Institute, 2019.