



**ITS**  
Institut  
Teknologi  
Sepuluh Nopember

**TUGAS AKHIR - KI141502**

## **TEMU KEMBALI CITRA MAKANAN MENGGUNAKAN REPRESENTASI MULTI TEXTON HISTOGRAM**

**IMAGINE CLARA ARABELLA**  
**NRP 5113100004**

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

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

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





**TUGAS AKHIR - KI141502**

## **TEMU KEMBALI CITRA MAKANAN MENGGUNAKAN REPRESENTASI MULTI TEXTON HISTOGRAM**

**IMAGINE CLARA ARABELLA  
NRP 5113100004**

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

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

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

***[Halaman ini sengaja dikosongkan]***



**FINAL PROJECT - KI141502**

# **FOOD IMAGE RETRIEVAL USING MULTI TEXTON HISTOGRAM REPRESENTATION**

**IMAGINE CLARA ARABELLA  
NRP 5113100004**

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

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

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

***[Halaman ini sengaja dikosongkan]***

## LEMBAR PENGESAHAN

### TEMU KEMBALI CITRA MAKANAN MENGGUNAKAN REPRESENTASI MULTI TEXTON HISTOGRAM

#### TUGAS AKHIR

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

Oleh:

**IMAGINE CLARA ARABELLA**  
**NRP : 5113 100 004**

Disetujui oleh Dosen Pembimbing Tugas Akhir:

Dr. Eng. Nanik Suciati, S.Kom., M.Kom  
NIP. 197208091995121001

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



**SURABAYA JULI, 2017**

***[Halaman ini sengaja dikosongkan]***



## **TEMU KEMBALI CITRA MAKANAN MENGGUNAKAN REPRESENTASI MULTI TEXTON HISTOGRAM**

Nama Mahasiswa : Imagine Clara Arabella  
NRP : 5113 100 004  
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***

*Temu kembali citra merupakan suatu metode pencarian citra dengan melakukan perbandingan antara citra query dengan citra yang terdapat dalam database berdasarkan informasi yang ada. Temu kembali citra untuk pengenalan makanan sangat dibutuhkan untuk pasien diet. Dalam tugas akhir ini diusulkan suatu metode temu kembali citra makanan berdasarkan input berupa citra makanan yang dibandingkan dengan database citra makanan yang ada.*

*Tugas akhir ini secara khusus membahas mengenai perancangan sebuah sistem temu kembali citra makanan dengan representasi Multi-texton Histogram (MTH). Proses pertama dilakukan deteksi orientasi tekstur menggunakan metode Sobel Edge Detection. Setelah itu dilakukan kuantisasi warna pada ruang warna RGB. Serta deteksi Texton untuk tahap ekstraksi fitur. Untuk mendapatkan kemiripan citra, dihitung jarak antar citra dengan menggunakan distance metric. Setelah didapatkan jarak antar citra, diurutkan dari yang terdekat sampai yang terjauh jarak citranya untuk menentukan temu kembali citra. Hasil yang didapat adalah berupa ditemukannya citra-citra yang mirip dengan citra query.*

*Berdasarkan uji coba yang dilakukan pada citra, hasil pencarian citra mirip dengan rata-rata nilai precision terbaik sebesar 40,50% dan recall terbaik sebesar 8,61% pada 18 level orientasi dan 64 level kuantisasi warna.*

***Kata kunci: Citra Makanan, Edge Detection, Multi Texton Histogram, Temu Kembali Citra.***

## **FOOD IMAGE RETRIEVAL USING MULTI TEXTON HISTOGRAM REPRESENTATION**

Student Name : Imagine Clara Arabella  
Registration Number : 5113 100 004  
Department : Teknik Informatika, FTIf ITS  
First Supervisor : Dr. Eng. Nanik Suciati, S.Kom., M.Kom.  
Second Supervisor : Dini Adni Navastara, S.Kom., M.Sc.

### ***Abstract***

*Image retrieval is an image search method by performing a comparison between the query image and the image contained in the database based on the existing information. Image retrieval for feeding recognition is essential for dietary patients. In this final project proposed a method of retrieval of food image based on input in the form of food image compared with existing food image database.*

*This final project specifically discusses the design of a food image retrieval system with a Multi-texton Histogram (MTH) representation. The first step is detection of texture orientation using Sobel Edge Detection method. After that is done the color quantization on RGB color space. As well as Texton Detection for its feature extraction stage. To get the image resemblance, calculated the distance between the image using distance metric. Having obtained the distance between images, sorted from the nearest to the farthest distance of its image to determine image retrieval. The results obtained are in the form of finding images similar to the image of the query.*

*Based on the experiments performed on the image, the best precision of image retrieval were obtained with average precision 40,5% and recall 8,61% at 18 level orientation and 64 level quantization.*

***Keywords: Edge Detection, Food Image, Multi Texton Histogram, Image Retrieval.***

## KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Segala puji bagi Allah SWT yang telah melimpahkan rahmat dan anugerah-Nya sehingga penulis dapat menyelesaikan tugas akhir yang berjudul **“TEMU KEMBALI CITRA MAKANAN MENGGUNAKAN REPRESENTASI MULTI-TEXTON HISTOGRAM”**.

Dengan pengerjaan Tugas Akhir ini, penulis bisa belajar lebih banyak untuk memperdalam dan meningkatkan apa yang telah didapatkan penulis selama menempuh perkuliahan di Teknik Informatika ITS. Dengan Tugas Akhir ini penulis juga dapat menghasilkan suatu implementasi dari apa yang telah penulis pelajari.

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

1. Ibu Dr. Eng. Nanik Suciati, S.Kom., M.Kom. dan Ibu Dini Adni Navastara, S.Kom., M.Sc. selaku dosen pembimbing penulis yang telah memberi ide, nasihat dan arahan sehingga penulis dapat menyelesaikan tugas akhir dengan tepat waktu.
2. Orang tua penulis Bapak Wasis Eko Utomo dan Ibu Antiq Wijayanti yang telah memberikan dukungan moral, spiritual dan material serta senantiasa memberikan doa demi kelancaran dan kemudahan penulis dalam mengerjakan tugas akhir.
3. Seluruh saudara kandung: dua adik (Imagine Aulia Rachma Diffa dan Imagine Aqeela Ar-rabbani) dan seluruh keluarga besar yang telah memberikan dukungan yang besar baik secara langsung maupun secara implisit.

4. Teman-teman di Lab LP: para admin yang telah banyak membantu memfasilitasi dalam pengerjaan dan telah menemani penulis.
5. Teman satu bimbingan Bu Dini dan teman satu bimbingan Bu Nanik yang sering bimbingan Bersama..
6. Sani, Razi, Afiif, Dewangga, Ekky, Yohana, Lusi, Dhea, Dian, Fany, Demsey, Luffi, Ghulam serta teman-teman angkatan 2013 yang telah membantu, berbagi ilmu, menjaga kebersamaan, dan memberi motivasi kepada penulis.
7. Teman-teman HMTC 2014/2015, khususnya PSDM yang telah memberi banyak kisah dan pengalaman berharga untuk penulis.
8. Teman-teman HMTC 2015/2015, khususnya PH Optimasi yang telah memberikan kisah lagi dan mengajarkan banyak hal kepada penulis.
9. Semua pihak yang telah turut membantu penulis dalam menyelesaikan Tugas Akhir ini.

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

Surabaya, Juli 2017

Penulis

## DAFTAR ISI

<b>LEMBAR PENGESAHAN</b> .....	<b>v</b>
<i>Abstrak</i> .....	<b>vii</b>
<i>Abstract</i> .....	<b>ix</b>
<b>KATA PENGANTAR</b> .....	<b>xi</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 Batasan Masalah .....	3
1.4 Tujuan Tugas Akhir .....	3
1.5 Manfaat Tugas Akhir .....	4
1.6 Metodologi .....	4
1.7 Sistematika Laporan.....	5
<b>BAB II DASAR TEORI</b> .....	<b>7</b>
2.1 Metode Temu Kembali Citra .....	7
2.2 Pengolahan Citra Digital .....	7
2.3 Content Based Image Retrieval.....	9
2.4 Ruang Warna ( <i>Color Space</i> ).....	10
2.5 Histogram Warna .....	10
2.6 Pendeteksian Tepi .....	11
2.7 <i>Texton</i> .....	13
2.8 <i>Distance Metric</i> .....	14
2.9 <i>Recall</i> dan <i>Precision</i> .....	15
2.10 Dataset UNICT FD-1200 .....	16
<b>BAB III ANALISIS DAN PERANCANGAN</b> .....	<b>19</b>
3.1 Lingkungan Desain dan Implementasi.....	19
3.2 Perancangan Data.....	19
3.2.1 Data Masukan.....	20
3.2.2 Data Pemrosesan .....	21
3.2.3 Data Keluaran.....	24

3.3	Perancangan Proses .....	24
3.3.1	Proses <i>Texture Orientation Detection</i> .....	26
3.3.2	Proses Kuantisasi Warna diruang warna RGB .....	27
3.3.3	Proses Deteksi <i>Texton</i> .....	28
3.3.4	Proses <i>Multi-texton Histogram</i> .....	29
3.3.5	Proses Perhitungan Kemiripan antar Citra .....	30
<b>BAB IV</b>	<b>IMPLEMENTASI.....</b>	<b>31</b>
4.1	Lingkungan Implementasi.....	31
4.2	Implementasi Program .....	31
4.2.1	Implementasi <i>Texture Orientation Detection</i> .....	31
4.2.2	Implementasi Kuantisasi Warna.....	33
4.2.3	Implementasi Deteksi <i>Texton</i> .....	34
4.2.4	Implementasi Fitur <i>Multi-texton Histogram</i> .....	35
4.2.5	Implementasi Perhitungan Kemiripan Citra.....	36
4.2.6	Implementasi Perhitungan Kemiripan Citra.....	37
<b>BAB V</b>	<b>UJI COBA DAN EVALUASI.....</b>	<b>39</b>
5.1	Lingkungan Uji Coba.....	39
5.2	Data Uji Coba.....	39
5.3	Hasil <i>Texture Orientation Detection</i> .....	39
5.4	Hasil Kuantisasi Warna pada Ruang RGB.....	40
5.5	Hasil <i>Texton Detection</i> .....	42
5.6	Hasil <i>Multi-texton Histogram</i> .....	43
5.7	Skenario Uji Coba.....	45
5.7.1	Uji Coba Berdasarkan Level Orientasi.....	45
5.7.2	Uji Coba Berdasarkan Level Kuantisasi Warna.....	45
5.7.3	Uji Coba Berdasarkan Jumlah Kategori.....	46
5.7.4	Uji Coba Berdasarkan Jumlah Citra Ditemukan .....	49
5.8	Evaluasi.....	49
5.8.1	Evaluasi Uji Coba Berdasarkan Level Orientasi .....	50
5.8.2	Evaluasi Uji Coba Berdasarkan Level Kuantisasi.....	50
5.8.3	Evaluasi Uji Coba Berdasarkan Jumlah Kategori .....	50
5.8.4	Evaluasi Uji Coba Berdasarkan Jumlah Citra yang Ditemukan .....	51
<b>BAB VI</b>	<b>KESIMPULAN DAN SARAN.....</b>	<b>53</b>
6.1	Kesimpulan .....	53



6.2 Saran .....	53
<b>DAFTAR PUSTAKA .....</b>	<b>55</b>
<b>LAMPIRAN.....</b>	<b>57</b>
<b>BIODATA PENULIS.....</b>	<b>69</b>

***[Halaman ini sengaja dikosongkan]***

## DAFTAR GAMBAR

<b>Gambar 2.1</b>	<b>Matrix Konvolusi Pendeteksi Tepi Sobel</b> .....	12
<b>Gambar 2.2</b>	<b>4 Tipe <i>Texton</i> didefinisikan dalam MTH</b> .....	13
<b>Gambar 2.3</b>	<b>Skema Pencarian <i>Texton</i> pada Citra</b> .....	14
<b>Gambar 2.4</b>	<b>Dataset Makanan UNICT-FD1200</b> .....	17
<b>Gambar 3.1</b>	<b>Diagram Alir Sistem</b> .....	25
<b>Gambar 3.2</b>	<b>Diagram Alir Proses Sobel <i>Edge Detection</i></b> .....	26
<b>Gambar 3.3</b>	<b>Diagram Alir Proses Kuantisasi Warna</b> .....	27
<b>Gambar 3.4</b>	<b>Diagram Alir Proses Deteksi <i>Texton</i></b> .....	28
<b>Gambar 3.5</b>	<b>Diagram Alir Proses <i>Multi-texton Histogram</i></b> ....	29
<b>Gambar 3.6</b>	<b>Diagram Alir Proses Perhitungan Kemiripan antar Citra</b> .....	30
<b>Gambar 5.1</b>	<b>Citra yang ditemukan berdasarkan 2 kategori</b> .	48
<b>Gambar 5.2</b>	<b>Citra yang ditemukan berdasarkan 8 kategori</b> .	49

***[Halaman ini sengaja dikosongkan]***

## DAFTAR TABEL

<b>Tabel 2.1</b> <i>Confusion Matrix Precision dan Recall</i> .....	15
<b>Tabel 3.1</b> Lingkungan Perancangan Perangkat Lunak .....	19
<b>Tabel 3.2</b> Citra Makanan.....	20
<b>Tabel 3.3</b> Nama File.....	22
<b>Tabel 3.4</b> Data Variabel .....	22
<b>Tabel 5.1</b> Hasil <i>Texture Orientation Detection</i> .....	40
<b>Tabel 5.2</b> Hasil Kuantisasi Warna .....	41
<b>Tabel 5.3</b> Hasil <i>Texton Detection</i> .....	42
<b>Tabel 5.4</b> Hasil <i>Multi-texton Histogram</i> .....	43
<b>Tabel 5.5</b> Hasil Uji Coba Level Orientasi.....	45
<b>Tabel 5.6</b> Hasil <i>Precision</i> Uji Coba Level Kuantisasi .....	46
<b>Tabel 5.7</b> Hasil <i>Recall</i> Uji Coba Level Kuantisasi .....	46
<b>Tabel 5.8</b> Hasil Uji Coba Jumlah Kategori .....	47
<b>Tabel 5.9</b> Hasil Uji Coba Jumlah Citra yang Ditemukan.....	49

***[Halaman ini sengaja dikosongkan]***

## DAFTAR KODE SUMBER

<b>Kode Sumber 4.1 Implementasi Sobel <i>Edge Detection</i> .....</b>	<b>32</b>
<b>Kode Sumber 4.2 Implementasi Kuantisasi Warna .....</b>	<b>33</b>
<b>Kode Sumber 4.3 Implementasi Deteksi Texton .....</b>	<b>34</b>
<b>Kode Sumber 4.4 Implementasi <i>Multi-texton Histogram</i> .....</b>	<b>35</b>
<b>Kode Sumber 4.5 Implementasi Perhitungan Kemiripan Citra .....</b>	<b>36</b>
<b>Kode Sumber 4.6 Implementasi Sorting .....</b>	<b>37</b>
<b>Kode Sumber 4.7 Implementasi Precision dan Recall .....</b>	<b>38</b>

***[Halaman ini sengaja dikosongkan]***



# **BAB I**

## **PENDAHULUAN**

Pada bab ini dijelaskan beberapa hal dasar yang meliputi latar belakang, tujuan, permasalahan, batasan masalah, metodologi serta sistematika penulisan tugas akhir, sehingga gambaran umum permasalahan dan pemecahan yang diambil dapat dipahami dengan baik.

### **1.1 Latar Belakang**

Informasi merupakan sebuah data yang telah diolah sehingga memiliki arti dan berupa fakta atau nilai yang bermanfaat. Berbagai macam informasi yang ada saat ini antara lain berupa dokumen, berita, gambar, video dan lain sebagainya.

Teknologi informasi adalah suatu teknologi yang digunakan untuk mengolah informasi menjadi lebih berkualitas. Informasi yang berkualitas yaitu informasi yang relevan, akurat dan dapat digunakan untuk pengambilan keputusan. Dengan semakin berkembangnya teknologi informasi pada masa kini, informasi pun menjadi semakin bervariasi dan dapat semakin mudah didapatkan. Seiring dengan berkembangnya informasi, permasalahan utama yang ada adalah cara pemilihan informasi yang berguna secara selektif dan relevan. Pemilihan informasi tidak mungkin dilakukan secara manual karena kumpulan informasi tersebut sangat besar. Dengan banyaknya informasi yang ada sekarang, dibutuhkan adanya pengolahan informasi yang baik. Salah satu metode yang dapat digunakan adalah sistem temu kembali informasi. Pada sistem yang dibuat ini, akan dibatasi dengan pencarian informasi hanya berupa citra. Salah satu teknik untuk melakukan pengolahan pencarian atau menemukan gambar sudah berkembang berbagai macam teknik. Terdapat pencarian yang menggunakan keyword atau teks, terdapat pula pencarian yang menggunakan *content based image retrieval*.

Beberapa teknik temu kembali informasi berbasis isi sudah banyak dikembangkan. Website yang mendukung sistem temu kembali citra salah satunya adalah *TineEye Reverse Image Search* yang menggunakan *image matching engine* yang memungkinkan untuk melakukan perbandingan citra berskala besar untuk berbagai kebutuhan [1]. Kemudian terdapat *Google Similar image* yang merupakan *Advanced Google Search Engine*, *Google* juga dapat melakukan pencarian dengan sebuah citra sebagai query.

Dewasa ini banyak hal dilakukan untuk mendapatkan tubuh yang ideal. Untuk memperoleh tubuh ideal, banyak orang melakukan diet ketat tanpa didasari pengetahuan kesehatan yang cukup baik. Kita ketahui bahwa diet yang tidak sehat dapat menyebabkan masalah kesehatan seperti diabetes, obesitas, serta dapat menimbulkan resiko bagi orang-orang yang memiliki alergi makanan. Oleh karena itu, diperlukan penilaian diet tentang makanan yang dikonsumsi untuk diet. Biasanya penilaian dilakukan sendiri yang menyebabkan hasil yang kurang akurat. Teknologi yang ada sekarang terutama di bidang *automatic images recognition* memungkinkan untuk dibangun sebuah sistem temu kembali dan klasifikasi citra makanan atau *food image retrieval and classification* untuk mengganti sistem penilaian diet yang konvensional yang menggunakan metode *anti textons* [2]. Data citra yang digunakan adalah citra makanan dari UNICT-FD1200 [3].

Pada tugas akhir ini dilakukan implementasi temu kembali citra makanan menggunakan metode *multi texton histogram* [4]. Dalam proses pengembangan sistem ini diperlukan beberapa tahapan yaitu mendapatkan *texture orientation Detection* dengan *sobel edge Detection*, mendapatkan kuantisasi warna pada ruang RGB, deteksi *texton*, menghitung representasi *multi-texton*, dan perhitungan kemiripan antar citra, setelah didapatkan kemiripan antar citra, maka dihitung akurasi menggunakan *precision* dan *recall*. Data yang digunakan adalah data citra makanan dari UNICT-FD1200. Diharapkan nilai akurasi dari sistem ini lebih

baik daripada penelitian yang sudah dilakukan sehingga bisa memberi hasil yang tepat.

## 1.2 Rumusan Masalah

Rumusan masalah yang diangkat dalam Tugas Akhir ini adalah sebagai berikut:

1. Bagaimana melakukan ekstraksi fitur data citra makanan dengan menggunakan *Multi-texton Histogram* ?
2. Bagaimana menghitung kemiripan citra menggunakan *distance metric* ?
3. Bagaimana mengevaluasi sistem temu kembali citra yang telah dibangun ?

## 1.3 Batasan Masalah

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

1. Dataset yang digunakan adalah dataset UNICT-FD1200 yang diambil dari *University of Catania*. Dataset berupa 4754 gambar makanan dengan 8 kelompok makanan.
2. Dataset citra makanan menggunakan format JPG dengan ukuran 320x240 piksel.
3. Metode diimplementasikan menggunakan MATLAB.

## 1.4 Tujuan Tugas Akhir

Tujuan tugas akhir ini adalah untuk mengimplemetasikan temu kembali citra makanan menggunakan metode *Multi-texton Histogram* dan perhitungan akurasi dengan algoritma *presicion* dan *recall*.

## 1.5 Manfaat Tugas Akhir

Dengan dibuatnya Tugas Akhir ini diharapkan bisa memudahkan pasien diet melakukan penilaian diet menggunakan citra makanan agar dapat menemukan citra makanan yang sesuai dengan menerapkan temu kembali citra makanan agar hasilnya lebih akurat.

## 1.6 Metodologi

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

### 1. Studi Literatur

Pada tahap ini dilakukan pencarian dan pemahaman informasi dan literatur baik melalui media internet maupun dengan melakukan studi pustaka pada jurnal-jurnal ilmiah cetak yang diperlukan untuk tahap implementasi program. Untuk mendukung implementasi program, didapatkan *database* citra makanan yang terdiri dari 4754 citra diperoleh dari UNICT-FD1200. Tahap ini diperlukan untuk membantu memahami penggunaan metode-metode terkait dengan sistem yang akan dibangun, antara lain : *Multi-texton Histogram*, *presicion* dan *recall*.

### 2. Analisis dan Desain Perangkat Lunak

Tahap ini meliputi perancangan sistem berdasarkan studi literatur dan pembelajaran konsep teknologi dari perangkat lunak yang ada. Tahap ini mendefinisikan alur dari implementasi. Langkah-langkah yang dikerjakan juga didefinisikan pada tahap ini. Pada tahapan ini dibuat *prototype* sistem, yang merupakan rancangan dasar dari sistem yang akan dibuat. Serta dilakukan perancangan antarmuka agar pengguna lebih mudah dalam menjalankan perangkat lunak. Sebagai input sistem adalah berupa citra makanan sebagai citra *query*. Untuk keluaran dihasilkan sederatan citra yang memiliki jarak

terhadap citra *query*. Semakin kecil jaraknya berarti citra semakin mirip.

### 3. Implementasi Perangkat Lunak

Sistem temu kembali citra makanan akan dibuat dengan bahasa pemrograman MATLAB menggunakan kaskas bantu IDE MATLAB (R2013a) pada platform *desktop*. Kaskas bantu pendukung lain diantaranya *Microsoft Paint* untuk dokumentasi, dan *Microsoft Excel* sebagai pengolah angka.

### 4. Uji Coba dan Evaluasi

Pada tahap ini dilakukan uji coba pada data yang telah dikumpulkan. Tahap ini digunakan untuk mengevaluasi kinerja program serta mencari masalah yang mungkin timbul saat program dievaluasi serta melakukan perbaikan jika terdapat kesalahan pada program.

## 1.7 Sistematika Laporan

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

### **Bab I**    **Pendahuluan**

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

### **Bab II**    **Dasar Teori**

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

### **Bab III**    **Analisis dan Perancangan**

Bab ini berisi tentang analisis dan perancangan desain sistem temu kembali citra makanan menggunakan representasi *Multi-texton Histogram*.

**Bab IV Implementasi**

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

**Bab V Uji Coba dan Evaluasi**

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

**Bab VI Kesimpulan dan Saran**

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

## **BAB II**

### **DASAR TEORI**

Bab ini berisi penjelasan teori-teori yang berkaitan dengan algoritma yang diajukan pada pengimplementasian program. Penjelasan ini bertujuan untuk memberikan gambaran secara umum terhadap program yang dibuat.

#### **2.1 Metode Temu Kembali Citra**

Citra dapat digunakan sebagai sumber informasi. Untuk memanfaatkan sumber informasi, maka diperlukan *database* citra. *Database* citra berfungsi untuk menampung informasi tentang citra. Dengan ketersediaan citra dalam jumlah besar, maka diperlukan metode pencarian yang berdaya guna untuk menemukan kembali citra *query* dari *database*.

Beberapa metode pencarian untuk menemukan kembali citra telah dikembangkan. Secara umum, metode temu kembali citra (*image retrieval method*) dapat diklasifikasikan menjadi dua kategori, yakni :

- a. Metode berbasis konsep (*concept based method*)  
Metode berbasis konsep menggambarkan isi citra melalui teks dan *field* terstruktur. Deskriptor yang digunakan adalah himpunan kata kunci dari fitur citra yang menjelaskan warna, bentuk dan lain-lain.
- b. Metode berbasis konten (*content based method*)  
Metode berbasis konten menggunakan fitur yang secara otomatis diekstraksi dari dalam citra seperti tekstur, warna, bentuk dan kendala spasial untuk menjelaskan citra (Li et al, 2005).

#### **2.2 Pengolahan Citra Digital**

Pengolahan citra merupakan proses pengolahan dan analisis citra yang banyak melibatkan persepsi visual. Proses ini

mempunyai ciri data masukan dan informasi keluaran yang berbentuk citra. Istilah pengolahan citra digital secara umum didefinisikan sebagai pemrosesan citra dua dimensi dengan komputer. Dalam definisi yang lebih luas, pengolahan citra digital juga mencakup semua data dua dimensi. Citra digital adalah barisan bilangan nyata maupun kompleks yang diwakili oleh bit-bit tertentu.

Pengolahan citra digital adalah bidang yang berkembang sangat pesat sejalan dengan kemajuan teknologi pada industri saat ini. Fungsi utama dari pengolahan citra digital adalah untuk memperbaiki kualitas dari citra sehingga citra dapat dilihat lebih jelas tanpa ada ketegangan pada mata, karena informasi penting diekstrak dari gambar yang dihasilkan harus jelas sehingga didapatkan hasil yang terbaik. Selain itu pengolahan citra digital digunakan untuk memproses data yang diperoleh dalam persepsi mesin, yaitu prosedur-prosedur yang digunakan untuk mengekstraksi informasi dari citra dalam bentuk yang cocok untuk proses komputer.

Proses pengolahan citra digital dengan menggunakan komputer digital terlebih dahulu mentransformasikan citra ke dalam bentuk besar-besaran diskrit dari nilai tingkat keabuan pada titik-titik elemen citra. Bentuk gambar ini disebut citra digital. Elemen-elemen citra digital apabila ditampilkan dalam layar monitor akan menempati sebuah ruang yang disebut dengan piksel (*picture elemen*). Teknik dan proses untuk mengurangi atau menghilangkan efek degradasi pada citra digital meliputi perbaikan citra (*image enhancement*), restorasi citra (*image restoration*), dan transformasi spasial (*spatial transformation*). Subyek lain dari pengolahan citra digital diantaranya adalah pengkodean citra (*image coding*), segmentasi citra (*image segmentation*), representasi dan deskripsi citra (*image representation and description*) [5].



## 2.3 Content Based Image Retrieval

*Content Based Image Retrieval* (CBIR) merupakan suatu teknik pencarian kembali citra yang mempunyai kemiripan karakteristik atau konten dari sebuah sekumpulan citra. Proses secara umum dari CBIR adalah citra yang menjadi *query* dilakukan proses ekstraksi fitur, begitu halnya dengan citra yang ada pada sekumpulan citra juga dilakukan proses seperti pada citra *query*. Parameter fitur citra yang dapat digunakan untuk *retrieval* pada sistem ini seperti histogram, susunan warna, tekstur, bentuk, tipe spesifik dari obyek, tipe event tertentu, nama individu, lokasi, emosi [6].

Citra *query* yang digunakan mempunyai level, yaitu :

- Level 1 : *retrieval* dengan *primitive feature*, seperti warna, bentuk, tekstur.
- Level 2 : *retrieval* dengan *derived feature*, seperti tipe obyek, individu obyek atau orang.
- Level 3 : *retrieval* dengan *abstract feature*, seperti nama event, tipe aktifitas, emotional, religius.

Penelitian dan pembangunan dalam CBIR mencakup keseluruhan topik. Pokok persoalan yang terjadi dalam pembangunan CBIR dapat diuraikan menjadi beberapa poin penting. Yaitu pemahaman citra yang dibutuhkan oleh user dan pencarian informasi. Pengidentifikasian cara yang sesuai dalam pencitraan content atau karakteristik dari citra. Proses ekstraksi fitur dari citra.

Ekstraksi fitur merupakan proses penting pada sistem CBIR. Karena hasil dari proses ini, perbedaan pada setiap citra dapat diketahui berdasarkan cirinya seperti ciri bentuk, ciri warna, ciri tekstur dan lain-lain.

Beberapa teknik yang dapat digunakan untuk ekstraksi fitur antara lain : teknik analisis komponen utama, besaran statistik, histogram warna, *wavelet transform* dan lain-lain.

Penyediaan tempat penyimpanan untuk sekumpulan citra juga sangat penting. Proses matching antara citra *query* dan citra yang ada pada sekumpulan citra. Proses juga merupakan proses pencocokan citra untuk memperoleh citra yang mempunyai kemiripan dengan citra *query*. Proses ini dilakukan dengan menghitung jarak antara dua citra yaitu citra *query* dan citra target pada sekumpulan citra. Parameter yang digunakan dalam perhitungan jarak berdasarkan pada hasil ekstraksi ciri [7].

## 2.4 Ruang Warna (*Color Space*)

Ruang warna adalah suatu organisasi dari warna tertentu. Berbagai warna dapat diciptakan oleh pigmen warna primer dan warna tersebutlah yang akan menentukan ruang warna tertentu. Ruang warna atau dikenal sebagai model warna adalah model matematika abstrak yang menggambarkan bagaimana warna dapat diwakili sebagai bilangan tupel [6]. Pada dasarnya, ruang warna merupakan penjabaran dari sistem koordinat dan sub-ruang. Setiap warna dalam sistem diwakili oleh satu titik. Ruang warna dapat hal ini memungkinkan untuk membuat suatu ruang warna secara bebas

Ruang warna dapat untuk dibuat secara bebas dengan aturan yang bebas pula. Namun ruang warna yang dibuat secara bebas akan sulit untuk dipahami oleh orang lain secara global. Maka dibutuhkan ruang warna yang disepakati oleh orang-orang secara global. Ada berbagai macam jenis ruang warna yang digunakan secara global, seperti ruang warna RGB, HSV, CMYK,  $L^*a^*b$  dan lain-lain. Diantara semua ruang warna, ruang warna RGB adalah ruang warna yang paling umum digunakan oleh orang-orang. Pada Tugas Akhir ini akan menggunakan RGB sebagai ruang warna dalam pemetaan warna dan proses-proses lainnya.

## 2.5 Histogram Warna

Citra terdiri dari piksel-piksel warna. Dalam pengolahan citra, intensitas citra dapat direpresentasikan ke dalam beberapa

jenis ruang warna seperti RGB, HSV,  $L^*a^*b$ , dsb. Tugas akhir ini menggunakan pembagian warna RGB dalam pemrosesan ekstraksi fitur. RGB adalah pembagian warna citra ke dalam 3 ruang warna yaitu merah, hijau dan biru. Setiap ruang warna RGB memiliki 256 nilai intensitas dengan rentang nilai dari 0-255

Secara umum histogram menyatakan frekuensi kemunculan atau peluang keberadaan parameter dalam domain. Histogram warna menyatakan frekuensi kemunculan atau peluang setiap warna *pixel* di dalam sebuah citra [8]. Untuk mengurangi waktu komputasi dan menghemat tempat penyimpanan, histogram warna menggunakan kuantisasi warna. Selain itu, kuantisasi warna juga dapat mengeliminasi komponen warna yang dapat dianggap sebagai *noise*. Banyaknya komponen kuantisasi (*bin*) dapat ditetapkan sesuai dengan kebutuhan pembuatan histogram. Operasi ambang batas sangat membantu dalam penghitungan frekuensi masing-masing *bin*.

## 2.6 Pendeteksian Tepi

Tepi (*edge*) adalah perubahan nilai intensitas derajat keabuan yang mendadak (besar) dalam jarak yang singkat. Perbedaan intensitas inilah yang menampakkan rincian atau batas objek pada citra. Tepi biasanya terdapat pada batas antara dua daerah berbeda pada suatu citra. Tepi dapat diorientasikan dengan suatu arah dan arah ini berbeda-beda bergantung pada perubahan intensitas.

Pendeteksian tepi (*edge Detection*) adalah operasi yang dijalankan untuk mendeteksi garis tepi (*edge*), yakni garis yang membatasi dua wilayah citra homogen berdasarkan tingkat kecerahan yang berbeda. Pendeteksian tepi merupakan salah satu langkah untuk meliputi informasi di dalam citra. Tepi menampilkan batas-batas objek, oleh karena itu tepi berguna dalam proses segmentasi dan identifikasi objek di dalam citra.

Tujuan operasi pendeteksian tepi adalah untuk memperjelas garis batas suatu objek dari latar-belakang di dalam citra. Karena tepi termasuk ke dalam komponen berfrekuensi tinggi, maka pendeteksian tepi dapat dilakukan dengan penapis lolos tinggi (*high pass filter*). Beberapa operator pendeteksi tepi yang umum digunakan, antara lain *Sobel*, *Prewitt*, *Roberts*, *Laplacian of a Gaussian (LoG)*, *Zero Crossings* dan *Canny* (Gonzalez *et al*, 2004).

Z1	Z2	Z3
Z4	Z5	Z6
Z7	Z8	Z9

a

-1	0	1
-2	0	2
-1	0	1

b

-1	-2	-1
0	0	0
1	2	1

c

**Gambar 2.1 Matrix Konvolusi Pendeteksi Tepi Sobel**

- (a) Matrix citra tetangga (b) Matrix konvolusi arah sumbu x  
(c) Matrix konvolusi arah sumbu y

Pendeteksi tepi *Sobel* (*Sobel edge detector*) merupakan salah satu metode pendeteksi tepi yang umum digunakan [9]. Pendeteksi tepi *Sobel* menggunakan dua buah matriks konvolusi berukuran  $3 \times 3$ . Matriks konvolusi pada *pixel-pixel* tetangga berukuran  $3 \times 3$ , yang diperlihatkan pada Gambar 2.1.a. Matriks konvolusi pertama digunakan untuk mengestimasi *gradient* pada arah sumbu x, diperlihatkan pada Gambar 2.1.b. Matriks konvolusi kedua digunakan untuk menentukan *gradient* pada arah sumbu y, diperlihatkan pada Gambar 2.1.c (Gonzalez *et al*, 2004).

Hasil operasi konvolusi dalam arah sumbu x terhadap citra dinyatakan dengan  $G_x$ , dan hasil operasi konvolusi dalam arah sumbu y terhadap citra dinyatakan dengan  $G_y$ . Sehingga dengan menggunakan matriks konvolusi maka nilai  $G_x$  dan  $G_y$  berturut turut dapat dihitung dengan Persamaan 2.3 dan 2.4.

$$G_x = (Z_7 + 2Z_8 + Z_9) - (Z_1 + 2Z_2 + Z_3) \quad (2.3)$$

$$G_y = (Z_3 + 2Z_6 + Z_9) - (Z_1 + 2Z_4 + Z_3) \quad (2.4)$$

Magnitudo ( $G$ ) dari *gradien* dapat dihitung dengan Persamaan 2.5 atau 2.6.

$$|G| = |G_x| + |G_y| \quad (2.5)$$

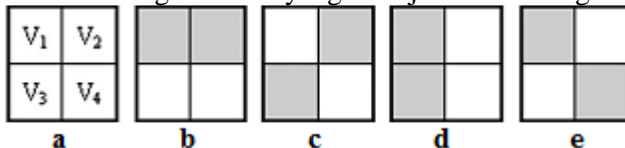
$$G = \sqrt{G_x^2 + G_y^2} \quad (2.6)$$

Sebuah *pixel* akan dianggap sebagai tepi (bernilai satu) jika nilai magnitudonya lebih besar dari nilai ambang (*threshold*) yang ditetapkan. Arah tepi ( $\theta$ ) dapat dihitung menggunakan Persamaan 2.7. Setelah nilai *edge direction* diperoleh, langkah selanjutnya adalah menentukan *pixel pixel* citra yang merupakan garis (*edge*). Sebuah *pixel* akan dianggap sebagai *edge* jika nilai magnitudonya lebih besar dari nilai *threshold* yang ditetapkan (Gonzalez *et al*, 2004).

$$\theta = \tan^{-1}(G_y/G_x) \quad (2.7)$$

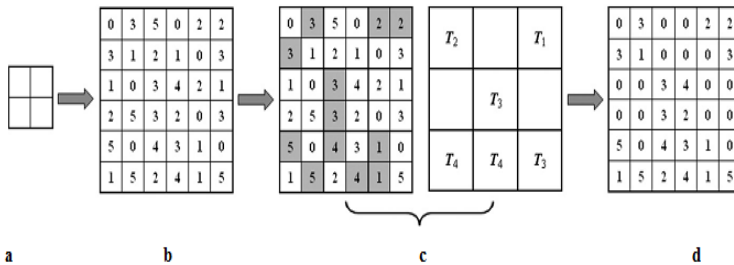
## 2.7 *Texton*

*Texton* adalah proses yang sangat berguna untuk analisis tekstur. Umumnya, *texton* didefinisikan sebagai satu set gumpalan atau pola yang muncul yang berbagi properti bersama diseluruh citra [10]. *Texton* yang digunakan adalah 4 tipe *texton* yang berukuran 2x2 dengan model yang ditunjukkan dalam gambar



**Gambar 2.2 4 Tipe *Texton* didefinisikan dalam MTH**  
 (a) 2x2 grid (b) T1 (c) T2 (d) T3 (e) T4

Langkah deteksi *texton* memanfaatkan 4 texton yang terkait dengan citra warna yang dikuantisasi. Langkah konvolusi dilakukan dari kiri ke kanan dan dari atas kebawah dengan dua piksel. Dimensi grid yang digunakan adalah piksel 2x2 yang ditandai dengan  $V_1, V_2, V_3, V_4$ . Apabila terdapat dua piksel yang sama, maka grid dideteksi sebagai *texton*. Apabila sebuah *texton* terdeteksi, nilai piksel asli dalam grid 2x2 tidak berubah. Selain itu, piksel berubah nilainya menjadi 0 [4].



**Gambar 2.3 Skema Pencarian *Texton* pada Citra**  
**(a) Model *texton* (b) Citra asli (c) Daerah *texton* dan tipe *texton* (d) Citra *texton***

## 2.8 *Distance Metric*

Dalam tugas akhir ini *distance metric* digunakan untuk mengukur kemiripan antar citra dengan mengukur jarak antara citra *query* dengan citra dalam database.

Setiap template citra pada database, sebuah vektor fitur  $M$ -dimensional  $T=[T_1, T_2, \dots, T_n]$  akan diekstrak dan disimpan didalam database.  $Q=[Q_1, Q_2, \dots, Q_n]$  adalah vektor fitur dari sebuah citra *query*, metrik jarak diantara citra ditunjukkan pada rumus dibawah :

$$D(T, Q) = \sum_{i=0}^n \frac{|T_i - Q_i|}{1 + T_i + Q_i} \quad (2.8)$$

Rumus diatas adalah rumus yang sama sederhana untuk menghitung jarak  $L_1$ , yang tidak membutuhkan operasi akar kuadrat atau kuadrat. Rumus tersebut dapat menghemat komputasi dan sangat cocok untuk dataset citra dengan skala besar.

## 2.9 *Recall dan Precision*

*Recall* dan *precision* merupakan parameter yang digunakan untuk mengukur keefektifan algoritma temu kembali informasi (*Information Retrieval*).

*Precision* adalah persentase rasio jumlah *record* relevan yang ditemukan terhadap jumlah *record* yang tidak relevan dan relevan. *Precision* digunakan untuk mengetahui tingkat ketepatan antara citra masukan dan citra hasil keuaran. Sedangkan *recall* adalah persentase rasio jumlah *record* relevan yang ditemukan terhadap jumlah total *record* yang relevan dalam database. Perhitungan *recall* digunakan untuk mengetahui tingkat keberhasilan sistem dalam menemukan kembali citra. *Confusion matrix* perhitungan *precision* dan *recall* ditunjukkan pada tabel 2.1.

**Tabel 2.1 *Confusion Matrix Precision dan Recall***

	Relevant	Not Relevant
Retrieved	TP	FP
Not Retrieved	FN	TN

TP (*True Positives*) yaitu citra yang relevan dan ditemukan kembali. FP (*False Positives*) adalah citra yang tidak relevan namun ditemukan kembali. FN (*False Negatives*) merupakan citra yang tidak relevan namun ditemukan kembali. TN (*True Negatives*) adalah citra yang tidak relevan dan tidak ditemukan kembali. Rumus perhitungan *precision* dan *recall* masing-masing dijelaskan pada persamaan 2.9 dan 2.10 berikut.

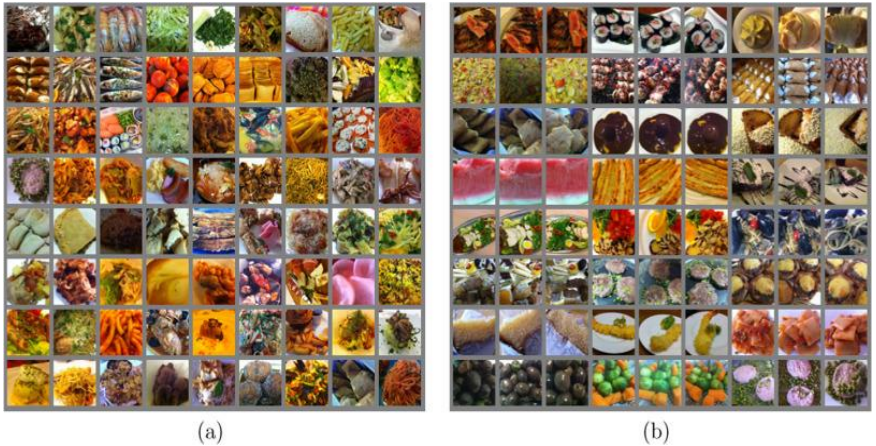
$$Precision = \frac{TP}{TP+FP} \times 100\% \quad (2.9)$$

$$Recall = \frac{TP}{TP+FN} \times 100\% \quad (2.10)$$

## 2.10 Dataset UNICT FD-1200

Dataset yang digunakan adalah UNICT FD-1200 yang berisi citra makanan. Dataset ini dari website milik "*University of Catania*" [3]. Data ini terdiri dari 4754 citra makanan dari 1200 makanan dari negara yang berbeda (Inggris, Jepang, India, Itali, Thailand, dan sebagainya). Setiap piring makanan difoto selama beberapa kali (rata-rata 4 kali) dan keseluruhan dataset mempunyai variabilitas secara geometrik dan fotometrik. Semua foto makanan diambil dalam kurun waktu 5 tahun terakhir dari foto makanan diambil menggunakan kamera ponsel dengan pengaturan seperti background yang berbeda dan memperhatikan kondisi cahaya. Citra dataset sudah diberi label secara manual kedalam delapan kategori, yaitu : *Appetizer, Main Course, Second Course, Single Course, Side Dish, Dessert, Breakfast, Fruit*.





**Gambar 2.4 Dataset Makanan UNICT-FD1200**  
**(a) Sampel dataset UNICT-FD1200 (b) 3 elemen dari 24 kelas dataset UNICT-FD1200**

***[Halaman ini sengaja dikosongkan]***

## **BAB III**

### **ANALISIS DAN PERANCANGAN**

Tugas akhir ini berfokus pada temu kembali citra makanan, dengan dataset yang diperoleh dari UNICT-FD1200. Bab ini membahas tentang perancangan perangkat lunak yang dibuat. Pada bab ini juga akan dijelaskan mengenai gambaran umum setiap proses utama sistem dalam diagram alir beserta penjelasannya.

#### **3.1 Lingkungan Desain dan Implementasi**

Subbab ini akan menjelaskan mengenai lingkungan desain dan implementasi perangkat lunak yang akan dibangun. Spesifikasi perangkat keras dan perangkat lunak yang digunakan dalam desain dan implementasi disebutkan dalam tabel 3.1.

**Tabel 3.1 Lingkungan Perancangan Perangkat Lunak**

Jenis	Spesifikasi
Prosesor	Intel (R) Core(TM) 2,40 Ghz
Memori	4,00 GB
Sistem Operasi	<i>Windows</i> 10 Pro 64 Bit
Pengembang	Matlab R2013a Ms. Word 2016

#### **3.2 Perancangan Data**













Pada sub bab ini dijelaskan mengenai perancangan data yang dibutuhkan untuk membangun perangkat lunak dalam melakukan ekstraksi fitur tekstur, warna dan deteksi *texton* yang meliputi data masukan, data proses dan data keluaran.





















### 3.2.1 Data Masukan

Data masukan yang diberikan oleh pengguna adalah citra makanan. Data masukan adalah citra makanan asli yang tersimpan dalam format JPG. Citra makanan tersebut berukuran 320x240 piksel. Citra dataset sudah diberi label secara manual kedalam delapan kategori, yaitu : *Appetizer, Main Course, Second Course, Single Course, Side Dish, Dessert, Breakfast, Fruit*.

Dataset makanan terdiri dari 416 citra yang didapatkan dari dataset UNICT-FD1200. Dari keseluruhan 416 citra makanan, dikelompokkan menjadi 8 kategori, jadi tiap kategori terdiri dari 52 citra yang mewakili 13 makanan yang berbeda, dimana setiap makanan difoto sebanyak 4 kali. 40 citra digunakan sebagai *testing* dan 376 citra digunakan sebagai *training*. Dibawah ini adalah tabel yang menunjukkan beberapa kelompok citra makanan.

**Tabel 3.2 Citra Makanan**

Kategori	1	2	3	4
Appetizer				
Breakfast				
Dessert				

Fruit				
Main Course				
Second Course				
Side Dish				
Single Course				

### 3.2.2 Data Pemrosesan

Data pemrosesan yang digunakan pada *Sobel Edge Detection* adalah variabel dari citra makanan. *Sobel edge Detection* diaplikasikan pada tiap kanal merah (R), hijau (G), dan biru (B) dari citra warna makanan untuk mendapatkan bentuk tekstur. Hasil dari pengaplikasian *sobel edge Detection* pada tiap kanal RGB akan digunakan pada perhitungan dot product lalu digunakan untuk mendapatkan nilai sudut.

Data pemrosesan pada kuantisasi warna di ruang warna RGB adalah data dari masing-masing kanal RGB citra yang

digunakan untuk mempermudah proses manipulasi warna. Masing-masing kanal dikuantisasi menjadi 4 bins sehingga terbentuk 64 warna. Data pemrosesan pada metode deteksi *texton* adalah hasil perhitungan dari kuantisasi warna. Citra hasil kuantisasi warna digunakan untuk mendeteksi *texton* dan membuat citra *texton*. Setelah itu data citra *texton* digunakan untuk proses mendapatkan *Multi-texton histogram*.

Pada aplikasi ini, terdapat data proses berdasarkan proses yang ada. Pada setiap proses akan menghasilkan data variabel yang menjadi masukan untuk proses berikutnya. Pembagian file dan variabel dapat dilihat pada tabel.

**Tabel 3.3 Nama File**

No	Nama File	Keterangan
1	Main.m	File yang berisi proses <i>texture orientation Detection</i> , kuantisasi warna, deteksi <i>texton</i> , dan <i>multi-texton histogram</i>
2	Distancejarak.m	File yang berisi proses perhitungan jarak, pengurutan jarak, dan perhitungan <i>precision</i> dan <i>recall</i> .

**Tabel 3.4 Data Variabel**

No	Nama Data	Tipe Data	Keterangan
1	Theta	Uint8	Menyimpan hasil ekstraksi fitur <i>sobel edge Detection</i>
2	ImageX	Uint8	Menyimpan hasil kuantisasi warna
3	Ori	array	Menyimpan hasil orientasi dalam array 2 dimensi

4	Texton	array	Menyimpan hasil deteksi texton dalam array 2 dimensi
5	MatrixH	array	Array 1 dimensi yang menyimpan hasil pencarian texton secara horizontal
6	MatrixV	array	Array 1 dimensi yang menyimpan hasil pencarian texton secara vertikal
7	MatrixRD	array	Array 1 dimensi yang menyimpan hasil pencarian texton secara diagonal kanan
8	MatrixLD	array	Array 1 dimensi yang menyimpan hasil pencarian texton secara diagonal kiri
9	MTH	array	Array 1 dimensi yang menyimpan hasil dari <i>multi-texton histogram</i>
10	TOTAL	Uint8	Menyimpan hasil MTH dari semua gambar yang diproses
12	MTH(file distancejarak.m)	array	Array yang menyimpan hasil dari nilai TOTAL yang berisi nilai MTH dari citra <i>testing</i> dari file .mat yang sudah tersimpan
13	dataMTH	array	Array yang menyimpan hasil dari nilai TOTAL yang berisi nilai MTH dari citra <i>training</i> dari file .mat yang sudah tersimpan
14	c	struct	<i>Struct</i> yang menyimpan label data <i>training</i>
15	d	struct	<i>Struct</i> yang menyimpan label data <i>testing</i>
16	dist	struct	<i>Struct</i> yang menyimpan hasil perhitungan jarak

17	res	struct	<i>Struct</i> yang berisi hasil sort per baris dari dist
18	perform	array	Array 2 dimensi yang berisi hasil <i>precision</i> dan <i>recall</i>

### 3.2.3 Data Keluaran

Data keluaran terakhir yang didapatkan oleh pengguna adalah citra makanan cocok dengan nilai jarak secara urut hasil pencocokan antara fitur yang diperoleh dari citra *query* dan citra *database*. Pada sistem ini, data keluaran adalah citra-citra yang mirip dengan data *query* sebanyak 10 citra.

### 3.3 Perancangan Proses

Pada subbab ini akan diuraikan perancangan proses yang bertujuan mengetahui hubungan antar proses beserta langkah-langkahnya pada setiap proses dalam membangun perangkat lunak Sistem temu kembali citra makanan menggunakan representasi *Multi-texton Histogram*.

Berikut adalah langkah-langkah umum proses yang akan dilakukan :

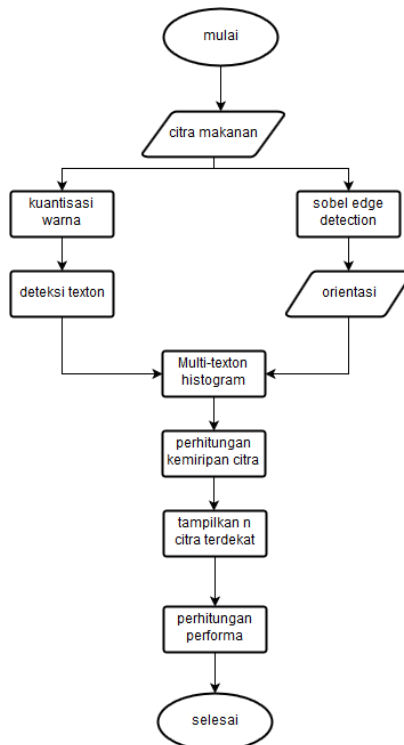
1. Data masukan yang diberikan pengguna adalah citra makanan.
2. Melakukan proses *texture orientation Detection* menggunakan *Sobel Edge Detection* yang digunakan untuk menghilangkan noise sebelum melakukan perhitungan deteksi tepi dan mendapatkan nilai sudut.
3. Melakukan proses kuantisasi warna pada ruang RGB hingga terbentuk 64 warna.
4. Melakukan deteksi *texton*. *Texton* dideteksi dengan cara memindai 2x2 grid dari piksel citra masukan,



dipindai dari atas ke bawah, kanan ke kiri, diagonal kanan dan diagonal kiri.

5. Melakukan proses representasi fitur dengan *Multi-texton Histogram* untuk mendapatkan nilai histogram citra.
6. Melakukan perhitungan kemiripan antar citra. Setelah didapatkan nilai kemiripan antar citra.
7. Perhitungan performa untuk mengetahui kinerja dari sistem yang telah dibuat.

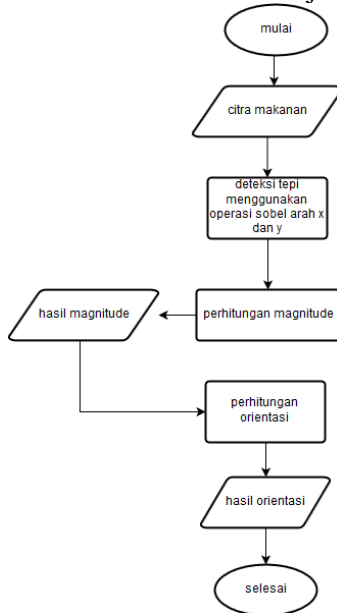
Gambar 3.1 merupakan gambaran umum mengenai proses-proses sehingga mendapatkan citra yang mirip dengan citra *query*.



**Gambar 3.1 Diagram Alir Sistem**

### 3.3.1 Proses *Texture Orientation Detection*

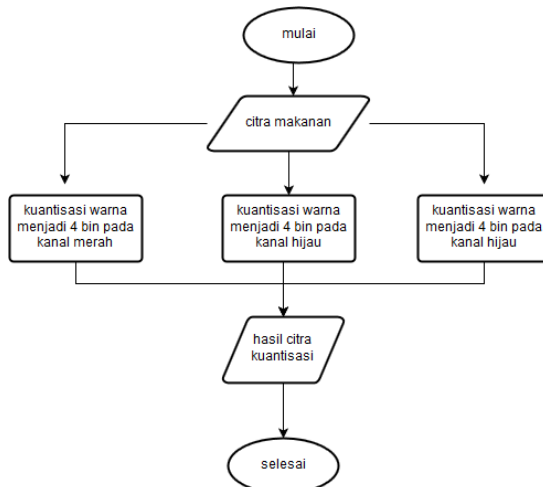
Sobel operator diaplikasikan terhadap setiap kanal merah, hijau, dan biru dari citra warna. Alasan diaplikasikannya Sobel operator adalah karena sobel operator tingkat kesensitifitasnya lebih rendah terhadap *noise* dibandingkan dengan operator *gradien* atau pendeteksi tepi yang lainnya menjadi sangat efisien [5]. *Gradien* sepanjang direksi x dan y dapat ditunjukkan oleh dua vektor  $a(R_x, G_x, B_x)$  dan  $b(R_y, G_y, B_y)$ , dimana  $R_x$  menunjukkan *gradien* dalam kanal R (merah) sepanjang direksi horizontal, dan seterusnya. Setelah didapatkan *gradien* sepanjang direksi x dan y dilanjutkan dengan proses mendapatkan *magnitude* dengan menggunakan rumus pada persamaan 2.5, 2.6 dan 2.7. Hasil dari perhitungan *magnitude* digunakan untuk mencari 18 orientasi dengan masing-masing 10 derajat. Gambaran umum mengenai proses *texture orientation detection* ditunjukkan pada gambar 3.2.



**Gambar 3.2 Diagram Alir Proses Sobel *Edge Detection***

### 3.3.2 Proses Kuantisasi Warna diruang warna RGB

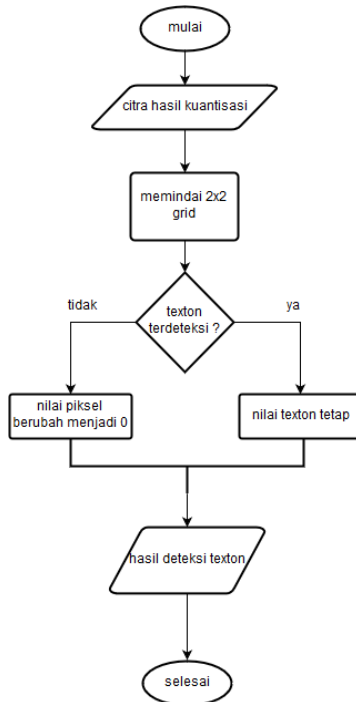
Sudah banyak diketahui bahwa warna mengandung informasi yang sangat penting untuk temu kembali citra atau pengenalan objek, bahkan dalam ketiadaan informasi bentuk. Dalam hal pemrosesan digital, ruang warna RGB paling sering digunakan dalam latihan dan itu sangat mudah. Untuk mengekstrak informasi warna dan menyederhanakan manipulasi, ruang warna RGB digunakan dan dikuantisasi menjadi 64 warna. Kanal RGB dikuantisasi secara merata menjadi 4 *bin* sehingga diperoleh 64 warna. Intensitas warna masing-masing dikelompokkan menjadi 4 kelompok dengan rentang intensitas 0-63, 64-127, 128-191, 192-255. Intensitas pada citra akan dirubah berdasarkan pembagian kelompok tersebut, angka 0 mewakili range intensitas 0-63, angka 1 mewakili range intensitas 64-127, angka 2 mewakili range intensitas 128-191 dan angka 3 mewakili range intensitas 192-255. Gambaran umum mengenai proses kuantisasi warna ditunjukkan pada gambar 3.3.



**Gambar 3.3 Diagram Alir Proses Kuantisasi Warna**

### 3.3.3 Proses Deteksi *Texton*

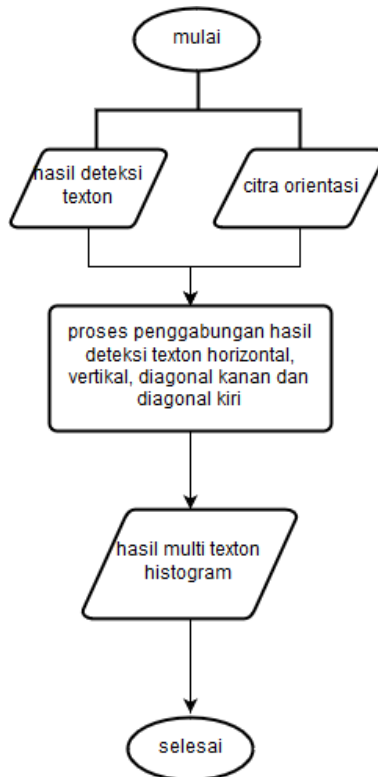
Langkah deteksi *texton* memanfaatkan 4 *texton* yang terkait dengan citra warna yang dikuantisasi. Langkah konvolusi dilakukan dari kiri ke kanan, dari atas kebawah, diagonal kiri dan diagonal kanan dengan dua piksel. Dimensi grid yang digunakan adalah piksel 2x2. Apabila terdapat dua piksel yang sama, maka grid dideteksi sebagai *texton*. Apabila sebuah *texton* terdeteksi, nilai piksel asli dalam grid 2x2 tidak berubah. Selain itu, piksel berubah nilainya menjadi 0. Gambaran umum mengenai proses deteksi *texton* ditunjukkan pada gambar 3.4.



**Gambar 3.4 Diagram Alir Proses Deteksi *Texton***

### 3.3.4 Proses *Multi-texton Histogram*

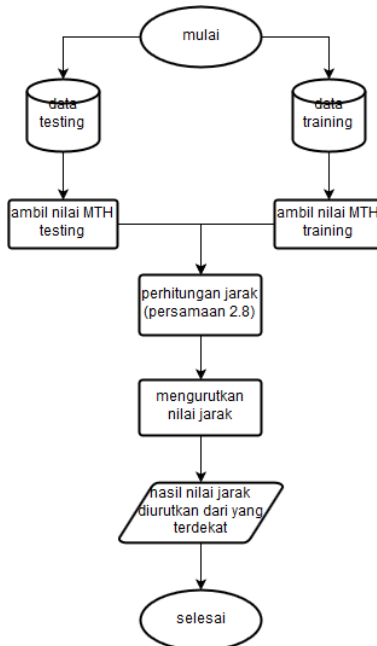
Hasil dari proses deteksi *texton* menghasilkan beberapa nilai yang didapatkan dari deteksi horizontal, vertikal, diagonal kanan dan diagonal kiri. Semua hasil tersebut diproses untuk mendapatkan hasil *multi-texton histogram*. Gambaran umum mengenai proses *multi-texton histogram* ditunjukkan pada gambar 3.5.



**Gambar 3.5** Diagram Alir Proses *Multi-texton Histogram*

### 3.3.5 Proses Perhitungan Kemiripan antar Citra

Perhitungan kemiripan antar citra digunakan untuk mengetahui kemiripan antara citra *query* dengan citra *database*. Semakin kecil jarak suatu citra dengan citra lainnya, maka semakin mirip citra tersebut. Hasil proses *Multi-texton Histogram* digunakan untuk mengetahui jarak antar citra. Perhitungan jarak menggunakan rumus pada persamaan 2.8. Jarak terdekat didapatkan dari hasil nilai jarak yang telah diurutkan berdasarkan nilai jarak terdekat ke nilai jarak terjauh. Gambaran umum mengenai proses perhitungan kemiripan antar citra ditunjukkan pada gambar 3.6.



**Gambar 3.6 Diagram Alir Proses Perhitungan Kemiripan antar Citra**

## **BAB IV IMPLEMENTASI**

Pada bab ini akan dibahas mengenai implementasi yang dilakukan berdasarkan rancangan yang telah dijabarkan pada bab sebelumnya. Pembahasan implementasi ini meliputi deskripsi lingkungan tahap implementasi sistem ini dilakukan, proses-proses pada tahap implementasi yang dikerjakan, beserta penjelasan fungsi-fungsinya dalam bentuk kode sumber.

### **4.1 Lingkungan Implementasi**

Subbab ini akan menjelaskan mengenai lingkungan implementasi perangkat lunak yang akan dibangun. Spesifikasi perangkat keras yang digunakan dalam implementasi perangkat lunak adalah prosesor berjenis Intel(R) Core(TM) i3-3110M CPU @ 2.40 GHz, dengan kapasitas memori (RAM) sebesar 4.00 GB. Sistem operasi yang digunakan adalah *Microsoft Windows 10 Pro 64 Bit*, sedangkan perangkat lunak yang digunakan dalam proses implementasi adalah *MATLAB R2013a*.

### **4.2 Implementasi Program**

Dalam subbab ini akan ditampilkan potongan kode program yang digunakan dalam temu kembali citra makanan dengan representasi Multi-texton Histogram. Potongan kode program yang akan ditampilkan dalam sub bab ini adalah potongan kode program untuk tahap *texture orientation detection*, kuantisasi warna, deteksi *texton*, representasi fitur *multi-texton histogram*, dan perhitungan jarak.

#### **4.2.1 Implementasi *Texture Orientation Detection***

Sobel operator diaplikasikan terhadap setiap kanal merah, hijau, dan biru dari citra warna. Kode program pada Kode Sumber

4.1 berisi fungsi untuk mengimplementasikan *sobel edge Detection*.

1	for i=2:Columns-1
2	for j=2:Rows-1
3	rx = abs((RGB(j+1 , i-1, 1) + 2 * RGB(j+1, i, 1) + RGB(j+1, i+1, 1)) - (RGB(j-1, i - 1, 1) + 2 * RGB(j-1, i, 1) + RGB(j-1, i + 1, 1)));
4	gx = abs((RGB(j+1 , i-1, 2) + 2 * RGB(j+1, i, 2) + RGB(j+1, i+1, 2)) - (RGB(j-1, i - 1, 2) + 2 * RGB(j-1, i, 2) + RGB(j-1, i + 1, 2)));
5	bx = abs((RGB(j+1 , i-1, 3) + 2 * RGB(j+1, i, 3) + RGB(j+1, i+1, 3)) - (RGB(j-1, i - 1, 3) + 2 * RGB(j-1, i, 3) + RGB(j-1, i + 1, 3)));
6	ry = abs((RGB(j-1 , i+1, 1) + 2 * RGB(j, i+1, 1) + RGB(j+1, i+1, 1)) - (RGB(j-1, i - 1, 1) + 2 * RGB(j, i-1, 1) + RGB(j+1, i - 1, 1)));
7	gy = abs((RGB(j-1 , i+1, 2) + 2 * RGB(j, i+1, 2) + RGB(j+1, i+1, 2)) - (RGB(j-1, i - 1, 2) + 2 * RGB(j, i-1, 2) + RGB(j+1, i - 1, 2)));
8	gy = abs((RGB(j-1 , i+1, 3) + 2 * RGB(j, i+1, 3) + RGB(j+1, i+1, 3)) - (RGB(j-1, i - 1, 3) + 2 * RGB(j, i-1, 3) + RGB(j+1, i - 1, 3)));
9	gxx = sqrt(double(rx * rx + gx * gx + bx * bx));
10	gyy = sqrt(double(ry * ry + gy * gy + by * by));
11	gxy = rx * ry + gx * gy + bx * by;
12	theta(j,i) = real(acosd(double(gxy)/double(gxx * gyy + 0.0001)));
13	end
14	End

**Kode Sumber 4.1 Implementasi Sobel Edge Detection**

Pada baris 3 sampai baris 8 merupakan potongan kode untuk proses sobel pada masing-masing kanal RGB. Pada baris 9 dan 10 mendapatkan *gradien* sepanjang x dan y, proses mendapatkan *magnitude* menggunakan rumus pada persamaan 2.5



dan 2.6. Pada baris 11 dilakukan perhitungan *dot product*. Pada baris 13 mendapatkan arah tepi, proses mendapatkan arah tepi dilakukan menggunakan rumus pada persamaan 2.7.

#### 4.2.2 Implementasi Kuantisasi Warna

Untuk mengekstrak informasi warna dan menyederhanakan manipulasi, ruang warna RGB digunakan dan dikuantisasi menjadi 64 warna. Kanal RGB dikuantisasi secara merata menjadi 4 *bins* sehingga diperoleh 64 warna.

1	for i=1:Rows
2	for j=1:Columns
3	.....
4	if (R >=0 && R <= 63) VI=0;
5	elseif (R >= 64 && R <= 127) VI=1;
6	elseif (R >= 128 && R <= 191) VI=2;
7	elseif (R >= 192 && R <= 255) VI=3;
8	
9	if (G >=0 && G <= 63) VI=0;
10	elseif (G >= 64 && G <= 127) SI=1;
11	elseif (G >= 128 && G <= 191) SI=2;
12	elseif (G >= 192 && G <= 255) SI=3;
13	
14	if (B >=0 && B <= 63) HI=0;
15	elseif (B >= 64 && B <= 127) HI=1;
16	elseif (B >= 128 && B <= 191) HI=2;
17	elseif (B >= 192 && B <= 255) HI=3;
18	
19	ImageX(i, j) = 16 * VI + 4 * SI + HI;
20	end
21	End

**Kode Sumber 4.2 Implementasi Kuantisasi Warna**

Pada baris 4 sampai baris 7 melakukan kuantisasi warna pada kanal R menjadi 4 bin. Pada baris 9 sampai baris 12

melakukan kuantisasi warna pada kanal G menjadi 4 *bin*. Pada baris 14 sampai bari 17 melakukan kuantisasi warna pada kanal B menjadi 4 *bin*. Pada baris 19 menambahkan jumlah kombinasi warna RGB.

### 4.2.3 Implementasi Deteksi *Texton*

Dari hasil kuantisasi warna, digunakan untuk proses deteksi *texton*. Sebelum melakukan proses deteksi *texton*, terlebih dahulu membuat matrix berisi nilai 0.

1	Texton = double(ones(Rows, Columns));
2	for i=2:Rows/2
3	for j=2:Columns/2
4	if (ImageX(2*i-1,2*j-1)== ImageX(2*i,2*j))
5	Texton(2*i-1,2*j-1) = ImageX(2*i-1, 2*j-1);
6	Texton(2*i, 2*j-1) = ImageX(2*i, 2*j-1);
7	Texton(2*i-1, 2*j) = ImageX(2*i-1,2*j);
8	Texton(2*i, 2*j) = ImageX(2*i, 2*j);
9	end
10	if (ImageX(2*i-1,2*j)==ImageX(2*i,2*j-1))
11	...
12	end
13	if (ImageX(2*i-1,2*j-1)==ImageX(2*i,2*j-1))
14	...
15	end
16	if (ImageX(2*i-1,2*j-1)==ImageX(2*i-1,2*j))
17	...
18	end
19	end
20	End

**Kode Sumber 4.3 Implementasi Deteksi *Texton***

Masukkan pada proses deteksi *texton* adalah hasil dari proses kuantisasi warna. Pada baris 1 dilakukan pembuatan matrix kosong *texton*. Pedeteksian *texton* dilakukan dengan memindai

matrix 2x2 grid secara vertikal, horizontal, diagonal kanan dan diagonal kiri. Pada baris 4 sampai baris 9 dilakukan pendeteksian texton secara diagonal kiri ke kanan pada piksel ImageX(3,3) dan ImageX(4,4). Pada baris 10 sampai baris 12 dilakukan pendeteksian texton secara diagonal kanan ke kiri pada piksel ImageX(3,4) dan ImageX(4,3). Pada baris 13 sampai baris 15 dilakukan pendeteksian texton secara vertikal pada piksel ImageX(3,3) dan ImageX(4,3). Pada baris 16 sampai baris 18 dilakukan pendeteksian texton secara horizontal pada piksel ImageX(3,3) dan ImageX(3,4). Pada baris 5 dilakukan pemindahan nilai piksel pada ImageX(3,3) ke piksel Texton(3,3). Pada baris 6 dilakukan pemindahan nilai piksel pada ImageX(4,3) ke piksel Texton(4,3). Pada baris 7 dilakukan pemindahan nilai piksel pada ImageX(3,4) ke piksel Texton(3,4). Pada baris 8 dilakukan pemindahan nilai piksel pada ImageX(4,4) ke piksel Texton(4,4). Pada baris 11, 14 dan 17 dilakukan proses yang sama seperti pada baris 5 sampai baris 8.

#### 4.2.4 Implementasi Fitur *Multi-texton Histogram*

Implementasi fitur *multi-texton histogram* dari hasil pendeteksian texton.

1	MTH = zeros(CSA+CSB,1);
2	for i=1: CSA+CSB
3	MTH(i) = (MatrixH(i) + MatrixV(i) + MatrixRD(i) + MatrixLD(i)) / 4.0;
4	end

##### **Kode Sumber 4.4 Implementasi *Multi-texton Histogram***

Pada baris 1 pembuatan matriks kosong sebanyak 82 baris dan 1 kolom. Pada baris 2 sampai baris 4 dilakukan penghitungan rata-rata nilai MTH. Untuk variabel CSA dan CSB menunjukkan nilai level kuantisasi dan level orientasi yang digunakan.

## 4.2.5 Implementasi Perhitungan Kemiripan Citra

Perhitungan jarak digunakan untuk mengetahui kemiripan antar citra ditunjukkan pada persamaan 2.8.

Database *training* dan *testing* dibutuhkan dalam proses perhitungan jarak. Dari database *training* dan *testing* diambil nilai MTH dari masing-masing untuk diproses.

1	For t=1:y
2	For j=1:totalY
3	Dist(j).jarak = 0;
4	For k=1:totalX
5	ValTotal = Total(k,j);
6	ValMTH = dataMTH(k,t);
7	dist(t,j) = sum(abs(valTotal-valMTH). / (1+valTotal+valMTH));
8	End
9	End
10	End

### Kode Sumber 4.5 Implementasi Perhitungan Kemiripan Citra

Dari kode sumber diatas, *for* sepanjang variabel *y* menunjukkan fungsi dilakukan sebanyak *size y* dari data *training*, begitu juga dengan *for* sepanjang variabel *totalY* dan *totalX*. Untuk nilai *totalY* adalah 82 karena merupakan elemen Histogram *Multi-texton*. Nilai dari masing-masing matrix *TOTAL* dan *MTH* disimpan kedalam variabel *valTotal* dan *valMTH*. Rumus perhitungan dari jarak adalah ditunjukkan pada variabel *dist* dan dapat ditunjukkan pada persamaan 2.8.

Setelah mendapatkan jarak dari masing-masing citra, nilai jarak diurutkan untuk mendapatkan citra mana yang jaraknya paling dekat dengan citra yang dicari dan citra mana yang jaraknya paling jauh dengan citra yang dicari.

1	Res = [];
2	For x=1:size(dist)
3	tmp = nestedsortStruct(dist(x,:), 'jarak');
4	res = [res; tmp];
5	end

#### Kode Sumber 4.6 Implementasi Sorting

Dari kode sumber diatas, baris 1 adalah membuat *array* kosong untuk menampung hasil *sorting* dari jarak. Selanjutnya dilakukan perulangan sebanyak *size* dari *dist* atau jarak yang telah didapatkan sebelumnya. Proses pengurutan pada baris 3 dan 4 dilakukan dengan mengurutkan nilai jarak pada masing-masing baris.

#### 4.2.6 Implementasi Perhitungan Kemiripan Citra

Setelah mendapatkan jarak citra terdekat dengan citra query dilakukan perhitungan *precision* dan *recall* untuk mengetahui performa dari sistem yang telah dibuat.

1	Performs = zeros(size(d,2),2);
2	For k=1: side(d,2)
3	LabelQuery=char(d{k}{2})
4	Tp=0, fp=0, fn=0;
5	For j=1:10
6	If strcmp(labelQuery(1,:),labelImRetrieval(1,:))==1
7	tp=tp+1;
8	else
9	fp=fp+1;
10	End
11	fn=fn;
12	Prec=(tp/(tp+fp))*100;
13	Rec=(tp/47)*100;
14	Performs(k,1)=prec;
15	Performs(k,2)=rec;

16	End
17	End

**Kode Sumber 4.7 Implementasi Precision dan Recall**

Dari kode sumber diatas, baris 1 dilakukan pembuatan matriks kosong dengan baris sebanyak citra *query* dengan 2 kolom. Pada baris 2 dilakukan perulangan sebanyak jumlah citra *query*. Selanjutnya dilakukan pengubahan tipe data menjadi char pada baris 3 dan inialisasi untuk variabel tp(true positive), fp(false positive), dan fn(false negative) dengan nilai 0 pada baris 4. Selanjutnya dilakukan perulangan sebanyak jumlah citra terdekat yang ingin dikeluarkan pada baris 5. Pada baris 6 sampai 9 dilakukan proses perhitungan tp dan fp dengan menyamakan kategori kelompok citra *query* dengan citra *training*. Apabila citra *query* dan citra *training* dalam kelompok yang sama, maka nilai tp bertambah 1, apabila tidak dalam kelompok yang sama maka nilai fp bertambah 1. Pada baris 12 dan 13 dilakukan perhitungan *precision* dan *recall*.

## **BAB V**

### **UJI COBA DAN EVALUASI**

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

#### **5.1 Lingkungan Uji Coba**

Lingkungan uji coba dan evaluasi merupakan komputer tempat uji coba perangkat lunak. Spesifikasi perangkat keras yang digunakan untuk melakukan uji coba perangkat lunak terdiri dari prosesor berjenis Intel(R) Core (TM) i3-3110M CPU @ 2.40 GHz, dengan kapasitas memori (RAM) sebesar 4.00 GB.

Sistem operasi yang digunakan adalah *Microsoft Windows* 10 Pro 64 Bit, serta *MATLAB R2013a*. Dokumentasi hasil uji coba dilakukan dengan menggunakan *Microsoft Paint*.

#### **5.2 Data Uji Coba**



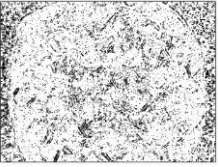


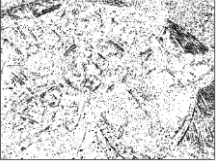


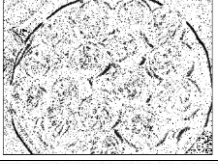


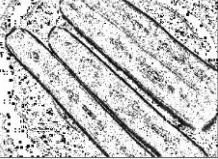
Pada tugas akhir ini, penulis menggunakan beberapa jenis data set sebagai masukan program. Dataset yang digunakan terdiri dari 40 citra makanan yang digunakan sebagai data *testing* dan 376 citra makanan digunakan sebagai data *training*. Dari keseluruhan 416 citra makanan, dikelompokkan menjadi 8 kategori, sehingga tiap kategori terdiri dari 52 citra yang mewakili 13 makanan yang berbeda, dimana setiap makanan difoto sebanyak 4 kali. 40 citra digunakan sebagai data *testing* dan 376 citra digunakan sebagai data *training*.

#### **5.3 Hasil *Texture Orientation Detection***

*Sobel* operator diaplikasikan terhadap setiap kanal merah, hijau, dan biru dari citra warna. Hasil dari *sobel edge Detection* digunakan untuk mengetahui tekstur dari citra. Setelah didapatkan

hasil *sobel edge detection*, dilakukan kuantisasi orientasi dengan membagi kedalam 18 orientasi dengan masing-masing 10 derajat. Beberapa citra hasil *sobel edge detection* dan *texture orientation detection* ditunjukkan pada tabel 5.1.

**Tabel 5.1 Hasil *Texture Orientation Detection***

<b>Citra Asli</b>	<b>Hasil Sobel Edge Detection</b>	<b>Hasil <i>Texture Orientation</i></b>
		
		
		
		

#### **5.4 Hasil Kuantisasi Warna pada Ruang RGB**

Langkah kuantisasi warna pada ruang RGB dilakukan dengan mengkuantisasi warna dari citra masukan. Citra RGB dikuantisasi menjadi 64 warna atau 4 bin. Kanal RGB dikuantisasi



secara merata menjadi 4 bins sehingga diperoleh 64 warna. Intensitas warna masing-masing dikelompokkan menjadi 4 kelompok dengan rentang intensitas 0-63, 64-127, 128-191, 192-255. Beberapa citra hasil kuantisasi warna ditunjukkan pada tabel 5.2.







**Tabel 5.2 Hasil Kuantisasi Warna**

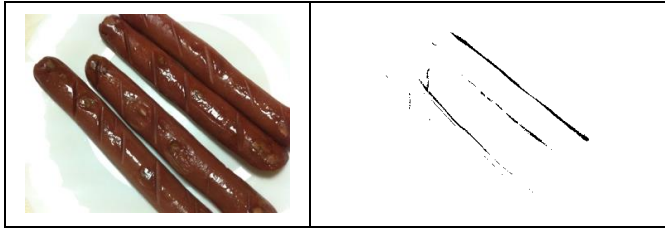
Citra Asli	Hasil Kuantisasi Warna
	
	
	
	

## 5.5 Hasil *Texton Detection*

Langkah deteksi *texton* memanfaatkan 4 *texton* yang terkait dengan citra warna yang dikuantisasi. Langkah konvolusi dilakukan dari kiri ke kanan dan dari atas kebawah dengan dua piksel. Hasil dari *texton detection* ini yang akan digunakan dalam proses selanjutnya yaitu *Multi-texton Histogram*. Beberapa citra hasil *texton detection* ditunjukkan pada tabel 5.3.

**Tabel 5.3 Hasil *Texton Detection***

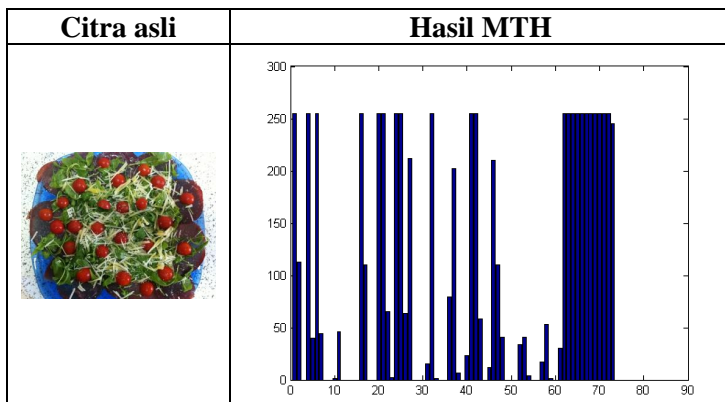
Citra Asli	Hasil <i>Texton Detection</i>
	
	
	

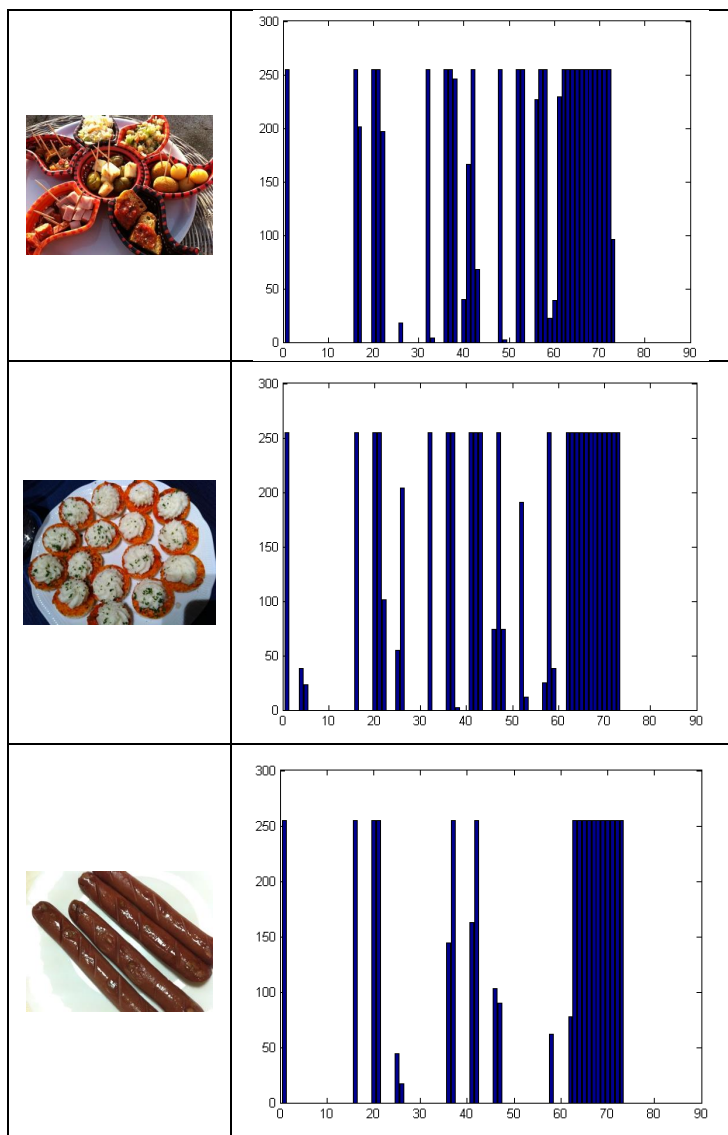


## 5.6 Hasil *Multi-texton Histogram*

*Multi-texton histogram* didapatkan dari proses sebelumnya yaitu *texture orientation detection*, kuantisasi warna, dan deteksi *texton*. Total MTH yang digunakan adalah sebanyak 82 yang terdiri dari 64 kuantisasi warna dan 18 orientasi. Beberapa citra hasil MTH akan ditunjukkan pada tabel 5.4.

**Tabel 5.4 Hasil *Multi-texton Histogram***





## 5.7 Skenario Uji Coba

Uji coba dilakukan untuk mengetahui nilai-nilai parameter untuk digunakan pada masing-masing proses.

Skenario pengujian terdiri dari tiga macam yaitu:

1. Uji coba berdasarkan level orientasi
2. Uji coba berdasarkan level kuantisasi warna
3. Uji coba berdasarkan jumlah kategori
4. Uji coba berdasarkan jumlah citra yang ditemukan

### 5.7.1 Uji Coba Berdasarkan Level Orientasi

Uji coba berdasarkan level orientasi yaitu uji coba kuantisasi orientasi menjadi 9 orientasi, 12 orientasi, 18 orientasi, 24 orientasi, dan 36 orientasi. Uji coba level orientasi diaplikasikan pada 8 kategori makanan. Dari tabel 5.5, menunjukkan nilai rata-rata *precision* dan *recall* terendah pada level orientasi 9 dan 12. Sedangkan pada level orientasi 18 menunjukkan nilai rata-rata *precision* dan *recall* tertinggi yaitu sebesar 40,50% *precision* dan 8,61% *recall*.

**Tabel 5.5 Hasil Uji Coba Level Orientasi**

<b>Level orientasi</b>	<b>Rata-rata Precision (%)</b>	<b>Rata-rata Recall (%)</b>
9	39,00	8,29
12	39,00	8,29
18	40,50	8,61
36	39,25	8,35

### 5.7.2 Uji Coba Berdasarkan Level Kuantisasi Warna

Uji coba berdasarkan level kuantisasi warna yaitu mengkuantisasi warna menjadi 64 atau 4 *bin* dan 128 atau 2 *bin* warna pada ruang warna RGB. Untuk kuantisasi menjadi

64, intensitas warna masing-masing dikelompokkan menjadi 4 kelompok dengan rentang intensitas 0-63, 64-127, 128-191, 192-255. Sedangkan untuk kuantisasi menjadi 128, intensitas warna masing-masing dikelompokkan menjadi 2 kelompok dengan rentang intensitas 0-127, dan 128-255. Tabel 5.6 dan 5.7 menunjukkan hasil uji coba level kuantisasi warna.

**Tabel 5.6 Hasil *Precision* Uji Coba Level Kuantisasi**

Level Kuantisasi	Hasil <i>Precision</i> (%)			
	Orientasi			
	9	12	18	36
64	39,00	39,00	40,50	39,25
128	21,50	21,50	21,50	21,50

**Tabel 5.7 Hasil *Recall* Uji Coba Level Kuantisasi**

Level Kuantisasi	Hasil <i>Recall</i> (%)			
	Orientasi			
	9	12	18	36
64	8,29	8,29	8,61	8,35
128	4,57	4,57	4,57	4,57

### 5.7.3 Uji Coba Berdasarkan Jumlah Kategori

Uji coba berdasarkan jumlah kategori dikelompokkan menjadi 8 kategori, 6 kategori, 4 kategori dan 2 kategori untuk mengetahui nilai *precision* yang paling besar. Pada uji coba jumlah kategori didapatkan bahwa semakin sedikit kategori yang digunakan maka semakin besar nilai *precision* yang dihasilkan. Hal ini karena tingkat kemiripan antar kategori yang tinggi. Hasil uji coba berdasarkan jumlah kategori dapat dilihat pada tabel 5.8.

Tabel 5.8 Hasil Uji Coba Jumlah Kategori

Jumlah Kategori	Precision (%)	Recall (%)
8	40,50	8,61
6	42,67	9,07
4	52,50	11,17
2	67,00	14,25

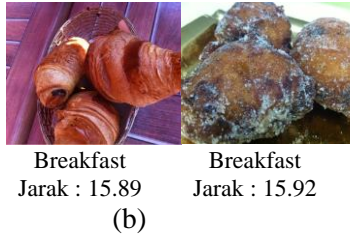
Pada uji coba berdasarkan 2 kategori, citra yang ditemukan nilai akurasinya lebih tinggi karena hanya 2 kategori saja yang digunakan. Dapat dilihat dari gambar 5.1, citra *query* yang digunakan adalah termasuk dalam kategori *breakfast*, dari 10 citra yang ditemukan, terdapat 1 citra yang bukan termasuk citra kategori *breakfast*.



Kategori : Breakfast

(a)

Breakfast  
Jarak : 9.12Breakfast  
Jarak : 9.66Breakfast  
Jarak : 12.84Breakfast  
Jarak : 13.90Breakfast  
Jarak : 14.96Fruit  
Jarak : 15.02Breakfast  
Jarak : 15.56Breakfast  
Jarak : 15.58



**Gambar 5.1 Citra yang ditemukan berdasarkan 2 kategori  
(a) citra query (b) citra yang ditemukan**

Pada uji coba berdasarkan 8 kategori, citra yang ditemukan nilai akurasinya lebih rendah karena terdapat 8 kategori yang digunakan dan tingkat kemiripan antar kategori yang tinggi. Dapat dilihat dari gambar 5.2, citra query yang digunakan adalah termasuk dalam kategori *fruit*, dari 10 citra yang ditemukan, hanya ditemukan 3 citra yang termasuk citra kategori *fruit*.



Kategori : fruit

(a)







Appetizer  
Jarak : 9.66

Second Course  
Jarak : 10.23

(b)

**Gambar 5.2 Citra yang ditemukan berdasarkan 8 kategori**  
(a) Citra query (b) citra yang ditemukan

### 5.7.4 Uji Coba Berdasarkan Jumlah Citra Ditemukan

Uji coba berdasarkan jumlah citra yang ditemukan yaitu dengan mengambil batas jumlah citra terdekat yang ditemukan. Pada uji coba ini, diambil 10, 7, dan 5 citra terdekat sebagai uji coba. Setelah diambil jarak jumlah citra terdekat, maka didapatkan nilai *precision* dan *recall*nya lalu dihitung rata-rata *precision* dan *recall*nya.

**Tabel 5.9 Hasil Uji Coba Jumlah Citra yang Ditemukan**

Citra yang ditemukan	<i>Precision</i> (%)	<i>Recall</i> (%)
10	40,50	8,61
7	48,00	7,12
5	55,50	5,90

## 5.8 Evaluasi

Pada subbab ini akan dijelaskan hasil serangkaian uji coba yang dilakukan dan kendala yang dihadapi selama proses pengerjaan. Evaluasi yang dilakukan adalah pada nilai hasil *precision* dan *recall*.

### 5.8.1 Evaluasi Uji Coba Berdasarkan Level Orientasi

Pada uji coba berdasarkan level orientasi didapatkan hasil nilai rata-rata *precision* terbaik pada level orientasi 18 dengan nilai rata-rata 40,50%. Hasil ini diuji coba pada 18 kategori dan 64 level kuantisasi warna. Hasil *recall* yang didapatkan pada uji coba pada 18 orientasi adalah sebesar 8,61%. Nilai rata-rata *precision* dan *recall* terendah didapatkan pada 9 dan 16 level orientasi dengan nilai rata-rata *precision* 39,00% dan nilai rata-rata *recall* 8,29%.

### 5.8.2 Evaluasi Uji Coba Berdasarkan Level Kuantisasi

Pada uji coba berdasarkan level kuantisasi didapatkan hasil nilai rata-rata *precision* terbaik pada level kuantisasi 64 dengan nilai rata-rata 40,50%. Sedangkan nilai rata-rata pada level kuantisasi 128 mempunyai nilai yang sama yaitu *precision* sebesar 21,50% dan *recall* sebesar 4,57%. Pada uji coba dengan level kuantisasi juga dilakukan perbandingan level orientasi. Nilai rata-rata *precision* dan *recall* terendah pada 64 level kuantisasi didapatkan pada 9 dan 16 level orientasi dengan nilai rata-rata *precision* 39,00% dan nilai rata-rata *recall* 8,29%.

### 5.8.3 Evaluasi Uji Coba Berdasarkan Jumlah Kategori

Pada uji coba berdasarkan jumlah kategori didapatkan percobaan dengan nilai rata-rata *precision* terbesar adalah pada percobaan menggunakan 2 kategori. Begitu pula dengan hasil nilai rata-rata *recall* didapatkan pada percobaan menggunakan 2 kategori yaitu 67,00% untuk *precision* dan 14,25% untuk *recall*. Hal ini dikarenakan pada kategori 8, 6, dan 4 terdapat banyak citra yang mirip namun berada pada kategori yang berbeda.

#### **5.8.4 Evaluasi Uji Coba Berdasarkan Jumlah Citra yang Ditemukan**

Pada uji coba berdasarkan jumlah citra yang ditemukan didapatkan nilai rata-rata *precision* terbesar pada jumlah citra 5 yaitu sebesar 55,50% namun nilai rata-rata *recall*-nya rendah, yaitu sebesar 5,90%. Dari tabel 5.9 diketahui bahwa dengan semakin kecilnya jumlah citra yang ditemukan, maka nilai rata-rata *precision* berbanding terbalik dengan nilai rata-rata *recall*. Hal ini dikarenakan perhitungan *precision* adalah data citra yang ditemukan dan relevan dibandingkan dengan data yang ditemukan. Sedangkan untuk *recall* adalah data yang ditemukan dan relevan dibandingkan dengan data yang terdapat dalam *database*.

***[Halaman ini sengaja dikosongkan]***

## **BAB VI**

### **KESIMPULAN DAN SARAN**

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

#### **6.1 Kesimpulan**

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

1. Jumlah kategori mempengaruhi hasil *precision* citra, hal ini dikarenakan banyak citra dalam dataset yang mirip namun dibedakan kategorinya.
2. Hasil citra yang ditemukan menunjukkan kemiripan berdasarkan *visual*, namun dikarenakan berbeda kategori makanan sehingga nilai *precision* bernilai rendah yaitu sebesar 40,50%.
3. Berdasarkan hasil uji coba level orientasi dan level kuantisasi warna pada 8 kategori, nilai rata-rata *precision* dan *recall* terbaik adalah sebesar 40,50% dan 8,61% yaitu pada level orientasi 18 dan level kuantisasi 64.
4. Sistem yang dibangun dapat menemukan kembali citra sesuai dengan citra *query*. Hal ini dibuktikan dengan rata-rata nilai *precision* dan *recall* yang cukup pada 2 kategori, yaitu *precision* sebesar 67,00% dan *recall* 14,25%.

#### **6.2 Saran**

Berikut ini merupakan beberapa saran untuk pengembangan penelitian yang akan datang, saran yang diberikan adalah sebagai berikut :

1. Mengelompokkan kembali kategori dalam dataset, citra yang serupa dikelompokkan menjadi satu. Hal ini perlu dilakukan

karena kemiripan citra antara satu kategori dengan kategori yang lain mempengaruhi nilai hasil *precision* dan *recall*.

2. Menggunakan metode perbandingan jarak yang lebih beragam untuk mengetahui metode apa yang paling baik untuk digunakan dalam kasus ini.
3. Menggunakan tipe operator *gradien* yang lebih beragam untuk mengetahui operator *gradien* mana yang lebih tepat untuk mendeteksi *gradien magnitude* dan orientasi.
4. Menggunakan ruang warna selain RGB dalam pembagian komponen warna untuk mengetahui ruang warna apa yang paling baik dengan menggunakan metode ini.

## DAFTAR PUSTAKA

- [1] "TineEye reverse image search engine.," Idee Inc., 2016. [Online]. Available: [www.ideeinc.com](http://www.ideeinc.com). [Accessed 14 Desember 2016].
- [2] G. M. Farinella, D. Allegra, M. Moltisanti, F. Stanco and S. Battiato, "Retrieval and classification of food image," *Computers in Biology and Medicine*, p. 17, 2016.
- [3] G. M. Farinella, D. Allegra, M. Moltisanti, F. Stanco and S. Battiato, "UNICT-FD1200," 6 July 2016. [Online]. Available: [iplab.dmi.unict.it/UNICT-FD1200/](http://iplab.dmi.unict.it/UNICT-FD1200/). [Accessed 14 Desember 2016].
- [4] G. H. Liu, L. Z. Y. K. Hou, Z. Y. Li and J. Y. Yang, "Image retrieval based on multi-texton histogram," *Pattern Recognition*, pp. 2380-2389, 2010.
- [5] F. Aziz, *Sistem Temu Kembali Citra Kain Berbasis Tekstur dan Wana*, Pekanbaru, 2013.
- [6] J. Eakins and M. Graham, "Education Line," JISC Technology Applications Programme , 23 November 1999. [Online]. Available: <http://www.leeds.ac.uk>. [Accessed 17 June 2017].
- [7] I. Hastuti, M. Hariadi and I. K. Edy Purnama, "Content Based Image Retrieval Berdasarkan Fitur Bentuk Menggunakan Metode Gradient Vector Flow Snake," *ISSN*, pp. 1979-2328, 2009.
- [8] M. J. Swain and D. H. Ballard, "Color Indexing," *International Journal of Computer Vision*, vol. 7, no. 1, pp. 11-32, 1991.
- [9] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, third ed, Prentice Hall, 2007.
- [10] N. Ulinnuha and H. Sa'dyah, "Sistem Temu Kembali Citra untuk E-Commerce Menggunakan Prosedur Pencarian Dua



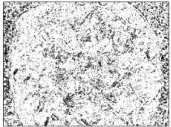

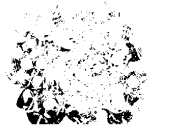
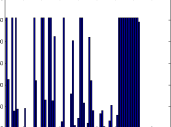


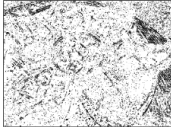


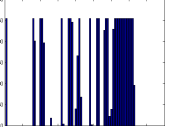

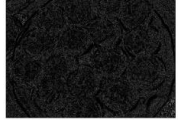
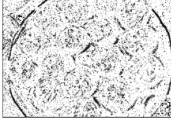


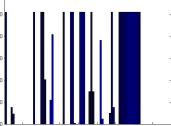
Fase dengan Fitur Histogram Multi Tekston," vol. 1, pp. 35-41, 2015.

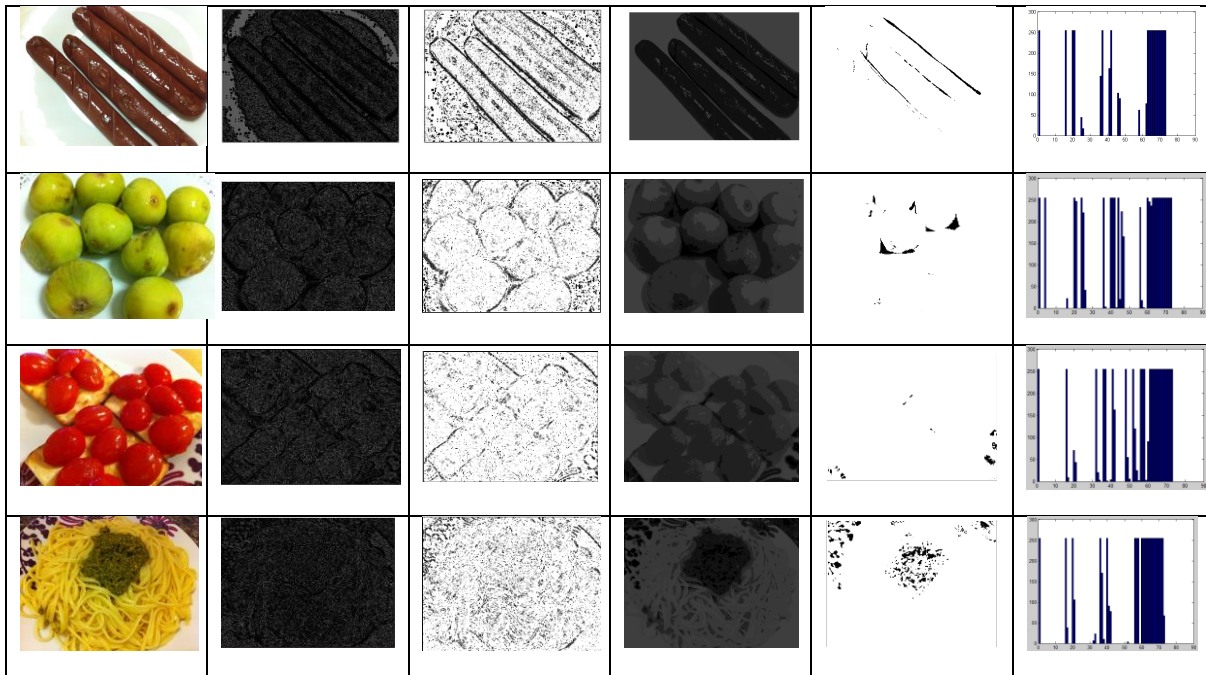
- [11] B. Julesz, "Textons, the elements of texture perception and their interactions," *Nature*, vol. 290, pp. 91-97, 1981.
- [12] B. Julesz, "Texton gradients: the texton theory revisited," *Biological Cybernetics*, vol. 54, pp. 245-251, 1986.
- [13] I. Sapuguh, D. O. Siahaan and C. Fatichah, "Sistem Temu Kembali Citra Gedung Berdasarkan Informasi Garis pada Bentuk Gedung," *Kursor*, vol. 5, 2009.

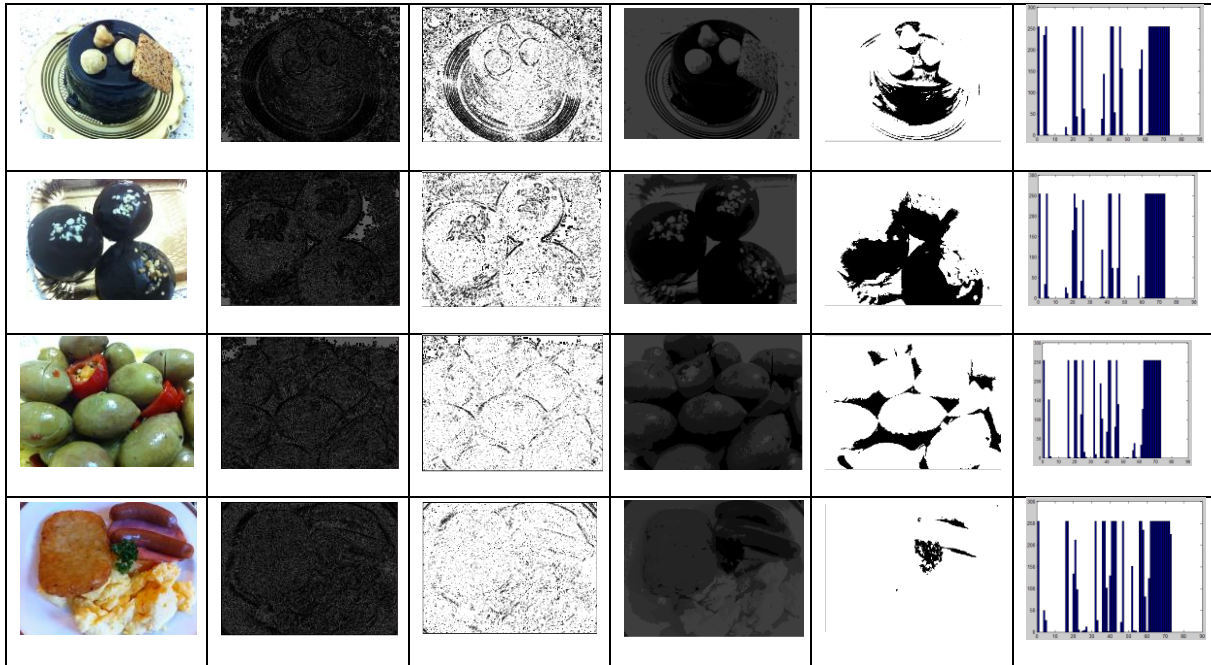


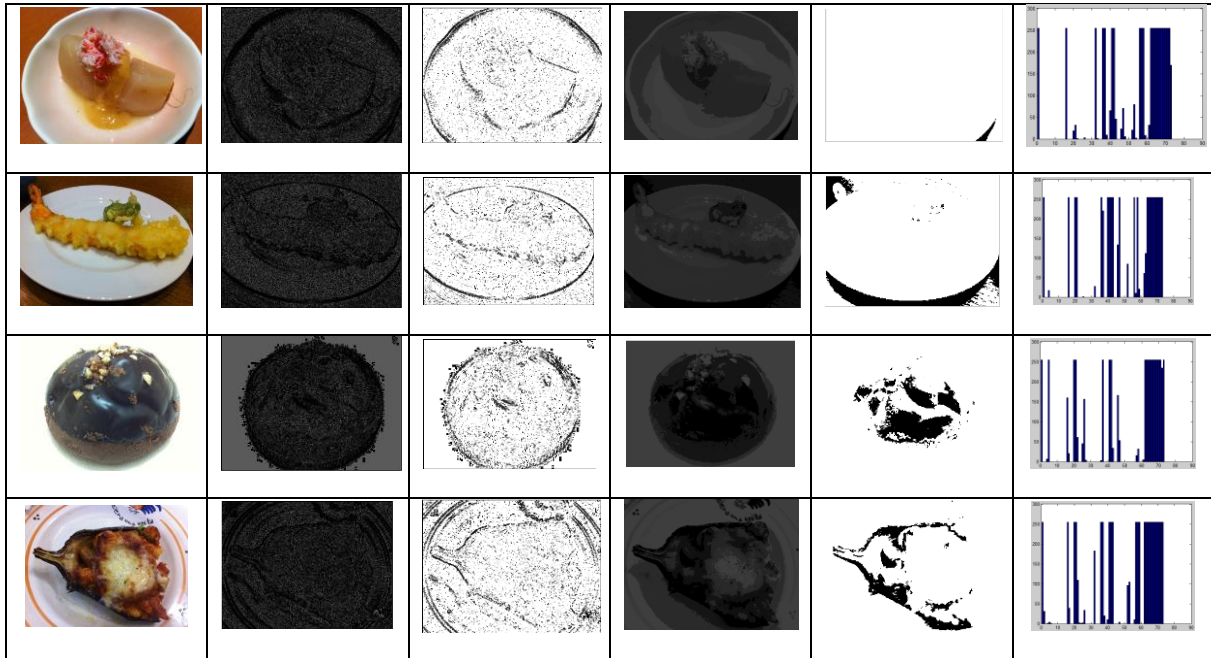
## LAMPIRAN

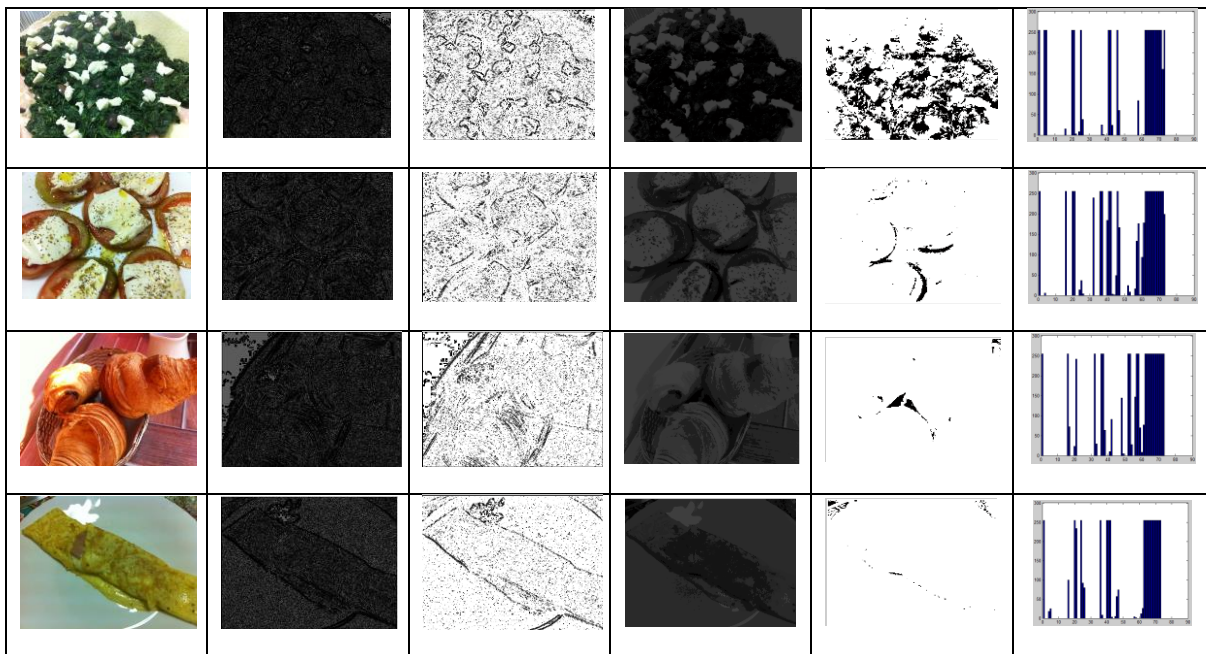
### A. Hasil Ekstraksi Data Tes 8 Kategori

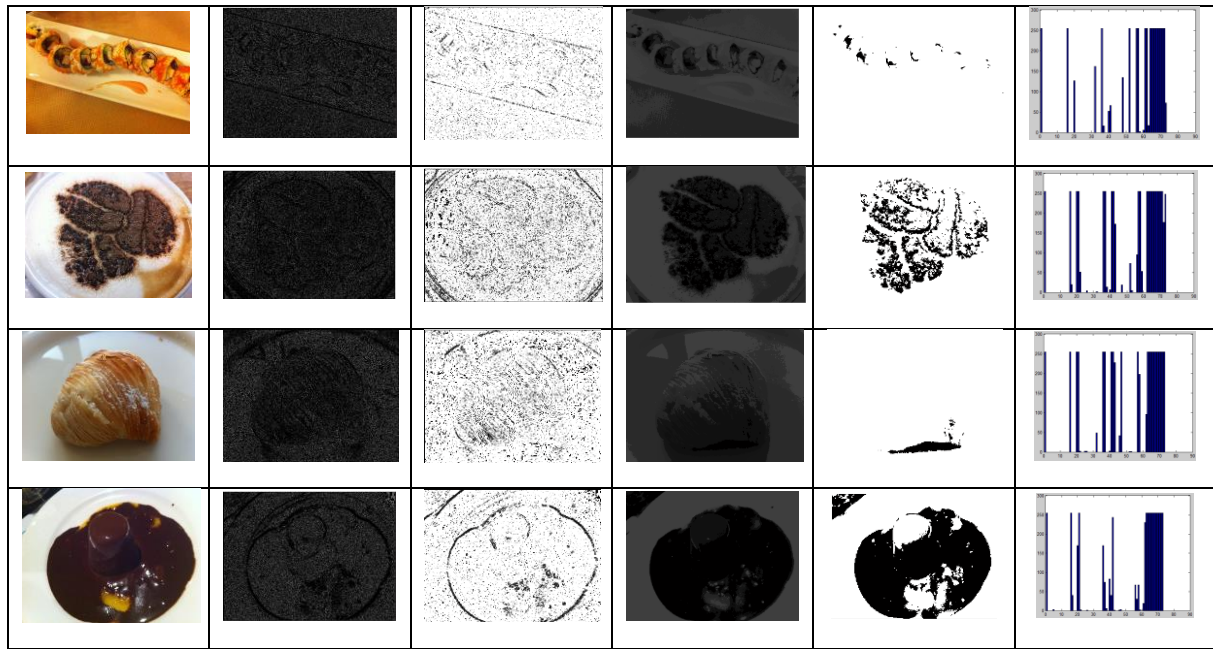
Citra Asli	<i>Sobel Edge</i>	<i>Orientation</i>	Kuantisasi Warna	<i>Texton Detection</i>	MTH
					
					
					

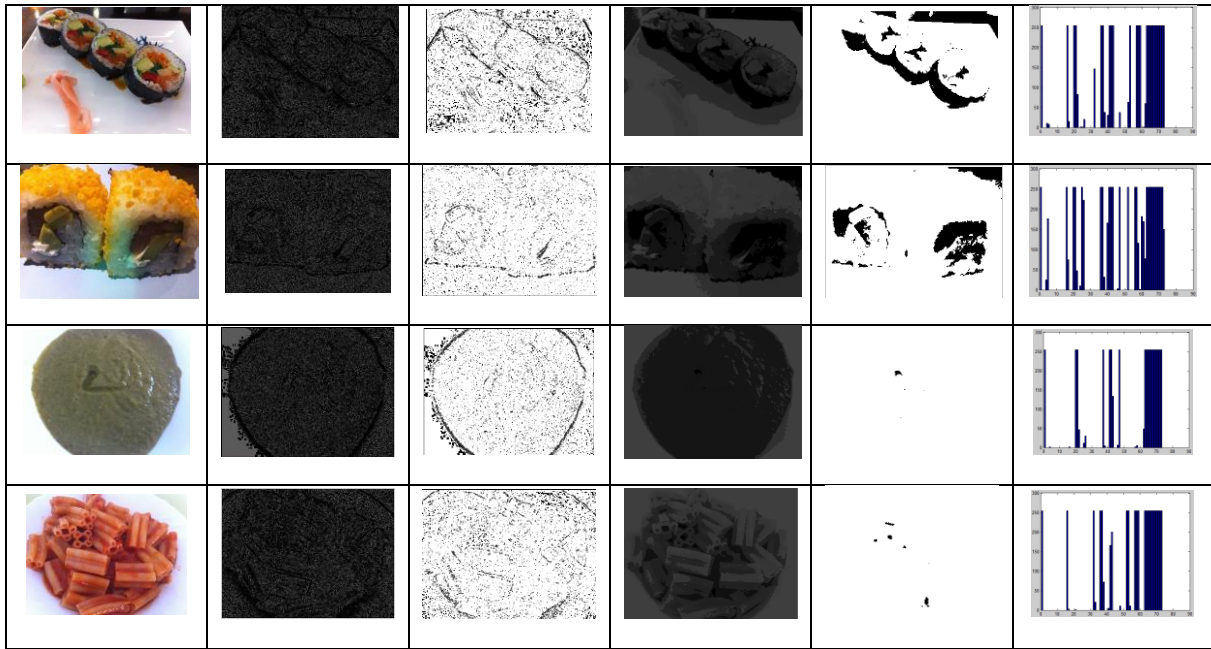


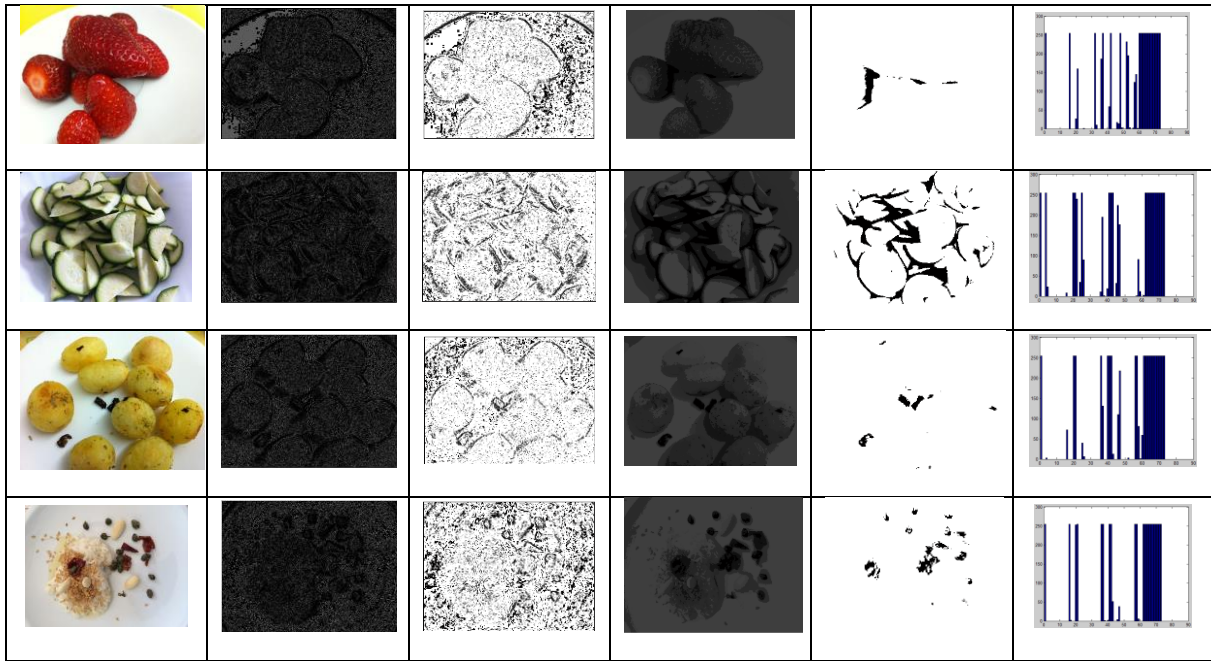




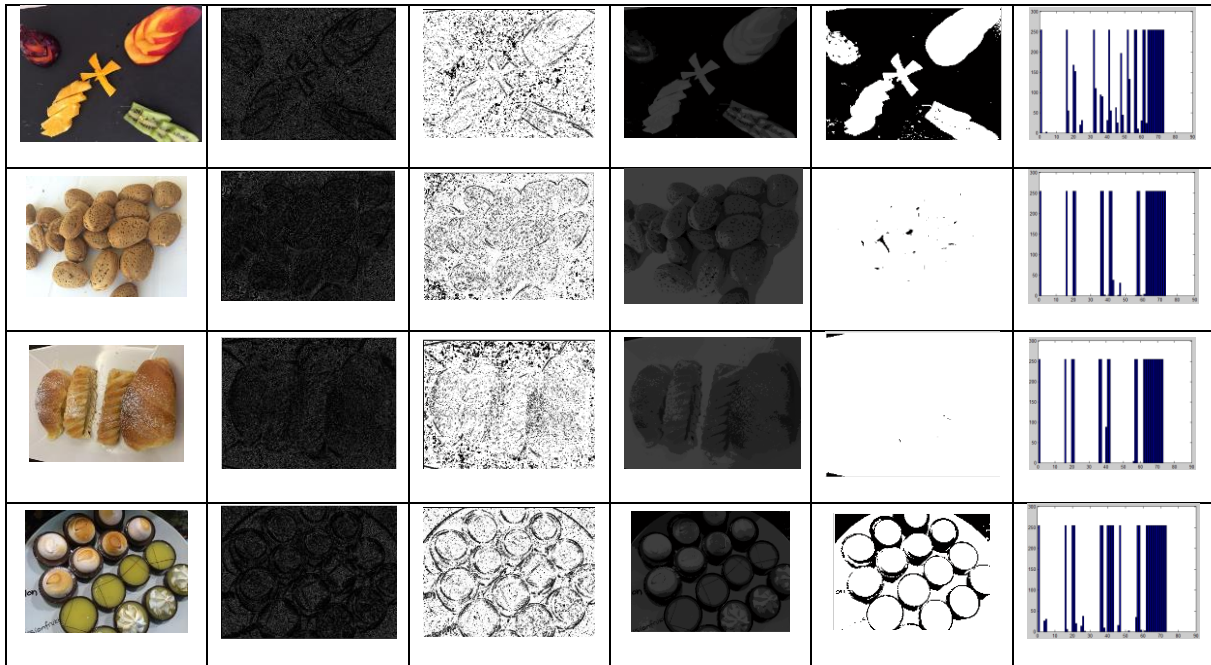


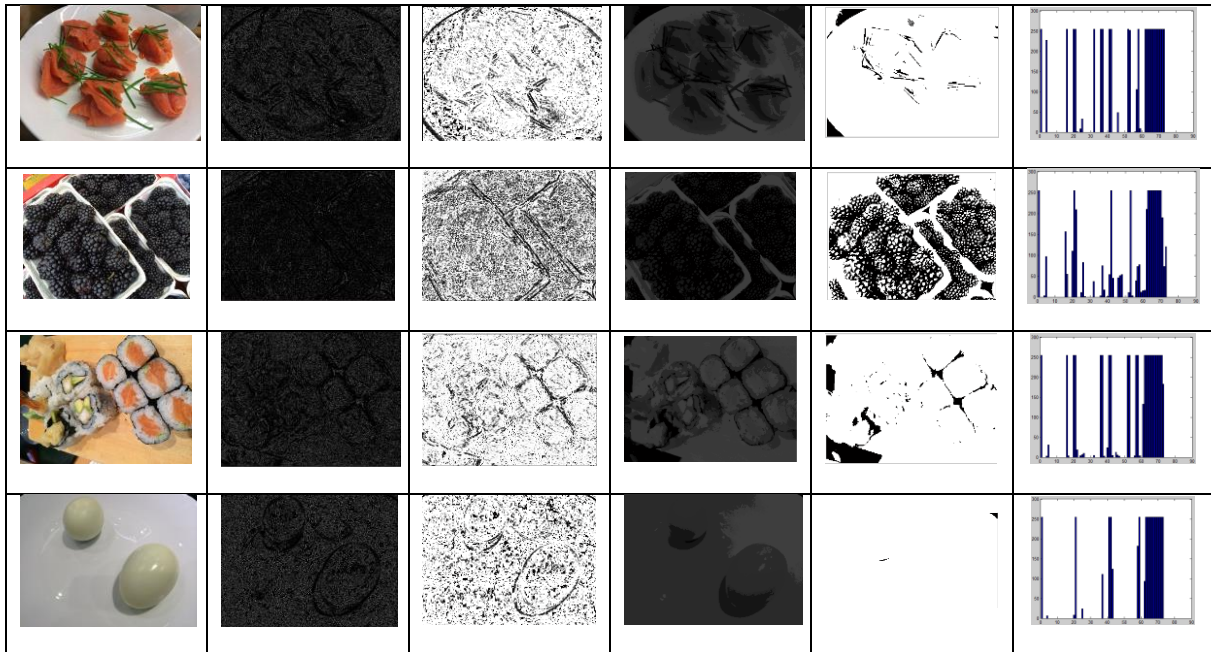


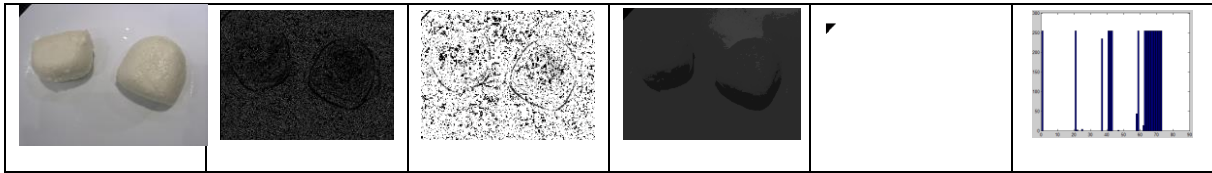












*[Halaman ini sengaja dikosongkan]*

## BIODATA PENULIS



Imagine Clara Arabella, lahir di Blora, pada tanggal 13 September 1995. Penulis menempuh pendidikan mulai dari SDN 3 Cepu (2001-2007), SMPN 4 Karangnom Klaten (2007-2010), SMAN 1 Karangnom Klaten (2010-2013) hingga terakhir Institut Teknologi Sepuluh Nopember Surabaya (2013-2017) di jurusan Teknik Informatika, Fakultas Teknologi Informasi angkatan tahun 2013.

Selama belajar di kampus Teknik Informatika, penulis berkesempatan menjadi admin laboratorium pemrograman pada tahun 2015. Selain mengikuti kegiatan akademik, penulis mengikuti kegiatan organisasi sekaligus mengembangkan minatnya pada bidang sumber daya manusia dengan berpartisipasi sebagai anggota Departemen Sumber Daya Mahasiswa HMTC (2014-2015) dan Sekretaris HMTC kabinet Optimasi (2015-2016).

Komunikasi dengan penulis dapat melalui email: **[imagineclaraarabella1@gmail.com](mailto:imagineclaraarabella1@gmail.com)**.