

**TUGAS AKHIR - KI141502**

**IMPLEMENTASI METODE KOMBINASI *HISTOGRAM OF ORIENTED GRADIENTS* DAN *HIERARCHICAL CENTROID* UNTUK *SKETCH BASED IMAGE RETRIEVAL***

**ATIKA FARADINA RANDA**  
**NRP 5112100193**

**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 2016**

***(Halaman ini sengaja dikosongkan)***



**UNDERGRADUATE THESIS - KI141502**

**IMPLEMENTATION OF HISTOGRAM OF  
ORIENTED GRADIENTS AND HIERARCHICAL  
CENTROID FOR SKETCH BASED IMAGE  
RETRIEVAL**

**ATIKA FARADINA RANDA  
NRP 5112100193**

**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  
INSTITUT TEKNOLOGI SEPULUH NOPEMBER  
SURABAYA 2016**

*(Halaman ini sengaja dikosongkan)*

## LEMBAR PENGESAHAN

### IMPLEMENTASI METODE KOMBINASI HISTOGRAM OF ORIENTED GRADIENTS DAN HIERARCHICAL CENTROID UNTUK SKETCH BASED IMAGE RETRIEVAL

#### TUGAS AKHIR

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

Oleh:

**ATIKA FARADINA RANDA**  
NRP : 5112 100 193

Disetujui oleh Pembimbing Tugas Akhir

1. Dr. Eng. Nanik Suciati, S.Komputer, M.Sc. (Pembimbing 1)  
NIP: 197104281994122001
2. Dini Adni Navastara, S.Komputer, M.Sc. (Pembimbing 2)  
NIP: 198510172015042001



**SURABAYA**  
**JUNI, 2016**

*(Halaman ini sengaja dikosongkan)*

# IMPLEMENTASI METODE KOMBINASI HISTOGRAM OF ORIENTED GRADIENTS DAN HIERARCHICAL CENTROID UNTUK SKETCH BASED IMAGE RETRIEVAL

**Nama Mahasiswa** : ATIKA FARADINA RANDA  
**NRP** : 5112100193  
**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

*Teknik pencarian gambar yang saat ini umum digunakan masih berbasis teks atau text based search seperti pada mesin pencarian Google Image, Yahoo, dan lain sebagainya. Namun metode ini masih kurang efektif karena nama dari sebuah file tidak dapat merepresentasikan isinya, oleh karena itu diperlukan pemilihan kata kunci yang benar-benar tepat agar hasil yang diinginkan dapat ditampilkan dengan baik.*

*Salah satu teknik pencarian gambar yang saat ini sedang diteliti adalah Sketch-Based Image Retrieval (SBIR). Dengan teknik ini user dapat menginputkan sketsa gambar atau user dapat menggambarkan obyek pada area yang disediakan lalu sistem akan melakukan pencocokkan sketsa dengan database gambar. Untuk mengimplementasikan teknik ini digunakan metode kombinasi Histogram of Oriented Gradient dan Hierarchical Centroid. Tahapan implementasi teknik tersebut yaitu, yang pertama melakukan preprocessing pada gambar dengan cara mendeteksi tepi obyek lalu membuat citra menjadi hitam putih. Yang kedua melakukan ekstraksi fitur menggunakan Histogram of Oriented Gradients dan Hierarchical Centroid dan menghasilkan fitur vektor. Yang terakhir menghitung jarak kedekatan antara gambar yang diuji dengan gambar yang terdapat dalam database*

*menggunakan Euclidean Distance. Hasil Euclidean Distance kemudian diurutkan secara ascending dan dikembalikan sejumlah gambar yang jaraknya terdekat. Hasil temu kembali menghasilkan nilai Average Normalized Modified Retrieval Rank sebesar 0,35 dan nilai presisi dan recall sebesar 78 % dan akurasi sebesar 96%.*

***Kata kunci: Sketch Based Image Retrieval, Histogram of Oriented Gradients, Hierarchical Centroid, Average Normalized Modified Retrieval Rank .***



# IMPLEMENTATION COMBINATION OF HISTOGRAM OF ORIENTED GRADIENTS AND HIERARCHICAL CENTROID FOR SKETCH BASED IMAGE RETRIEVAL

**Student's Name** : ATIKA FARADINA RANDA  
**Student's ID** : 5112100193  
**Department** : Teknik Informatika FTIF-ITS  
**First Advisor** : Dr.Eng. Nanik Suciati, S.Kom.,  
M.Kom.  
**Second Advisor** : Dini Adni Navastara, S.Kom., M.Sc.

## Abstract

*Technique that commonly used in search image is text based search as well as in the searching machine like Google image, Yahoo an so on. But this method is not really effective because name of an image can not represented the content, therefore it is necessary keyword selection that can displayed result properly.*

*One of image search technique that currently being studied is Sketch Based Image Retrieval (SBIR). With this technique, user can input sketch image or user can sketch image in the area provided and then the system will match a sketch image with database image. To implement this technique used combination of method Histogram of Oriented Gradients and Hierarchical Centroid. First perform preprocessing on the image by detecting the edge of the object and make an image to black and white, Then exctrated feature using Histogram of Oriented Gradients and Hierarchical Centroid and generate vector features. Last, calculates the distance between test image and database image using euclidean distance and sort the result in ascending and retrieve a number of images that have high proximity. The result of the test of the system obtained 0,35 of Average Normalized Modified Retrieval Rank, precission and recall 78% and also accuracy 96%.*

***Keywords : Sketch Based Image Retrieval, Histogram of Oriented Gradients, Hierarchical Centroid, Average Normalized Modified Retrieval Rank.***

## KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Alhamdulillahirabbil'alamin, segala puji bagi Allah SWT, yang telah melimpahkan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul **“Implementasi Metode Kombinasi *Histogram Of Oriented Gradients* Dan *Hierarchical Centroid* Untuk *Sketch Based Image Retrieval*”** yang merupakan salah satu syarat dalam menempuh ujian sidang guna memperoleh gelar Sarjana Komputer. Bagi penulis, pengerjaan Tugas Akhir ini merupakan sebuah pengalaman yang berharga. Selama pengerjaan Tugas Akhir, penulis bisa belajar lebih banyak untuk memperdalam dan meningkatkan apa yang telah didapatkan penulis selama menjalani perkuliahan di Teknik Informatika ITS dan Tugas Akhir ini adalah implementasi dari apa yang telah penulis pelajari.

Selesaiannya Tugas Akhir ini tidak lepas dari bantuan dan dukungan beberapa pihak. Sehingga pada kesempatan ini penulis mengucapkan syukur dan terima kasih kepada:

1. Allah SWT dan Nabi Muhammad SAW.
2. Bapak Sugiman dan Ibu Siti Aningsih selaku orang tua penulis yang selalu memberikan dukungan doa, moral, dan material yang tak terhingga kepada penulis sehingga penulis dapat menyelesaikan Tugas Akhir ini. Serta selalu memberikan semangat dan motivasi pada penulis dalam mengerjakan Tugas Akhir ini.
3. Ibu Dr.Eng. Nanik Suciati, S.Kom., M.Kom. dan Dini Adni Navastara, S.Kom., M.Sc selaku pembimbing I dan pembimbing II yang telah membimbing dan membantu penulis serta memberikan motivasi dalam menyelesaikan Tugas Akhir ini dengan sabar.
4. Ibu Umi Laili Yuhana S.Kom, M.Sc. selaku dosen wali penulis yang telah memberikan arahan, masukan dan motivasi kepada penulis.

5. Bapak Darlis Herumurti, S.Kom., M.Kom. selaku Kepala Jurusan Teknik Informatika ITS, Bapak Radityo Anggoro, S.Kom.,M.Sc. selaku koordinator TA, dan segenap dosen Teknik Informatika yang telah memberikan ilmunya serta staf karyawan Jurusan Teknik Informatika ITS yang telah memberikan bantuan demi kelancaran admisnistrasi penulis selama kuliah.
6. Bian, Yaya, Uti, Zhi, Farah, Natasha, Kelly, Hendy, Leli dan Mei yang telah memberi semangat serta menjadi teman seperjuangan dalam menyelesaikan Tugas Akhir.
7. Serta semua pihak yang telah turut membantu penulis dalam menyelesaikan Tugas Akhir ini.

Penulis menyadari bahwa Tugas Akhir ini masih memiliki banyak kekurangan. Sehingga, penulis mengharapkan kritik dan saran yang membangun dari pembaca untuk perbaikan ke depannya.

Surabaya, Juni 2016

# DAFTAR ISI

<b>LEMBAR PENGESAHAN</b> .....	<b>v</b>
<b>Abstrak</b> .....	<b>vii</b>
<b>Abstract</b> .....	<b>ix</b>
<b>DAFTAR ISI</b> .....	<b>xiii</b>
<b>DAFTAR GAMBAR</b> .....	<b>xvii</b>
<b>DAFTAR TABEL</b> .....	<b>xix</b>
<b>DAFTAR KODE SUMBER</b> .....	<b>xxi</b>
<b>BAB I PENDAHULUAN</b> .....	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	3
1.3 Tujuan .....	3
1.4 Manfaat.....	3
1.5 Batasan Masalah.....	3
1.6 Metodologi .....	4
1.6.1 Penyusunan Proposal .....	4
1.6.2 Studi Literatur .....	4
1.6.3 Implementasi Perangkat Lunak.....	5
1.6.4 Pengujian dan Evaluasi.....	5
1.6.5 Penyusunan Buku .....	5
1.7 Sistematika Penulisan Laporan Tugas Akhir .....	5
<b>BAB II TINJAUAN PUSTAKA</b> .....	<b>7</b>
2.1 Sketsa .....	7
2.2 Sketch-Based Image Retrieval .....	8
2.3 Deteksi Tepi .....	9
2.4 Metode Sobel.....	10
2.5 Histogram of Oriented Gradients .....	11
2.6 Hierarchical Centroid .....	18
2.7 Euclidean Distance.....	20
2.8 Performa .....	21
2.8.1 Average Normalized Modified Retrieval Rank .....	21
2.8.2 Confussion Matrix .....	22
2.8.2.1 Presisi .....	22
2.8.2.2 Recall.....	23

2.8.2.3	Akurasi .....	23
<b>BAB III</b>	<b>PERANCANGAN PERANGKAT LUNAK .....</b>	<b>25</b>
3.1	Desain Metode Secara Umum .....	25
3.2	Perancangan data .....	27
3.2.1	Data Masukan .....	27
3.2.2	Data Keluaran .....	27
3.3	Preprocessing.....	28
3.4	Ekstraksi Fitur .....	30
3.4.1	Ekstraksi Fitur dengan HOG.....	30
3.4.2	Ekstraksi Fitur dengan Hierarchical Centroid.....	32
3.5	Perancangan Antar Muka .....	33
<b>BAB IV</b>	<b>IMPLEMENTASI.....</b>	<b>37</b>
4.1	Lingkungan Implementasi .....	37
4.2	Implementasi .....	37
4.2.1	Implementasi Menggambar sketsa.....	38
4.2.2	Implementasi Pemrosesan Preprocessing Data Masukan.....	40
4.2.3	Implementasi Ekstraksi fitur dengan HOG.....	41
4.2.4	Implementasi Ekstraksi Fitur menggunakan Hierarchical Centroid.....	44
4.2.5	Implementasi Penghitungan jarak dengan Euclidean Distance.....	46
<b>BAB V</b>	<b>HASIL UJI COBA DAN EVALUASI .....</b>	<b>49</b>
5.1	Lingkungan Pengujian.....	49
5.2	Data Pengujian .....	49
5.3	<i>Preprocessing</i> Gambar .....	50
5.4	Skenario Uji Coba .....	52
5.4.1	Skenario Uji Coba 1.....	52
5.4.1.1	Nilai ANMRR.....	52
5.4.1.2	Nilai Presisi, Recall, dan Akurasi .....	60
5.4.2	Skenario Uji Coba 2.....	61
<b>BAB VI</b>	<b>KESIMPULAN DAN SARAN .....</b>	<b>65</b>
6.1	Kesimpulan.....	65
6.2	Saran.....	66
<b>DAFTAR PUSTAKA .....</b>		<b>67</b>

<b>A.</b>	<b>LAMPIRAN.....</b>	<b>69</b>
	<b>BIODATA PENULIS.....</b>	<b>85</b>

*(Halaman ini sengaja dikosongkan)*



## DAFTAR GAMBAR

Gambar 2.1 Contoh gambar sketsa.....	7
Gambar 2.2 Hasil citra berbasis sketsa.....	8
Gambar 2.3 Susunan piksel-piksel disekitar piksel $(i,j)$ .....	10
Gambar 2.4 Operator Sobel.....	10
<i>Gambar 2.5 Gambar Asli</i> .....	12
Gambar 2.6 Gambar Setelah Dihitung Besar Gradiennya.....	12
Gambar 2.7 Panjang Bin Histogram.....	13
<i>Gambar 2.8 Citra yang Telah Dibagi Menjadi Cell</i> .....	14
Gambar 2.9 Nilai matriks $I_x$ pada Cell ke-1.....	14
Gambar 2.10 Nilai matriks $I_y$ pada Cell ke-1.....	14
Gambar 2.11 Nilai Gradien pada Cell ke-1.....	15
Gambar 2.12 Nilai Orientasi Sudut Gradien pada Cell ke-1.....	15
Gambar 2.13 Histogram Cell ke-1.....	15
<i>Gambar 2.14 Histogram Tiap Cell pada Citra</i> .....	16
Gambar 2.15 Ilustrasi Blok.....	16
Gambar 2.16 Nilai Histogram pada blok kesatu.....	17
Gambar 2.17 Nilai Histogram yang telah dinormalisasi pada blok kesatu.....	17
Gambar 2.18 Centroid pada Suatu Bidang.....	19
Gambar 2.19 Struktur Nilai Centroid dalam Bentuk Tree.....	19
Gambar 2.20 Contoh Letak centroid (a) Letak Centroid pada Kedalaman 2 (b) Letak Centroid pada Kedalaman 3.....	20
Gambar 3.1 Diagram Alir Sistem.....	26
Gambar 3.2 Contoh Data Masukan.....	27
Gambar 3.3 Contoh Data Keluaran pada Sistem.....	28
Gambar 3.4 Diagram Alir Preprocessing Sistem.....	29
Gambar 3.5 Gambar Preprocesiing (a) Citra RGB (b) Citra yang diubah menjadi grayscale (c) citra yang telah dideteksi tepi.....	29
Gambar 3.6 Digram Alir proses HOG.....	31
Gambar 3.7 Gambar HOG (a) Citra Hitam Putih (b) Citra yang Telah dihitung Gradiennya (c) Visualisasi Fitur HOG.....	31
Gambar 3.8 Diagram Alir Hierarchical Centroid.....	32

Gambar 3.9 Gambar Hierarchical Centroid (a) Citra Hitam Putih (b) Letak centroid pada kedalaman 1 (c) Letak centroid pada kedalaman 2 (d) Letak centroid pada kedalaman 3 .....	33
Gambar 3.10 Desain Antar Muka Sistem.....	34
Gambar 4.1 Tampilan sketsa gambar .....	40
Gambar 5.1 Gambar Asli Data Training .....	50
Gambar 5.2 Gambar Data Training Setelah dilakukan Preprocessing.....	51
Gambar 5.3 Gambar Asli Data Testing .....	51
Gambar 5.4 Gambar Data Testing Setelah Dilakukan Preprocessing.....	51
Gambar 5.5 Fitur HOG pada Gambar Testing .....	53
Gambar 5.6 Letak Centroid (a) Kedalaman 2 (b) Kedalaman 3 (c) Kedalaman 4 (d) Kedalaman 5 (e) Kedalaman 6.....	55
Gambar 5.7 Keluaran program Queri ke 16 .....	56
Gambar 5.8 Keluaran program Queri ke 28 .....	56
Gambar 5.9 Grafik Perbandingan Nilai ANMRR .....	62
Gambar 5.10 Grafik Perbandingan Nilai Presisi, Recall, dan Akurasi .....	63

## DAFTAR TABEL

Tabel 2.1 Confussion Matrix Dua Kelas .....	22
Tabel 3.1 Tabel Penjelasan Antar Muka Sistem .....	34
Tabel 4.1 Lingkungan Implementasi Perangkat Lunak.....	37
Tabel 5.1 Spesifikasi Lingkungan Pengujian .....	49
Tabel 5.2 Performa ANMRR pada kedalaman yang berbeda .....	53
Tabel 5.3 Tabel Nilai Presisi Recall dan Akurasi.....	60
Tabel 5.4 Performa ANMRR pada masing masing metode .....	61
Tabel 5.5 Performa Presisi, Recall dan Akurasi pada Masing Masing Metode.....	62
Tabel A.1 Queri Gambar .....	69
Tabel A.2 Hasil Keluaran Queri ke-16.....	71
Tabel A.3 Hasil Keluaran Queri ke-28.....	74
Tabel A.4 Nilai Performa pada Kedalaman 2 metode HOGHC..	77
Tabel A.5 Nilai Performa pada Kedalaman 3 metode HOGHC..	78
Tabel A.6 Nilai Performa pada Kedalaman 4 metode HOGHC..	80
Tabel A.7 Nilai Performa pada Kedalaman 5 metode HOGHC..	81
Tabel A.8 Nilai Performa pada Kedalaman 6 metode HOGHC..	82

*(Halaman ini sengaja dikosongkan)*

## DAFTAR KODE SUMBER

Kode Sumber 4.1 Kode program sketsa gambar .....	39
Kode Sumber 4.2 Kode Program Proses Preprocessing dengan Sobel.....	41
Kode Sumber 4.3 Kode Program Menghitung Magnitude dan Sudut Citra.....	41
Kode Sumber 4.4 Kode Program Proses Perhitungan Orientasi Sel Histogram .....	42
Kode Sumber 4.5 Kode Program Proses Normalisasi Fitur Blok43	
Kode Sumber 4.6 Kode Program Proses Normalisasi Fitur HOG .....	43
Kode Sumber 4.7 Kode Program Menghitung Nilai Centroid ....	44
Kode Sumber 4.7 Kode Program Menghitung Nilai Centroid (lanjutan) .....	45
Kode Sumber 4.8 Kode Program Normalisasi fitur Hierarchical Centroid.....	46
Kode Sumber 4.9 Kode Program Euclidean Distance.....	46

*(Halaman sengaja dikosongkan)*

# BAB I

## PENDAHULUAN

Dalam bab ini dijelaskan mengenai alasan yang melatar belakangi pembuatan sistem. Selain itu terdapat juga tujuan dan manfaat dalam pembuatan Tugas Akhir, serta metodologi dan sistematika penulisan Tugas Akhir ini sehingga dapat menjadi penunjang dalam pembuatan sistem.

### 1.1 Latar Belakang

Penelitian *image retrieval* pada awalnya dilakukan dengan berbasis teks dari citra yang akan dicari. Seperti pada halnya mesin pencarian gambar seperti Google Images, Yahoo, dan lain-lain masih menggunakan metode berbasis teks atau text-based search yaitu mengembalikan citra yang sesuai dengan kata kunci yang dimasukkan [1]. Dengan kompleksitas informasi yang dimiliki oleh suatu citra user tidak berharap mendapatkan kecocokan dengan pasti antara queri atau kata kunci yang dimasukkan dan citra yang akan ditampilkan kembali dari database. Untuk itu teknik pencarian gambar berbasis teks ini belum sepenuhnya efektif karena nama dari sebuah file tidak dapat merepresentasikan isinya. Oleh karena itu diperlukan pemilihan kata kunci yang benar-benar tepat agar hasil yang diinginkan dapat ditampilkan. Untuk mengatasi masalah tersebut diperlukan pendekatan alternatif. Penelitian tentang *image retrieval* sudah banyak dilakukan. Salah satunya yang sedang populer adalah menggunakan queri berupa citra hasil jepretan kamera. Dengan metode seperti itu membutuhkan kompleksitas waktu dan komputasi yang besar.

*Content Based Image Retrieval* (CBIR) bertujuan untuk mempermudah dan mempercepat pencarian informasi bersarkan pada informasi citra dari citra yang mirip dengan kriteria tertentu yang diinginkan dari sekumpulan image yang ada. Karakteristik yang dihasilkan berupa bentuk, warna, tekstur dan lain-lain sesuai

dengan citra yang diinginkan. Salah satu perkembangan *Content Based Image Retrieval* yaitu *Sketch Based Image Retrieval*. Dengan *Sketch Based Image Retrieval* (SBIR) user dapat menggambarkan obyek pada area yang telah disediakan lalu sistem akan mencocokkan sketsa dengan database. Penggunaan queri dengan sketsa ini dapat sangat berguna dan efisien pada kehidupan sehari-hari, karena user dapat mengingat kembali suatu obyek dengan bantuan sketsa dari obyek tersebut. Teknologi SBIR dapat berguna diberbagai bidang seperti mengidentifikasi pelaku kejahatan dari sketsa wajahnya, mengidentifikasi bentuk tato, grafiti, lukisan dan lain sebagainya. Image retrieval dengan sketsa ini penelitiannya masih jarang dilakukan dari pada metode CBIR yang telah memiliki variasi deskriptor untuk mendapatkan fitur [2]. Ada beberapa penelitian yang telah dilakukan sebelumnya yaitu menggunakan *Query by Visual Example* (QVE) yang menggunakan *edge map* sebagai sketsa citra dan citra tes dipetakan dalam representasi yang abstrak. Proses ini membutuhkan dana yang banyak dan masih belum didukung oleh banyak *software* [3]. Metode yang lain yaitu dengan melakukan pendekatan dari bentuk citra. Pendekatan tersebut ternyata memberikan hasil yang bagus [4] dengan menggabungkan metode *Histogram of Oriented Gradient* dan *Hierarchical Centroid* [5]. Dengan banyaknya kegunaan dari teknik SBIR ini maka topik tugas akhir ini akan menerapkan teknik tersebut dengan metode *Histogram of Oriented Gradient* dan *Hierarchical Centroid*.

Oleh karena itu untuk menyelesaikan permasalahan tersebut, pada tugas akhir ini akan dimplementasikan metode kombinasi *Histogram of Oriented Gradients* dan *Hierarchical Centroid* (HOG-HC). Teknik yang dilakukan yaitu melakukan preprocessing dengan cara membuat citra menjadi hitam putih lalu mendeteksi bagian tepi dari citra. Setelah didapatkan citra hasil preprocessing lalu dilakukan ekstraksi fitur menggunakan metode *Histogram of Oriented Gradients* dan *Hierarchical Centroid* (HOG-HC) dan selanjutnya menghitung jarak kedekatan



menggunkan *euclidean distance* dan mengembalikan citra yang jaraknya terdekat.

## 1.2 Rumusan Masalah

Rumusan masalah yang diangkat dalam Tugas Akhir ini dapat dipaparkan sebagai berikut:

1. Bagaimana implementasi mendeteksi tepi obyek menggunakan metode Sobel?
2. Bagaimana melakukan ekstraksi fitur dengan menggunakan metode kombinasi *Histogram of Oriented Gradients* dan *Hierarchical Centroid*?
3. Bagaimana performa dan hasil yang ditampilkan dari metode kombinasi *Histogram of Oriented Gradients* dan *Hierarchical Centroid* untuk *Sketch-Based Image Retrieval*?
4. Apakah metode kombinasi *Histogram of Oriented Gradients* dan *Hierarchical Centroid* lebih efektif dibanding metode lain seperti *Histogram of Oriented Gradients (HOG)*, *Hierarchical Centroid*, *Edge Histogram Descriptor (EHD)*, dan *Region Properties*?

## 1.3 Tujuan

Tujuan dari pembuatan Tugas Akhir ini adalah mengimplementasi metode *Histogram of Oriented Gradients* dan *Hierarchical Centroid* untuk *Sketch-Based Image Retrieval*

## 1.4 Manfaat

Tugas Akhir ini diharapkan dapat menyediakan alat bantu citra berbasis sketsa sebagai salah satu alternatif dalam menyelesaikan permasalahan pencarian gambar dari sketsa.

## 1.5 Batasan Masalah

Permasalahan yang dibahas dalam Tugas Akhir ini memiliki beberapa batasan, yaitu sebagai berikut:

1. Implementasi algoritma menggunakan bahasa pemrograman matlab R2014a.

2. Dataset yang digunakan yaitu citra gambar natural yang berjumlah 1000 citra yang didapat dari dari *Centre for Vision, Speech and Signal Processing*, University of Surrey, United Kingdom [6].
3. Dataset citra terdiri dari 10 macam citra atau kelas yang masing-masing kelas berjumlah 100 citra. Kelas kelas tersebut yaitu, bintang laut, menara Big Ben, menara Burj Al-arab, sepeda, London Eye, gerbang Arc de Triomphe, Piramida Mesir, gedung Sydney Opera, jembatan Sydney, pantai.
4. Ukuran piksel citra 134 x 100.
5. Tempat menggambar sketsa berukuran 402 x 300.
6. Ukuran minimal sketsa 134 x 100 dan ukuran maksimum sketsa 402 x 300.

## **1.6 Metodologi**

Pembuatan Tugas Akhir ini dilakukan dengan menggunakan metodologi sebagai berikut:

### **1.6.1 Penyusunan Proposal**

Tahap awal Tugas Akhir ini adalah menyusun proposal Tugas Akhir. Pada proposal, diajukan gagasan untuk melakukan implementasi terhadap metode *Histogram of Oriented Gradients* dan *Hierarchical Centroid* untuk *Sketch-Based Image Retrieval* dengan melakukan preprocessing menggunakan sobel. Kemudian dihitung jarak kedekatannya menggunakan euclidean distance. Hasil akhir adalah Citra yang memiliki jarak terdekat.

### **1.6.2 Studi Literatur**

Pada tahap ini dilakukan untuk mencari informasi dan studi literatur apa saja yang dapat dijadikan referensi untuk membantu pengerjaan Tugas Akhir ini. Informasi didapatkan dari buku dan literatur yang berhubungan dengan metode yang digunakan. Informasi yang dicari adalah *Histogram of Oriented Gradients*, *Hierarchical Centroid*, *Sobel* dan *Sketch Based Image Retrieval*

### **1.6.3 Implementasi Perangkat Lunak**

Implementasi merupakan tahap untuk membangun metode-metode yang sudah diajukan pada proposal Tugas Akhir. Untuk membangun algoritma yang telah dirancang sebelumnya, maka dilakukan implementasi dengan menggunakan suatu perangkat lunak yaitu Matlab R2014a.

### **1.6.4 Pengujian dan Evaluasi**

Pada tahap ini algoritma yang telah disusun diuji coba dengan menggunakan data uji coba yang ada. Data uji coba tersebut diuji coba dengan menggunakan suatu perangkat lunak dengan tujuan mengetahui kemampuan metode yang dipakai. Dan mengevaluasi hasil tugas akhir dengan paper pendukung yang ada. Parameter yang diujicobakan adalah nilai rangking yang dihasilkan pada saat menghitung jarak kedekatan

### **1.6.5 Penyusunan Buku**

Pada tahap ini disusun buku sebagai dokumentasi dari pelaksanaan Tugas Akhir yang mencakup seluruh konsep, teori, implementasi, serta hasil yang telah dikerjakan.

## **1.7 Sistematika Penulisan Laporan Tugas Akhir**

Buku Tugas Akhir ini merupakan laporan secara lengkap mengenai Tugas Akhir yang telah dikerjakan baik dari sisi teori, rancangan, maupun implementasi sehingga memudahkan bagi pembaca dan juga pihak yang ingin mengembangkan lebih lanjut. Sistematika penulisan buku Tugas Akhir secara garis besar antara lain:

### **Bab I Pendahuluan**

Bab ini berisi penjelasan latar belakang, rumusan masalah, batasan masalah dan tujuan pembuatan Tugas Akhir. Selain itu, metodologi pengerjaan dan sistematika

penulisan laporan Tugas Akhir juga terdapat di dalamnya.

## **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 Perancangan Perangkat Lunak**

Bab ini berisi penjelasan tentang rancangan dari sistem yang akan dibangun.

## **Bab IV Implementasi**

Bab ini berisi penjelasan implementasi dari rancangan yang telah dibuat pada bab sebelumnya. Implementasi disajikan dalam bentuk *code* secara keseluruhan disertai dengan penjelasannya.

## **Bab V Uji Coba Dan Evaluasi**

Bab ini berisi penjelasan mengenai data hasil percobaan dan pembahasan mengenai hasil percobaan yang telah dilakukan.

## **Bab VI Kesimpulan Dan Saran**

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

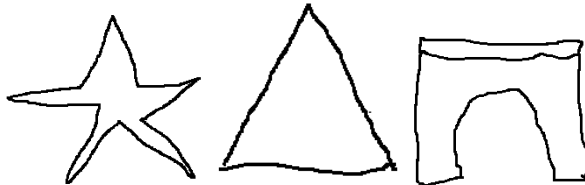
## BAB II TINJAUAN PUSTAKA

Dalam bab ini akan dijelaskan mengenai teori-teori yang merupakan dasar dari pembangunan sistem. Selain itu terdapat penjelasan yang menunjang pengerjaan Tugas Akhir ini sehingga dapat memberikan gambaran secara umum sistem yang akan dibangun.

### 2.1 Sketsa

Sketsa adalah karya yang berupa gambar yang biasanya tidak dimasukkan sebagai hasil karya akhir. Sketsa juga berarti lukisan awal yang kasar, ringan, semata-mata garis besar ataupun belum selsai, yang digunakan sebagai percobaan atau tanda yang digunakan dalam gambar biasa, catatan singkat tanpa bagian-bagian kecil yang mengemukakan gagasan tentang sesuatu [7]. Tujuan dibuatnya sketsa adalah merekam atau mengembangkan gagasan untuk dipakai kemudian atau dapat juga digunakan sebagai cara singkat menggambarkan citra, gagasan atau prinsip. Contoh sketsa dapat dilihat pada Gambar 2.1. Jenis-jenis sketsa antara lain:

- Gambar garis besar yaitu sketsa yang membuat garis-garis bentuk sederhana tanpa rincian dan tidak selesai.
- Sketsa cepat yaitu sketsa yang menggunakan beberapa garis saja untuk menampilkan citra suatu sketsa yang sudah selesai.
- Studi citra yaitu sketsa yang berupa coretan dengan cepat dan kurang terperinci hanya menunjukkan bentuk global.



*Gambar 2.1 Contoh gambar sketsa*

## 2.2 Sketch-Based Image Retrieval

Pada perkembangan teknologi saat ini, database dan internet menjadi alat yang digunakan untuk melakukan pencarian terhadap suatu informasi. Umumnya pencarian menggunakan query berupa teks atau tulisan. Walaupun banyak mesin pencarian yang dapat menangani query dan menyimpan informasi yang banyak, tetapi dalam pencarian berupa gambar masih belum efisien, karena informasi berupa visual atau gambar tidak dapat diekspresikan secara tekstual. Masalah ini dapat diatasi dengan teknologi *Content Based Image Retrieval* (CBIR) [8]. CBIR merupakan teknik yang



*Gambar 2.2 Hasil citra berbasis sketsa*

dilakukan dalam pencarian citra sesuai dengan input dari user. Input dapat berupa sketsa atau gambar yang spesifik. Proses CBIR yang pertama yaitu mengekstrak citra menjadi fitur dan menyimpannya, kemudian dilakukan perbandingan citra dari database dan mengembalikan hasil yang sesuai dengan input.

Salah satu metode CBIR yang saat ini dikembangkan adalah metode *Sketch-Based Image Retrieval* (SBIR). Sistem ini menggunakan input berupa gambar sketsa yang digambar sendiri oleh user sehingga lebih fleksibel. SBIR ini pertama kali dipelajari tahun 1990 dan di perkenalkan pada program *Query by Image Content* (QBIC) [9] dan *VisualSEEK* [10]. Pada sistem ini pengguna atau user menggambar sketsa dari sebuah objek pada area yang telah ditentukan. Selanjutnya dari sketsa yang telah diinputkan dicari gambar yang mirip dengan sketsa. Teknologi SBIR dapat berguna diberbagai bidang seperti mengidentifikasi pelaku kejahatan dari sketsa wajahnya, mengidentifikasi bentuk tato, grafiti, lukisan dan lain sebagainya [11]. Pada Gambar 2.2 merupakan contoh hasil citra berbasis sketsa.

### 2.3 Deteksi Tepi

Pertemuan antara bagian obyek dan bagian latar belakang obyek disebut tepi obyek. Pada pengolahan citra, tepi obyek ditandai oleh titik yang nilai keabuannya memiliki perbedaan yang cukup besar dengan titik yang di sebelahnya. Sedangkan deteksi tepi obyek adalah operasi yang dijalankan untuk mendeteksi garis tepi yang membatasi dua wilayah citra homogen yang memiliki tingkat kecerahan yang berbeda [12]. Adapun tujuan dari deteksi tepi adalah sebagai berikut:

1. Untuk mendeteksi garis tepi yang membatasi dua wilayah citra, obyek dan latar belakang obyek.
2. Untuk menemukan perubahan intensitas yang berbeda dalam sebuah bidang citra.
3. Untuk meningkatkan penampakan garis batas suatu daerah atau obyek di dalam citra.
4. Untuk memperbaiki detail citra yang kabur, yang terjadi karena error atau adanya proses akuisisi citra.

Deteksi tepi bermanfaat dalam berbagai bidang misalnya dalam bidang kedokteran adalah untuk menentukan stadium kanker, mendeteksi tepi citra USG janin, mendeteksi karies pada gigi. Pada bidang lainnya, deteksi tepi digunakan untuk aplikasi pengenalan plat kendaraan, aplikasi pengenalan sidik jari, dan lain sebagainya. Deteksi tepi merupakan komponen berfrekuensi tinggi, sehingga dibutuhkan *High Pass Filter* (HPF). Biasanya operator yang digunakan untuk mendeteksi tepi yang pertama adalah operator berbasis *gradient* (turunan pertama), yaitu operator Roberts, Sobel, dan Prewitt. Sedangkan untuk mendeteksi tepi kedua adalah operator berbasis turunan kedua, yaitu operator Laplacian dan operator *Laplacian of Gaussian*. Turunan pertama menghasilkan tepi yang lebih tebal, sedangkan turunan kedua menghasilkan tepi yang lebih tipis.

## 2.4 Metode Sobel

Metode sobel merupakan metode yang menggunakan dua buah *kernel* yang berukuran 3x3 piksel untuk penghitungan gradient sehingga perkiraan gradient berada tepat di tengah jendela [13]. Misalkan susunan piksel-piksel disekitar piksel  $(i,j)$  seperti pada Gambar 2.3, maka besar gradien yang dihitung menggunakan operator sobel dapat dihitung dengan Persamaan (2.1)

$$G = \sqrt{S_x^2 + S_y^2} \quad (2.1)$$

Dimana  $G$  adalah besar gradient operator sobel,  $S_x$  adalah gradient sobel horizontal dan  $S_y$  adalah gradient sobel horizontal

$a_0$	$a_1$	$a_2$
$a_7$	$(i,j)$	$a_3$
$a_6$	$a_5$	$a_4$

Gambar 2.3 Susunan piksel-piksel disekitar piksel  $(i,j)$

Besar gradient di titik tengah kernel dan turunan parsial dapat dihitung menggunakan Persamaan (2.2) dan Persamaan (2.3) sebagai berikut

$$S_x = (a_2 + ca_3 + a_4) - (a_0 + ca_7 + a_6) \quad (2.2)$$

$$S_y = (a_0 + ca_1 + a_2) - (a_6 + ca_5 + a_4) \quad (2.3)$$

Dimana  $c$  adalah konstanta yang bernilai 2. Gambar 2.4 menampilkan nilai operator  $S_x$  dan  $S_y$ . Operator tersebut diimplemetasikan dengan cara konvolusi.

$S_x$	<table border="1" style="display: inline-table;"> <tbody> <tr><td>-1</td><td>0</td><td>1</td></tr> <tr><td>-2</td><td>0</td><td>2</td></tr> <tr><td>-1</td><td>0</td><td>1</td></tr> </tbody> </table>	-1	0	1	-2	0	2	-1	0	1
-1	0	1								
-2	0	2								
-1	0	1								

$S_y$	<table border="1" style="display: inline-table;"> <tbody> <tr><td>1</td><td>2</td><td>1</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>-1</td><td>-2</td><td>-1</td></tr> </tbody> </table>	1	2	1	0	0	0	-1	-2	-1
1	2	1								
0	0	0								
-1	-2	-1								

Gambar 2.4 Operator Sobel



## 2.5 Histogram of Oriented Gradients

*Histogram of Oriented Gradient* (HOG) umumnya digunakan dalam pengenalan obyek dan mendeteksi pejalan kaki. HOG adalah deskriptor berbasis window yang mendeteksi pada titik interest. Metode ini menghitung nilai gradien dalam daerah tertentu pada suatu citra. Setiap citra memiliki karakteristik yang ditunjukkan oleh distribusi gradien yang diperoleh dengan membagi citra kedalam daerah kecil yang disebut *cell*. Tiap *cell* disusun dari sebuah histogram dari sebuah gradien. Kombinasi dari histogram ini dijadikan sebagai deskriptor yang mewakili sebuah obyek. Langkah dalam penghitungan deskriptor HOG dibagi dalam beberapa tahap [14].

### 1. Menghitung gradien citra

Gradien citra dapat dihitung dengan membagi citra kedalam beberapa area kecil sesuai dengan areanya. Area ini dapat dikatakan sebagai window yang membagi  $n \times n$  grid. Gradien citra dapat dihitung dengan Persamaan (2.4)

$$|G| = \sqrt{I_x^2 + I_y^2} \quad (2.4)$$

Dimana  $I$  adalah citra *graylevel*.  $I_x$  merupakan matrik terhadap sumbu- $x$  dan  $I_y$  merupakan matrik terhadap sumbu- $y$ .  $I_x$  dan  $I_y$  dapat dihitung dengan Persamaan (2.5).

$$I_x = I * D_x, \quad I_y = I * D_y \quad (2.5)$$

$D_x$  adalah *mask*  $[-1 \ 0 \ 1]$ , sedangkan  $D_y$  adalah *mask*  $\begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$

masing-masing dihitung dengan cara konvolusi. Kemudian gradien ditransformasi ke dalam kordinat sumbu dengan sudut diantara  $0$  sampai  $180^0$  yang disebut orientasi gradien. Orientasi gradien ( $\theta$ ) dapat dihitung dengan Persamaan (2.6). Ilustrasi dari langkah pertama dapat dilihat pada Gambar 2.5 dan Gambar 2.6. Gambar 2.5 merupakan gambar asli sedangkan Gambar 2.6 merupakan gambar yang telah dihitung besar gradiennya.

$$\theta = \arctan\left(\frac{I_x}{I_y}\right) \quad (2.6)$$



*Gambar 2.5 Gambar Asli*



*Gambar 2.6 Gambar Setelah Dihitung Besar Gradiennya*

## 2. Menghitung histogram dari orientasi gradien tiap *cell*

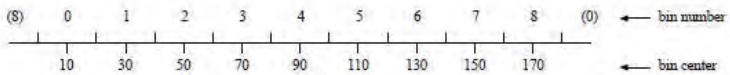
Citra dibagi menjadi *cell* ukuran sel 8 x 8 piksel. Setiap *cell* dihitung nilai histogramnya dari nilai orientasi gradien. Bin yang dipakai 9 bin ( $B=9$ ). Penentuan nilai histogram dilakukan dengan melakukan *voting* terhadap masing-masing *cell* pada citra. Gambar 2.7 merupakan gambar 9 bin histogram yang digunakan pada *voting* untuk menentukan kontribusi nilai histogram. Bin yang dipakai berukuran sembilan, dengan nilai bin dari 0 sampai B-1.

Kontribusi nilai histogram ( $v$ ) dapat diperoleh dari Persamaan (2.7) dan Persamaan (2.8). Dimana  $\mu$  merupakan

besar gradien pada piksel,  $c$  merupakan nilai tengah sudut pada bin,  $\theta$  adalah sudut orientasi gradien pada piksel,  $w$  adalah lebar dari nilai tengah sudut yaitu  $w = \frac{180}{B}$  dan  $B$  adalah panjang bin histogram yang digunakan.

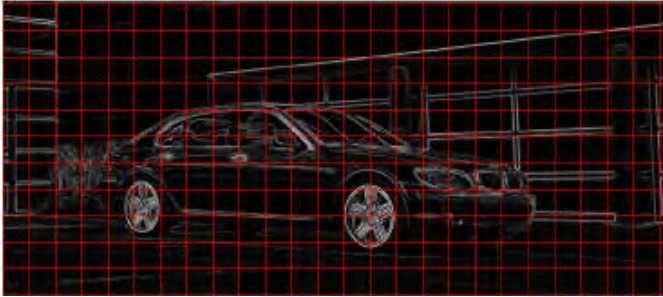
$$v_j = \mu \frac{c_{j=1} - \theta}{w} \text{ untuk bin ke } j = \left(\frac{\theta}{w} - \frac{1}{2}\right) \bmod B \quad (2.7)$$

$$v_{j+1} = \mu \frac{\theta - c_j}{w} \text{ untuk bin ke } j = j + 1 \bmod B \quad (2.8)$$



Gambar 2.7 Panjang Bin Histogram

Ilustrasinya pembagian citra menjadi *cell* dapat dilihat pada Gambar 2.8. Sedangkan Gambar 2.9 merupakan gambar matriks  $I_x$  pada *cell* kesatu. Nilai rentang  $I_x$  yaitu antara -1 sampai 1. *Cell* kesatu berada pada pojok kiri atas Gambar 2.8. Gambar 2.10 merupakan gambar matriks  $I_y$  pada *cell* kesatu. Nilai rentang  $I_y$  sama dengan  $I_x$  yaitu antara -1 sampai 1. Setelah diketahui nilai  $I_x$  dan  $I_y$  pada citra, kemudian dihitung besar gradien dan orientasi sudut gradiennya. Gambar 2.11 merupakan matriks besar gradien pada *cell* kesatu dan Gambar 2.12 merupakan matriks besar orientasi sudut pada *cell* kesatu. Nilai orientasi sudut memiliki rentang 0 sampai 180. Dari nilai gradien dan orientasi sudut gradien dapat dihitung nilai kontribusi histogramnya menggunakan Persamaan 2.7 dan Persamaan 2.8. Gambar 2.13 merupakan gambar visualisasi histogram pada *cell* kesatu. Sedangkan Gambar 2.14 merupakan gambar histogram pada masing-masing *cell*.



*Gambar 2.8 Citra yang Telah Dibagi Menjadi Cell*

0	-1	0	1	0	0	0	0
0	-1	0	1	0	0	0	0
0	-1	0	1	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	-1
-1	0	1	0	0	0	0	-1
-1	0	1	0	0	0	0	0
-1	0	1	0	0	0	0	0

*Gambar 2.9 Nilai matriks  $I_x$  pada Cell ke-1*

0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	0	1	0	0	0	0
0	0	-1	0	0	0	0	0
0	0	-1	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

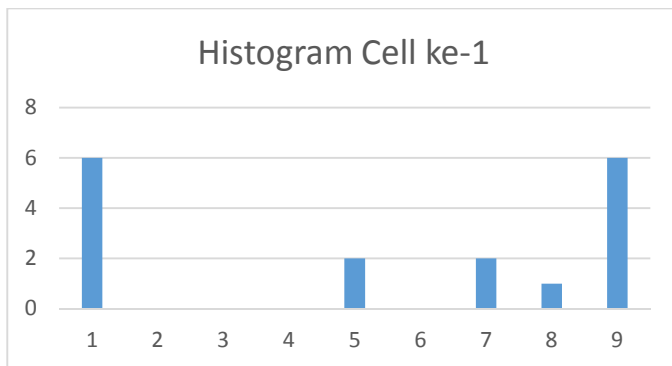
*Gambar 2.10 Nilai matriks  $I_y$  pada Cell ke-1*

0	1	0	1	0	0	0	0
0	1	0	1.41	0	0	0	0
0	1	0	1.41	0	0	0	0
0	0	1	0	0	0	0	0
0	0	1	0	0	0	0	1
1	0	1	0	0	0	0	1
1	0	1	0	0	0	0	0
1	0	1	0	0	0	0	0

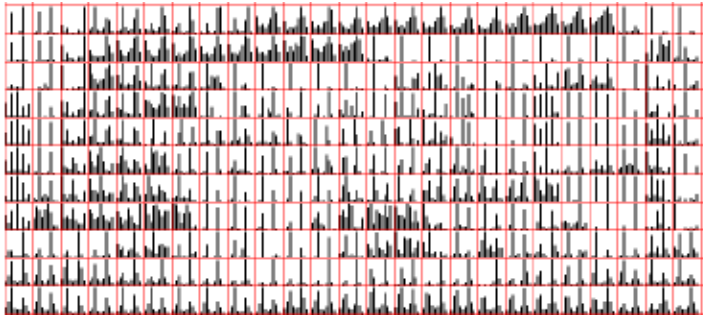
*Gambar 2.11 Nilai Gradien pada Cell ke-1*

0	0	0	180	0	0	0	0
0	0	0	135	0	0	0	0
0	0	0	135	0	0	0	0
0	0	90	0	0	0	0	0
0	0	90	0	0	0	0	0
0	0	180	0	0	0	0	0
0	0	180	0	0	0	0	0
0	0	180	0	0	0	0	0

*Gambar 2.12 Nilai Orientasi Sudut Gradien pada Cell ke-1*



*Gambar 2.13 Histogram Cell ke-1*

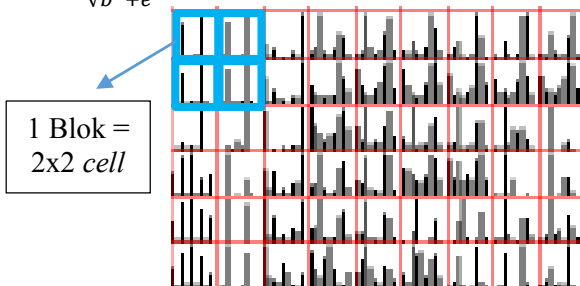


Gambar 2.14 Histogram Tiap Cell pada Citra

3. Melakukan normalisasi pada fitur blok

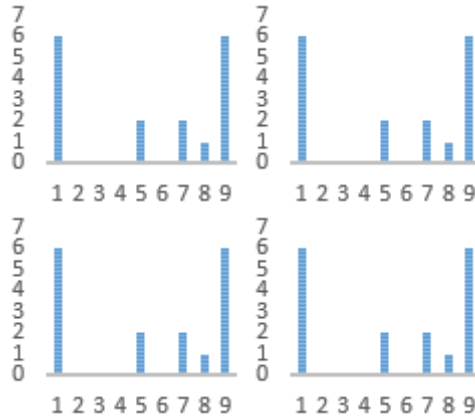
Citra dibagi menjadi blok-blok berukuran  $2 \times 2$  cell dapat dilihat pada Gambar 2.15. Fitur blok didapatkan dari nilai histogram pada  $2 \times 2$  cell. Nilai histogram pada satu cell digabungkan dengan cell lain menjadi satu *vector* dan kemudian dilakukan normalisasi. Nilai normalisasi fitur blok didapat dari Persamaan (2.7). Fitur blok dinormalisasi untuk mengurangi efek perubahan kecerahan obyek pada satu blok. Variabel  $b$  merupakan nilai blok fitur dan variabel  $e$  merupakan bilangan positif yang bernilai kecil untuk mencegah pembagian dengan 0.

$$b = \frac{b}{\sqrt{b^2 + e}} \quad (2.7)$$

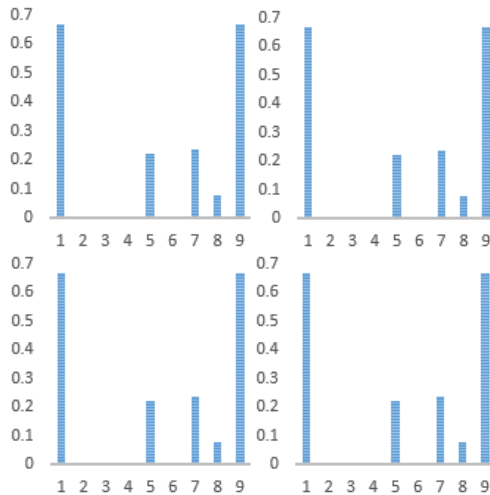


Gambar 2.15 Ilustrasi Blok

Nilai histogram pada satu blok dapat dilihat pada Gambar 2.16. Sedangkan nilai normalisasi histogram pada satu blok dapat dilihat pada Gambar 2.17.



Gambar 2.16 Nilai Histogram pada blok kesatu



Gambar 2.17 Nilai Histogram yang telah dinormalisasi pada blok kesatu

#### 4. Melakukan normalisasi fitur HOG

Fitur-fitur blok digabungkan menjadi satu menjadi sebuah fitur HOG. Kemudian fitur HOG dinormalisasi agar kecerahan citra menjadi sama. Normalisasi dilakukan melalui Persamaan (2.8). Variabel  $h$  merupakan nilai fitur HOG dan variabel  $e$  merupakan bilangan positif yang bernilai kecil untuk mencegah pembagian dengan 0.

$$h = \frac{h}{\sqrt{\|h\|^2 + e}} \quad (2.8)$$

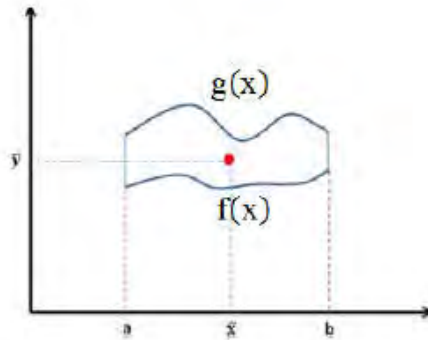
### 2.6 Hierarchical Centroid

Bentuk merupakan fitur keterangan yang signifikan karena memberikan penjelasan suatu obyek menggunakan karakteristik yang paling penting. Metode *Hierarchical Centroid* adalah deskriptor sederhana dengan mendapatkan pusat massa dari suatu obyek. Pusat massa pada bentuk sederhana seperti persegi, lingkaran, dan segitiga mudah didapatkan dan sederhana untuk menghitungnya. Namun, untuk mendapatkan pusat massa pada bentuk yang tidak beraturan sulit didapatkan dan perhitungan lokasinya sulit. Oleh karena itu, untuk mendapatkan pusat massanya dilakuakn dengan menghitung jarak rata-rata di setiap arah dan menyatakan proporsi dari total luas obyek. Setiap titik pada perubahan ukuran bentuk dikenal sebagai *moment* [15].

Gambar 2.18 menunjukkan *centroid* dari area citra dimana  $f(x)$  dan  $g(x)$  berbentuk kurva. Untuk menghitung nilai pusat massanya pertama-tama hitung luas bidang dari obyek dihitung menggunakan Persamaan (2.10)

$$A = \int_a^b [f(x) - g(x)] dx \quad (2.10)$$

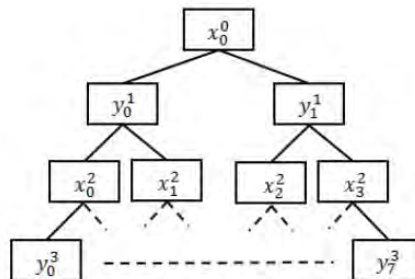




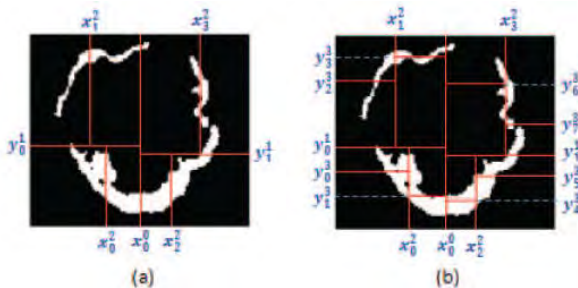
Gambar 2.18 Centroid pada Suatu Bidang

Variabel  $A$  merupakan luas bidang pada obyek. Sedangkan variabel  $a$  merupakan batas minimal dari obyek pada sumbu- $x$  dan variabel  $b$  merupakan batas maksimal dari obyek pada sumbu- $x$ . Nilai  $a$  dan  $b$  diilustrasikan pada Gambar 2.18. Nilai *centroid* dari suatu obyek dapat direpresentasikan sebagai  $x$  dan  $y$  diperoleh dari Persamaan (2.11). Nilai fitur *Hierarchical Centroid* dapat direpresentasikan ke dalam bentuk *tree* seperti pada Gambar 2.19. Letak centroid pada sebuah citra dapat dilihat pada Gambar 2.20.

$$\begin{aligned}
 x &= \frac{1}{A} \int_a^b [x (f(x) - g(x))] dx \\
 y &= \frac{1}{A} \int_a^b \frac{1}{2} [x (f(x)^2 - g(x)^2)] dx
 \end{aligned}
 \quad (2.11)$$



Gambar 2.19 Struktur Nilai Centroid dalam Bentuk Tree



Gambar 2.20 Contoh Letak centroid (a) Letak Centroid pada Kedalaman 2 (b) Letak Centroid pada Kedalaman 3

Sehingga jika melihat dari struktur tree pada Gambar 2.19 yang terbentuk untuk fitur *Hierarchical Centroid* saat berada pada kedalaman 1 adalah  $[y_0^1, x_0^0, y_1^1]$ . Sedangkan kedalaman 2 nilai fiturnya adalah  $[x_0^2, y_0^1, x_1^2, x_2^2, y_1^1, x_3^2]$  dan seterusnya.

## 2.7 Euclidean Distance

*Euclidean distance* adalah perhitungan jarak dari 2 buah titik dalam *Euclidean space*. *Euclidean space* diperkenalkan oleh seorang matematikawan dari Yunani sekitar tahun 300 S.M untuk mempelajari hubungan antara sudut dan jarak. *Euclidean space* adalah bilangan *Cartesian* pada  $n$ -ruang, yang terdiri dari bilangan real  $(x_1, x_2, \dots, x_n)$ . *Euclidean* ini berkaitan dengan teorema Pythagoras dan biasanya diterapkan pada 1, 2 dan 3 dimensi. Tapi juga sederhana jika diterapkan pada dimensi yang lebih tinggi [16].

Cara mencari jarak euclidean antara 2 titik  $p$  dan  $q$  dapat dilihat pada Persamaan (2.12). Dimana pada kordinat Cartesian  $p=(p_1, p_2, \dots, p_n)$  dan  $q=(q_1, q_2, \dots, q_n)$ . Variabel  $d$  merupakan jarak antar dua titik. Variabel  $p$  dan  $q$  merupakan titik yang akan dicari jaraknya, sedangkan  $n$  merupakan banyaknya titik yang akan dicari jaraknya.

$$d(p, q) = d(q, p) = \sqrt{\sum_{i=1}^n (q_i - p_i)^2} \quad (2.12)$$

## 2.8 Performa

Performa pengukuran metode dilakukan untuk mengetahui metode yang dipakai apakah lebih baik dari metode yang lain. Pengukuran metode yang dipakai yaitu *Average Normalized Modified Retrieval Rank*, Presisi, Recall, dan Akurasi.

### 2.8.1 Average Normalized Modified Retrieval Rank

*Average Normalized Modified Retrieval Rank* (ANMRR) adalah salah satu metode dalam melakukan pengukuran performa perangkaian pada CBIR sistem. ANMRR didefinisikan dari penelitian MPEG-7. Nilai dari ANMRR dinormalisasi [17]. Langkah dalam mencari nilai ANMRR adalah sebagai berikut:

- Langkah pertama mencari nilai rata-rata ranking dari banyaknya query dapat dicari dari Persamaan (2.13)

$$AVR(q) = \sum_{k=1}^{NG(k)} \frac{Rank(k)}{NG(q)} \quad (2.13)$$

Dimana variabel  $AVR$  merupakan nilai rata-rata *ranking* pada query ke- $q$ ,  $NG(q)$  adalah jumlah *ground truth* gambar pada query ke- $q$ . Jika citra yang ditampilkan terdapat pada  $K$  yang pertama maka  $Rank(k) = R$ , jika tidak  $Rank(k) = 1,25 * K$ . Variabel  $R$  adalah *ranking* citra pada  $K$  pertama yang ditampilkan.

- $K$  adalah *top-ranking* yang ditampilkan. Dimana nilai  $K$  dapat diperoleh dari Persamaan (2.14).

$$K(q) = \min\{X * NG(q), X * GMT\} \quad (2.14)$$

Jika  $NG(q) > 50$  maka  $X = 2$  jika lebih dari 50 maka  $X = 4$ . Parameter  $X$  didefinisikan oleh MPEG-7.  $GMT$  merupakan nilai maksimum *ground truth* untuk semua query didalam dataset.

- Kemudian mencari nilai modified retrieval rank ( $MRR(q)$ ), diperoleh dari persamaan (2.15)

$$MRR(q) = AVR(q) - 0.5 \times (1 + NG(q)) \quad (2.15)$$

- Melakukan normalisasi pada ranking *retrieval* ( $NMRR(q)$ ) agar data tidak ada yang dominan. Normalisasi didapat dari Persamaan (2.16).

$$NMRR(q) = \frac{MRR(q)}{1.25 \times K - 0.5 \times (1 + NG(q))} \quad (2.16)$$

- Langkah terakhir dihitung nilai rata-rata dari normalisasi semua query atau nilai *ANMRR*. Dihitung dengan Persamaan (2.17)

$$ANMRR = \frac{1}{Q} \sum_{q=1}^Q NMRR(q) \quad (2.17)$$

$Q$  adalah banyaknya query yang diujikan. Nilai *NMRR* dan *ANMRR* memiliki rentang nilai diantara  $[0,1]$ , dimana semakin kecil nilai *NMRR* dan *ANMRR* berarti performa dari suatu metode tersebut lebih bagus.

## 2.8.2 Confusion Matrix

*Confusion matrix* adalah suatu metode yang biasa digunakan untuk melakukan perhitungan akurasi pada konsep data mining. *Confusion Matrix* memiliki informasi hasil *relevant* dan *retrieved*. Kelas *relevant* merupakan gambar yang benar, sedangkan *retrieved* merupakan gambar yang ditampilkan. Performa suatu sistem biasanya dievaluasi menggunakan *Confusion Matrix*. Tabel 2.1 menunjukkan *Confusion Matrix* pada dua kelas. Nilai yang bisa dihitung menggunakan *Confusion Matrix* yaitu akurasi, presisi, dan recall.

**Tabel 2.1 Confusion Matrix Dua Kelas**

		Relevant	
		1	0
Retrieved	1	TP	FN
	0	FP	TN

### 2.8.2.1 Presisi

Presisi merupakan ukuran kedekatan antar serangkaian hasil analisis yang diperoleh dari beberapa kali pengukuran pada sampel homogen yang sama [18]. Presisi juga memiliki arti kata lain yaitu tingkat ketepatan antara informasi yang diminta oleh

pengguna dengan jawaban yang diberikan oleh system. Nilai presisi didapatkan dari tabel *confussion matrix* yang dihasil saat dilakukan pengujian. Nilai presisi dapat dihitung dari Persamaan (2.18)

$$\text{Presisi} = \frac{TP}{TP+FP} \quad (2.18)$$

Nilai *TP* dan *FP* didapat dari tabel *confussion matrix* yang merupakan *True Positive* dan *False Positive*.

### 2.8.2.2 Recall

Recall adalah tingkat keberhasilan sistem dalam menemukan kembali sebuah informasi [19]. Nilai *recall* didapatkan dari tabel *confussion matrix* yang dihasil saat dilakukan pengujian. Nilai *recall* dapat dihitung dari Persamaan (2.19)

$$\text{Recall} = \frac{TP}{TP+FN} \quad (2.19)$$

Nilai *TP* dan *FN* didapat dari tabel *confussion matrix* yang merupakan *True Positive* dan *False Negative*.

### 2.8.2.3 Akurasi

Akurasi adalah ukuran seberapa dekat suatu hasil pengukuran dengan nilai yang benar atau diterima dari kuantitas besaran yang diukur [18]. Nilai akurasi didapatkan dari tabel *confussion matrix* yang dihasil saat dilakukan pengujian. Nilai akurasi dapat dihitung dari Persamaan (2.20)

$$\text{Akurasi} = \frac{TP+TN}{TP+TN+FP+FN} \quad (2.20)$$

Nilai *TP*, *TN*, *FP* dan *FN* didapat dari tabel *confussion matrix* yang merupakan *True Positive*, *True Negative*, *False Positive*, dan *False Negative*.

*(Halaman sengaja dikosongkan)*

## **BAB III**

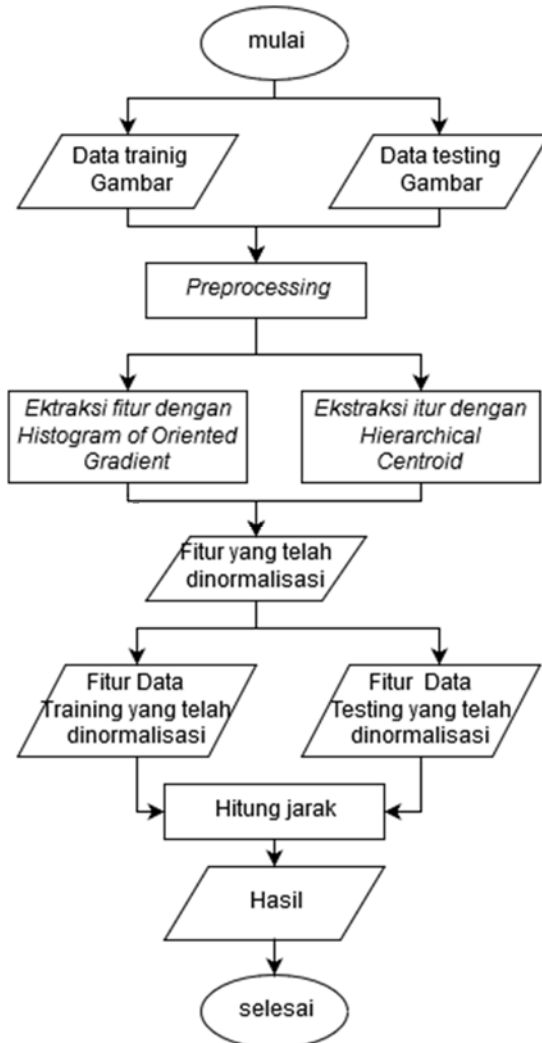
### **PERANCANGAN PERANGKAT LUNAK**

Pada bab ini akan dijelaskan mengenai perancangan sistem perangkat lunak yang akan dibuat. Perancangan yang dijelaskan meliputi data dan proses. Data yang dimaksud adalah data yang akan diolah dalam perangkat lunak baik digunakan sebagai pelatihan maupun pengujian sehingga tujuan Tugas Akhir ini bisa tercapai. Selain itu akan dijelaskan juga desain metode secara umum pada sistem.

#### **3.1 Desain Metode Secara Umum**

Pada Tugas Akhir ini akan dibangun suatu sistem yang dapat melakukan pencarian gambar di dalam dataset. Input berupa sketsa gambar yang akan dicari kemiripannya dengan gambar yang ada dalam dataset. Gambar yang ada pada dataset terdiri dari 10 kelas gambar yaitu, bintang laut, menara Big Ben, menara Burj Al-arab, sepeda, London Eye, gerbang Arc de Triomphe, Piramida Mesir, gedung Sydney Opera, jembatan Sydney, pantai. Gambar pada dataset dan gambar sketsa dilakukan preprocessing untuk mencari tepi gambar dan menjadikan gambar menjadi gambar binery. Kemudian dilakukan ekstraksi fitur dan diukur jarak kedekatannya lalu ditampilkan gambar yang jarak kedekatannya paling kecil. Diagram alir proses ini ditunjukkan pada Gambar 3.1. Ekstraksi fitur yang digunakan pada Tugas Akhir ini adalah *Histogram of Oriented Gradient* (HOG) dan *Hierarchical Centroid*. Kedua ekstraksi fitur ini berdasarkan pada bentuk gambar. Metode HOG merupakan fitur untuk mencari orientasi dari gambar, sedangkan metode *Hierarchical Centroid* mencari nilai centroid secara hirarki. Ekstraksi fitur dilakukan pada dataset gambar dan gambar sketsa. Selanjutnya dilakukan pengukuran jarak kedekatan antara dataset gambar dan gambar sketsa menggunakan *euclidean distance*. Setelah itu dilakukan perankingan untuk mencari nilai yang paling kecil. Hasil akhir

dari program ini adalah gambar yang memiliki nilai *similarity* paling kecil. Yang terakhir dilakukan pengukuran performa dari metode ekstraksi fitur.



Gambar 3.1 Diagram Alir Sistem

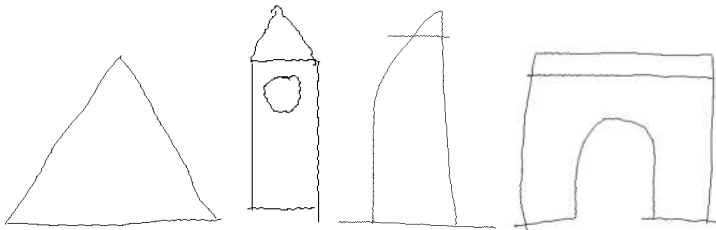


## 3.2 Perancangan data

Pada sub bab ini akan dijelaskan mengenai data yang digunakan sebagai masukan perangkat lunak untuk selanjutnya diolah dan dilakukan pengujian sehingga menghasilkan data keluaran yang diharapkan.

### 3.2.1 Data Masukan

Data masukan adalah data yang digunakan sebagai masukan awal dari sistem. Data yang digunakan pada sistem ini adalah sketsa gambar. Input gambar sketsa juga dapat dimasukkan atau digambar sendiri oleh user yang kemudian ukuran piksel sketsa diubah menjadi 134x100 piksel lalu dilakukan *preprocessing*. Selain itu juga terdapat data *testing* sketsa gambar. Data *testing* sketsa gambar berjumlah 30 buah gambar dari 10 kelas yang berbeda dan 2 dari gambar yang tidak terdapat pada kelas. Sedangkan data training yang diolah terdapat 1000 buah gambar dari 10 kelas yang berbeda. Setiap kelasnya terdiri dari 100 gambar. Data masukan diambil dari *Centre for Vision, Speech and Signal Processing*, University of Surrey, United Kingdom [18]. Contoh data masukan dapat dilihat pada Gambar 3.2

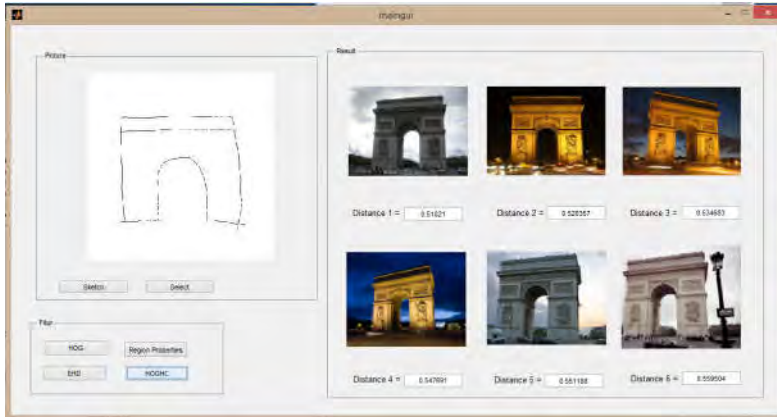


Gambar 3.2 Contoh Data Masukan

### 3.2.2 Data Keluaran

Data masukan akan dilakukan ekstraksi fitur dengan menggunakan metode *Histogram of Oriented Gradient* dan *Hierarchical Centroid*. Terdapat dua data yaitu data pembelajaran atau *training* dan data pengujian atau *testing*. Kedua data tersebut

dilakukan ekstraksi fitur. Setelah mendapatkan fitur data *training* dan fitur data *testing*, kemudian dilakukan perhitungan jarak dengan menggunakan *euclidean distance* dan selanjutnya dilakukan perangkingan data secara *ascending*. Keluaran dari sistem ini adalah 6 gambar dengan nilai *distance* yang paling kecil. Contoh data keluaran dapat dilihat pada Gambar 3.3

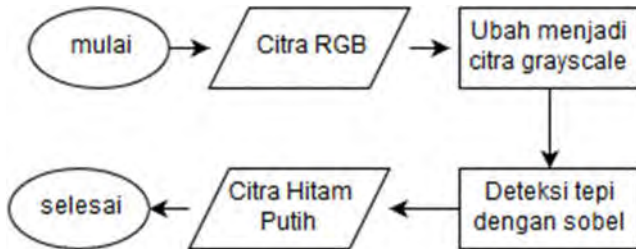


Gambar 3.3 Contoh Data Keluaran pada Sistem

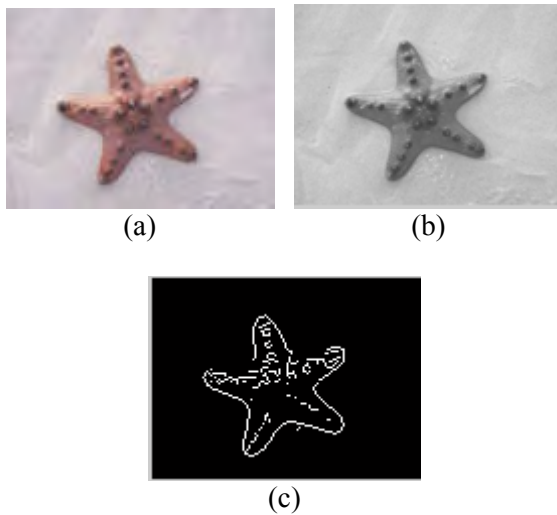
### 3.3 Preprocessing

*Preprocessing* merupakan cara untuk mengubah data menjadi bentuk yang cocok untuk pencarian gambar. *Preprocessing* dilakukan pada data *testing* dan data *training* dengan cara mendeteksi tepi dari obyek menggunakan metode sobel. Input citra berupa citra *Red Green Blue* (RGB) yang kemudian diubah menjadi citra *grayscale* atau citra abu-abu supaya lebih mudah untuk dicari tepinya karena dimensinya menjadi berkurang. Citra RGB memiliki 3 dimensi, dimensi merah, hijau dan biru. Sedangkan jika diubah menjadi *grayscale*, dimensinya berkurang menjadi 2 dimensi. Keluaran dari proses ini adalah citra biner atau citra hitam putih yang telah dideteksi tepinya dan selanjutnya akan diproses pada ekstraksi fitur. Langkah

preprocessing digambarkan pada diagram alir Gambar 3.4. Sedangkan gambar hasil tiap langkah dapat dilihat pada Gambar 3.5.



Gambar 3.4 Diagram Alir Preprocessing Sistem



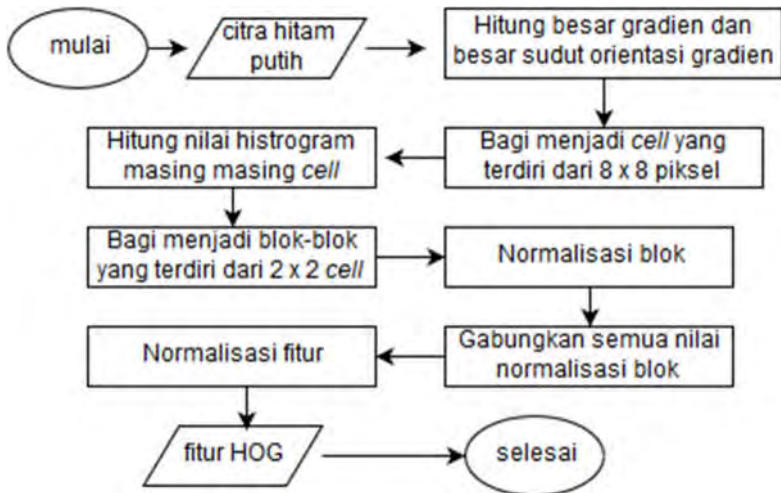
Gambar 3.5 Gambar Preprocessing (a) Citra RGB (b) Citra yang diubah menjadi grayscale (c) citra yang telah dideteksi tepi

### 3.4 Ekstraksi Fitur

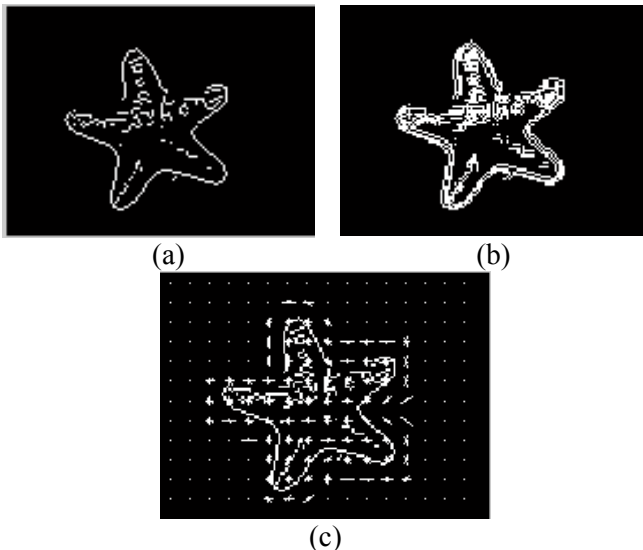
Ekstraksi fitur dilakukan untuk menghasilkan fitur vektor yang akan digunakan sebagai penghitungan jarak kedekatannya antara data pembelajaran atau data *training* dan data uji atau data *testing* dan bertujuan untuk mendapatkan informasi-informasi penting dari citra, sehingga dapat membedakan citra satu dan yang lainnya. Ekstraksi fitur dilakukan pada dua macam data yaitu data *testing* dan data *training*. Citra yang dilakukan ekstraksi fitur adalah citra hasil *preprocessing* yaitu citra biner atau citra hitam putih.

#### 3.4.1 Ekstraksi Fitur dengan HOG

Terdapat 4 tahap dalam melakukan ekstraksi fitur dengan metode *Histogram of Oriented Gradients* (HOG). Yang pertama menghitung gradien dari masing masing citra. Langkah ini bertujuan untuk mengetahui besar gradien dan besar orientasi sudut gradien pada citra. Yang kedua menghitung orientasi sel histogram. Langkah ini bertujuan untuk membagi citra menjadi 8 x 8 sel yang kemudian dihitung nilai orientasi histogramnya. Yang ketiga melakukan normalisasi pada blok dari 2 x 2 sel. Gabungkan 4 sel histogram pada setiap blok menjadi satu block fitur dan kemudian dinormalisasi menggunakan *Euclidean norm*. Tujuan dilakukannya normalisasi blok ini adalah untuk mengurangi efek akibat perubahan kecerahan citra pada objek yang sama. Yang terakhir fitur HOG, hasil normalisasi dari blok digabungkan keseluruhannya dan kemudian dilakukan normalisasi lagi. Normalisasi terakhir membuat fitur HOG menjadi nilai yang independen terhadap perubahan kecerahan citra. Secara lebih lengkap proses HOG digambarkan pada Gambar 3.6. Gambar 3.7 merupakan gambar visualisasi proses ekstraksi fitur dengan HOG.



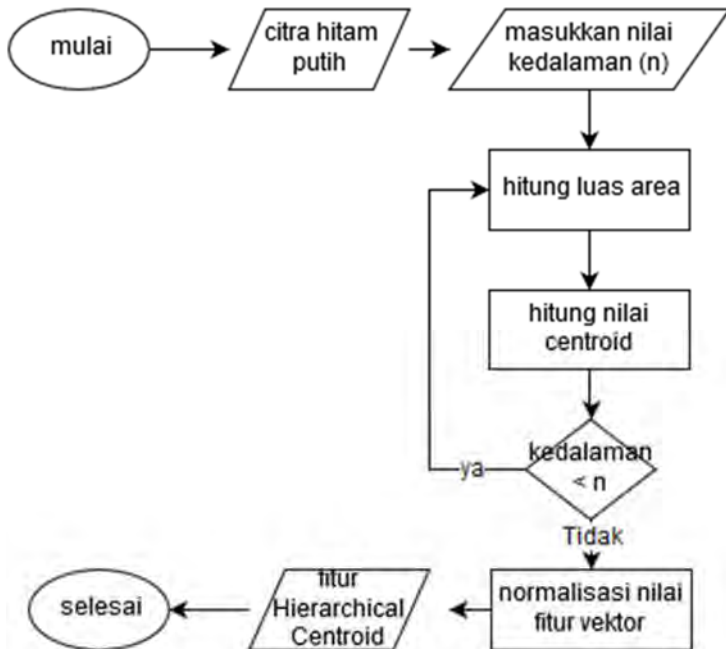
Gambar 3.6 Digram Alir proses HOG



Gambar 3.7 Gambar HOG (a) Citra Hitam Putih (b) Citra yang Telah dihitung Gradiennya (c) Visualisasi Fitur HOG

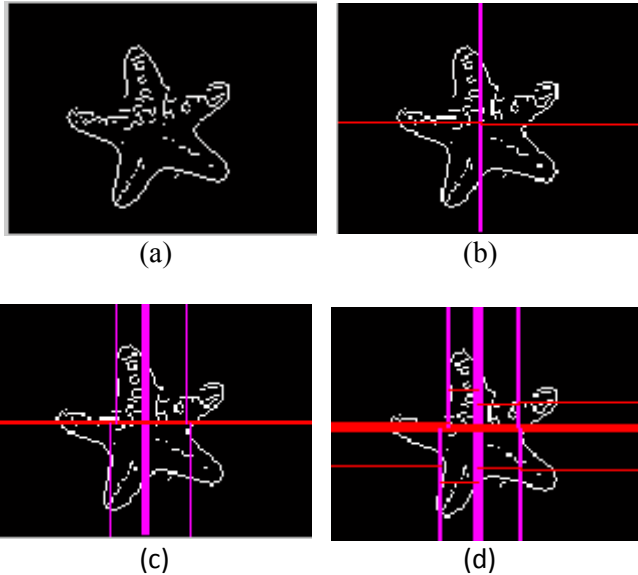
### 3.4.2 Ekstraksi Fitur dengan Hierarchical Centroid

Langkah dalam melakukan ekstraksi fitur dengan *Hierarchical Centroid* adalah yang pertama citra input berupa citra hitam putih dihitung luas area yang digunakan pada perhitungan centroid, kemudian hitung nilai centroid pada sumbu x. Setelah itu citra dibagi menjadi citra sebelah kanan dan citra sebelah kiri lakukan rekursif untuk mencari nilai centroid. Rekursif berhenti jika telah mencapai kedalaman yang telah ditentukan. Setelah mendapatkan titik-titik centroid yang merupakan fitur dari *Hierarchical Centroid*, kemudian fitur dilakukan normalisasi untuk menghilangkan nilai yang lebih dominan. Gambar diagram alir dari dari proses *Hierarchical Centroid* dapat dilihat pada Gambar 3.8.



Gambar 3.8 Diagram Alir Hierarchical Centroid

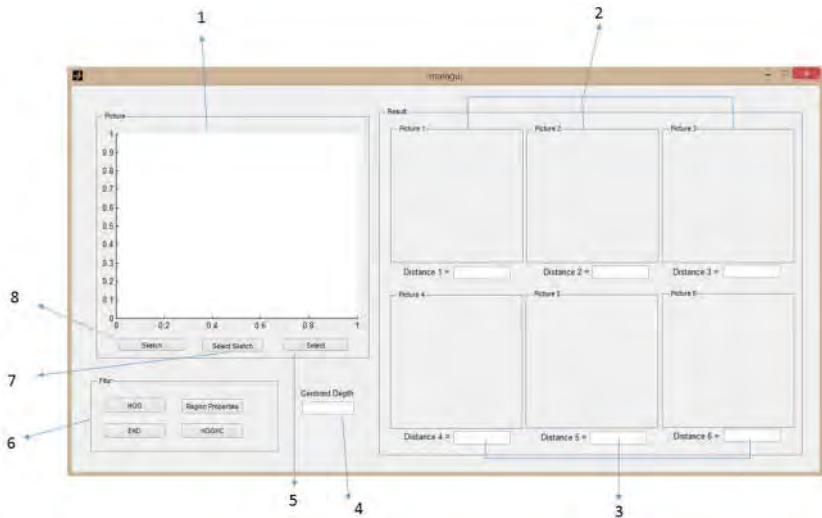
Gambar 3.9 merupakan gambar letak centroid pada kedalaman 1, 2 dan 3. Terlihat pada gambar bahwa letak centroid pada masing-masing kedalaman membentuk suatu struktur *tree*.



*Gambar 3.9 Gambar Hierarchical Centroid (a) Citra Hitam Putih (b) Letak centroid pada kedalaman 1 (c) Letak centroid pada kedalaman 2 (d) Letak centroid pada kedalaman 3*

### 3.5 Perancangan Antar Muka

Sistem ini memiliki antar muka agar *user* dapat dengan mudah untuk melakukan pencarian gambar. Antar muka sistem terdiri dari input gambar yang berupa sketsa dan output berupa gambar yang memiliki jarak kedekatan yang paling kecil dengan gambar pada database. Gambar 3.10 menampilkan perancangan antar muka pada sistem. Penjelasan dari setiap nomer dapat dilihat pada Tabel 3.1



Gambar 3.10 Desain Antar Muka Sistem

Tabel 3.1 Tabel Penjelasan Antar Muka Sistem

Nomor	Nama	Fungsi
1	Tampilan sketsa	Menampilkan sketsa gambar yang dipilih
2	Hasil gambar	Menampilkan 6 hasil gambar yang memiliki nilai kedekatan kecil sesuai pengukuran dengan fitur yang dipilih
3	Hasil Jarak	Menampilkan nilai jarak kedekatan antara input sketsa dengan gambar yang terdapat dalam database
4	Input Kedalaman	Memasukkan nilai kedalam yang dipakai untuk menghitung fitur <i>Hierarchical Centroid</i>
5	Pilih Gambar	Memilih gambar sketsa
6	Tombol Fitur	Pilihan fitur yang digunakan untuk ekstraksi fitur terhadap gambar sketsa dan dataset gambar



**Tabel 3.1 Tabel Penjelasan Antar Muka Sistem (lanjutan)**

<b>Nomor</b>	<b>Nama</b>	<b>Fungsi</b>
7	Pilih Gambar Sketsa	Memilih gambar sketsa yang telah dibuat dari sistem
8	Tombol Sketsa	Menampilkan tempat untuk menggambar sketsa

*(Halaman ini sengaja dikosongkan)*

## BAB IV IMPLEMENTASI

Bab ini berisi penjelasan mengenai implementasi dari perancangan yang sudah dilakukan pada bab sebelumnya. Implementasi berupa kode sumber untuk membangun program.

### 4.1 Lingkungan Implementasi

Implementasi metode kombinasi *Histogram of Oriented Gradient* dan *Hierarchical Centroid* untuk *Sketch Based Image Retrieval* menggunakan spesifikasi perangkat keras dan perangkat lunak seperti yang ditunjukkan pada Tabel 4.1.

**Tabel 4.1 Lingkungan Implementasi Perangkat Lunak**

Perangkat	Jenis Perangkat	Spesifikasi
Perangkat Keras	Prosesor	Intel(R) Core(TM) i7-4700HQ CPU @ 2.40GHz
	Memori	4 GB
Perangkat Lunak	Sistem Operasi	Windows 8.1 Pro
	Perangkat Pengembang	MATLAB R2014a

### 4.2 Implementasi

Pada sub bab implementasi ini menjelaskan mengenai pembangunan perangkat lunak secara detail dan menampilkan kode sumber yang digunakan mulai tahap *preprocessing* hingga klasifikasi. Pada tugas akhir ini data yang digunakan seperti yang telah dijelaskan di bab sebelumnya 32 data *testing* dan 1000 data *training*.

#### 4.2.1 Implementasi Menggambar sketsa

Kode Sumber 4.1 merupakan kode program untuk sketsa gambar. Baris 1 sampai 7 merupakan fungsi untuk menampilkan kotak untuk menggambar sketsa. Fungsi *figure* pada baris 2 digunakan untuk menampilkan *graphic user interface*. Baris 3 menunjukkan ukuran kotak menggambar. *XLim* merupakan limit sumbu x yaitu antara 0 sampai 0,1 sedangkan *YLim* merupakan limit untuk sumbu y yaitu antara 0 sampai 0,1. Saat user men-*click* mouse pada *figure window* maka dipanggil fungsi *down*. Baris 8 sampai 16 case *down* yang merupakan fungsi saat user men-*click* mouse. Baris 12 variabel *curpos* mengambil nilai *currentpoint* pada saat pertama kali menggambar. Pada baris ke 10 variabel *fig* sama dengan *gcbf*. *Gcbf* merupakan fungsi yang menangani fungsi *figure* saat terjadi eksekusi pada properti *callback*, saat tidak terjadi *callback* pada fungsi *figure* maka *gcbf* akan mengembalikan menjadi matriks kosong. Selanjutnya pada baris 16 menunjukkan jika terjadi pergerakan pada mouse maka akan memanggil fungsi *move* sedangkan saat user melepaskan tekanan pada mouse maka akan memanggil fungsi *up*.

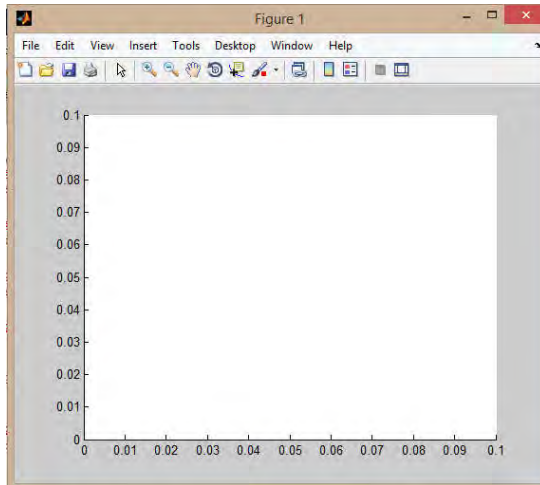
Baris ke 17 sampai 24 case *move* merupakan fungsi terjadi pergerakan pada mouse saat mouse di-*click*. Fungsi ini menyimpan info *UserData* seperti nilai *currentpoint*, nilai x dan y. Nilai x dan y saat terjadi perubahan disimpan dalam variabel *info.x* dan *info.y*. Baris ke 25 sampai 27 merupakan case *up* yang berfungsi saat user melepaskan *click* pada mouse. Kotak menggambar sketsa ditunjukkan pada Gambar 4.1

```

1 case 'init'
2     fig = figure('DoubleBuffer','on','back','off');
3     info.ax = axes('XLim',[0 0.1],'YLim',[0 0.1]);
4     info.drawing = [];
5     info.x = [];
6     info.y = [];
7     set(fig,'UserData',info,...
8         'WindowButtonDownFcn',[mfilename,'down'])
9 case 'down'
10    myname = mfilename;
11    fig = gcbf;
12    info = get(fig,'UserData');
13    curpos = get(info.ax,'CurrentPoint');
14    info.x = curpos(1,1);
15    info.y = curpos(1,2);
16    info.drawing = line(info.x,info.y,'Color','k');
17    set(fig,'UserData',info,...
18        'WindowButtonMotionFcn',[myname,'move'],...
19        'WindowButtonUpFcn',[myname,'up'])
20 case 'move'
21    fig = gcbf;
22    info = get(fig,'UserData');
23    curpos = get(info.ax,'CurrentPoint');
24    info.x = [info.x;curpos(1,1)];
25    info.y = [info.y;curpos(1,2)];
26    set(info.drawing,'XData',info.x,'YData',info.y)
27    set(fig,'UserData',info)
28 case 'up'
29    fig = gcbf;
30    set(fig,'WindowButtonMotionFcn','',...
31        'WindowButtonUpFcn','')

```

*Kode Sumber 4.1 Kode program sketsa gambar*



*Gambar 4.1 Tampilan sketsa gambar*

#### **4.2.2 Implementasi Pemrosesan Preprocessing Data Masukan**

Kode Sumber 4.2 merupakan kode program preprocessing menggunakan operator sobel. Baris 1 merupakan fungsi untuk merubah gambar RGB menjadi gambar grayscale. Tujuannya adalah untuk mengurangi dimensi fitur menjadi 2 dimensi agar lebih mudah dilakukan ekstraksi fitur. Baris 2 merupakan fungsi untuk menghasilkan nilai threshold pada operator sobel. Operator sobel terdapat pada fungsi edge. Fungsi edge adalah fungsi untuk mendeteksi tepi suatu obyek. Selanjutnya gambar dibuat menjadi gambar biner atau gambar hitam putih dengan nilai threshold yang baru. Nilai threshold yang baru berfungsi untuk mempertajam tepi pada obyek sehingga tepi yang dihasilkan lebih detail dan terlihat jelas.

1	<code>gray = rgb2gray(I);</code>
2	<code>[~, threshold] = edge(gray, 'sobel');</code>
3	<code>fudgeFactor = .5;</code>
4	<code>BWs = edge(gray, 'sobel', threshold * fudgeFactor);</code>

*Kode Sumber 4.2 Kode Program Proses Preprocessing dengan Sobel*

### 4.2.3 Implementasi Ekstraksi fitur dengan HOG

Langkah selanjutnya adalah melakukan ekstraksi fitur menggunakan HOG. Parameternya yang digunakan citra hitam putih hasil preprocessing menggunakan sobel. Kemudian dilakukan penghitungan magnitude dan sudut dari citra. Kode Sumber 4.3 merupakan kode program perhitungan magnitude dan sudut citra. Magnitude dan sudut didapatkan dari perhitungan yang telah dijelaskan pada bab 2.

1	<code>%konvolusi gradien sumbu x dan y</code>
2	<code>for i=1:baris-2</code>
3	<code>    Iy(i,:)=(img(i,:)-img(i+2,:));</code>
4	<code>end</code>
5	<code>for i=1:kolom-2</code>
6	<code>    Ix(:,i)=(img(:,i)-img(:,i+2));</code>
7	<code>end</code>
8	<code>magnitude=sqrt(Ix.^2 + Iy.^2); % menghitung magnitude</code>
9	<code>sudut=atand(Ix./Iy); % orientasi sudut sudut=imadd(sudut,90); % membuat range sudut(0,180)</code>
10	

*Kode Sumber 4.3 Kode Program Menghitung Magnitude dan Sudut Citra*

Setelah mendapatkan nilai magnitude dan sudut citra, dilakukan pembagian sel menjadi 8 x 8 piksel. Sel-sel tersebut kemudian dibagi menjadi blok-blok berukuran 2 x 2 sel dan

dilakukan normalisasi terhadap blok tersebut. Sebelum dinormalisasi setiap sel dilakukan perhitungan orientasi histogramnya. Penentuan nilai histogram berdasarkan *voting* dengan *bi-linear interpolation*. Proses perhitungan histogram dapat dilihat pada Kode Sumber 4.4.

```

% Vote dan Binning Process (Bi-Linear Interpolation)
1  if cell_sudut(m,n)>=0 && cell_sudut(m,n)<=10
    histogram(1)=histogram(1)+
2  cell_mag(m,n)*(cell_sudut(m,n)+10)/20;
    histogram(9)=histogram(9)+ cell_mag(m,n)*(10-
3  cell_sudut(m,n))/20;
4  elseif cell_sudut(m,n)>10 && cell_sudut(m,n)<=30
    histogram(1)=histogram(1)+ cell_mag(m,n)*(30-
5  ll_sudut(m,n))/20;
    histogram(2)=histogram(2)+
6  cell_mag(m,n)*(cell_sudut(m,n)-10)/20;
7  ...
8  elseif cell_sudut(m,n)>170 && cell_sudut(m,n)<=180
    histogram(9)=histogram(9)+ cell_mag(m,n)*(190-
9  cell_sudut(m,n))/20;
    histogram(1)=histogram(1)+
10 cell_mag(m,n)*(cell_sudut(m,n)-170)/20;
11 end

```

*Kode Sumber 4.4 Kode Program Proses Perhitungan Orientasi Sel Histogram*

Hasil dari proses perhitunagan orientasi sel histogram akan menghasilkan fitur blok. Fitur blok ini masih perlu dinormalisasi karena terdapat nilai yang overlapping sehingga mempengaruhi pada tingkat kecerahan pada citra. Normalisasi dilakukan dengan cara *euclidean norm* seperti yang terlihat pada baris 1. Kemudian setiap blok fitur digabungkan menjadi satu untuk membentuk fitur HOG. Proses normalisasi dapat dilihat pada Kode Sumber 4.5.



	<code>%Normalisasi block fitur L2-norm</code>
	<code>norm_fitur =</code>
1	<code>block_fitur/(sqrt(norm(block_fitur)^2+.001));</code>
2	<code>fitur = [fitur norm_fitur];</code>

*Kode Sumber 4.5 Kode Program Proses Normalisasi Fitur Blok*

Proses yang terakhir dilakukan adalah menggabungkan semua fitur blok yang telah dinormalisasi dan dilanjutkan dengan melakukan normalisasi pada fitur HOG menggunakan *euclidean norm* seperti yang telah dijelaskan pada bab 2. Baris ke 3 merupakan kode normalisasi dengan *euclidean norm*. Baris ke 4 sampai 8 merupakan kode untuk menghilangkan besar gradien yang lebih dari 0,2 dan merubahnya menjadi 0,2 hal ini dilakukan supaya nilai gradient yang besar tidak terlalu dominan. Selanjutnya dilakukan normalisasi lagi dengan *euclidean norm*. Proses normalisasi HOG dapat dilihat pada Kode Sumber 4.6.

1	<code>fitur(isnan(fitur))=0;</code>
2	<code>%Normalisasi fitur vektor</code>
3	<code>fitur=fitur/(sqrt(norm(fitur)^2+.001));</code>
4	<code>for z=1:length(fitur)</code>
5	<code>    if fitur(z)&gt;0.2</code>
6	<code>        fitur(z)=0.2;</code>
7	<code>    End</code>
8	<code>End</code>
9	<code>fitur=fitur/(sqrt(norm(fitur)^2+.001));</code>

*Kode Sumber 4.6 Kode Program Proses Normalisasi Fitur HOG*

Pada kode sumber diatas, fitur yang dihasilkan ditampung kedalam data berbentuk `.mat` sehingga setiap gambar hanya dilakukan satu kali perhitungan fitur. Hal ini juga untuk mempengaruhi kompleksitas waktu dalam menjalankan program.

#### 4.2.4 Implementasi Ekstraksi Fitur menggunakan Hierarchical Centroid

Kode Sumber 4.7 merupakan kode program untuk menghitung ekstraksi fitur kedua yaitu dengan metode *Hierarchical Centroid*. Metode ini dilakukan dengan melakukan perhitungan terhadap centroid obyek pada citra. Paramater fungsi ini adalah citra hitam putih hasil dari *preprocessing* menggunakan metode sobel, kedalaman dari hirarki centroidnya dan maksimal kedalaman hirarki centroidnya. Langkah pertama yang dilakukan adalah dengan memasukkan kedalaman dari hirarki centroidnya untuk mendeklarasikan nilai maksimal kedalaman hirarki centroidnya. Kemudian menghitung luas dari obyek pada citra yang ditampilkan pada baris ke 5 dan menghitung nilai centroid pada citra yang ditampilkan pada baris ke 7 sampai 15. Setelah mendapatkan nilai centroid pada sumbu x untuk level pertama maka citra akan dibagi menjadi dua citra bagian kanan dan citra bagian kiri yang ditampilkan pada baris ke 16 sampai 17. Dan selanjutnya dilakukan proses rekursif terhadap citra bagian kanan dan citra bagian kiri sesuai dengan kedalaman hirarki yang diinginkan yang ditampilkan pada baris 18 dan 19.

```

1 function p = hierarchicalCentroidRec(im, depth,
2   maxDepth)
3   if depth > maxDepth
4     p = [];
5   Else
6     area = sum(sum(im));
7     [rows,cols] = size(im);
8     if cols == 1    % menghitung nilai centroid-x
9       centroid = 0.5;
10      elseif area == 0
11        centroid = cols/2;
12      elseif rows == 1

```

Kode Sumber 4.7 Kode Program Menghitung Nilai Centroid

```

12 centroid = (im*[1:cols]')/area;
13 Else
14 centroid = sum(im)*[1:cols]'/area;
15 End
16 leftIm = im(:,1:floor(centroid));
17 rightIm = im(:,ceil(centroid):end);
18 pLeft = hierarchicalCentroidRec(leftIm' ,
19 depth+1, maxDepth);
20 pRight = hierarchicalCentroidRec(rightIm' ,
21 depth+1, maxDepth);
22 % Update jarak sehingga masih beralasi dengan
23 gambar
24 if size(pRight,1) > 1
25 ind = find((1-mod(depth,2))-
26 mod(pRight(2:end,2),2)) + 1;
27 pRight(ind,1) = pRight(ind,1) + ceil(centroid) -
28 1;
29 pRight(1,1) = pRight(1,1) - 1;
30 end
31 p = [centroid, depth; pLeft; pRight];
32 end
33 end

```

*Kode Sumber 4.8 Kode Program Menghitung Nilai Centroid  
(lanjutan)*

Setelah didapatkan nilai centroid dari citra dan citra transpose, dilakukan normalisasi data agar nilainya tidak ada yang dominan. Proses normalisasi dapat dilihat pada kode sumber 4.8. Pada kode sumber 4.8, fitur yang dihasilkan ditampung kedalam data berbentuk .mat sehingga setiap gambar hanya dilakukan satu kali perhitungan fitur. Hal ini juga untuk mempengaruhi kompleksitas waktu dalam menjalankan program.

```

1 function[vec levels] = hierarchicalCentroid1(im, d,
2 plotFlag)
3     p = hierarchicalCentroidRec(im, 1, d);
4     meany = ([1:size(im,1)]*sum(im,2))/sum(sum(im));
5     indVer = (mod(p(:,2),2) == 1);
6     % Normalisasi lokasi:
7     p(indVer,1) = p(indVer,1) - p(1,1);
8     p(~indVer,1) = p(~indVer,1) - meany;
9     % Normalisasi ukuran (keeping aspect ratio):
10    p(:,1) = p(:,1)/size(im,1);
11    vec = p(2:end,1)';
12    levels = p(2:end,2)';
13 end

```

*Kode Sumber 4.9 Kode Program Normalisasi fitur Hierarchical Centroid*

#### 4.2.5 Implementasi Penghitungan jarak dengan Euclidean Distance

Setelah didapatkan fitur dari data *training* dan data *testing* kemudian dilakukan perhitungan jarak kedekatan menggunakan *euclidean distance*. Proses tersebut dapat dilihat pada kode sumber 4.9. Baris 4 sampai 7 merupakan rumus untuk mendapatkan jarak antara data *training* dan data *testing*.

```

1 for n=1 : length(training.hasil)
2     training.hasil{n,2} = 0;
3     for m = 1 : size(training.fiturHOG,2)
4         temp = ((training.fiturHOG(n,m) - fitur1(1,m)).^2);
5         training.hasil{n,2} = training.hasil{n,2} + temp;
6     end
7     training.hasil{n,3} = sqrt(training.hasil{n,2});
8 end
9 hasilakhir = sortrows(training.hasil,3);

```

*Kode Sumber 4.10 Kode Program Euclidean Distance*

Dari hasil perhitungan jarak dengan *euclidean distance* kemudian dilakukan *sorting* secara *ascending* untuk mendapatkan nilai *similarity* yang paling kecil. Baris ke 9 merupakan fungsi untuk melakukan pengurutan secara *ascending*. Hasil dari program ini adalah menampilkan citra yang memiliki tingkat kemiripan yang tinggi.

*(Halaman ini sengaja dikosongkan)*

## BAB V HASIL UJI COBA DAN EVALUASI

Bab ini berisi penjelasan mengenai skenario uji coba dan evaluasi pada proses pencarian gambar dengan metode *Histogram of Oriented Gradients* dan *Hierarchical Centroid*. Hasil uji coba didapatkan dari implementasi pada bab 4 dengan skenario yang berbeda. Bab ini berisikan pembahasan mengenai lingkungan pengujian, data pengujian, dan uji kinerja.

### 5.1 Lingkungan Pengujian

Lingkungan pengujian pada uji coba permasalahan pencarian gambar menggunakan *Histogram of Oriented Gradients* dan *Hierarchical Centroid* menggunakan spesifikasi perangkat keras dan perangkat lunak seperti yang ditunjukkan pada Tabel 5.1.

**Tabel 5.1 Spesifikasi Lingkungan Pengujian**

Perangkat	Jenis Perangkat	Spesifikasi
Perangkat Keras	Prosesor	Intel(R) Core(TM) i7-4700HQ CPU @ 2.40GHz
	Memori	4 GB
Perangkat Lunak	Sistem Operasi	Windows 8.1 Pro
	Perangkat Pengembang	MATLAB R2014a

### 5.2 Data Pengujian

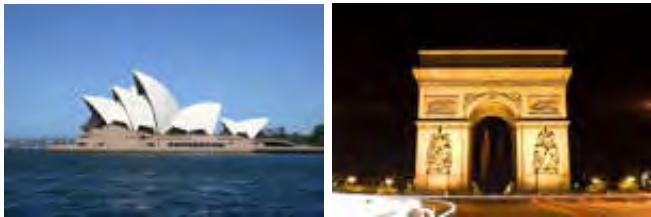
Subbab ini menjelaskan mengenai data yang digunakan pada uji coba. Seperti yang telah dijelaskan sebelumnya, data berasal dari Data training gambar diambil dari *Centre for Vision, Speech and Signal Processing*, University of Surrey, United Kingdom.

Data yang akan diolah adalah 1000 buah gambar yang terdiri 10 macam gambar atau kelas. Data testing terdiri dari 30 gambar sketsa dari 10 kelas gambar.

Gambar sketsa dan gambar training kemudian dilakukan *preprocessing* untuk membuat gambar dapat dengan mudah diolah yaitu dengan mendeteksi tepi pada gambar. Proses *preprocessing* dilakukan dengan metode sobel. Keluaran dari *preprocessing* gambar hitam putih. Kemudian gambar hitam putih dilakukan ekstraksi fitur untuk menghasilkan vektor fitur menggunakan *Histogram of Oriented Gradients* dan *Hierarchical Centroid*. Pada metode *Hierarchical Centroid* terdapat beberapa kedalaman hirarki centroid yang diuji yang digunakan pada skenario uji coba untuk mendapatkan hasil yang optimal. Dari fitur vektor yang didapat antara data testing dan data training dilakukan perhitungan jarak menggunakan *euclidean distance*

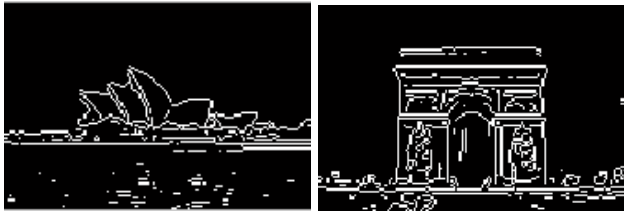
### 5.3 Preprocessing Gambar

Pada tahap *preprocessing* gambar dilakukan untuk mendeteksi tepi pada obyek menggunakan metode sobel. Gambar 5.1 dan 5.2 merupakan gambar pada data training sebelum dan sesudah dilakukan deteksi tepi. Serta gambar 5.3 dan 5.4 merupakan gambar pada data testing sebelum dan sesudah dilakukan deteksi tepi.

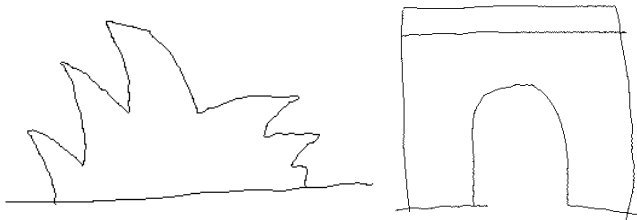


*Gambar 5.1 Gambar Asli Data Training*

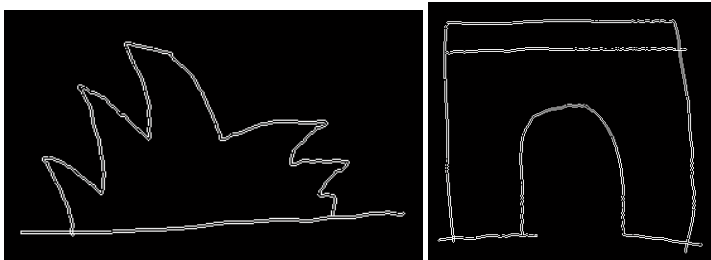




*Gambar 5.2 Gambar Data Training Setelah dilakukan Preprocessing*



*Gambar 5.3 Gambar Asli Data Testing*



*Gambar 5.4 Gambar Data Testing Setelah Dilakukan Preprocessing*

Pada Gambar 5.1 dan 5.2 menunjukkan gambar data *training* sebelum dan sesudah dilakukan *preprocessing* dengan metode sobel. Sedangkan Gambar 5.3 dan 5.4 menunjukkan gambar data *testing* sebelum dan sesudah dilakukan *preprocessing* dengan metode sobel. Untuk semua data testing dapat dilihat pada lampiran halaman 65. Terlihat perbedaan pada gambar, gambar

setelah dilakukan preprocessing menjadi gambar hitam putih dan hanya terlihat bagian tepi pada obyek. Tetapi untuk gambar training deteksi tepi masih kurang bagus karena masih terdapat noise pada gambar sehingga mempengaruhi pada ekstraksi fitur.

#### 5.4 Skenario Uji Coba

Sebelum melakukan uji coba, perlu ditentukan skenario yang akan digunakan dalam uji coba. Melalui skenario ini, perangkat akan diuji apakah sudah berjalan dengan benar dan bagaimana performa pada masing-masing skenario. Dan membandingkan skenario manakah yang memiliki hasil lebih baik. Terdapat 2 macam skenario uji coba, yaitu :

1. Perhitungan nilai performa dengan metode *Histogram of Oriented Gradients* dan *Hierarchical Centroid* dengan kedalaman hirarki 2, 3, 4, 5, dan 6.
2. Perbandingan nilai performa dengan metode *Histogram of Oriented Gradients* dan *Hierarchical Centroid*, metode *Histogram of Oriented Gradients*, metode *Region properties*, dan *Edge of Histogram Descriptor*.

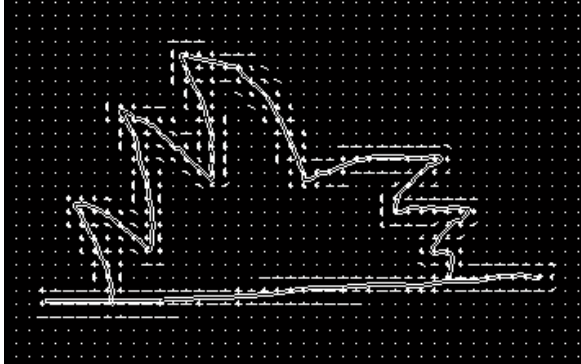
##### 5.4.1 Skenario Uji Coba 1

Skenario uji coba 1 adalah perhitungan Perhitungan nilai performa dengan metode *Histogram of Oriented Gradients* dan *Hierarchical Centroid* dengan kedalaman hirarki 2, 3, 4, 5, dan 6. Performa yang dihitung adalah nilai *Average Normalized Modified Retrieval Ranking* (ANMRR), nilai presisi, recall dan nilai akurasi. Setiap kedalaman diuji pada 30 query data testing.

##### 5.4.1.1 Nilai ANMRR

Perhitungan nilai ANMRR bertujuan untuk mengetahui nilai dari perfoma metode yang diujikan. Hasil uji ANMRR pada 30 query dengan metode *Histogram of Oriented Gradients* dan *Hierarchical Centroid* dengan kedalaman hirarki 2, 3, 4, 5, dan 6 dapat dilihat pada Tabel 5.2. Gambar 5.5 menampilkan fitur HOG pada gambar *testing*. Fitur HOG yang didapatkan berjumlah 5940.

Nilai tersebut didapat dari 11 *cell* horizontal dan 15 *cell* vertikal, dan dengan 4 blok serta 9 bin histogram. Sehingga fitur yang dihasilkan berjumlah  $11 \times 15 \times 4 \times 9 = 5940$  fitur. Rentang nilai fitur HOG bernilai dari 0 sampai 1 karena telah dilakukan normalisasi.



*Gambar 5.5 Fitur HOG pada Gambar Testing*

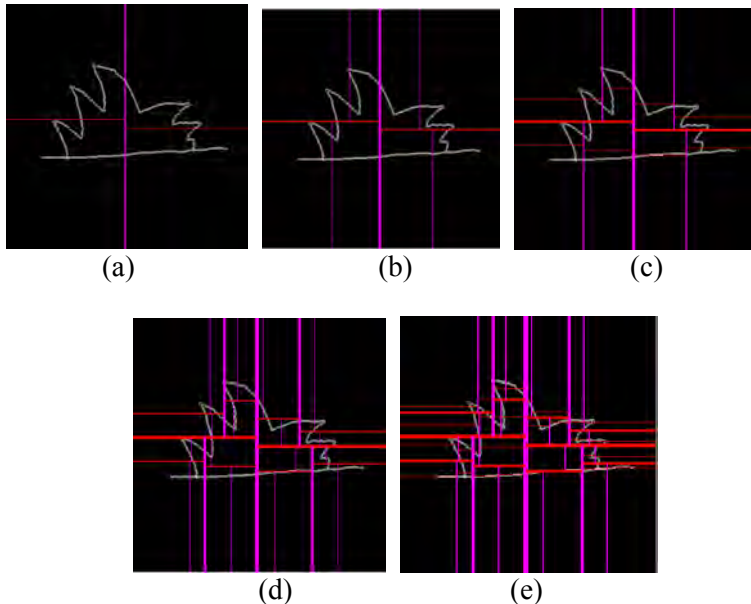
**Tabel 5.2 Performa ANMRR pada kedalaman yang berbeda**

Queri	Kedalaman					Rata-rata
	2	3	4	5	6	
1	0,469	0,344	0,347	0,347	0,359	0,373
2	0,432	0,258	0,265	0,272	0,286	0,303
3	0,292	0,204	0,209	0,215	0,224	0,229
4	0,373	0,341	0,330	0,343	0,347	0,347
5	0,388	0,280	0,252	0,264	0,258	0,288
6	0,549	0,530	0,482	0,476	0,449	0,497
7	0,604	0,532	0,561	0,611	0,680	0,598
8	0,593	0,567	0,576	0,599	0,616	0,590
9	0,675	0,388	0,598	0,687	0,877	0,645
10	0,706	0,527	0,734	0,807	0,924	0,740
11	0,833	0,657	0,679	0,752	0,788	0,742

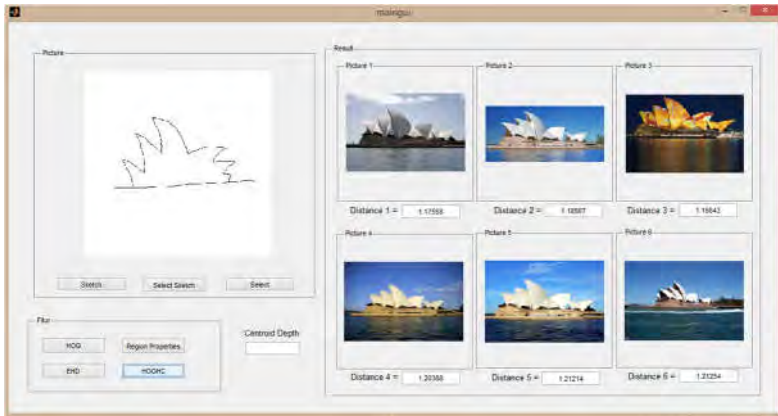
Queri	Kedalaman					Rata-rata
	2	3	4	5	6	
12	0,755	0,586	0,695	0,758	0,873	0,733
13	0,531	0,333	0,310	0,378	0,376	0,386
14	0,436	0,275	0,253	0,230	0,204	0,280
15	0,437	0,173	0,171	0,206	0,205	0,238
16	0,722	0,627	0,653	0,693	0,724	0,684
17	0,373	0,370	0,375	0,378	0,397	0,379
18	0,570	0,520	0,522	0,520	0,527	0,532
19	0,238	0,207	0,218	0,520	0,271	0,291
20	0,288	0,260	0,275	0,274	0,299	0,279
21	0,307	0,293	0,306	0,321	0,359	0,317
22	0,443	0,149	0,164	0,190	0,234	0,236
23	0,626	0,506	0,494	0,498	0,501	0,525
24	0,569	0,260	0,299	0,363	0,414	0,381
25	0,403	0,328	0,325	0,325	0,325	0,341
26	0,294	0,179	0,189	0,201	0,236	0,220
27	0,319	0,151	0,154	0,159	0,165	0,190
28	0,256	0,273	0,300	0,356	0,433	0,324
29	0,329	0,163	0,192	0,251	0,392	0,265
30	0,397	0,270	0,278	0,369	0,440	0,351
ANMRR	0,474	<b>0,352</b>	0,374	0,412	0,439	0,410

Sedangkan untuk fitur *Hierarchical Centroid* pada kedalaman 2 menghasilkan 4 fitur, kedalaman 3 menghasilkan 12 fitur, kedalaman 4 menghasilkan 28 fitur, kedalaman 5 menghasilkan 60 fitur, dan kedalaman 6 menghasilkan 126 fitur. Berdasarkan hasil yang ditunjukkan pada perhitungan performa ANMRR diatas menunjukkan bahwa pada kedalaman 3 menghasilkan nilai ANMRR yang paling baik yaitu 0,352. Sedangkan pada kedalaman 2 menghasilkan nilai ANMRR 0,474.

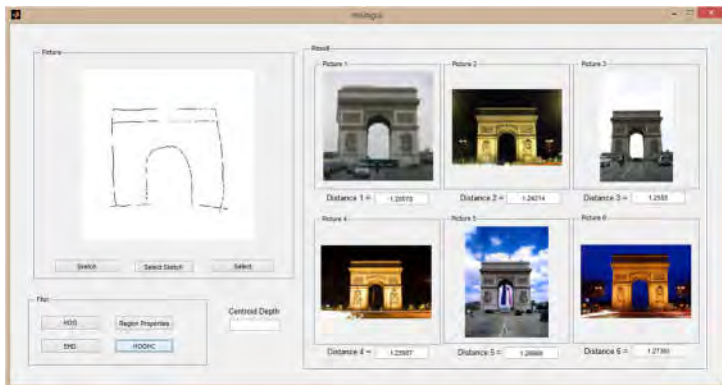
Pada kedalaman 4 menghasilkan nilai 0,374, pada kedalaman 5 menghasilkan nilai 0,412 dan pada kedalaman 6 menghasilkan nilai 0,439. Hal ini menunjukkan bahwa letak centroid yang paling baik ada pada kedalaman 3 terlihat pada Gambar 5.6 yang menampilkan perbandingan letak centroid untuk masing-masing kedalaman. Dapat dilihat pada Gambar 5.6 bahwa semakin tinggi kedalamannya letak centroidnya semakin tidak akurat, banyak letak centroidnya ada yang melebihi dari gambar obyek. Sehingga pada kedalaman 3 menampilkan letak centroid yang lebih baik dari pada kedalaman yang lain. Untuk hasil keluaran dari program dapat dilihat pada Gambar 5.7 dan Gambar 5.8 secara rinci dapat dilihat pada lampiran halaman 67.



Gambar 5.6 Letak Centroid (a) Kedalaman 2 (b) Kedalaman 3 (c) Kedalaman 4 (d) Kedalaman 5 (e) Kedalaman 6

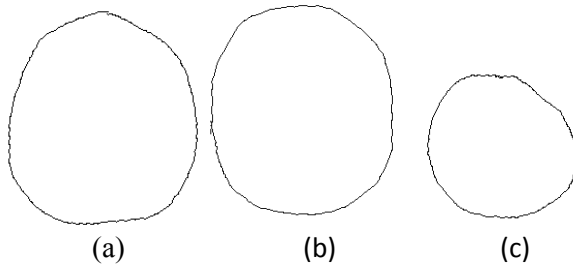


*Gambar 5.7 Keluaran program Queri ke 16*



*Gambar 5.8 Keluaran program Queri ke 28*

Pada Tabel 5.2 dapat dilihat bahwa nilai ANMRR yang terbesar ada pada queri ke 10, 11 dan 12 jika dilihat pada gambar, gambar ke 10, 11 dan 12 merupakan gambar berbentuk lingkaran dapat dilihat pada Gambar 5.9.



Gambar 5.9 Gambar *Queri* (a) *Queri* 10 (b) *Queri* 11 (c) *Queri* 12

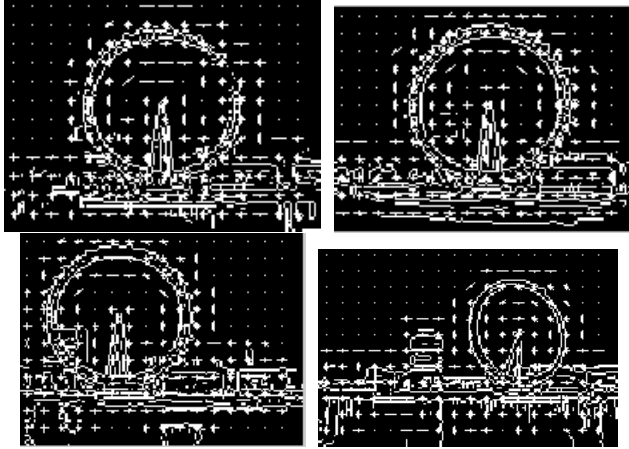
Hal tersebut dikarenakan pada saat melakukan preprocessing pada gambar training hasil tidak terlalu bagus karena masih banyak *noise* yang terdapat pada gambar training. Contoh hasil *preprocessing* pada gambar training untuk *queri* 10, 11 dan 12 dapat dilihat pada gambar 5.10.



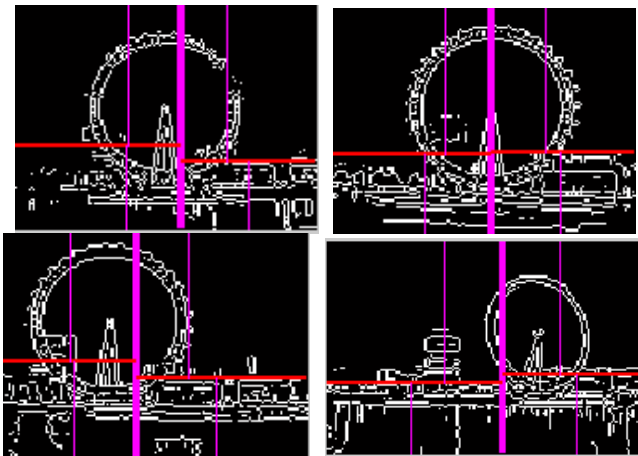
Gambar 5.10 Contoh hasil *preprocessing*

Noise yang terdapat pada hasil preprocessing dapat mengganggu pada saat dilakukan ekstraksi fitur. Pada saat melakukan ekstraksi fitur dengan HOG *noise* tersebut dapat terdeteksi sebagai obyek, sehingga obyek yang sebenarnya menjadi tidak dominan. Contoh terdapat pada Gambar 5.11 untuk hasil ekstraksi fitur dengan HOG dan Gambar 5.12 untuk ekstraksi fitur

dengan *Hierarchical Centroid*. Hal tersebut membuat fitur HOG dan HC menjadi tidak akurat. Hasil keluaran dari program pada query 10, 11, dan 12 ditampilkan pada Gambar 5.13, Gambar 5.14, dan Gambar 5.15.

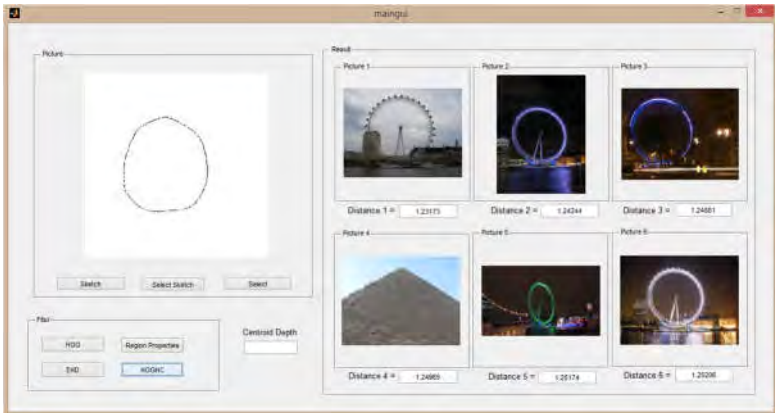


Gambar 5.11 Gambar Hasil Ekstraksi fitur HOG

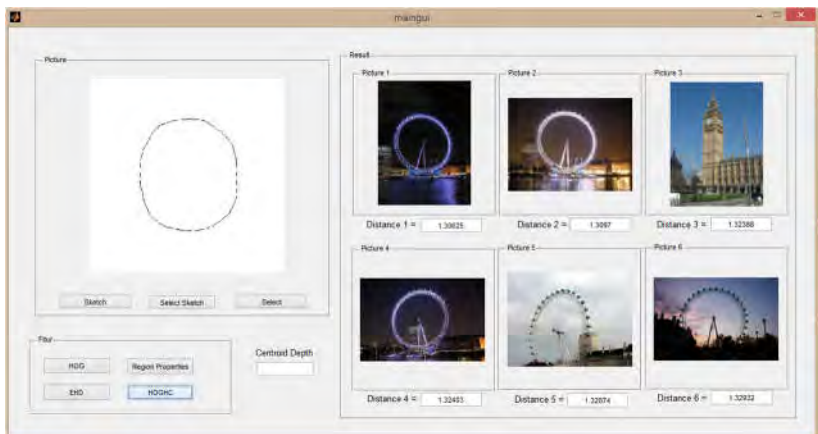


Gambar 5.12 Gambar Hasil Ekstraksi Fitur Hierarchical Centroid

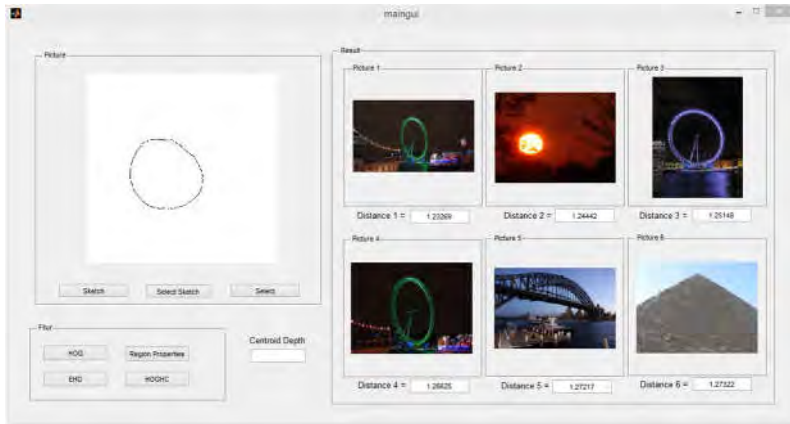




*Gambar 5.13 Hasil Keluaran Query ke-10*



*Gambar 5.14 Hasil Keluaran Query ke-11*



Gambar 5.15 Hasil Keluaran Query ke-12

#### 5.4.1.2 Nilai Presisi, Recall, dan Akurasi

Dari Confusion matrix yang dihasilkan dapat diperoleh nilai presisi, recall dan akurasinya. Nilai presisi menunjukkan banyaknya gambar yang relevan dari sekumpulan gambar yang ditampilkan. Nilai recall menunjukkan banyaknya gambar yang ditampilkan dari sejumlah gambar yang relevan di dalam database gambar. Sedangkan akurasi menunjukkan keakuratan metode dalam menampilkan gambar yang sesuai. Tabel 5.3 menampilkan nilai presisi, recall, dan akurasi pada kedalaman 2, 3, 4, 5, dan 6

Tabel 5.3 Tabel Nilai Presisi Recall dan Akurasi

Kedalaman	Presisi	Recall	Akurasi
2	0,731	0,731	0,939
<b>3</b>	<b>0,785</b>	<b>0,785</b>	<b>0,957</b>
4	0,779	0,779	0,956
5	0,762	0,762	0,952
6	0,756	0,756	0,951

Dari tabel diatas dapat dilihat bahwa nilai presisi, recall dan akurasi yang paling tinggi terdapat pada kedalaman 3 sebesar 0,785 untuk presisi dan recall serta 0,957 untuk akurasi. Nilai presisi dan recall sama karena jumlah gambar yang ditampilkan ada 100 gambar dan jumlah gambar setiap kelas ada 100 gambar. Untuk rincian nilai presisi, recall dan akurasi pada masing masing queri ditampilkan pada lampiran halaman 73. Hal ini menunjukkan bahwa nilai presisi, recall dan akurasi sebanding dengan nilai ANMRR. Semakin bagus nilai ANMRR maka nilai presisi, recall dan akurasi juga akan semakin bagus.

#### 5.4.2 Skenario Uji Coba 2

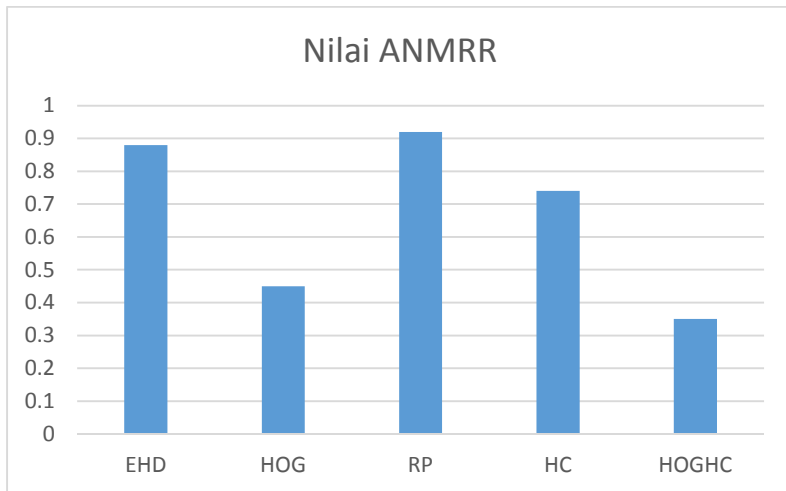
Pada skenario uji coba 2 dilakukan perhitungan nilai *Average Normalized Modified Retrieval Rank* (ANMRR) dengan metode *Histogram of Oriented Gradients* dan *Hierarchical Centroid*, metode *Histogram of Oriented Gradients*, metode *Region properties*, dan *Edge of Histogram Descriptor*. Uji coba ini dilakukan untuk membandingkan apakah metode HOGHC lebih baik dari metode lainnya. Tabel 5.4 dan Tabel 5.5 menampilkan nilai performa dari masing masing metode.

**Tabel 5.4 Performa ANMRR pada masing masing metode**

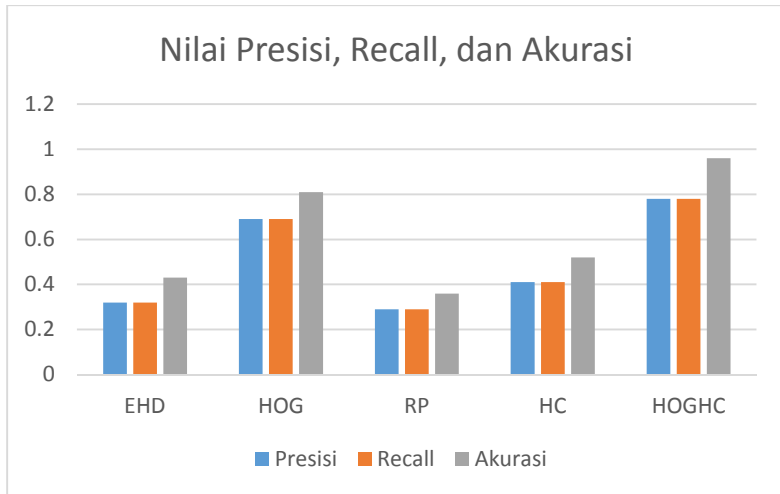
Metode	ANMRR
EHD	0,88
HOG	0,45
RP	0,92
HC	0,74
<b>HOGHC</b>	<b>0,35</b>

**Tabel 5.5 Performa Presisi, Recall dan Akurasi pada Masing Masing Metode**

Metode	Presisi	Recall	Akurasi
EHD	0,32	0,32	0,43
HOG	0,69	0,69	0,81
RP	0,29	0,29	0,36
HC	0,41	0,41	0,52
<b>HOGHC</b>	<b>0,78</b>	<b>0,78</b>	<b>0,96</b>



*Gambar 5.16 Grafik Perbandingan Nilai ANMRR*





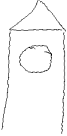
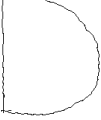

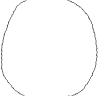

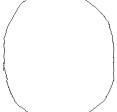



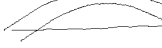

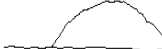


Gambar 5.17 Grafik Perbandingan Nilai Presisi, Recall, dan Akurasi

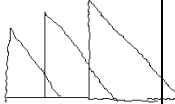







Pada Tabel 5.5 dan Gambar 5.16 terlihat bahwa nilai ANMRR paling kecil terdapat pada metode kombinasi *Histogram of Oriented Gradient* dan *Hierarchical Centroid* sebesar 0.52. Sedangkan pada metode *Edge of Histogram Descriptor* nilai ANMRR sebesar 0.88, pada metode *Histogram of Oriented Gradients* sebesar 0.58, *Region Properties* sebesar 0.92 dan metode *Hierarchical Centroid* sebesar 0,74. Hal ini menunjukkan bahwa menggunakan metode HOGHC dapat menghasilkan perankingan lebih baik dari metode lain yang umumnya digunakan pada permasalahan Sketch Based Image Retrieval. Sedangkan pada Tabel 5.6 dan Gambar 5.17 terlihat bahwa nilai presisi, recall, dan akurasi tertinggi terdapat pada metode HOGHC, sehingga HOGHC lebih baik digunakan dalam pencarian gambar berbasis sketsa dari pada metode-metode yang lainnya seperti *Histogram of Oriented Gradient*, *Edge of Histogram Descriptor*, *Region Properties*, dan *Hierarchical Centroid*



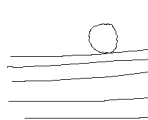
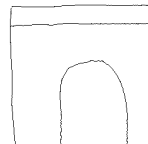
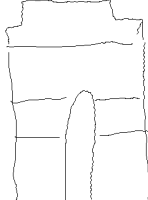

*(Halaman ini sengaja dikosongkan)*

## A. LAMPIRAN

Tabel A.1 Queri Gambar







No	Gambar	Nama Gambar	No	Gambar	Nama Gambar
1		Bigben-1.png	8		Burj-2.png
2		Bigben-2.png	9		Burj-3.png
3		Bigben-3.png	10		EyeLondon-1.png
4		Bintang-1.png	11		EyeLondon-2.png
5		Bintang-2.png	12		EyeLondon-3.png
6		Bintang-3.png	13		Jembatan-1.png
7		Burj-1.png	14		Jembatan-2.png
			15		Jembatan-3.png
			16		Opera-1.png







No	Gambar	Nama Gambar
17		Opera-2.png
18		Opera-3.png
19		Piramida-1.png
20		Piramida-2.png
21		Piramida-3.png
22		Sepeda-1.png
23		Sepeda-2.png
24		Sepeda-3.png

No	Gambar	Nama Gambar
25		Sunset-1.png
26		Sunset-2.png
27		Sunset-3.png
28		Triumph-1.png
29		Triumph-2.png
30		Triumph-3.png




Tabel A.2 Hasil Keluaran Query ke-16





Urutan	Gambar	Kelas	Jarak
1		opera	1.142
2		opera	1.142
3		opera	1.159
4		opera	1.163
5		opera	1.168
6		opera	1.203







Urutan	Gambar	Kelas	Jarak
7		opera	1.207
8		opera	1.226
9		opera	1.226
10		opera	1.227
11		opera	1.230
12		opera	1.251

Urutan	Gambar	Kelas	Jarak
13		opera	1.254
14		opera	1.262
15		opera	1.276
16		opera	1.278
17		opera	1.282
18		opera	1.292
19		opera	1.296

Urutan	Gambar	Kelas	Jarak
20		opera	1.299

**Tabel A.3 Hasil Keluaran Query ke-28**

Urutan	Gambar	Kelas	Jarak
1		'triumph'	1.200
2		'triumph'	1.206
3		'triumph'	1.216
4		'triumph'	1.217

Urutan	Gambar	Kelas	Jarak
5		'triumph'	1.220
6		'triumph'	1.222
7		'triumph'	1.225
8		'triumph'	1.226
9		'triumph'	1.227
10		'triumph'	1.229

Urutan	Gambar	Kelas	Jarak
11		'triumph'	1.229
12		'triumph'	1.232
13		'triumph'	1.235
14		'triumph'	1.240
15		'triumph'	1.242
16		'triumph'	1.245
17		'triumph'	1.246

Urutan	Gambar	Kelas	Jarak
18		'triumph'	1.246
19		'triumph'	1.251
20		'triumph'	1.252

**Tabel A.4 Nilai Performa pada Kedalaman 2 metode HOGHC**

Queri	Nmrr	Presisi	Recall	Akurasi
1	0,469	0,700	0,700	0,940
2	0,432	0,730	0,730	0,946
3	0,292	0,810	0,810	0,962
4	0,373	0,920	0,920	0,984
5	0,388	0,890	0,890	0,978
6	0,549	0,740	0,740	0,740
7	0,604	0,520	0,520	0,904
8	0,593	0,390	0,390	0,878
9	0,833	0,340	0,340	0,868
10	0,706	0,650	0,650	0,930
11	0,675	0,650	0,650	0,930
12	0,755	0,570	0,570	0,914

Queri	Nmrr	Presisi	Recall	Akurasi
13	0,531	0,740	0,740	0,948
14	0,436	0,840	0,840	0,968
15	0,437	0,770	0,770	0,954
16	0,722	0,390	0,390	0,878
17	0,373	0,720	0,720	0,944
18	0,570	0,560	0,560	0,912
19	0,238	0,850	0,850	0,970
20	0,288	0,760	0,760	0,952
21	0,307	0,760	0,760	0,952
22	0,443	0,930	0,930	0,986
23	0,626	0,800	0,800	0,960
24	0,569	0,880	0,880	0,976
25	0,403	0,870	0,870	0,974
26	0,294	0,870	0,870	0,974
27	0,319	0,790	0,790	0,958
28	0,256	0,880	0,880	0,976
29	0,329	0,790	0,790	0,958
30	0,397	0,820	0,820	0,964
Rata-rata	0,474	0,731	0,731	0,939

**Tabel A.5 Nilai Performa pada Kedalaman 3 metode HOGHC**

Queri	Nmrr	Presisi	Recall	Akurasi
1	0,344	0,790	0,790	0,958
2	0,258	0,780	0,780	0,956
3	0,204	0,890	0,890	0,978
4	0,341	0,960	0,960	0,992
5	0,280	0,930	0,930	0,986



Queri	Nmrr	Presisi	Recall	Akurasi
6	0,530	0,780	0,780	0,956
7	0,532	0,620	0,620	0,924
8	0,567	0,430	0,430	0,886
9	0,657	0,390	0,390	0,878
10	0,527	0,760	0,760	0,952
11	0,388	0,770	0,770	0,954
12	0,586	0,730	0,730	0,946
13	0,333	0,840	0,840	0,968
14	0,275	0,930	0,930	0,986
15	0,173	0,930	0,930	0,986
16	0,627	0,420	0,420	0,884
17	0,370	0,730	0,730	0,946
18	0,520	0,570	0,570	0,914
19	0,207	0,870	0,870	0,974
20	0,260	0,740	0,740	0,948
21	0,293	0,750	0,750	0,950
22	0,149	0,930	0,930	0,986
23	0,506	0,910	0,910	0,982
24	0,260	0,920	0,920	0,984
25	0,328	0,900	0,900	0,980
26	0,179	0,890	0,890	0,978
27	0,151	0,890	0,890	0,978
28	0,273	0,830	0,830	0,966
29	0,163	0,860	0,860	0,972
30	0,270	0,810	0,810	0,962
Rata-rata	0,352	0,785	0,785	0,957

**Tabel A.6 Nilai Performa pada Kedalaman 4 metode HOGHC**

Queri	nmrr	presisi	recall	akurasi
1	0.347	0.780	0.780	0.956
2	0.265	0.780	0.780	0.956
3	0.209	0.890	0.890	0.978
4	0.330	0.970	0.970	0.994
5	0.252	0.930	0.930	0.986
6	0.482	0.790	0.790	0.958
7	0.561	0.640	0.640	0.928
8	0.576	0.440	0.440	0.888
9	0.679	0.390	0.390	0.878
10	0.734	0.690	0.690	0.938
11	0.598	0.690	0.690	0.938
12	0.695	0.660	0.660	0.932
13	0.310	0.880	0.880	0.976
14	0.253	0.930	0.930	0.986
15	0.171	0.930	0.930	0.986
16	0.653	0.430	0.430	0.886
17	0.375	0.730	0.730	0.946
18	0.522	0.560	0.560	0.912
19	0.218	0.840	0.840	0.968
20	0.275	0.740	0.740	0.948
21	0.306	0.740	0.740	0.948
22	0.164	0.930	0.930	0.986
23	0.494	0.920	0.920	0.984
24	0.299	0.920	0.920	0.984
25	0.325	0.880	0.880	0.976
26	0.189	0.890	0.890	0.978

Queri	nmrr	presisi	recall	akurasi
27	0.154	0.890	0.890	0.978
28	0.300	0.830	0.830	0.966
29	0.192	0.860	0.860	0.972
30	0.278	0.820	0.820	0.964
Rata-rata	0.374	0.779	0.779	0.956

**Tabel A.7 Nilai Performa pada Kedalaman 5 metode  
HOGHC**

Queri	nmrr	presisi	recall	akurasi
1	0.347	0.780	0.780	0.956
2	0.272	0.790	0.790	0.958
3	0.215	0.870	0.870	0.974
4	0.343	0.970	0.970	0.994
5	0.264	0.930	0.930	0.986
6	0.476	0.800	0.800	0.960
7	0.611	0.590	0.590	0.918
8	0.599	0.410	0.410	0.882
9	0.752	0.340	0.340	0.868
10	0.807	0.620	0.620	0.924
11	0.687	0.640	0.640	0.928
12	0.758	0.580	0.580	0.916
13	0.378	0.870	0.870	0.974
14	0.230	0.930	0.930	0.986
15	0.206	0.920	0.920	0.984
16	0.693	0.440	0.440	0.888
17	0.378	0.720	0.720	0.944
18	0.520	0.570	0.570	0.914
19	0.520	0.570	0.570	0.914

Queri	nmrr	presisi	recall	akurasi
20	0.274	0.840	0.840	0.968
21	0.321	0.740	0.740	0.948
22	0.190	0.920	0.920	0.984
23	0.498	0.920	0.920	0.984
24	0.363	0.910	0.910	0.982
25	0.325	0.890	0.890	0.978
26	0.201	0.880	0.880	0.976
27	0.159	0.910	0.910	0.982
28	0.356	0.830	0.830	0.966
29	0.251	0.860	0.860	0.972
30	0.369	0.810	0.810	0.962
Rata-rata	0.412	0.762	0.762	0.952

**Tabel A.8 Nilai Performa pada Kedalaman 6 metode HOGHC**

Queri	nmrr	presisi	recall	akurasi
1	0.359	0.790	0.790	0.958
2	0.286	0.800	0.800	0.960
3	0.224	0.870	0.870	0.974
4	0.347	0.970	0.970	0.994
5	0.258	0.930	0.930	0.986
6	0.449	0.800	0.800	0.960
7	0.680	0.610	0.610	0.922
8	0.616	0.400	0.400	0.880
9	0.788	0.330	0.330	0.866
10	0.924	0.490	0.490	0.898
11	0.877	0.520	0.520	0.904
12	0.873	0.510	0.510	0.902

Queri	nmrr	presisi	recall	akurasi
13	0.376	0.870	0.870	0.974
14	0.204	0.930	0.930	0.986
15	0.205	0.920	0.920	0.984
16	0.724	0.430	0.430	0.886
17	0.397	0.710	0.710	0.942
18	0.527	0.570	0.570	0.914
19	0.271	0.820	0.820	0.964
20	0.299	0.750	0.750	0.950
21	0.359	0.710	0.710	0.942
22	0.234	0.920	0.920	0.984
23	0.501	0.930	0.930	0.986
24	0.414	0.910	0.910	0.982
25	0.325	0.910	0.910	0.982
26	0.236	0.880	0.880	0.976
27	0.165	0.900	0.900	0.980
28	0.433	0.820	0.820	0.964
29	0.392	0.850	0.850	0.970
30	0.440	0.820	0.820	0.964
Rata-rata	0.439	0.756	0.756	0.951

*(Halaman ini sengaja dikosongkan)*

## BAB VI KESIMPULAN DAN SARAN

Bab ini berisikan kesimpulan yang dapat diambil dari hasil uji coba yang telah dilakukan. Selain kesimpulan, terdapat juga saran yang ditujukan untuk pengembangan perangkat lunak nantinya.

### 6.1 Kesimpulan

Kesimpulan yang didapatkan berdasarkan hasil uji coba pencarian gambar dengan metode Histogram of Oriented Gradients dan Hierarchical Centroid adalah sebagai berikut:

1. Implementasi *preprocessing* menggunakan metode Sobel menghasilkan hasil yang kurang bagus karena ada bagian tepi yang tidak terdeteksi sebagai tepi dan masih terdapat noise sehingga mengganggu dalam proses ekstraksi fitur. Hal tersebut terjadi khususnya pada gambar data *training*.
2. Implementasi metode kombinasi *Histogram of Oriented Gradient* dan *Hierarchical Centroid* dapat digunakan sebagai metode dalam pencarian gambar berbasis sketsa.
3. Kedalaman nilai centroid yang baik digunakan yang menghasilkan nilai ANMRR, presisi, recall dan akurasi yang terbaik ada pada kedalaman 3.
4. Nilai ANMRR pada kedalaman 3 sebesar 0,35. Sedangkan rata-rata nilai presisi dan recall sebesar 0,78. Serta nilai akurasi tertinggi sebesar 0,96
5. Metode kombinasi *Histogram of Oriented Gradients* dan *Hierarchical Centroid* menghasilkan nilai ANMRR, presisi, recall, dan akurasi lebih baik dari pada metode lainnya yang diuji seperti metode *Histogram of Oriented Gradients*, *Hierarchical Centroid*, *Edge Histogram Descriptor* dan *Region Properties*.

## 6.2 Saran

Saran yang diberikan terkait pengembangan pada Tugas Akhir ini adalah:

1. Menambah metode pada saat melakukan preprocessing agar hasil yang ditampilkan lebih baik. Bisa juga dilakukan segmentasi pada gambar.
2. Mengkombinasikan metode HOG dengan metode lain karena jika dilihat dari percobaan metode HC tidak terlalu bagus walaupun lebih dari metode lain yang diuji
3. Mengurangi fitur HOG yang tidak terlalu berpengaruh karena metode HOG menghasilkan fitur yang terlalu banyak sehingga fitur HOG tidak terlalu dominan dibanding fitur lain.
4. Dilakukan pembobotan pada fitur HOG dan HC agar fitur HOG tidak menjadi dominan.
5. Sketsa sebaiknya dapat langsung ditampilkan pada GUI supaya memudahkan pengguna dalam mengoperasikan sistem.



## DAFTAR PUSTAKA

- [1] J. M. Saavedra dan B. Bustos, "An improved histogram of edge local orientations for sketch-based image retrieval," *Pattern Recognition*, pp. 432-441, 2010.
- [2] J. P. Eakins, "Trademark image retrieval," dalam *S.W. Lew (Ed), Principle of visual Information Retrieval*, New York, Springer, 2001, pp. 319-354.
- [3] T. Kato, T. Kurita, N. Otsu dan K. Hirata, "A sketch retrieval method for full color image database-query by visual example," dalam *Proc. of the 11th IAPR International Conf on Computer Vision and Applications*, 1992.
- [4] A. K. Jain dan A. Vailaya, "A case study with trademark image databases," *Pattern recognition*, vol. 31, no. 9, pp. 1369-1390, 1998.
- [5] N. M. Asiri, N. AlHumaidi dan N. AIOsaim, "Combination of Histogram of Oriented Combination of Histogram of Oriented Sketch-Based Image Retrieval," 2015.
- [6] R. Hu dan J. Collomosse, "A Performance Evaluation of Gradient Field HOG Descripor for Sketch Based Image Retrieval," *Computer Vision and Image Understanding (CVIU)*, 2013.
- [7] D. Harper, "Online Etymology Dictionary," [Online]. Available: <http://www.etymonline.com/index.php?search=s sketch&searchmode=none>. [Diakses 31 May 2016].
- [8] A. Miguelena Bada, G. de Jesus Hoyos Rivera dan A. Marin Hernandez, "Garabato: A proposal of a sketch-based Image Retrieval system for the Web," dalam *Electronics, Communications and Computers (CONIELECOMP), International Conference*, 2014.

- [9] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Hiang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele dan P. Yanker, "Query by image and video content: the QBIC system," *IEEE Computer*, vol. 28, pp. 23-32, 2002.
- [10] J. R. Smith dan S. F. Chang, "VisualSEEK: a fully automated content based image query system," *ACM Multimedia*, pp. 97-98, 1996.
- [11] B. Szanto, P. Pozsegovics, Z. Vamossy dan S. Z. Sergyan, "Sketch4Match - Content-based Image Retrieval System Using Sketches," dalam *9th IEEE Symposium on Applied Machine Intelligence and Informatics*, Smoleric, Slovakia, 2011.
- [12] D. Ziou dan S. Tabbone, "Edge detection techniques: An overview," *International Journal of Pattern Recognition and Image Analysis*, vol. 8, no. 4, p. 537-559, 1998.
- [13] P. A. Setiyono, "Menganalisa Perbandingan Deteksi Tepi Antara Metode Sobel dan Metode Robet," 2009.
- [14] N. Dalal dan B. Triggs, "Histograms of Oriented Gradients for Human Detection," dalam *one-Alps, 655 avenue de l'Europe*, Montbonnot 38334, France, 2005.
- [15] N. M. Asiri, N. AL-Humaidi dan N. AlOsaim, "Combination of Histogram of Oriented Gradients and Hierarchical Centroid for Sketch-Based Image Retrieval," *IEEE*, pp. 149-152, 2015.
- [16] E. Deza dan M. M. Deza, "Encyclopedia of Distances," *Springer*, p. 94, 2009.
- [17] MPEG-7, "Subjective Evaluation of the MPEG-7 Retrieval Accuracy Measure (ANMRR)," 2000.
- [18] J. R. Taylor, "An Introduction to Error Analysis: The Study of Uncertainties in Physical Measurements," dalam *University Science Books*, 1999, pp. 128-129.
- [19] R. Baeza-Yates dan B. Ribeiro-Neto, "Modern Information Retrieval," dalam *NY: ACM Press*, New York, 1999.

## **BIODATA PENULIS**



Atika Faradina Randa lahir di Jember 19 Juli 1994. Penulis menempuh pendidikan formal dimulai dari SD Al-Furqan Jember (2000-2006), SMP Negeri 2 Jember (2006-2009), SMA Negeri 1 Jember (2009-2012) dan S1 Teknik Informatika ITS (2012-2016). Bidang studi yang diambil oleh penulis pada saat berkuliah di Teknik Informatika ITS adalah Komputasi Cerdas dan Visi (KCV). Penulis aktif dalam organisasi seperti Himpunan Mahasiswa Teknik Computer-Informatika (2013-2014) dan Keluarga Muslim Informatika (2013-2014). Penulis juga aktif dalam berbagai kegiatan kepanitiaan yaitu SCHEMATICS 2013 divisi kesekretariatan dan SCHEMATICS 2014 divisi NST. Penulis dapat dihubungi melalui email: [atikafaradina@gmail.com](mailto:atikafaradina@gmail.com)

*(Halaman ini sengaja dikosongkan)*