



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - KI141502

KLASIFIKASI *STYLE* LUKISAN MENGGUNAKAN SELF-ORGANIZING MAPS (SOM)

REZA ANDRIYUNANTO
5113100140

Dosen Pembimbing
Dr.Eng. Nanik Suciati, S.Kom., M.Kom.
Dini Adni Navastara, S.Kom., M.Sc.

JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2017



TUGAS AKHIR - K1141502

KLASIFIKASI *STYLE* LUKISAN MENGGUNAKAN SELF-ORGANIZING MAPS (SOM)

REZA ANDRIYUNANTO
5113100140

Dosen Pembimbing I
Dr.Eng. Nanik Suciati, S.Kom., M.Kom.

Dosen Pembimbing II
Dini Adni Navastara, S.Kom., M.Sc.

JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya, 2017

[Halaman ini sengaja dikosongkan]



FINAL PROJECT - KI141502

ART *STYLE* CLASSIFICATION USING SELF-ORGANIZING MAPS (SOM)

REZA ANDRIYUNANTO
5113100140

Supervisor I
Dr.Eng. Nanik Suciati, S.Kom., M.Kom.

Supervisor II
Dini Adni Navastara, S.Kom., M.Sc.

DEPARTMENT OF INFORMATICS
FACULTY OF INFORMATION TECHNOLOGY
Sepuluh Nopember Institute of Technology
Surabaya, 2017

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN

**KLASIFIKASI *STYLE* LUKISAN MENGGUNAKAN
SELF-ORGANIZING MAPS (SOM)**

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Rumpun Mata Kuliah Komputasi Cerdas dan Visi
Program Studi S-1 Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh:

REZA ANDRIYUNANTO
NRP: 5113 100 140

Disetujui oleh Dosen Pembimbing Tugas Akhir:

Dr.Eng. Nanik Suciati, S.Kom., M.Kom.
NIP. 197104281994122001



(Pembimbing 1)

Dini Adni Navastara, S.Kom., M.S.
NIP. 198510172015042001

(Pembimbing 2)

SURABAYA
JUNI, 2017

[Halaman ini sengaja dikosongkan]

KLASIFIKASI *STYLE* LUKISAN MENGGUNAKAN SELF-ORGANIZING MAPS (SOM)

Nama Mahasiswa : Reza Andriyunanto
NRP : 5113 100 140
Jurusan : Teknik Informatika, FTIf ITS
Dosen Pembimbing 1 : Dr.Eng. Nanik Suciati, S.Kom., M.Kom.
Dosen Pembimbing 2 : Dini Adni Navastara, S.Kom., M.Sc.

Abstrak

Dalam pengenalan *style* citra lukisan, umumnya dilakukan secara subjective berdasarkan literasi yang ada. *Style* lukisan sendiri ada banyak, mulai dari abstrak sampai *style* modern. Untuk mengenali *style* lukisan seperti itu dibutuhkan fitur objective yang merepresentasikan setiap lukisan berdasarkan *style*.

Tugas akhir ini mengusulkan sebuah metode klasifikasi menggunakan self-organizing maps untuk citra lukisan. Dilakukan ekstraksi fitur sebanyak 50 yang mewakili fitur lokal dan fitur global. Dimana fitur global mencakup fitur warna, sedangkan pada fitur lokal mencakup fitur warna dan fitur komposisi. Pada fitur lokal dibutuhkan proses untuk mendapatkan segmen pada citra, sehingga dilakukan segmentasi dengan menggunakan *K-Means clustering* lalu dilakukan filtering dengan median filter untuk memperbaiki hasil segmentasi. Hasil dari fitur tadi akan diolah dengan menggunakan self-organizing maps sehingga menghasilkan pengelompokan berdasarkan *style* lukisan.

Ujicoba yang dilakukan pada tugas akhir ini menunjukkan bahwa dengan *learning rate* 0.2 dan *k* pada *K-Means* ketika segmentasi 8 menghasilkan akurasi klasifikasi sebesar 53%.

Kata kunci: Self-Organizing Map, *style* lukisan, *K-Means clustering*

[Halaman ini sengaja dikosongkan]

ART STYLE CLASSIFICATION USING SELF-ORGANIZING MAPS

Student Name : Reza Andriyunanto
Registration Number : 5113 100 140
Department : Informatics, FTIf ITS
First Supervisor : Dr.Eng. Nanik Suciati, S.Kom.,
M.Kom.
Second Supervisor : Dini Adni Navastara, S.Kom., M.Sc.

Abstract

In the introduction of image painting style, generally done in a subjective based on existing literacy. Style painting itself there are many, ranging from abstract to modern style. To recognize the style of painting like that required feature objective that represents each painting based on style.

This final project proposes a classification method using self-organizing maps for painting images. 50 feature extractions are performed that represent local features and global features. Where the global features include color features, while in local features include color features and composition features. On the local features required process to get the segment in the image, so it is done segmentation by using K-Means clustering then done filtering with median filter to improve the result of segmentation. The results of these features will be processed by using self-organizing maps to produce groupings based on painting style.

The experiments performed on this final project showed that with learning rate of 0.2 and k on K-Means when segmentation 8 resulted in classification accuracy of 53%.

Keywords: Self-Organizing Map, painting style, K-Means clustering

[Halaman ini sengaja dikosongkan]

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Segala puji bagi Allah SWT yang telah melimpahkan rahmat dan anugerah-Nya sehingga penulis dapat menyelesaikan tugas akhir yang berjudul “**Klasifikasi Style Lukisan Menggunakan Self-Organizing Maps (SOM)**”.

Buku tugas akhir ini disusun dengan harapan dapat memberikan manfaat dalam pengenalan *style* lukisan lebih lanjut. Selain itu, penulis berharap dapat memberikan kontribusi positif bagi kampus Teknik Informatika ITS.

Dalam perancangan, pengerjaan, dan penyusunan tugas akhir ini, penulis banyak mendapatkan bantuan dari berbagai pihak. Penulis ingin mengucapkan terima kasih kepada:

1. Ibu Dr.Eng. Nanik Suciati, S.Kom., M.Kom. selaku dosen pembimbing 1 yang telah memberi ide, nasihat, harapan dan arahan sehingga penulis dapat menyelesaikan tugas akhir dengan tepat waktu.
2. Ibu Dini Adni Navastara, S.Kom., M.Sc. selaku dosen pembimbing 2 yang telah memberi ide, semangat dan arahan sehingga penulis dapat menyelesaikan tugas akhir dengan tepat waktu.
3. Orang tua penulis Bapak Joko Puryanto dan Mama Yuniarti yang telah memberikan dukungan moral, spiritual dan material serta senantiasa memberikan doa demi kelancaran dan kemudahan penulis dalam mengerjakan tugas akhir.
4. Seluruh saudara kandung: dua kakak (mbak Lyana, mas Satria) dan keluarga serta seluruh keluarga besar yang telah memberikan dukungan yang besar baik secara langsung maupun secara tidak langsung.
5. Teman-teman Administrator Lab KCV 2013 (Haqiqi, Ihsan, Nela, Nida) yang sering saling menyemangati dan menjadi teman *refreshing* dalam pengerjaan tugas akhir ini.

6. Teman-teman di Lab KCV: para admin yang telah banyak membantu memfasilitasi dalam pengerjaan serta teman-teman user TA di KCV, yang sudah menemani penulis.
7. Teman teman (Gurat, Ibnu, Ery, Hilman, Ifut) yang selalu mengingatkan mengenai tugas akhir baik tersirat maupun tersurat dan saling memberi semangat dalam pengerjaannya.
8. Teman-teman mahasiswa angkatan 2013 lain yang sama-sama mengarungi empat tahun masa perkuliahan bersama penulis, yang telah memberi penulis banyak pelajaran hidup.
9. Pihak-pihak lain yang tidak bisa penulis sebutkan satu-persatu.

Penulis menyadari masih ada kekurangan dalam penyusunan tugas akhir ini. Penulis mohon maaf atas kesalahan, kelalaian maupun kekurangan dalam penyusunan tugas akhir ini. Kritik dan saran yang membangun dapat disampaikan sebagai bahan perbaikan ke depan.

Surabaya, Juni 2017

Penulis

DAFTAR ISI

LEMBAR PENGESAHAN	v
Abstrak	vii
Abstract	ix
KATA PENGANTAR	xi
DAFTAR ISI	xiii
DAFTAR GAMBAR	xvi
DAFTAR TABEL	xvii
DAFTAR KODE SUMBER	xviii
1 BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah.....	2
1.4 Tujuan Tugas Akhir.....	3
1.5 Manfaat Tugas Akhir.....	3
1.6 Metodologi.....	3
1.7 Sistematika Laporan.....	4
2 BAB II DASAR TEORI	7
2.1 Style Lukisan.....	7
2.1.1 Ekspresionisme.....	7
2.1.2 Impresionisme.....	8
2.1.3 Surealisme.....	8
2.1.4 Post-Impressionism.....	8
2.2 Sistem Koordinat Warna.....	11
2.2.1 Sistem Ruang Warna HSV.....	11
2.3 Ekstraksi Fitur.....	12
2.3.1 Ekstraksi Fitur Global.....	12
2.3.2 Ekstraksi Fitur Lokal.....	13
2.4 Histogram.....	15
2.5 Filter Median.....	16
2.6 Normalisasi.....	17
2.7 K-Means Clustering.....	17
2.8 Self-Organizing Maps.....	18

2.9	Confusion Matrix	20
3	BAB III ANALISIS DAN PERANCANGAN.....	23
3.1	Tahap Analisis.....	23
3.1.1	Deskripsi Umum.....	23
3.1.2	Spesifikasi Kebutuhan Sistem	23
3.1.3	Analisis Permasalahan	24
3.1.4	Analisis Permasalahan Klasifikasi.....	25
3.2	Tahap Perancangan	26
3.2.1	Perancangan Sistem.....	26
3.2.2	Perancangan Data	28
3.2.3	Perancangan Proses	29
4	BAB IV IMPLEMENTASI.....	37
4.1	Lingkungan Implementasi.....	37
4.1.1	Perangkat Keras	37
4.1.2	Perangkat Lunak	37
4.2	Implementasi Tahap Ekstraksi fitur	38
4.2.1	Implementasi Ekstraksi fitur global.....	38
4.2.2	Implementasi Segmentasi	39
4.2.3	Implementasi Filtering dan Pengambilan Segmen	39
4.3	Implementasi Ekstraksi Fitur Lokal	39
4.4	Implementasi Proses Normalisasi	41
4.5	Implementasi Proses Klasifikasi	42
5	BAB V UJI COBA DAN EVALUASI.....	45
5.1	Lingkungan Uji Coba.....	45
5.2	Data Uji Coba.....	45
5.3	Uji Coba Penentuan Parameter learning rate SOM.....	45
5.4	Uji Coba Parameter k pada K-Means.....	47
5.5	Evaluasi	49
5.5.1	Evaluasi Uji Coba Parameter Learning rate.....	50
5.5.2	Evaluasi Uji Coba Parameter K pada K-Means.....	51
6	BAB VI KESIMPULAN DAN SARAN	53
6.1	Kesimpulan	53
6.2	Saran.....	53

DAFTAR PUSTAKA	55
LAMPIRAN.....	57
BIODATA PENULIS.....	83

DAFTAR GAMBAR

Gambar 2.1 Lukisan ekspresionisme [11].....	9
Gambar 2.2 Lukisan impresionisme [11]	9
Gambar 2.3 Lukisan surealisme [11]	10
Gambar 2.4 Lukisan post-impresionism [11]	10
Gambar 2.5 Histogram	15
Gambar 2.6 Contoh filter median.....	16
Gambar 2.7 Ilustrasi K-Means Clustering (a-f menunjukan proses klastering dengan K-Means)	18
Gambar 2.8 Struktur Self-Organizing Map.....	20
Gambar 2.9 Confusion Matrix berdasar Area	21
Gambar 3.1 Struktur Map.....	26
Gambar 3.2 Diagram Alur Sistem.....	28
Gambar 3.3 Contoh citra masukan	29
Gambar 3.4 Gambar dengan ruang warna HSV	30
Gambar 3.5 Histogram dari Gambar 3.4	31
Gambar 3.6 Diagram alur ekstraksi fitur global.....	32
Gambar 3.7 Hasil K-Means.....	32
Gambar 3.8 Hasil median filter	33
Gambar 3.9 Diagram alur segmentasi	33
Gambar 3.10 Diagram alur ekstraksi fitur lokal.....	34
Gambar 3.11 (a) Alur proses training, (b) Alur proses testing	35
Gambar 3.12 Struktur SOM untuk klasifikasi.....	36
Gambar 5.1 k-means dengan k=6.....	48
Gambar 5.2 k-means dengan k=8.....	49
Gambar 5.3 k-means dengan k=10.....	49
Gambar 5.4 Akurasi learning rate dengan K-Means pada k=8.....	50
Gambar 5.5 Perbandingan variasi warna pada setiap citra	52

DAFTAR TABEL

Tabel 5.1 Akurasi setiap learning rate pada K-Means dengan k= 8.....	46
Tabel 5.2 Akurasi setiap kelas pada data training	47
Tabel 5.3 Akurasi setiap kelas pada data testing.....	47
Tabel 5.4 Hasil akurasi dari K-Means dengan k=8 dan k=10	48

DAFTAR KODE SUMBER

Kode Sumber 4.1 Implementasi rata rata <i>hue</i> dan <i>saturation</i>	38
Kode Sumber 4.2 Implementasi nilai histogram	38
Kode Sumber 4.3 Implementasi segmentasi citra	39
Kode Sumber 4.4 Implementasi <i>median filter</i>	39
Kode Sumber 4.5 Implementasi pengambilan region dan fitur	40
Kode Sumber 4.6 Implementasi Pengambilan fitur komposisi.....	40
Kode Sumber 4.7 Implementasi Pengambilan Fitur lokal warna	41
Kode Sumber 4.8 Implementasi Normalisasi	42
Kode Sumber 4.9 Implementasi Proses klasifikasi	43

BAB I PENDAHULUAN

Pada bab ini dibahas hal-hal yang mendasari tugas akhir. Bahasan meliputi latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, metodologi, dan sistematika laporan tugas akhir.

1.1 Latar Belakang

Pada ilmu yang mendalami sejarah sebuah lukisan, *style* sebuah lukisan terus berubah. Terdapat beberapa faktor yang mempengaruhi sebuah *style* lukisan yang mana diantaranya adalah perubahan waktu, perbedaan tempat pelukis berada, serta keadaan psikologis pelukis merubah *style* lukisan yang pelukis lukiskan. *Style* yang berbeda dari lukisan yang berasal dari eropa dan asia juga menjadi penentu penting untuk mengetahui pelukis dari sebuah lukisan tertentu. Dari kasus tersebut, terdapat banyak faktor yang mempengaruhi dalam proses pengelompokan lukisan, dan faktor tersebut nantinya akan disebut dengan sebutan fitur[1].

Setiap pelukis memiliki ciri khas yang digambarkan pada lukisan mereka masing masing. Pada sisi yang lain, sejarawan seni menganalisa dan mengelompokan seni lukisan berdasarkan banyaknya literatur penelitian yang berhubungan dengan seni lukisan. Metode tersebut akan memunculkan sebuah keterbatasan dalam intepretasi fitur yang tidak nampak dengan jelas, karena mereka akan menggunakan opini *subjective*. Oleh karena itu perlu dikembangkan sebuah fitur *objective* yang akan digunakan untuk mengelompokan seni lukisan berdasarkan metode ilmiah.

Pada sebuah penelitian, Lombardi [2] menggolongkan seni lukisan berdasarkan *style* lukisan. Lombardi mengekstrak fitur menjadi cahaya, garis dan warna serta menggolongkan seni lukisan dengan k-nearest neighbor, sedangkan untuk analisa dan visualisasinya menggunakan melalui unsupervised learning, hierarchical clustering, self-organizing maps. Jafarpour dkk [3] dalam penelitiannya mengajukan untuk analisa *style* dari lukisan

dengan menggunakan dual-tree complex wavelet transforms dan Hidden Markov Trees. Zujovic dkk [4] dalam penelitiannya, menggunakan 5 *style* lukisan : *abstract*, *expressionism*, *cubism*, *impressionism*, *pop art*, dan *realism*. Fitur yang diekstrak menggunakan *edge map* sebagai fitur *gray-level* dan histogram sebagai fitur warna pada ruang HSV. Dengan penelitian yang sudah ada, maka untuk pengelompokan seni lukisan dimungkinkan untuk dilakukan dengan menggunakan metode klasifikasi. Klasifikasi sendiri adalah metode pembelajaran data untuk memprediksi nilai dari sebuah atribut.

Pada tugas akhir ini akan digunakan 4 *style* lukisan yang digunakan untuk penggolongan lukisan yang diantaranya adalah *expressionism*, *impresionism*, *post-impresionism*, serta *surrealism*. Sedangkan untuk metode klasifikasinya menggunakan metode Self-Organizing Maps (SOM) yang mana akan memproses 50 fitur yang diekstraksi dari warna dan komposisi. Keuntungan penggunaan metode ini adalah untuk menampilkan data dimensional tinggi kedalam data dimensional rendah. SOM disini digunakan untuk mengklasifikasi lukisan.

1.2 Rumusan Masalah

Rumusan masalah yang diangkat dalam tugas akhir ini dapat dipaparkan sebagai berikut:

1. Bagaimana melakukan ekstraksi fitur pada lukisan?
2. Bagaimana melakukan klasifikasi terhadap fitur yang didapat dari citra lukisan dengan menggunakan *Self-Organizing Map*?
3. Bagaimana melakukan evaluasi terhadap hasil klasifikasi citra lukisan?

1.3 Batasan Masalah

Permasalahan yang dibahas dalam tugas akhir ini memiliki beberapa batasan antara lain:

1. Dataset total yang digunakan adalah sebanyak 160 citra lukisan

2. *Style* lukisan yang digunakan adalah *expressionism*, *impresionism*, *post-impresionism*, serta *surrealism*.
3. Metode diimplementasikan menggunakan bahasa python dengan IDE Spyder.
4. Metode evaluasi yang digunakan adalah *K-Fold Cross Validation*

1.4 Tujuan Tugas Akhir

Tujuan tugas akhir ini adalah melakukan klasifikasi lukisan berdasarkan *style* lukisan menggunakan *Self-Organizing Maps*.

1.5 Manfaat Tugas Akhir

Manfaat dari tugas akhir ini adalah menghasilkan sistem untuk mengklasifikasikan citra lukisan berdasarkan *style* lukisan.

1.6 Metodologi

Tahapan-tahapan yang dilakukan dalam pengerjaan tugas akhir ini adalah sebagai berikut:

1. Studi Literatur

Dalam pengerjaan tugas akhir ini dilakukan penggalian data dan studi literatur yang didapat dari sejumlah referensi. Referensi yang dimaksud berasal dari artikel yang dipublikasikan oleh jurnal dan pencarian melalui media internet yang membahas informasi yang dibutuhkan seperti *K-Means clustering*, ruang warna HSV, dan *Self-Organizing Map*.

2. Analisis dan Desain Perangkat Lunak

Pada tahap ini dilakukan rancang bangun perangkat lunak yang dibuat. Dengan menentukan citra lukisan sebagai masukan, sistem akan melakukan segmentasi, perbaikan citra, ekstraksi fitur dan proses klasifikasi. Hasil dari proses sistem tersebut adalah klasifikasi data lukisan berdasarkan *style* lukisan.

3. Implementasi Perangkat Lunak

Sistem yang diajukan akan dibuat dengan bahasa pemrograman python menggunakan kakas bantu IDE spyder 3 pada platform *desktop*. *Library* yang digunakan untuk mendukung pengerjaan adalah *scikit-learn*, *opencv* serta *numpy*. Kakas bantu pendukung lain diantaranya *Inkscape* untuk dokumentasi, dan *Microsoft Excel* sebagai pengolah angka.

4. Uji Coba dan Evaluasi

Pada tahapan ini, dilakukan uji coba terhadap parameter-parameter yang digunakan pada ekstraksi fitur dan klasifikasi. Hasil dari ujicoba dilakukan evaluasi dengan menghitung nilai akurasi dan presisi.

1.7 Sistematika Laporan

Buku tugas akhir ini bertujuan untuk mendapatkan gambaran dari pengerjaan tugas akhir ini. Selain itu, diharapkan dapat berguna untuk pembaca yang tertarik untuk melakukan pengembangan lebih lanjut. Secara garis besar, buku tugas akhir terdiri atas beberapa bagian seperti berikut:

Bab I Pendahuluan

Bab yang berisi mengenai latar belakang, tujuan, dan manfaat dari pembuatan tugas akhir. Selain itu permasalahan, batasan masalah, metodologi yang digunakan, dan sistematika penulisan juga merupakan bagian dari bab ini.

Bab II Dasar Teori

Bab ini berisi penjelasan secara detail mengenai dasar-dasar penunjang dan teori-teori yang digunakan untuk mendukung pembuatan tugas akhir ini.

Bab III Analisis dan Perancangan

Bab ini berisi tentang analisis dan perancangan desain sistem klasifikasi citra lukisan berdasarkan *style* lukisan.

Bab IV Implementasi

Bab ini membahas implementasi dari desain yang telah dibuat pada bab sebelumnya. Penjelasan berupa kode yang digunakan untuk proses implementasi.

Bab V Uji Coba dan Evaluasi

Bab ini membahas tahap-tahap uji coba. Kemudian hasil uji coba dievaluasi untuk kinerja dari aplikasi yang dibangun.

Bab VI Kesimpulan dan Saran

Bab ini merupakan bab terakhir yang menyampaikan kesimpulan dari hasil uji coba yang dilakukan dan saran untuk pengembangan aplikasi ke depannya.

[Halaman ini sengaja dikosongkan]

BAB II

DASAR TEORI

Pada bab ini diuraikan mengenai dasar-dasar teori yang digunakan dalam pengerjaan tugas akhir dengan tujuan memberikan gambaran secara umum terhadap penelitian yang dikerjakan. Bab ini berisi penjelasan mengenai ekstraksi fitur yang terdiri dari fitur lokal dan fitur global, segmentasi yang menggunakan k-mens clustering dan perbaikan dengan medianblur untuk *preprocessing* sebelum dilakukan ekstraksi fitur lokal, serta metode self-organizing maps untuk klasifikasi data.

2.1 Style Lukisan

Lukisan berupa dua dimensi, dimana biasanya diterapkan pada media kanvas, kertas ataupun kaca. Dalam melukis, peralatan yang biasanya dipakai adalah cat minyak, cat poster, cat air dan sebagainya. Dalam lukisan terdapat beberapa *style* yang sangat beragam yang biasa dinamakan aliran atau *style*. *Style* lukisan menggambarkan ciri khas dan keunikan dari karya seni lukis tersebut. Dalam tugas akhir ini digunakan 4 *style* lukisan yang digunakan untuk klasifikasi, yakni ekspresionisme, impresionisme, surealisme dan *post-impresionism*.

2.1.1 Ekspresionisme

Style lukisan ekspresionisme menggambarkan perasaan yang timbul dari kejadian dari luar yang diterima oleh jiwa disamping dari pancaindera. *Style* lukisan ini beberapa mengacu pada sejumlah seniman dari jerman, serta dari negara austria, perancis dan rusia yang mana aktif pada masa sebelum perang dunia I[5]. Contoh dari lukisan ekspresionisme adalah seperti yang tertera pada Gambar 2.1 Lukisan ekspresionisme.

2.1.2 Impresionisme

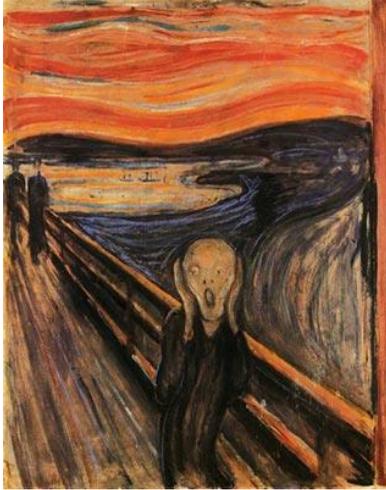
Style lukisan impresionisme ini pertama berkembang di Prancis pada akhir abad 19 dan awal abad 20. Lukisan impresionisme dihasilkan oleh sekelompok orang yang memiliki beragam pendekatan dan teknik dalam melukis. Karakteristik lukisan impresionisme adalah mencoba untuk melukiskan objek dengan akurat dan objektif pada keadaan yang sebenarnya pada efek cahaya dan transien warna[6]. Contoh gambar impresionisme seperti yang tertera pada Gambar 2.2 Lukisan impresionisme.

2.1.3 Surealisme

Lukisan surealisme adalah lukisan yang mementingkan aspek bawah sadar manusia dan nonrasional dalam pencitraannya. Setiap pelukis memiliki hasil eksplorasinya sendiri-sendiri sehingga pelukis diberikan kebebasan dalam mengutarakan fantasinya baik dalam keadaan sadar maupun tidak sadar[7]. Lukisan surealisme dapat dilihat pada Gambar 2.3 Lukisan surealisme.

2.1.4 *Post-Impressionism*

Style post-impressionism muncul ketika adanya penolakan terhadap keterbatasan pada lukisan impresionisme. Istilah *post-impressionism* diciptakan oleh kritikus seni Inggris Roger Fry pada akhir abad 19. Pada lukisan *post-impressionism* lebih mendukung ekspresi yang ambisius dan teknik sapuan dalam lukisan yang digunakan adalah kuas warna yang pendek. Karya pelukis *post-impressionism* menjadi dasar bagi tren kontemporer dan modernisme pada awal abad 20[8]. Contoh lukisan *post-impressionism* terlihat pada Gambar 2.4 Lukisan *post-impressionism*.



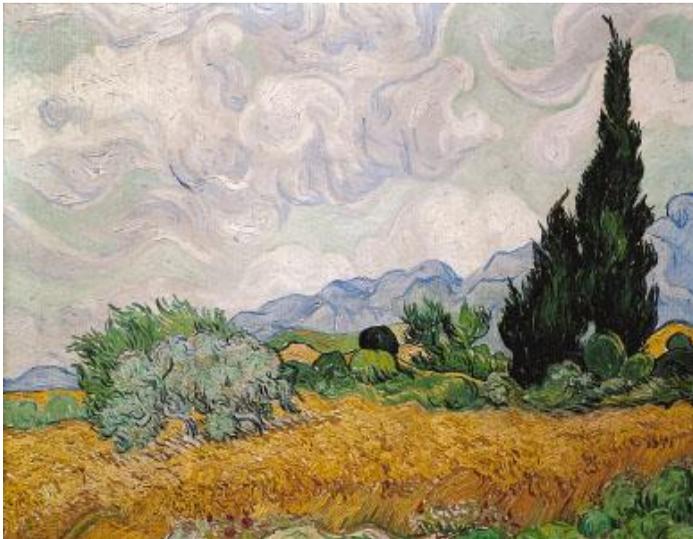
Gambar 2.1 Lukisan ekspresionisme [11]



Gambar 2.2 Lukisan impresionisme [11]



Gambar 2.3 Lukisan surealisme [11]



Gambar 2.4 Lukisan post-impressionism [11]

2.2 Sistem Koordinat Warna

Terdapat beberapa sistem koordinat warna yang diciptakan untuk memenuhi tujuan tertentu. Menurut T. Young, warna tertentu dapat diciptakan dengan penggabungan 3 warna utama dengan persentase tertentu seperti pada persamaan (2.1) [9].

$$C = aC_1 + bC_2 + cC_3 \quad (2.1)$$

Pada tugas akhir ini akan digunakan sistem koordinat warna berdasarkan persepsi warna. Pada umumnya sistem koordinat warna berdasarkan persepsi warna memasukkan tiga atribut yang disebut *hue*, *saturation* dan *brightness*. *Hue* sendiri digunakan untuk membedakan warna (misal merah, kuning, biru) serta menjadi penentu tingkat kemerahan atau kebiruan dari suatu cahaya. *Saturation* sendiri digunakan untuk mengukur persentase warna putih yang ditambahkan pada warna murni seperti warna merah memiliki saturasi tinggi sedangkan merah muda memiliki saturasi rendah. *Brightness* merepresentasikan intensitas cahaya.

Ada beberapa jenis sistem ruang warna pada sistem koordinat warna berdasarkan persepsi warna, dimana pada tugas akhir ini akan digunakan ruang warna *hue*, *saturation*, *value* (HSV).

2.2.1 Sistem Ruang Warna HSV

Pada sistem ruang warna HSV digunakan 3 komponen utama yakni *hue*, *saturation* dan *value* atau *brightness*. *Hue* adalah sudut dari 0 sampai 360 yang menunjukkan warna dimana 0 derajat adalah merah, 60 derajat adalah kuning, 120 derajat adalah hijau, 180 derajat adalah cyan, 240 derajat adalah biru, dan 300 derajat adalah magenta. *Saturation* adalah ukuran untuk menunjukkan seberapa besar kemurnian suatu warna. Semakin sedikit warna putih yang ada pada suatu warna maka saturasi warna tersebut akan semakin tinggi. Komponen ketiga dari HSV adalah *value* atau bisa juga disebut intensitas. Intensitas ini menunjukkan seberapa besar kecerahan dari suatu warna yang dapat bernilai 0 sampai 1 [9].

Sistem ruang warna HSV ini bagus ketika digunakan untuk mendapatkan informasi mengenai warna serta intensitas pada suatu warna pada suatu citra. Dengan demikian maka sistem ruang warna HSV dapat mengidentifikasi warna serta level intensitas yang mana sangat berguna ketika digunakan untuk proses segmentasi.

2.3 Ekstraksi Fitur

Citra lukisan memiliki karakteristik pada garis, warna dan tingkat keterangan yang berbeda antar lukisan dan dikelompokkan pada setiap *style* lukisan. Dari karakteristik tersebut, maka dilakukan ekstraksi fitur warna dan fitur komposisi untuk ekstraksi fitur. Dalam ekstraksi fitur dibedakan menjadi 2 buah jenis berdasarkan area yang diakses yakni fitur global dan fitur lokal.

2.3.1 Ekstraksi Fitur Global

Pada ekstraksi fitur global, area yang diakses untuk mendapatkan fitur adalah keseluruhan dari citra yang digunakan. Terdapat 23 fitur yang diekstrak pada ekstraksi fitur global. Komponen yang diekstrak pada area global adalah fitur warna yang didapat dari nilai histogram serta rata rata nilai *hue* dan *saturation*.

Fitur 1 menunjukkan persamaan untuk mendapatkan nilai rata rata *hue* dari citra seperti yang tampak dari persamaan (2.2).

$$f_1 = \frac{1}{MN} \sum_n \sum_m I_h(m, n) \quad (2.2)$$

Fitur 2 menunjukkan persamaan untuk mendapatkan nilai rata rata *saturation* dari citra seperti yang tampak dari persamaan (2.3).

$$f_2 = \frac{1}{MN} \sum_n \sum_m I_s(m, n) \quad (2.3)$$

Dimana M dan N menunjukkan panjang dan lebar citra sedangkan I_h dan I_s menunjukkan nilai *hue* dan *saturation* pada posisi piksel (m, n) .

Fitur 3 menunjukkan persamaan untuk mendapatkan jumlah dominasi warna dalam histogram dari sebuah citra seperti yang tampak pada persamaan (2.4).

$$f_3 = \text{jumlah dari } \{i | h_H(i) > c \cdot q\} \quad (2.4)$$

Dimana setiap i , dicari nilai dari bin histogram ke i dari komponen *hue* yang lebih besar dari c dikali q dimana c adalah koefisien yang bernilai 0,1 dan q adalah nilai bin tertinggi dalam histogram tersebut. Sehingga didapat jumlah warna dominan dari suatu citra tersebut.

Fitur 4 sampai dengan fitur 23 menunjukkan nilai dari setiap bin histogram nilai *hue* dari suatu citra untuk mengetahui distribusi warna dari citra tersebut seperti pada persamaan (2.5).

$$f_{3+i} = h_H(i) \quad i = 1,2,\dots,20 \quad (2.5)$$

2.3.2 Ekstraksi Fitur Lokal

Pada ekstraksi fitur lokal dilakukan ekstraksi fitur pada area/region tertentu. Area ini dibagi berdasarkan warna yang ada pada suatu citra. Pada fitur lokal ini dibagi menjadi 2, yakni ekstraksi fitur komposisi dan fitur warna.

2.3.2.1 Fitur Komposisi

Pada fitur komposisi di region lokal, dilakukan pencarian nilai *moment* dengan pendekatan statistik. Persamaan *moment* yang digunakan adalah *moment* dengan level 1,2 dan 3[1], [9]. *Moment* level 1 adalah pencarian mean, yakni rata rata dari posisi suatu region. Persamaan ini digunakan untuk mencari pusat dari suatu region. *Moment* level 1 dapat terlihat seperti persamaan (2.6) dan (2.7) untuk setiap masing masing koordinat x dan y atau koordinat horizontal dan vertikal.

$$f_{23+j} = \frac{\sum_{k \in \text{region}_j} x_k}{\text{area region}_j} \quad (2.6)$$

$$f_{26+j} = \frac{\sum_{k \in \text{region}_j} y_k}{\text{area region}_j} \quad (2.7)$$

Dengan x_k adalah nilai setiap posisi x pada region tersebut dan y_k adalah posisi y pada region tersebut. *Moment* level 2 adalah pencarian varian. Varian disini dicari untuk mengetahui persebaran region dari titik pusat region. Pencarian nilai varian dapat dicari dari persamaan (2.8).

$$f_{30+j} = \frac{\sum_{k \in \text{region}_j} [(x_k - \bar{x})^2 - (y_k - \bar{y})^2]}{\text{area region}_j} \quad (2.8)$$

Dengan \bar{x} adalah koordinat pusat x pada region tersebut dan \bar{y} adalah koordinat pusat y pada region tersebut. *Moment* level 3 adalah pencarian *skewness*. *Skewness* disini digunakan untuk pencarian simetri dari region. Persamaan untuk mencari *skewness* terlihat seperti pada persamaan (2.9).

$$f_{32+j} = \frac{\sum_{k \in \text{region}_j} [(x_k - \bar{x})^3 - (y_k - \bar{y})^3]}{\text{area region}_j} \quad (2.9)$$

Dimana j pada 3 persamaan diatas berada direntang 1 sampai 3. Sehingga yang diambil adalah 3 region terbesar.

2.3.2.2 Fitur Warna

Pada fitur warna di region lokal, dilakukan pencarian nilai rata rata dari *hue*, *saturation* dan *value* dari suatu region. Dimulai dari pencarian rata rata nilai *hue* untuk mendapatkan warna yang digunakan pada region tersebut. Persamaan yang digunakan untuk mencari rata rata *hue* adalah seperti yang tampak pada persamaan (2.10).

$$f_{35+j} = \frac{\sum_{(m,n) \in \text{region}_j} I_H(m,n)}{\text{area region}_j} \quad (2.10)$$

Sedangkan rata rata *saturation* pada regional dicari untuk mendapatkan seberapa gelap atau terang karena pengaruh warna

putih dari regional tersebut. Persamaan untuk mencari rata rata *saturation* seperti yang nampak pada persamaan (2.11).

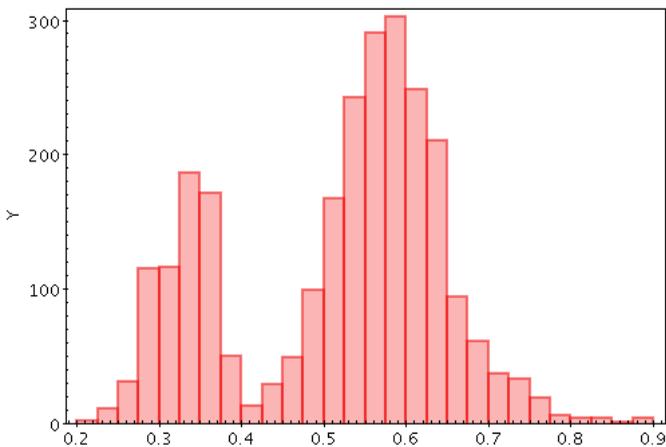
$$f_{40+j} = \frac{\sum_{(m,n) \in region_j} I_s(m,n)}{area\ region_j} \quad (2.11)$$

Untuk komponen ketiga yakni *value*. Rata rata *value* digunakan untuk mencari intensitas dari regional tersebut, sehingga dapat mengidentifikasi seberapa kecerahan dari suatu regional yang dicari dengan persamaan (2.12).

$$f_{45+j} = \frac{\sum_{(m,n) \in region_j} I_v(m,n)}{area\ region_j} \quad (2.12)$$

2.4 Histogram

Histogram merupakan grafik yang menggambarkan persebaran nilai dalam suatu data, dalam kasus citra, histogram menampilkan nilai dari piksel-piksel dalam citra. Histogram berfungsi untuk mengetahui bahwa citra tersebut terang atau gelap dari bentuk grafik yang ada. Apabila posisi puncak grafik dominan di sisi kanan maka dapat dipastikan citra tersebut terang, dan juga



Gambar 2.5 Histogram

sebaliknya, apabila puncak grafik dominan di sisi kiri maka citra tersebut gelap[9].

Histogram ditampilkan dalam bentuk grafik 2 dimensi. Sumbu x pada histogram menyatakan rentang nilai yang dimiliki oleh data, sedangkan sumbu y pada histogram menyatakan banyaknya data yang muncul pada masing masing rentang nilai data pada suatu citra tersebut. Berikut adalah contoh histogram seperti pada Gambar 2.5 Histogram.

2.5 Filter Median

Filter median adalah suatu sekumpulan piksel yang memuat piksel ganjil, dimana nilai tengah dari sekumpulan tadi diubah dengan nilai median dari sekumpulan piksel tersebut. Nilai median didapat dengan mengurutkan nilai piksel dari besar ke kecil atau sebaliknya, lalu diambil nilai piksel yang berada ditengah dari kumpulan piksel tersebut.

Filter median berguna untuk menghilangkan *noise* berupa *salt* dan *papper*. Teknik ini mampu mengurangi *noise* lebih baik daripada metode *linear smoothing* dengan ukuran yang sama[9].

Pada tugas akhir ini filter median digunakan untuk menghilangkan *noise* yang terdapat dari hasil segmentasi *K-Means*. Metode ini perlu dilakukan untuk mendapatkan region warna tanpa adanya *noise* pada citra.

123	125	126	130	140
122	124	126	127	135
118	120	150	125	134
119	115	119	123	133
111	116	110	120	130

Neighbourhood values:
115, 119, 120, 123, 124,
125, 126, 127, 150

Median value: 124

Gambar 2.6 Contoh filter median

2.6 Normalisasi

Normalisasi adalah teknik untuk menstandarkan rentang data sehingga tidak ada satu atau lebih atribut yang terlalu dominan atas atribut yang lain. Pada tugas akhir ini, normalisasi dilakukan pada dataset yang didapat dari hasil ekstraksi fitur.

Normalisasi skala adalah teknik penstandaran data pada suatu rentang tertentu, umumnya pada rentang 0 sampai 1. Diketahui nilai maksimal dari data fitur sebagai X_{max} dan nilai minimal dari data fitur adalah X_{min} . Data yang akan dinormalisasi disimbolkan dengan X [9].

Nilai hasil normalisasi yang dilambangkan dengan huruf Y didapatkan dari rumus perhitungan normalisasi skala ditunjukkan pada persamaan (2.13) berikut:

$$Y = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (2.13)$$

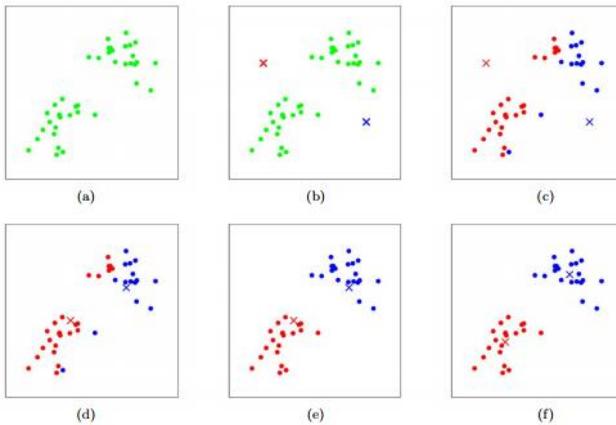
2.7 K-Means Clustering

K-Means Clustering adalah metode tidak terbimbing yang bertujuan membentuk kluster pada data yang belum maupun tidak diketahui klasternya. *K-Means Clustering* membagi data menjadi beberapa bagian.

K-Means clustering mengambil beberapa titik acak sebagai titik pusat tiap kluster. Tiap titik pada data lalu dievaluasi jaraknya dengan masing-masing titik pusat kluster hingga dapat ditemukan tiap titik adalah anggota kluster apa. Dalam perhitungan matematis J adalah jarak antara titik $x_i^{(j)}$ ke titik pusat c_j [1]. Nilai J dapat dihitung dengan formula (2.14) berikut:

$$J = \sum_{j=1}^k \sum_{i=1}^n \left\| x_i^{(j)} - c_j \right\|^2 \quad (2.14)$$

Langkah selanjutnya, data yang sudah dikluster dilakukan penghitungan kembali titik pusat kluster untuk mendapat titik pusat baru menggantikan titik pusat sebelumnya. Titik pusat baru didapatkan dari rata-rata atribut seluruh anggota kluster. Setelah titik pusat baru didapat, dilakukan perulangan untuk menghitung jarak pada tiap titik ke titik-titik pusat kluster yang baru. *K-Means clustering* berhenti ketika titik pusat tiap kluster tidak berubah nilainya ketika dilakukan penghitungan kembali titik pusat kluster. Proses *K-Means clustering* diilustrasikan pada Gambar 2.7



Gambar 2.7 Ilustrasi *K-Means Clustering* (a-f menunjukkan proses klastering dengan *K-Means*)

2.8 Self-Organizing Maps

SOM adalah sebuah teknik visualisasi data yang ditemukan oleh Profesor Teuvo Kohonen yang mana memperkecil dimensi data dengan menggunakan self-organizing neural networks. Metode ini digunakan untuk mempermudah dalam mempelajari data dengan dimensi yang sangat besar. Selain memperkecil data, SOM juga membagi data berdasarkan kelompok yang memiliki persamaan data, sehingga bisa dikatakan SOM menyelesaikan 2 masalah, yakni memperkecil dimensi data serta menampilkan

kesamaan data. SOM sendiri merupakan salah satu metode di dalam Jaringan Syaraf Tiruan yang menggunakan unsupervised learning[9]. Berikut adalah proses training dan klasifikasi dengan menggunakan SOM digambarkan pada Gambar 2.8 Struktur .

Algoritma Self-Organizing Maps disini dibagi menjadi 2, yakni untuk pelatihan dan pengujian. Untuk proses pelatihan dilakukan seperti yang tertera sebagai berikut :

1. Setiap data inputan dilakukan penghitungan jarak dengan setiap neuron pada map. Penghitungan jarak dilakukan dengan persamaan (2.15)

$$J_{(x,w_m)} = \sum_{i=1}^n (x_i - w_{mi})^2 \quad (2.15)$$

Dimana x adalah data masukan dan w_m adalah neuron ke m pada map.

2. Jarak terkecil pada penghitungan setiap data menjadi neuron pemenang, dimana neuron pemenang ini akan dievaluasi bobotnya dengan persamaan (2.16)

$$w_{baru} = w_{lama} + lr \times \theta \times (x - w_{lama}) \quad (2.16)$$

Dimana w_{baru} adalah bobot baru yang akan menggantikan w_{lama} yang merupakan bobot lama pada neuron pemenang dengan lr adalah *learning rate*, θ adalah lebar tetangga serta x adalah data pelatihan.

3. Setelah langkah 1 dan 2 dilakukan pada setiap data, dilakukan penurunan tingkat pembelajaran seiring iterasi dengan persamaan (2.17).

$$\alpha(t) = \alpha_i \left(1 - \frac{t}{t_{max}}\right) \quad (2.17)$$

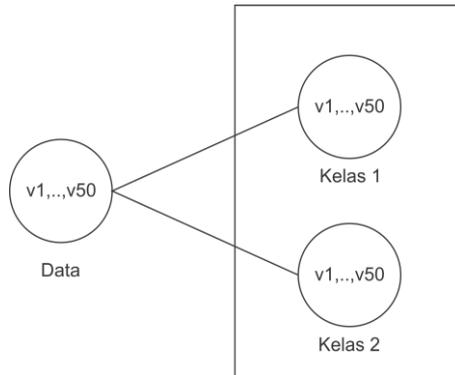
Dimana $\alpha(t)$ adalah *learning rate* pada setiap iterasi (t) dengan α_i adalah *learning rate* awal.

4. Selain itu dilakukan perubahan lebar tetangga (θ) dengan persamaan (2.18).

$$\theta(t) = \theta_i \left(\frac{\theta_f}{\theta_i} \right)^{\frac{t}{t_{max}}} \quad (2.18)$$

$\theta(t)$ adalah lebar tetangga yang akan berkurang seiring dengan iterasi (t). Dimana θ_i adalah nilai awal lebar tetangga dan θ_f adalah nilai akhir tetangga.

Setelah dilakukan proses pelatihan yang menghasilkan map dengan bobot terakhir, dilakukan proses pengujian. Bobot terakhir pada map digunakan sebagai model klasifikasi. Untuk proses pengujian sama dengan proses pelatihan, akan tetapi hanya dilakukan pada tahap 1. Setelah didapatkan neuron pemenang, maka neuron tersebut yang akan menjadi kelas untuk data masukan tersebut. Gambaran untuk struktur SOM dapat dilihat pada Gambar 2.8 Struktur *Self-Organizing Map*.



Gambar 2.8 Struktur *Self-Organizing Map*

2.9 Confusion Matrix

Confusion matrix adalah sebuah statistik klasifikasi yang menyimpan informasi mengenai prediksi kelas dan kelas asli. *Confusion matrix* banyak digunakan untuk menguji performa dari suatu metode klasifikasi [10].

Dari *confusion matrix* bisa didapatkan berbagai informasi mengenai performa *classifier*, salah satunya akurasi. Akurasi

adalah jumlah total data yang prediksi kelas hasil klasifikasinya sesuai dengan data *ground truth* [2]. Rumus perhitungan dari akurasi ditunjukkan pada persamaan (2.19)

$$Akurasi = \frac{TP+TN}{TP+FP+FN+TN} \quad (2.19)$$

	P' (Predicted)	n' (Predicted)
P (Actual)	True Positive (TP)	False Negative (FN)
n (Actual)	False Positive (FP)	True Negative (TN)

Gambar 2.9 *Confusion Matrix* berdasar Area

[Halaman ini sengaja dikosongkan]

BAB III

ANALISIS DAN PERANCANGAN

Bab analisis dan perancangan berisi analisis kebutuhan dan perancangan aplikasi yang akan dibangun. Tahap analisis membahas mengenai analisis kebutuhan yang menjadi dasar dari tahap perancangan.

3.1 Tahap Analisis

Tahap analisis mendefinisikan kebutuhan yang akan dipenuhi dalam pembangunan aplikasi klasifikasi menggunakan self-organizing map. Selain itu dijelaskan pula alasan pengerjaan masing-masing tahap pada tugas akhir ini.

3.1.1 Deskripsi Umum

Pada tugas akhir ini dibangun aplikasi untuk melakukan klasifikasi menggunakan self-organizing map. Data masukan yang digunakan adalah citra digital lukisan. Data keluaran dari aplikasi adalah klasifikasi citra lukisan berdasarkan *style*.

Aplikasi ini diharapkan dapat digunakan untuk membantu peneliti menganalisa hubungan citra lukisan dengan citra lukisan lainnya berdasarkan kedekatan *style* masing masing citra dan membangun fitur objektif dari citra lukisan.

3.1.2 Spesifikasi Kebutuhan Sistem

Pada aplikasi klasifikasi citra lukisan menggunakan self-organizing maps dibutuhkan beberapa tahapan untuk mendapatkan klasifikasi yang baik. Tahapan tersebut antara lain:

1. Ekstraksi Fitur Global

Ekstraksi fitur global dilakukan untuk mendapatkan data dari keseluruhan nilai piksel citra.

2. Ekstraksi Fitur Lokal

Ekstraksi fitur lokal dilakukan untuk mendapatkan data dari sebagian data dari citra yang mewakili karakteristik citra.

3. Klasifikasi

Klasifikasi dilakukan untuk klasifikasi data berdasarkan kelas.

3.1.3 Analisis Permasalahan

Dalam pengidentifikasian citra lukisan dibutuhkan analisa dengan fitur objektif untuk menghindari nilai nilai subjektif. Maka dari itu dibutuhkan fitur dari citra lukisan yang bisa mewakili karakteristik citra lukisan tersebut sehingga bisa diklasifikasikan menurut kedekatan *style* dari citra lukisan tersebut.

Untuk mencapai hasil klasifikasi yang akurat, terdapat beberapa permasalahan yang dapat menurunkan akurasi klasifikasi. Permasalahan-permasalahan tersebut dijelaskan sebagai berikut:

3.1.3.1 Analisis Permasalahan Ekstraksi Fitur Global

Dalam citra lukisan, hal yang mencolok adalah warna, setelah itu kontur dan detail lukisan. Warna dalam citra lukisan memiliki karakteristiknya pada masing masing citra lukisan. Dibutuhkan ruang warna yang bisa merepresentasikan warna dengan baik. Dibutuhkan histogram untuk mengetahui seberapa banyak sebuah nilai warna pada suatu citra digunakan. Hasil histogram yang berupa jumlah nilai dari tiap rentang intensitas warna ini menggambarkan seberapa banyak warna dalam rentang tertentu yang digunakan oleh suatu citra lukisan sehingga dapat digunakan untuk menyimpulkan bahwa suatu citra dominan menggunakan rentang intensitas warna tertentu.

Rata rata nilai dari piksel juga dibutuhkan untuk mengetahui gambaran kasar dari keseluruhan citra. Dalam model citra HSV *hue* dan *saturation* menggambarkan warna dan jumlah warna keabuan dalam citra lukisan.

3.1.3.2 Analisis Permasalahan Ekstraksi Fitur Lokal

Citra lukisan memiliki karakteristik yang berbeda beda, terutama pada kontur dan hanya terletak pada beberapa wilayah pada citra. Untuk mendapatkan segmen tersebut diperlukan pengolahan citra lukisan agar terlihat perbedaan antar segmen pada citra lukisan.

Pengambilan segmen pada citra dilakukan dengan menggunakan *K-Means* clustering yang berguna mengelompokkan warna dalam citra lukisan menjadi K kluster. Dengan begitu akan didapat K kluster yang merupakan wilayah yang akan diambil fiturnya nanti.

Hasil segmentasi pada pengambilan wilayah masih memperlihatkan *salt and papper noise*. Diperlukannya filtering pada hasil segmentasi untuk menghilangkan *noise* tersebut. Cara yang dapat dilakukan adalah dengan menerapkan median filter yang berguna untuk menghilangkan *salt and papper noise* yang diakibatkan proses segmentasi sebelumnya.

Setelah didapat segmen pada citra, maka ekstraksi fitur baru dapat dilakukan. Fitur yang dapat digali pada setiap wilayah adalah posisi koordinat pusat massa, dari posisi ini menunjukkan keberadaan wilayah tersebut pada keseluruhan gambar citra. Pada setiap wilayah juga diambil nilai varian, hal ini bertujuan untuk mendapatkan seberapa jauh distribusi wilayah dari titik pusat wilayah, semakin besar nilai varian berarti semakin besar jangkauan wilayah tersebut dan juga sebaliknya semakin kecil nilai varian, berarti semakin kecil jangkauan wilayah tersebut. Selain varian, diperlukan juga nilai *skewness*, *skewness* ini digunakan untuk mendapatkan derajat asimetris dari wilayah tersebut.

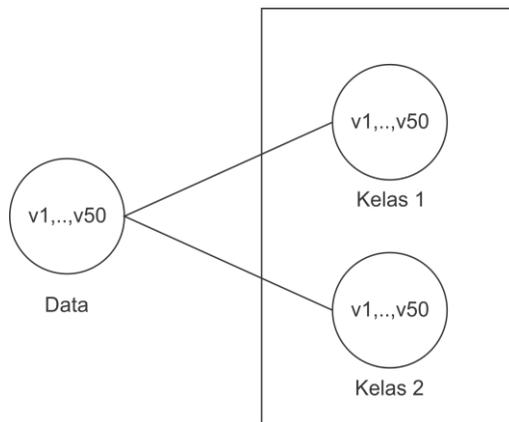
Pada setiap wilayah juga diperlukan nilai rata-rata *hue*, *saturation* dan *value* brightness untuk memperlihatkan dominasi warna, level keabuan dan brightness dari setiap piksel.

3.1.4 Analisis Permasalahan Klasifikasi

Klasifikasi diperlukan untuk mengklasifikasikan sebuah citra masukan ke dalam kategori *style* yang mana. Untuk pendekatan analisa digunakan metode self-organizing maps. Metode ini digunakan karena bisa merepresentasikan vektor dimensi tinggi pada dimensi rendah, sehingga dengan 50 fitur yang diekstraksi sebelumnya dapat direpresentasikan dalam ukuran dimensi rendah.

Klasifikasi dilakukan pada map ukuran 2x1 dengan 50 nilai pada masing-masing neuron seperti yang terlihat pada Gambar 3.1

Struktur Map. Pada tugas akhir ini dilakukan klasifikasi biner sehingga akan membandingkan 1 *style* lukisan dengan masing masing yang lain yang totalnya akan terdapat 6 bagian klasifikasi. Klasifikasi biner dilakukan karena tidak adanya informasi yang diperoleh ketika dilakukan pengelompokan pada map, sehingga klasifikasi dilakukan berpasangan dan penentuan kelas baru bisa dilakukan setelah hasil klasifikasi keluar dengan membandingkan data *ground truth* sehingga terlihat pada neuron mana suatu kelas terkelompok.



Gambar 3.1 Struktur Map

3.2 Tahap Perancangan

Tahap perancangan dilakukan untuk merancang proses secara keseluruhan berdasarkan fungsionalitas dan kebutuhan dari aplikasi klasifikasi *style* lukisan.

3.2.1 Perancangan Sistem

Perancangan sistem dilakukan untuk menggambarkan proses secara keseluruhan dari aplikasi klasifikasi *style* lukisan. Untuk dapat menggunakan aplikasi ini diperlukan citra lukisan dalam bentuk digital untuk menjadi masukan aplikasi ini.

Sistem klasifikasi pada tugas akhir ini memiliki tiga proses utama. Proses pertama adalah ekstraksi fitur lokal. Proses kedua adalah ekstraksi fitur global yang didalamnya diperlukan segmentasi untuk pengambilan segmen dari citra. Proses ketiga adalah klasifikasi seperti yang terdapat pada Gambar 3.2 Diagram Alur Sistem.

Citra digital lukisan didapat dari dataset yang disediakan oleh sebuah galeri seni pada halaman internet[11]. Setelah itu dilakukan ekstraksi fitur yang dibagi menjadi 2 jenis, yakni ekstraksi fitur lokal dan ekstraksi fitur global.

Sebelum dilakukan ekstraksi fitur, citra masukan dikonversi terlebih dahulu ke ruang warna HSV. Pada tahapan ekstraksi fitur global dilakukan perhitungan nilai rata rata *hue* dan *saturation* yang diambil pada setiap piksel pada citra lukisan. Distribusi warna diperoleh dari pembuatan histogram. Dari nilai histogram tersebut diambil nilai dari setiap bin. Untuk mendapatkan warna dominan dari citra dilakukan perhitungan dengan memilih warna yang nilainya lebih dari sepersepuluh nilai maksimal dari histogram.

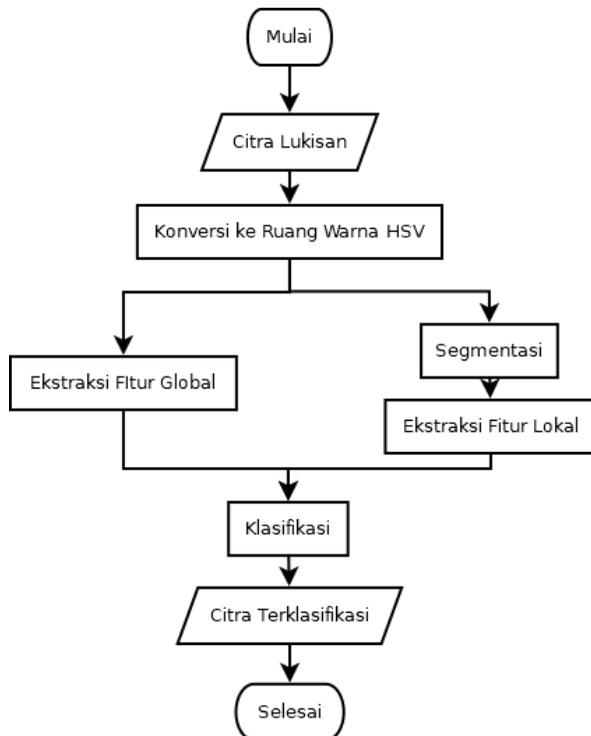
Tahapan ekstraksi fitur lokal dilakukan untuk mendapatkan fitur dari setiap wilayah dari citra, maka dari itu diperlukan segmentasi dari citra terlebih dahulu. Dilakukan proses *K-Means* clustering untuk mendapatkan masing masing wilayah. Untuk memperbaiki hasil wilayah dilakukan filtering dengan menggunakan median filter. Median filter disini berguna untuk menghilangkan *salt and papper noise* dari hasil klastering sehingga dapat memperjelas area segmentasi.

Setelah didapat wilayah wilayah pada citra masukan, dilakukan ekstraksi fitur lokal pada wilayah tersebut. Fitur pertama adalah pencarian titik pusat dari wilayah tersebut yang berupa koordinat titik horizontal dan koordinat titik vertikal. Selanjutnya pencarian nilai varian dan *skewness* dari wilayah tersebut. Fitur diatas dilakukan pada 3 wilayah terbesar. Setelah itu penghitungan nilai rata rata dari nilai *hue*, *saturation* dan *value* yang berada pada wilayah tersebut. Untuk fitur ini dilakukan pada 5 wilayah terbesar.

Tahapan klasifikasi dilakukan setelah semua fitur telah berhasil diekstrak. Klasifikasi dilakukan dengan menggunakan self-organizing maps.

3.2.2 Perancangan Data

Perancangan data dilakukan untuk memastikan pengoperasian aplikasi berjalan dengan benar. Data masukan (*input*) adalah data yang diperlukan dalam pengoperasian aplikasi dan data keluaran (*output*) adalah data yang akan digunakan oleh pengguna.



Gambar 3.2 Diagram Alur Sistem

Data masukan adalah data yang diproses oleh aplikasi klasifikasi *style* lukisan. Data yang digunakan sebagai masukan adalah citra digital lukisan. Berikut adalah contoh gambar masukan seperti yang ada pada Gambar 3.3 Contoh citra masukan.

Data keluaran aplikasi klasifikasi *style* lukisan merupakan klasifikasi data berdasarkan kelas dengan pendekatan analisis melalui visualisasi.



Gambar 3.3 Contoh citra masukan

3.2.3 Perancangan Proses

Perancangan proses dilakukan untuk memberikan gambaran mengenai setiap proses yang terdapat pada aplikasi klasifikasi *style* lukisan.

3.2.3.1 Ekstraksi Fitur Global

Ekstraksi fitur global dilakukan menggunakan data citra digital lukisan. Data input tersebut diubah model warnanya menjadi model warna HSV. Berikut adalah citra dengan model

warna HSV seperti pada Gambar 3.4 Gambar dengan ruang warna HSV.

Citra dengan model warna HSV tersebut diakses nilai *hue* dan *saturation* pada setiap pikselnya untuk mendapatkan rata rata kasar dari keseluruhan piksel citra lukisan.

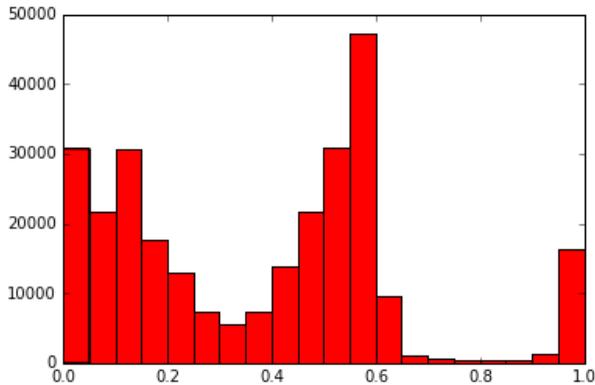
Metode histogram diperlukan untuk mendapatkan nilai distribusi intensitas nilai *hue* dan *saturation* dalam citra seperti pada persamaan (2.2) dan (2.3). Pada histogram seperti pada Gambar 3.5 Histogram dilakukan penghitungan untuk melihat nilai *hue* dominan dengan membandingkan nilai setiap bin histogram dengan sepersepuluh nilai terbesar histogram dari Gambar 3.4 seperti pada persamaan (2.4). Setelah itu nilai *hue* per *bins* digunakan untuk mendapatkan persebaran nilai seperti persamaan (2.5). Diagram alur dari proses ini seperti pada Gambar 3.6 Diagram alur ekstraksi fitur global.



Gambar 3.4 Gambar dengan ruang warna HSV

3.2.3.2 Ekstraksi fitur lokal

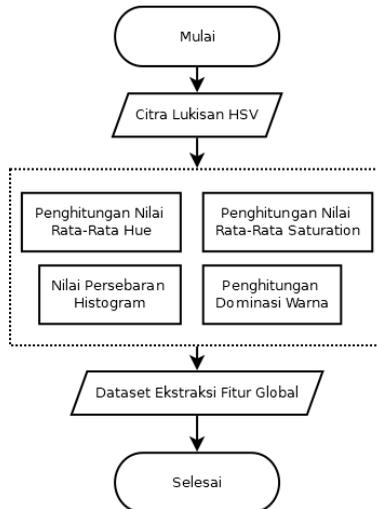
Proses ekstraksi fitur lokal dilakukan untuk pengambilan fitur pada setiap segmen pada citra lukisan. Dilakukan clustering untuk mendapatkan segmen dari citra lukisan. Metode clustering yang dipakai disini adalah *K-Means* clustering. Data hasil segmentasi seperti pada Gambar 3.7 Hasil *K-Means* dilakukan filtering untuk memperjelas area segmentasi. Untuk metode filtering yang dilakukan adalah median filter yang berguna menghilangkan salt and papper *noise* pada citra. Hasil dari filtering dapat dilihat pada Gambar 3.8 Hasil median filter sedangkan alur pada proses segmentasi dapat dilihat pada Gambar 3.9 Diagram alur segmentasi.



Gambar 3.5 Histogram dari Gambar 3.4

Hasil segmentasi dilakukan pengambilan fitur komposisi yang terdiri dari posisi koordinat pusat masa dari setiap wilayah untuk mengetahui posisi wilayah pada keseluruhan citra seperti pada persamaan (2.6) dan (2.7). Selain itu juga dicari varian wilayah untuk mengetahui persebaran wilayah seperti pada persamaan (2.8) dan *skewness* dari tiap wilayah seperti pada persamaan (2.9). Wilayah yang dipakai untuk fitur ini adalah 3 wilayah terbesar. Dari setiap wilayah juga dicari nilai rata rata *hue*, *saturation* dan *value* brightness. Untuk fitur ini, wilayah yang

dipakai adalah 5 wilayah terbesar seperti pada persamaan (2.10), (2.11) dan (2.12). Alur proses ekstraksi fitur lokal ini dapat dilihat pada Gambar 3.10 Diagram alur ekstraksi fitur lokal.



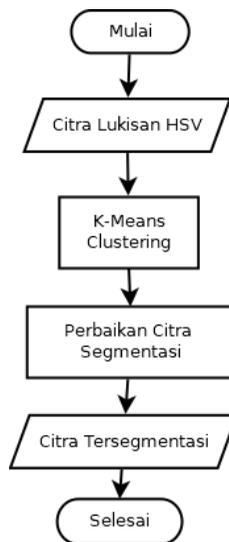
Gambar 3.6 Diagram alur ekstraksi fitur global



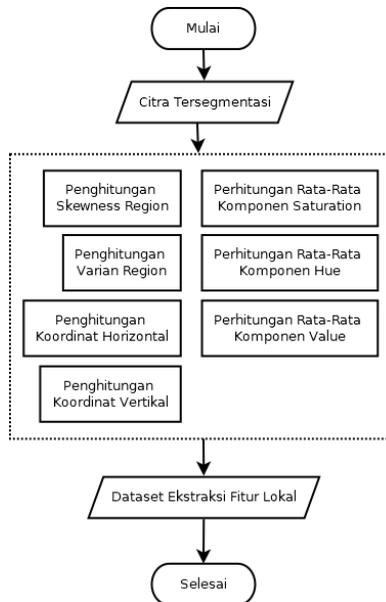
Gambar 3.7 Hasil *K-Means*



Gambar 3.8 Hasil median filter



Gambar 3.9 Diagram alur segmentasi

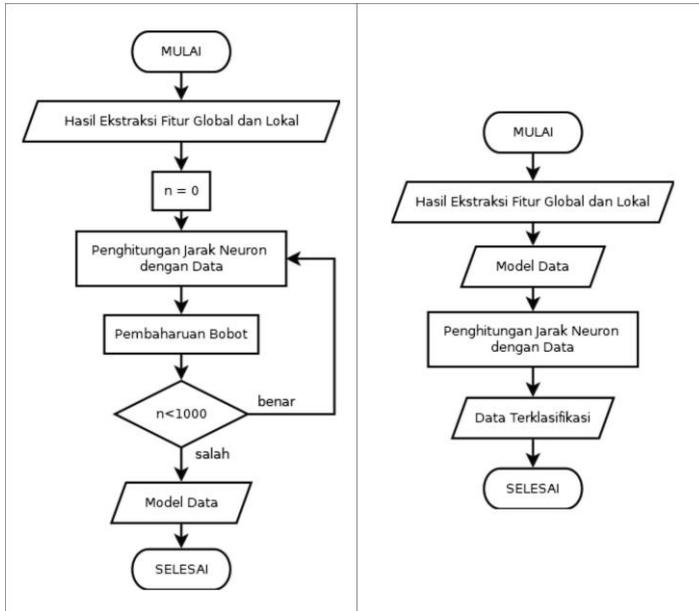


Gambar 3.10 Diagram alur ekstraksi fitur lokal

3.2.3.3 Klasifikasi

Proses klasifikasi *style* lukisan adalah tahapan terakhir dari aplikasi ini. Dari fitur yang telah diekstraksi sebelumnya dapat dilakukan proses klasifikasi. Metode klasifikasi yang digunakan adalah self-organizing maps. Pertama yang dilakukan untuk klasifikasi adalah proses training, hasil ekstraksi fitur yang awalnya $1 \times 1 \times 50$ dimensi untuk setiap citranya dipetakan ke neuron $2 \times 1 \times 50$. Neuron pada awalnya di inisiasi dengan nilai random untuk inputan awal, lalu nilai fitur dihitung kedekatannya dengan pembobotan. Dari hasil pembobotan tadi dicari nilai paling kecil yang juga menggambarkan kedekatan fitur tersebut dengan neuron. Setelah itu dilakukan update bobot neuron pada pemetaan tersebut. Setelah itu dilakukan iterasi untuk memperbaiki hasil pengelompokan fitur tadi sehingga menghasilkan model klasifikasi. Alur proses training ini dapat dilihat pada Gambar 3.11 (a) Alur proses training, (b) Alur proses testing

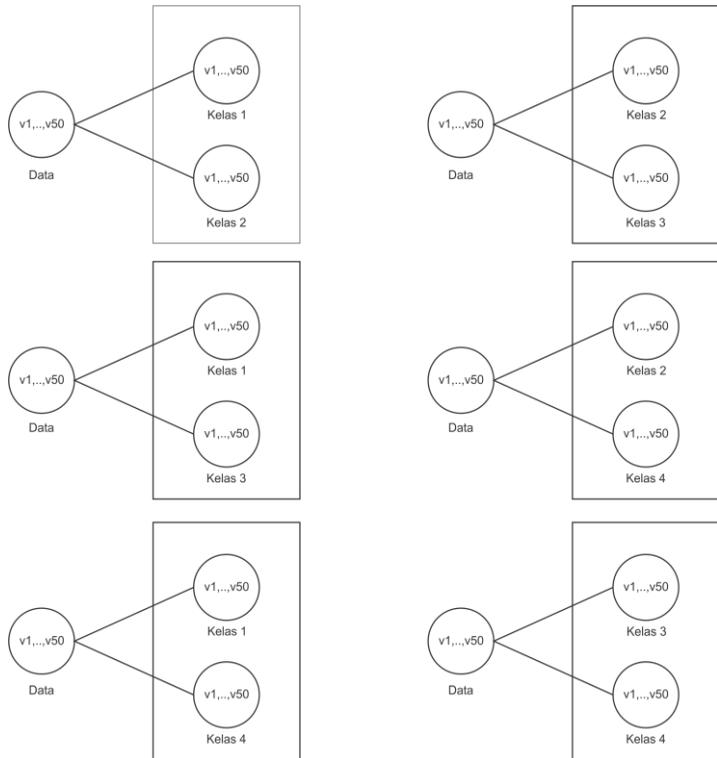
Selanjutnya dilakukan proses testing, proses testing sama dengan proses training, akan tetapi tidak dilakukan iterasi. Nilai terkecil menggambarkan kedekatan fitur dengan neuron, dimana neuron tersebut menjadi kelas bagi fitur tadi. Alur proses testing ini dapat dilihat pada Gambar 3.11 (a) Alur proses training, (b) Alur proses testing.



(a)

(b)

Gambar 3.11 (a) Alur proses training, (b) Alur proses testing



Gambar 3.12 Struktur SOM untuk klasifikasi

Proses klasifikasi ini menggunakan klasifikasi biner, dimana proses klasifikasi dibandingkan pada setiap 2 kelas, sehingga untuk klasifikasi 4 kelas dibutuhkan 6 proses klasifikasi untuk menentukan kelas dari data masukan yang ingin diklasifikasi. Dari 6 klasifikasi tersebut maka akan ditentukan suatu data masukan akan masuk pada kelas mana dengan membandingkan kelas yang paling dekat dengan data masukan pada hasil tiap klasifikasi. Jarak ketetanggaan (θ) yang digunakan adalah 1 karena jumlah neuron yang berjumlah 2 saja. Gambaran struktur SOM untuk klasifikasi dapat dilihat pada Gambar 3.12 Struktur SOM untuk klasifikasi.

BAB IV IMPLEMENTASI

Pada bab ini diuraikan mengenai implementasi perangkat lunak dari rancangan metode yang telah dibahas pada Bab III meliputi kode program dalam perangkat lunak. Selain itu, implementasi dari tiap proses, parameter masukan, keluaran, dan beberapa keterangan yang berhubungan dengan program juga dijelaskan.

4.1 Lingkungan Implementasi

Objek citra yang akan diolah pada implementasi tugas akhir ini adalah sebuah citra *digital* lukisan dengan tipe gambar Joint Photographic Group (JPG).

Dalam implementasi algoritma tersebut, digunakan perangkat-perangkat sebagai berikut:

4.1.1 Perangkat Keras

Lingkungan implementasi pada tugas akhir ini adalah sebuah *personal computer* (PC). Perangkat PC yang digunakan adalah tipe desktop *mini tower* bertipe LENOVO H530S dengan sebuah layar monitor.

Spesifikasi dari PC yang digunakan pada tugas akhir ini adalah: memiliki prosesor Intel Core i3 3240 dengan kecepatan 3,4 GHz dan *Random Access Memory* (RAM) untuk proses menjalankan program sebesar 4,00 GB.

4.1.2 Perangkat Lunak

Lingkungan implementasi pada tugas akhir ini adalah sebuah *personal computer* (PC). Spesifikasi PC dari sisi perangkat lunak menggunakan *IDE spyder 2*.

Selain itu, pada tugas akhir ini dalam melakukan pengolahan angka didukung dengan *software Microsoft Excel*.

4.2 Implementasi Tahap *Ekstraksi fitur*

Tahap *ekstraksi fitur* mengambil fitur sebanyak 50 fitur, yang mana 23 fitur berasal dari ekstraksi fitur global dan 27 fitur dari ekstraksi fitur lokal. Pada ekstraksi fitur lokal terdapat proses segmentasi dan filtering dahulu sebelum dilakukan ekstraksi fitur.

Penjelasan dari masing-masing tahapan tersebut adalah sebagai berikut:

4.2.1 Implementasi Ekstraksi fitur global

Tahap ini mengambil nilai citra dengan model warna HSV, yakni nilai *hue* dan *saturation*, dijumlahkan dahulu seperti pada baris 1-4 lalu diambil nilai rata ratanya pada baris 5-6. Tahapan ini mengimplementasikan persamaan (2.2) dan (2.3).

```

1. for x in range(0,dims[0]):
2.     for y in range (0,dims[1]):
3.         sum_hue = sum_hue + imhsv[x][y][0]
4.         sum_sat = sum_sat + imhsv[x][y][1]
5. Avg_hue = sum_hue/(dims[0]*dims[1])
6. Avg_sat = sum_sat/(dims[0]*dims[1])

```

Kode Sumber 4.1 Implementasi rata rata *hue* dan *saturation*

Pada tahap ini dibuat histogram pada baris 1 untuk diambil nilai bin. Sedangkan perhitungan banyaknya nilai bin seperti pada persamaan (2.4) didapat dari baris kode 2-4 yang mana akan disimpan pada *variable hue_dominated* .

```

1. (n, bins, patches) = plt.hist(z, bins=20)
2. for x in range(0,20):
3.     if n[x] > constant*bin_max:
4.         hue_dominated += 1

```

Kode Sumber 4.2 Implementasi nilai histogram

4.2.2 Implementasi Segmentasi

Segmentasi yang digunakan disini adalah *K-Means* clustering. *K-Means* mengelompokan berdasarkan warna. *K-Means* disini menggunakan library dari opencv pada python.

K-Means disini menggunakan kriteria maksimal iterasi 10 kali atau ketika nilai clustering sudah mencapai 1. Lalu kluster yang dipakai adalah 10 kluster.

```
1. criteria = (cv2.TERM_CRITERIA_EPS + cv2.TERM_CRITERIA_MAX_ITER, 10, 1.0)
2. ret,label,center=cv2.kmeans(image,kluster,None,criteria,10,cv2.KMEANS_PP_CENTERS)
```

Kode Sumber 4.3 Implementasi segmentasi citra

4.2.3 Implementasi *Filtering* dan Pengambilan Segmen

Filtering dilakukan dengan tujuan untuk menghilangkan detail yang mengganggu pada citra, memperhalus citra dan mempertegas area segmentasi. *Filtering* yang digunakan disini adalah median filter dengan menggunakan library dari opencv seperti pada baris 1. Lalu untuk mendapatkan segmen yang sudah difiltering tadi dengan menggunakan regionprops seperti pada baris ke 2.

```
1. median = cv2.medianBlur(v,25)
2. props = regionprops(median)
```

Kode Sumber 4.4 Implementasi *median filter*

4.3 Implementasi Ekstraksi Fitur Lokal

Setelah didapat segmen yang akan digunakan pada ekstraksi fitur, maka perlu dicari 5 nilai terbesar dari segmen yang terbentuk dari proses segmentasi seperti pada baris 1-2. Setelah itu dicari nilai kumulative seperti pada persamaan (2.6) dan (2.7) dari masing masing x dan y pada segmen yang bersangkutan seperti pada baris 3-5. Lalu dibagi dengan ukuran area segmen seperti pada baris 6-11.

```

1. regionsort = sorted(range(len(bigregion)), key=lambda k: bigregion[k])
2. regionsort = regionsort[::-1]
3. rg1 = np.sum(props[regionsort[0]].coords, axis=0)
4. rg2 = np.sum(props[regionsort[1]].coords, axis=0)
5. rg3 = np.sum(props[regionsort[2]].coords, axis=0)
6. f24 = rg1[0]/props[regionsort[0]].area
7. f25 = rg2[0]/props[regionsort[1]].area
8. f26 = rg3[0]/props[regionsort[2]].area
9. f27 = rg1[1]/props[regionsort[0]].area
10. f28 = rg2[1]/props[regionsort[1]].area
11. f29 = rg3[1]/props[regionsort[2]].area

```

Kode Sumber 4.5 Implementasi pengambilan region dan fitur

Untuk fitur lokal *skewness* dan varian wilayah didapat dengan menghitung sesuai dengan persamaan (2.8) dan (2.9) seperti pada baris 1-6.

```

1. rg1 = np.sum((props[regionsort[0]].coords - props[regionsort[0]].centroid)*(props[regionsort[0]].coords - props[regionsort[0]].centroid), axis=0)
2. rg2 = np.sum((props[regionsort[1]].coords - props[regionsort[1]].centroid)*(props[regionsort[1]].coords - props[regionsort[1]].centroid), axis=0)
3. rg3 = np.sum((props[regionsort[2]].coords - props[regionsort[2]].centroid)*(props[regionsort[2]].coords - props[regionsort[2]].centroid), axis=0)
4.
5. f30 = rg1[0]/props[regionsort[0]].area
6. f31 = rg2[0]/props[regionsort[1]].area
7. f32 = rg3[0]/props[regionsort[2]].area
8.
9. f33 = rg1[1]/props[regionsort[0]].area
10. f34 = rg2[1]/props[regionsort[1]].area
11. f35 = rg3[1]/props[regionsort[2]].area

```

Kode Sumber 4.6 Implementasi Pengambilan fitur komposisi

Untuk fitur selanjutnya dicari nilai rata rata *hue saturation* dan *value brightness* dari wilayah dengan cara menjumlahkan

masing masing *hue*, *saturation*, dan *value* brightness seperti pada persamaan (2.10), (2.11) dan (2.12) yang ditunjukkan pada baris ke 1-7. Lalu dibagi dengan luas area region. Disini diambil 5 region terbesar.

```

1. for w in range(0,5):
2.     for x in range(0,len(props[regionsort[w]].coord
   s)):
3.         xi = props[regionsort[w]].coords[x][0]
4.         yi = props[regionsort[w]].coords[x][1]
5.         h = h + hsv[xi][yi][0]
6.         s = s + hsv[xi][yi][1]
7.         v = v + hsv[xi][yi][2]
8. = fh[0]/props[regionsort[0]].area
9. f37 = fh[1]/props[regionsort[1]].area
10. f38 = fh[2]/props[regionsort[2]].area
11. f39 = fh[3]/props[regionsort[3]].area
12. f40 = fh[4]/props[regionsort[4]].area
13. f41 = fs[0]/props[regionsort[0]].area
14. f42 = fs[1]/props[regionsort[1]].area
15. f43 = fs[2]/props[regionsort[2]].area
16. f44 = fs[3]/props[regionsort[3]].area
17. f45 = fs[4]/props[regionsort[4]].area
18. f46 = fv[0]/props[regionsort[0]].area
19. f47 = fv[1]/props[regionsort[1]].area
20. f48 = fv[2]/props[regionsort[2]].area
21. f49 = fv[3]/props[regionsort[3]].area
22. f50 = fv[4]/props[regionsort[4]].area

```

Kode Sumber 4.7 Implementasi Pengambilan Fitur lokal warna

4.4 Implementasi Proses *Normalisasi*

Setelah proses ekstraksi fitur, data yang didapat perlu dilakukan normalisasi dahulu. Normalisasi yang digunakan adalah dengan rentang 0-1 seperti pada baris 1-5 berikut.

```

1. for x in range (0,50):
2.     max = data[:,x].max()
3.     min = data[:,x].min()
4.     for y in range (0,len(data)):

```

```
5. data[y,x] = (data[y,x]-min)/(max-min)
```

Kode Sumber 4.8 Implementasi Normalisasi

4.5 Implementasi Proses Klasifikasi

Tahap klasifikasi pada tugas akhir ini dilakukan dengan menggunakan self-organizing maps (SOM). SOM dilakukan iterasi berulang untuk mendapatkan hasil yang bagus. Variable kelas_hasil dan test digunakan untuk menyimpan hasil dari training dan testing klasifikasi. Lalu pada setiap data training dilakukan penghitungan bobot seperti pada persamaan (2.15) dengan menggunakan jarak euclidean, lalu pada setiap hasil dibandingkan dengan hasil sebelumnya untuk mencari nilai terkecil seperti pada baris 4-12. Setelah didapat neuron dengan nilai terkecil, maka dilakukan evaluasi dengan menghitung bobot baru dengan persamaan (2.16) seperti pada baris 14. Testing dilakukan setelah proses training selesai. Setelah didapat bobot neuron terakhir, maka bobot tersebut digunakan untuk menghitung jarak dengan data baru seperti pada baris ke 21. Setelah itu, neuron dengan jarak terkecil menjadi kelas bagi data baru tersebut.

```
1. def som(weight, data_train, learning_rate, theta, i
   terasi):
2.         kelas_hasil = []
3.         test = []
4.         for iterasi in range(0,iterasi):
5.             for y in range(data_train.shape
   [0]):
6.                 maxi = 999999
7.                 for x in range(weight.shape
   [0]):
8.                     eu_d = np.sum(np.square
   (weight[x,:]-data_train[y,:]))
9.                     if eu_d < maxi:
10.                        maxi = eu_d
11.                        kelas = x
12.
13.                 #evaluate
```

```
14.         weight[kelas,:] = weight[kelas,:] + learning_rate*sigma*(data_train[y,:] - weight[kelas,:])
15.         if iterasi == 999 :
16.             kelas_hasil.append(kelas)
17.
18.     for y in range(data_test.shape[0]):
19.         maxi = 999999
20.         for x in range(weight.shape[0]):
21.             eu_d = np.sum(np.square(weight[x,:]-data_test[y,:]))
22.             if eu_d < maxi:
23.                 maxi = eu_d
24.                 kelas = x
25.             test.append(kelas)
26.     return weight, kelas_hasil, test
```

Kode Sumber 4.9 Implementasi Proses klasifikasi

[Halaman ini sengaja dikosongkan]

BAB V

UJI COBA DAN EVALUASI

Dalam bab ini dibahas mengenai hasil uji coba sistem yang telah dirancang dan dibuat. Uji coba dilakukan untuk mengetahui kinerja sistem dengan lingkungan uji coba yang telah ditentukan.

5.1 Lingkungan Uji Coba

Lingkungan uji coba pada tugas akhir ini adalah sebuah *personal computer* (PC). Spesifikasi PC dari sisi perangkat keras adalah memiliki prosesor Intel Core i3 3240 dengan kecepatan 3,4 GHz dan memori untuk proses sebesar 4,00 GB. PC yang digunakan memiliki sistem operasi Windows 8.

Pada sisi perangkat lunak, uji coba pada tugas akhir ini dilakukan dengan menggunakan *IDE Spyder*.

5.2 Data Uji Coba

Data uji coba yang digunakan sebagai masukan adalah citra *digital* lukisan. Total citra yang digunakan adalah 160 citra dengan ukuran beragam.

Untuk menguji kebenaran dari hasil klasifikasi, dilakukan *K-Fold Cross Validation* pada setiap parameter ujicoba.

Uji coba dilakukan untuk mengetahui nilai-nilai parameter yang tepat untuk digunakan pada masing-masing proses. Nilai parameter yang tepat penting untuk diketahui karena penggunaan parameter yang tepat akan memberikan hasil yang terbaik pada keluaran tiap proses.

Skenario pengujian terdiri dari dua macam yaitu:

1. Uji coba penentuan parameter *learning rate SOM*
2. Uji coba penentuan parameter *K* pada *K-Means*

5.3 Uji Coba Penentuan Parameter *learning rate SOM*

Pada parameter ini diujicobakan *learning rate*. *Learning rate* berguna ketika akan digunakan update bobot pada neuron.

Pada tugas akhir ini nilai learning rate pada setiap iterasi tidak ada perubahan.

Nilai parameter yang digunakan berkisar antara 0.1 sampai dengan 1. Hasil klasifikasi dengan nilai parameter tersebut akan dilakukan penghitungan akurasi untuk melihat performa dari setiap nilai *learning rate*. Berikut adalah hasil dari percobaan dengan dataset *K-Means* dengan $k=8$ seperti yang terlihat pada Tabel 5.1 Akurasi setiap *learning rate* pada *K-Means* dengan $k= 8$.

Tabel 5.1 Akurasi setiap *learning rate* pada *K-Means* dengan $k= 8$

<i>Learning Rate</i>	Akurasi <i>TRAINING</i>	Akurasi <i>TEST</i>
0.1	52%	51%
0.2	64%	53%
0.3	66%	49%
0.4	74%	48%
0.5	76%	48%
0.6	70%	48%
0.7	69%	45%
0.8	66%	51%
0.9	69%	50%
1	54%	50%

Sedangkan pada hasil pengujian pada setiap kelasnya dimana dilakukan klasifikasi biner sehingga klasifikasi dilakuakn pada setiap 2 data, sehingga menghasilkan 6 hasil klasifikasi terlihat pada Tabel 5.2 Akurasi setiap kelas pada data *training* dan

Tabel 5.3 Akurasi setiap kelas pada data testing. Pengujian tersebut menggunakan *K-Fold cross validation* dengan $k=8$ sehingga terdapat 5 data *testing* dan 35 data *training* pada setiap kelasnya untuk melakukan klasifikasi sehingga terdapat total 10 data *testing* dan 70 data *training* pada setiap proses klasifikasi.

Tabel 5.2 Akurasi setiap kelas pada data training

LR	Ekspresionism / Impresionism	Ekspresionism / Post-Impresionism	Ekspresionism / Surrealism	Impresionism / Post-Impresionism	Impresionism / Surrealism	Post-Impresionism / Surrealism
0.1	52%	50%	56%	50%	54%	51%
0.2	52%	90%	58%	59%	56%	67%
0.3	58%	90%	62%	60%	63%	64%
0.4	77%	89%	85%	61%	66%	67%
0.5	78%	89%	76%	77%	68%	71%
0.6	68%	76%	82%	67%	69%	58%
0.7	74%	71%	80%	68%	67%	57%
0.8	67%	74%	59%	64%	70%	61%
0.9	74%	66%	73%	71%	70%	62%
1	58%	50%	54%	50%	53%	58%

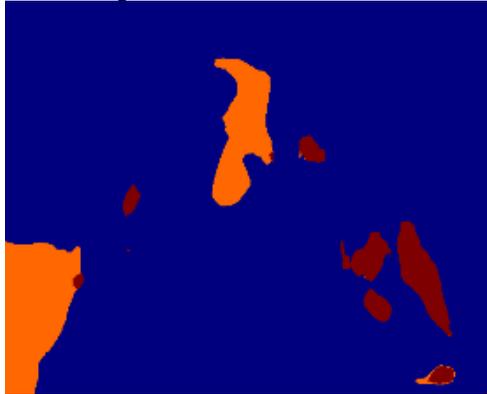
Tabel 5.3 Akurasi setiap kelas pada data testing

LR	Ekspresionism / Impresionism	Ekspresionism / Post-Impresionism	Ekspresionism / Surrealism	Impresionism / Post-Impresionism	Impresionism / Surrealism	Post-Impresionism / Surrealism
0.1	48%	54%	44%	54%	61%	48%
0.2	55%	45%	66%	38%	61%	53%
0.3	39%	48%	68%	41%	49%	50%
0.4	40%	45%	61%	39%	54%	48%
0.5	34%	36%	66%	50%	46%	53%
0.6	45%	39%	46%	56%	49%	50%
0.7	40%	45%	49%	46%	39%	49%
0.8	50%	49%	50%	50%	53%	53%
0.9	48%	46%	54%	50%	56%	44%
1	50%	50%	51%	50%	48%	49%

5.4 Uji Coba Parameter k pada K -Means

Uji coba ini dilakukan untuk mendapatkan nilai parameter k pada K -Means yang digunakan pada proses segmentasi. Nilai k berpengaruh terhadap pengelompokan warna. Pada ujicoba ini

dilakukan pada k dengan nilai 6, 8 dan 10. Berikut adalah hasil segmentasi pada masing masing k yang terlihat pada Gambar 5.1 k -means dengan $k=6$, Gambar 5.2 k -means dengan $k=8$ dan Gambar 5.3 k -means dengan $k=10$.

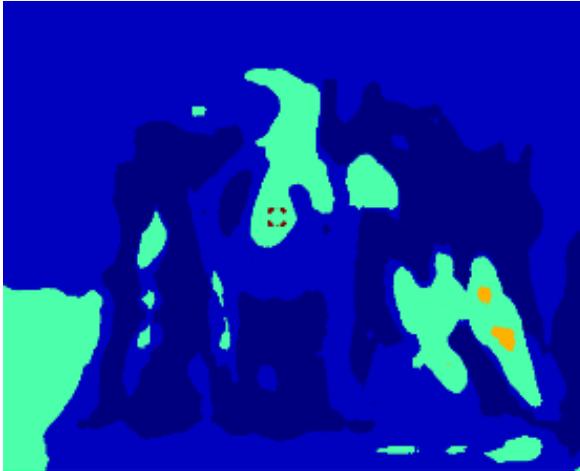


Gambar 5.1 k -means dengan $k=6$

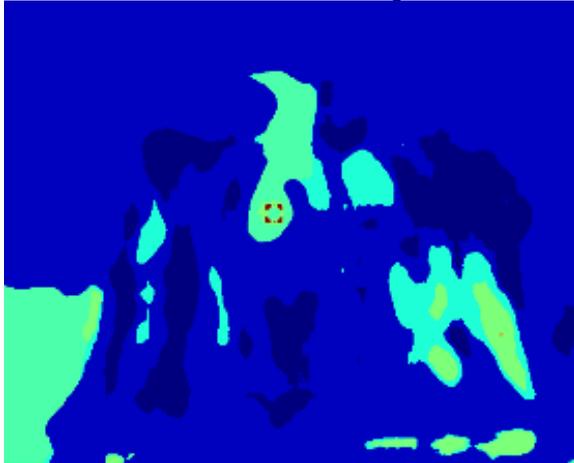
Pada pengujian ini, pada hasil k-means tersebut sudah dilakukan proses perbaikan segmentasi dengan menggunakan filter median untuk menghilangkan noise berupa salt dan papper yang mana akan mempengaruhi daerah yang akan digunakan untuk ekstraksi fitur lokal nantinya. Untuk akurasi dari setiap K-Means dilakukan dengan learning rate bernilai 0.2. Hasil akurasi dapat dilihat pada Gambar 5.4 Akurasi *learning rate* dengan *K-Means* pada $k=8$. Pada $k=6$ akurasi tidak dapat dilakukan karena jumlah wilayah tidak lebih dari 5 sesuai dengan *paper*[1].

Tabel 5.4 Hasil akurasi dari K-Means dengan $k=8$ dan $k=10$

k	Akurasi <i>TRAINING</i>	Akurasi <i>TEST</i>
6	-	-
8	64%	53%
10	51%	50%



Gambar 5.2 *k-means* dengan $k=8$



Gambar 5.3 *k-means* dengan $k=10$

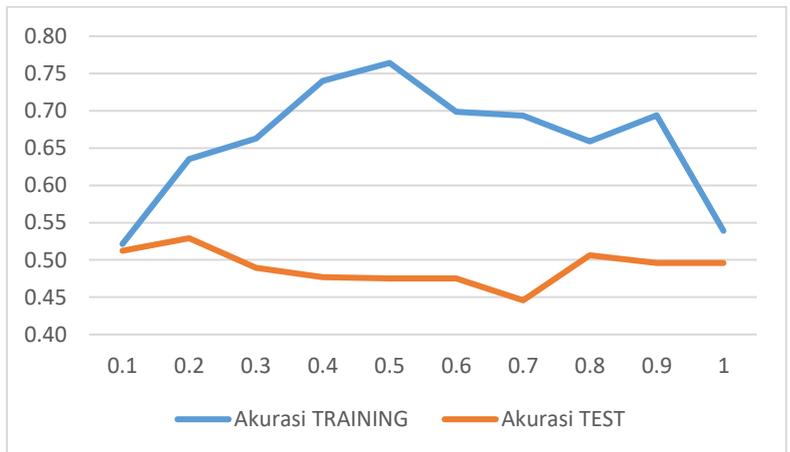
5.5 Evaluasi

Pada subbab ini akan dijelaskan hasil dari serangkaian uji coba yang dilakukan dan kendala yang dihadapi selama proses

pengerjaan. Evaluasi pada tahap klasifikasi adalah nilai *parameter learning rate* dan *k* pada *K-Means*.

5.5.1 Evaluasi Uji Coba Parameter *Learning rate*

Pada uji coba dengan parameter *learning rate*, digunakan rentang *learning rate* 0.1 sampai dengan 1. *Learning rate* pada data training berpola naik keatas ketika menjauhi 0.1 dan menurun ketika mendekati 1 dengan puncak akurasi terbaik pada 0.5 pada *K-Means* dengan $k=8$. Pada data *testing* justru menurun dari *learning rate* 0.1 dan kembali naik ketika mendekati *learning rate* 1.



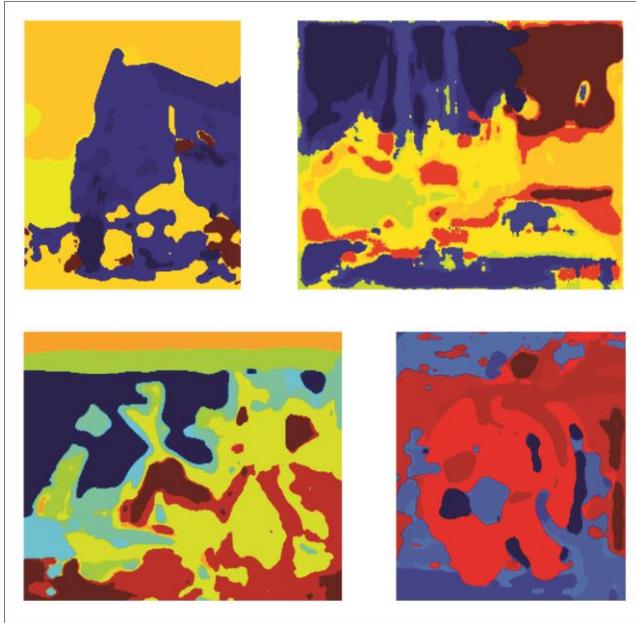
Gambar 5.4 Akurasi *learning rate* dengan *K-Means* pada $k=8$

Dari hasil diatas disimpulkan bahwa *learning rate* terbaik untuk proses klasifikasi terdapat pada angka 0.2 pada dataset *K-Means* dengan $k=8$ dengan angka 53% data *testing* dapat diklasifikasikan dengan benar pada kelasnya. Akan tetapi yang menjadi permasalahan adalah jauhnya rentang akurasi data training dengan data testing yang dapat dikatakan bahwa model yang digunakan adalah *overfitting* yang mana terlalu eksklusif bagi data

training sehingga tidak dapat mengklasifikasikan dengan baik untuk data baru. Permasalahan ini dapat diselesaikan dengan penambahan data yang lebih banyak atau dengan mensesederhanakan model.

5.5.2 Evaluasi Uji Coba Parameter k pada K -Means

Pada proses uji coba parameter k pada K -Means, terdapat perbedaan hasil pada citra hasil segmentasi dimana citra dikelompokkan berdasarkan warna. Ketika proses segmentasi terdapat proses perbaikan segmentasi menggunakan median filter yang mana memungkinkan pengurangan wilayah warna dikarenakan proses filtering tersebut. Pada K -means dengan $k=6$ seperti pada Gambar 5.1 k -means dengan $k=6$ terlihat bahwa dari citra sebelumnya yang dilakukan proses K -Means yang mengelompokkan nilai pada citra berdasarkan warna menjadi 6. Pada hasil akhir terlihat bahwa wilayah warna yang dihasilkan tereduksi menjadi 3 wilayah warna saja, sedangkan pada K -Means dengan $k=8$ seperti pada gambar Gambar 5.2 k -means dengan $k=8$ terlihat memiliki 5 wilayah warna begitu juga dengan $k=10$ seperti pada Gambar 5.3 k -means dengan $k=10$ yang memiliki 5 wilayah warna. Wilayah warna ini sendiri dibutuhkan untuk proses ekstraksi fitur lokal dimana membutuhkan 5 wilayah warna untuk mewakili karakteristik citra lukisan ini sendiri. Jumlah dari wilayah yang dibutuhkan ditetapkan menjadi 5 karena dibutuhkan kemungkinan variasi warna yang berbeda karena dengan memiliki 5 wilayah warna akan memiliki 120 kemungkinan variasi warna dari lukisan yang dirasa cukup dari total 360 derajat warna dengan perbedaan nilai saturasi dan kecerahan. Sedangkan dengan nilai dibawahnya, seperti 4 maupun 3, dinilai kurang bisa mewakili perbedaan karakteristik dari citra lukisan karena hanya memunculkan 6 dan 24 kemungkinan variasi warna. Perbandingan variasi warna dapat dilihat pada Gambar 5.5 Perbandingan variasi warna pada setiap citra.



Gambar 5.5 Perbandingan variasi warna pada setiap citra

Sedangkan pada sisi akurasi, yang dapat dibandingkan pada bagian ini adalah k-means dengan $k=8$ dan $k=10$. Pada akurasi, dilakukan dengan learning rate 0.2. Terlihat bahwa K-Means dengan $k=8$ memiliki nilai yang lebih besar daripada dengan $k=10$. Disana terlihat bahwa pengelompokan warna pada 8 wilayah lebih optimal daripada pengelompokan warna pada 10 wilayah disebabkan karena semakin banyak pengelompokan warna akan semakin memperkecil wilayah yang seharusnya dominan dan bisa menjadi karakteristik dari citra, dan pada $k=8$ wilayah berdasarkan warna dapat direpresentasikan dengan baik sesuai dengan warna dan wilayahnya layaknya yang terlihat pada citra asil, dan pengelompokan warna yang memiliki nilai yang dekat dapat dikelompokkan dengan sesuai.

BAB VI

KESIMPULAN DAN SARAN

Bab ini membahas mengenai kesimpulan yang dapat diambil dari hasil uji coba yang telah dilakukan sebagai jawaban dari rumusan masalah. Selain itu juga terdapat saran yang ditujukan untuk pengembangan penelitian lebih lanjut.

6.1 Kesimpulan

Kesimpulan yang diperoleh dari uji coba dan evaluasi adalah sebagai berikut:

1. Hasil akurasi klasifikasi terbaik ketika *Learning rate* 0.2 dengan nilai akurasi klasifikasi adalah 53%.
2. K-Means dengan $k=6$ tidak dapat dilakukan karena pada perbaikan citra terdapat kemungkinan hilangnya wilayah sehingga wilayah warna pada hasil segmentasi kurang dari 5.
3. Sistem dapat menghasilkan akurasi terbaik pada *style* lukisan Ekspresionism dan Surrealism.

6.2 Saran

Saran yang dapat diberikan dalam pengujian metode klasifikasi *style* lukisan adalah sebagai berikut:

1. Diperlukan penambahan data untuk mengatasi masalah overfit yang teridentifikasi pada dataset dengan K untuk *K-Means* 8
2. Dilakukan penambahan fitur pada citra untuk memperbaiki nilai akurasi dan fitur yang belum terekplor seperti fitur nilai kutosis.

[Halaman ini sengaja dikosongkan]

DAFTAR PUSTAKA

- [1] S.-G. Lee and E.-Y. Cha, “Style classification and visualization of art painting’s genre using self-organizing maps,” *Hum.-Centric Comput. Inf. Sci.*, vol. 6, no. 1, Dec. 2016.
- [2] T. Lombardi, “The Classification of *Style* in Fine-Art Painting,” Pace University.
- [3] S. Jafarpour, G. Polatkan, E. Brevdo, S. Hughes, A. Brasoveanu, and I. Daubechies, “Stylistic analysis of painting using wavelets and machine learning.”
- [4] J. Zujovic, L. Gandy, S. Friedman, B. Pardo, and T. N. Pappas, “Classifying Paintings by Artistic Genre: An Analysis of Features & Classifiers.”
- [5] “Expressionism | artistic *style*,” *Encyclopedia Britannica*. [Online]. Available: <https://www.britannica.com/art/Expressionism>. [Accessed: 06-Jul-2017].
- [6] “Impressionism | art,” *Encyclopedia Britannica*. [Online]. Available: <https://www.britannica.com/art/Impressionism-art>. [Accessed: 06-Jul-2017].
- [7] “Surrealism | art and literature,” *Encyclopedia Britannica*. [Online]. Available: <https://www.britannica.com/art/Surrealism>. [Accessed: 06-Jul-2017].
- [8] “Post-Impressionism | art | Britannica.com.” [Online]. Available: <https://www.britannica.com/art/Post-Impressionism>. [Accessed: 15-Jun-2017].
- [9] D. Putra, *Pengolahan Citra Digital*. Penerbit Andi, 2010.
- [10] “Confusion Matrix.” [Online]. Available: http://www2.cs.uregina.ca/~dbd/cs831/notes/confusion_matrix/confusion_matrix.html. [Accessed: 06-Jul-2017].

- [11] “Web Gallery of Art, searchable fine arts image database.” [Online]. Available: <http://www.wga.hu/>. [Accessed: 09-Jul-2017].

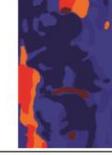
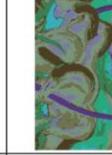
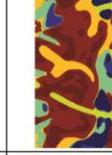
LAMPIRAN

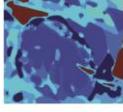
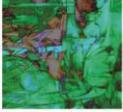
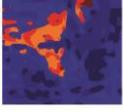
A. Hasil Ekstraksi Fitur

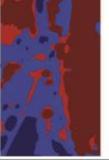
	Fitur																										
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25		
Data 1	0.95	0.11	0.82	0.07	0.12	0.01	0	0	0	0	0	0	0	0	0	0.4	0.5	0.48	0.56	0.64	0.56	0.28	0.08	0.17	0.17		
.....	0.83	0.21	0.64	0.04	0.16	0.01	0	0	0.01	0.01	0.13	0.31	0.12	0.11	0.08	0.22	0.79	0.2	0.24	0.11	0.11	0.06	0.01	0.21	0.46		
.....	0.91	0.12	0.45	0.05	0.07	0.01	0	0.01	0.01	0.01	0.01	0.02	0.04	0.08	0.28	0.52	0.37	0.45	0.28	0.2	0.14	0.13	0.03	0.49	0.57		
.....	0.97	0.03	0.82	0.11	0.09	0.02	0.01	0.01	0.01	0.01	0	0.02	0	0.01	0.01	0.26	0.8	1	0.95	0.86	0.69	0.41	0.1	0.08	0.47		
.....	0.95	0.13	0.09	0.02	0.03	0.04	0.01	0.01	0.01	0.01	0.01	0	0	0.01	0.03	1	0.78	0.17	0.1	0.1	0.05	0.05	0.01	0.48	0.62		
Data n	0.77	0.09	0.82	0.23	0.26	0.06	0.02	0.01	0.01	0	0	0	0	0	0	0	0.01	0.2	0.59	0.76	1	0.93	0.82	0.21	0.12	0.17	
Data 1	0.42	0.7	0.63	0.48	0.16	0.31	0.11	0.68	0.13	0.57	0.87	0.86	0.67	0.35	0.64	0.11	0.01	0.13	0.15	0.14	0.76	0.86	0.7	0.83	0.62		
.....	0.9	0.05	0.43	0.05	0.19	0.25	0.1	0.19	0.16	0.1	0.91	0.74	0.55	0.07	0.45	0.18	0.07	0.15	0.35	0.29	0.05	0.11	0.14	0.56	0.35		
.....	0.2	0.7	0.9	0.08	0.08	0.17	0.01	0.38	0.46	0.09	0.79	0.68	0.66	0.63	0.66	0.21	0.16	0.08	0.04	0.03	0.67	0.58	0.28	0.56	0.76		
.....	0.46	0.37	0.73	0.49	0.1	0.23	0.23	0.94	0.35	0.42	0.79	0.72	0.67	0.73	0.46	0.04	0.05	0.07	0.05	0.08	0.84	0.58	0.37	0.7	0.52		
.....	0.69	0.18	0.39	0.6	0.41	0.55	0.43	0.26	0.18	0.06	0.7	0.67	0.64	0.75	0.78	0.17	0.09	0.04	0.27	0.13	0.4	0.63	0.75	0.36	0.6		
Data n	0.55	0.66	0.96	0.1	0.07	0.08	0.19	0.25	0.19	0.66	0.7	0.78	0.2	0.66	0.16	0.04	0.05	0.11	0.07	0.1	0.45	0.62	0.2	0.38	0.52		

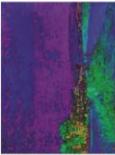
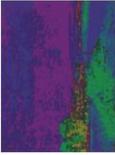
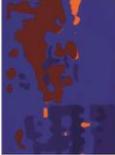
B. Hasil Segmentasi dan Klasifikasi

Kelas Ekspresionism dan Impresionism

No	Citra Asli	Citra HSV	K-Means	Filtering	Kelas Aktual	Kelas Sistem
1					Ekspresionism	Impresionism
2					Ekspresionism	Impresionism
3					Ekspresionism	Ekspresionism
4					Ekspresionism	Ekspresionism

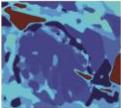
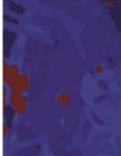
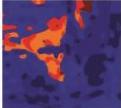
No	Citra Asli	Citra HSV	K-Means	Filtering	Kelas Aktual	Kelas Sistem
5					Ekspresionism	Ekspresionism
6					Ekspresionism	Ekspresionism
7					Ekspresionism	Ekspresionism
8					Ekspresionism	Ekspresionism

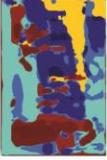
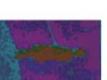
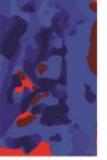
No	Citra Asli	Citra HSV	K-Means	Filtering	Kelas Aktual	Kelas Sistem
9					Impressionism	Impressionism
10					Impressionism	Impressionism
11					Impressionism	Impressionism
12					Impressionism	Impressionism

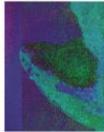
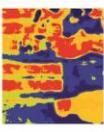
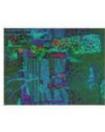
No	Citra Asli	Citra HSV	K-Means	Filtering	Kelas Aktual	Kelas Sistem
13					Impressionism	Impressionism
14					Impressionism	Impressionism
15					Impressionism	Impressionism
16					Impressionism	Impressionism

Kelas Ekspresionism dan Post-Impresionism

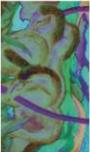
No	Citra Asli	Citra HSV	K-Means	Filtering	Kelas Aktual	Kelas Sistem
1					Ekspresionism	Post-Impresionism
2					Ekspresionism	Post-Impresionism
3					Ekspresionism	Ekspresionism
4					Ekspresionism	Ekspresionism

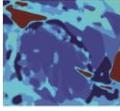
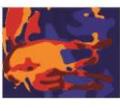
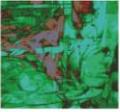
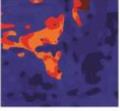
No	Citra Asli	Citra HSV	K-Means	Filtering	Kelas Aktual	Kelas Sistem
5					Ekspresionism	Ekspresionism
6					Ekspresionism	Ekspresionism
7					Ekspresionism	Ekspresionism
8					Ekspresionism	Ekspresionism

No	Citra Asli	Citra HSV	K-Means	Filtering	Kelas Aktual	Kelas Sistem
9					Post-Impressionism	Post-Impressionism
10					Post-Impressionism	Post-Impressionism
11					Post-Impressionism	Post-Impressionism
12					Post-Impressionism	Post-Impressionism

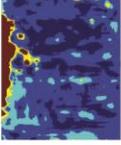
No	Citra Asli	Citra HSV	K-Means	Filtering	Kelas Aktual	Kelas Sistem
13					Post-Impresionism	Post-Impresionism
14					Post-Impresionism	Post-Impresionism
15					Post-Impresionism	Post-Impresionism
16					Post-Impresionism	Post-Impresionism

Kelas Ekspresionism dan Surrealism

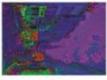
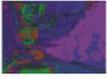
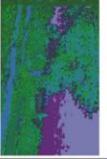
No	Citra Asli	Citra HSV	K-Means	Filtering	Kelas Aktual	Kelas Sistem
1					Ekspresionism	Ekspresionism
2					Ekspresionism	Surrealism
3					Ekspresionism	Ekspresionism
4					Ekspresionism	Ekspresionism

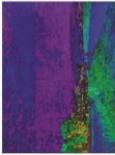
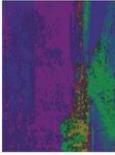
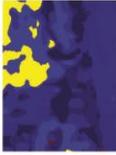
No	Citra Asli	Citra HSV	K-Means	Filtering	Kelas Aktual	Kelas Sistem
5					Ekspresionism	Ekspresionism
6					Ekspresionism	Ekspresionism
7					Ekspresionism	Ekspresionism
8					Ekspresionism	Ekspresionism

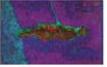
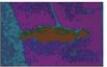
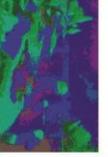
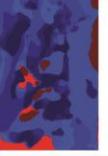
No	Citra Asli	Citra HSV	K-Means	Filtering	Kelas Aktual	Kelas Sistem
9					Surrealism	Surrealism
10					Surrealism	Surrealism
11					Surrealism	Surrealism
12					Surrealism	Surrealism

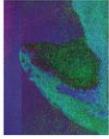
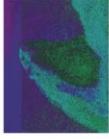
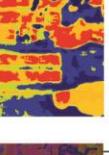
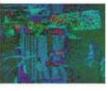
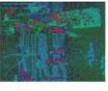
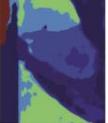
No	Citra Asli	Citra HSV	K-Means	Filtering	Kelas Aktual	Kelas Sistem
13					Surrealism	Surrealism
14					Surrealism	Surrealism
15					Surrealism	Surrealism
16					Surrealism	Surrealism

Kelas Impressionism dan Post-Impressionism

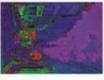
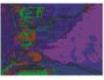
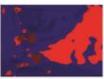
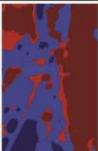
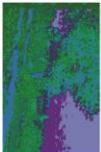
No	Citra Asli	Citra HSV	K-Means	Filtering	Kelas Aktual	Kelas Sistem
1					Impressionism	Impressionism
2					Impressionism	Impressionism
3					Impressionism	Impressionism
4					Impressionism	Impressionism

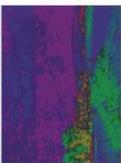
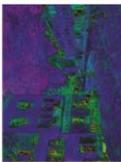
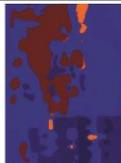
No	Citra Asli	Citra HSV	K-Means	Filtering	Kelas Aktual	Kelas Sistem
5					Impressionism	Impressionism
6					Impressionism	Impressionism
7					Impressionism	Impressionism
8					Impressionism	Impressionism

No	Citra Asli	Citra HSV	K-Means	Filtering	Kelas Aktual	Kelas Sistem
9					Post-Impressionism	Post-Impressionism
10					Post-Impressionism	Post-Impressionism
11					Post-Impressionism	Impressionism
12					Post-Impressionism	Post-Impressionism

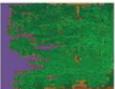
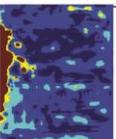
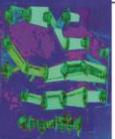
No	Citra Asli	Citra HSV	K-Means	Filtering	Kelas Aktual	Kelas Sistem
13					Post-Impressionism	Post-Impressionism
14					Post-Impressionism	Impressionism
15					Post-Impressionism	Impressionism
16					Post-Impressionism	Impressionism

Kelas Impressionism dan Surrealism

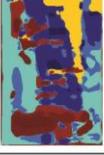
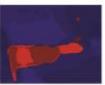
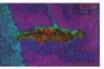
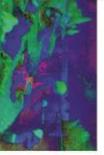
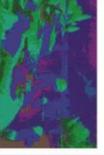
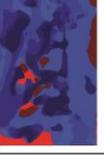
No	Citra Asli	Citra HSV	K-Means	Filtering	Kelas Aktual	Kelas Sistem
1					Impressionism	Impressionism
2					Impressionism	Impressionism
3					Impressionism	Impressionism
4					Impressionism	Impressionism

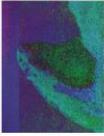
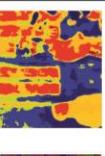
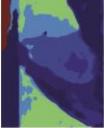
No	Citra Asli	Citra HSV	K-Means	Filtering	Kelas Aktual	Kelas Sistem
5					Impressionism	Impressionism
6					Impressionism	Impressionism
7					Impressionism	Impressionism
8					Impressionism	Impressionism

No	Citra Asli	Citra HSV	K-Means	Filtering	Kelas Aktual	Kelas Sistem
9					Surrealism	Surrealism
10					Surrealism	Surrealism
11					Surrealism	Surrealism
12					Surrealism	Surrealism

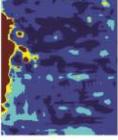
No	Citra Asli	Citra HSV	K-Means	Filtering	Kelas Aktual	Kelas Sistem
13					Surrealism	Surrealism
14					Surrealism	Surrealism
15					Surrealism	Surrealism
16					Surrealism	Surrealism

Kelas Post-Impressionism dan Surrealism

No	Citra Asli	Citra HSV	K-Means	Filtering	Kelas Aktual	Kelas Sistem
1					Post-Impressionism	Post-Impressionism
2					Post-Impressionism	Surrealism
3					Post-Impressionism	Post-Impressionism
4					Post-Impressionism	Surrealism

No	Citra Asli	Citra HSV	K-Means	Filtering	Kelas Aktual	Kelas Sistem
5					Post-Impressionism	Post-Impressionism
6					Post-Impressionism	Surrealism
7					Post-Impressionism	Surrealism
8					Post-Impressionism	Surrealism

No	Citra Asli	Citra HSV	K-Means	Filtering	Kelas Aktual	Kelas Sistem
9					Surrealism	Surrealism
10					Surrealism	Surrealism
11					Surrealism	Surrealism
12					Surrealism	Surrealism

No	Citra Asli	Citra HSV	K-Means	Filtering	Kelas Aktual	Kelas Sistem
13					Surrealism	Surrealism
14					Surrealism	Surrealism
15					Surrealism	Surrealism
16					Surrealism	Surrealism

[Lembar ini sengaja dikosongkan]

BIODATA PENULIS



Reza Andriyunanto, lahir di Karanganyar, pada tanggal 1 Juni 1995. Penulis menempuh pendidikan mulai dari SDN I Mojogedang (2001-2007), SMPN 1 Karanganyar (2007-2010), SMAN 1 Karanganyar (2010-2013) hingga terakhir Institut Teknologi Sepuluh Nopember Surabaya (2013-2017) di jurusan Teknik Informatika, Fakultas Teknologi Informasi angkatan tahun 2013.

Selama belajar di kampus Teknik Informatika, penulis berkesempatan menjadi administrator Laboratorium Komputasi Cerdas dan Visi pada tahun 2015 dan 2016. Selain mengikuti kegiatan akademik, penulis mengikuti kegiatan organisasi sebagai anggota Internal Affairs BEM FTIf (2014-2015) dan staff ahli Departemen Hubungan Kelembagaan Keluarga Muslim Informatika (2015-2016).

Penulis memiliki bidang minat Komputasi Cerdas Visi (KCV) dengan fokus studi pada bidang *image processing* dan *web programming*. Komunikasi dengan penulis dapat melalui email: **andriyunantoreza@gmail.com**.