

TUGAS AKHIR - KI141502

**IMPLEMENTASI DETEKSI *COPY-MOVE FORGERY*  
PADA CITRA DENGAN MENGGUNAKAN METODE  
*HISTOGRAM OF ORIENTED GRADIENTS (HOG)***

NI LUH MADE ASRI MULYASARI  
5113100085

Dosen Pembimbing  
Dr. Eng. Nanik Suciati, S.Kom., M.Kom.  
Arya Yudhi Wijaya, S.Kom., M.Kom.

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





**TUGAS AKHIR - KI141502**

**IMPLEMENTASI DETEKSI *COPY-MOVE FORGERY*  
PADA CITRA MENGGUNAKAN METODE  
*HISTOGRAM OF ORIENTED GRADIENTS (HOG)***

NI LUH MADE ASRI MULYASARI  
5113100085

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

Dosen Pembimbing II  
Arya Yudhi Wijaya, S.Kom., M.Kom.

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

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



**FINAL PROJECT - KI141502**

# **IMPLEMENTATION OF COPY-MOVE IMAGE FORGERY DETECTION USING HISTOGRAM OF ORIENTED GRADIENTS (HOG)**

**NI LUH MADE ASRI MULYASARI  
5113100085**

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

**Supervisor II  
Arya Yudhi Wijaya, S.Kom., M.Kom.**

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

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

## LEMBAR PENGESAHAN

### **IMPLEMENTASI DETEKSI COPY-MOVE FORGERY PADA CITRA MENGGUNAKAN METODE *HISTOGRAM OF ORIENTED GRADIENTS (HOG)***

#### **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

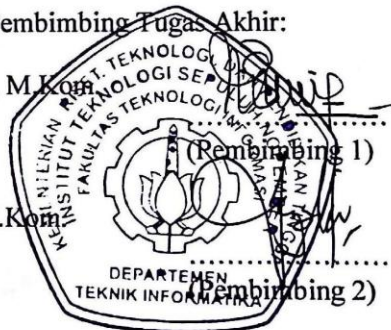
Oleh:

**NI LUH MADE ASRI MULYASARI**  
**NRP: 5113 100 085**

Disetujui oleh Dosen Pembimbing Tugas Akhir:

Dr. Eng. Nanik Suciati, S.Kom., M.Kom  
NIP. 19710428 199412 2 001

Arya Yudhi Wijaya, S.Kom., M.Kom  
NIP. 19840904 201012 1 002



**SURABAYA**  
**JULI, 2017**

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



# **IMPLEMENTASI DETEKSI *COPY-MOVE FORGERY* PADA CITRA MENGGUNAKAN METODE *HISTOGRAM OF ORIENTED GRADIENTS (HOG)***

Nama Mahasiswa : Ni Luh Made Asri Mulyasari  
NRP : 5113 100 085  
Jurusan : Teknik Informatika, FTIf ITS  
Dosen Pembimbing 1 : Dr. Eng. Nanik Suciati, S.Kom., M.Kom.  
Dosen Pembimbing 2 : Arya Yudhi Wijaya, S.Kom., M.Kom.

## ***Abstrak***

*Pemalsuan citra terdiri dari bermacam-macam jenis yang salah satunya adalah melakukan duplikasi suatu bagian gambar kemudian ditempelkan pada bagian lain di gambar tersebut. Hal ini disebut sebagai copy-move forgery atau pemalsuan pada citra. Karena bagian yang diduplikasi berasal dari citra yang sama, maka property-properti seperti noise, warna dan tekstur akan kompatibel dengan seluruh citra dan dengan demikian menjadi lebih sulit untuk dibedakan dan deteksi secara kasat mata.*

*Tugas akhir ini mengusulkan sebuah sistem deteksi pemalsuan copy-move pada citra dengan menggunakan metode Histogram of Oriented Gradients (HOG). Data citra pertama dibagi ke dalam blok-blok yang saling overlap kemudian setiap blok di ekstraksi dengan HOG yang menghasilkan fitur vektor. Kemudian deteksi copy-move dicari dengan menghitung jarak antar fitur setiap blok, blok-blok yang memiliki nilai jarak nol merupakan daerah yang mengalami copy-move.*

*Uji coba dilakukan terhadap 20 citra yang diambil dari CoMoFod database. Berdasarkan hasil uji coba menunjukkan bahwa metode ini dapat memberikan hasil deteksi copy-move yang akurat dengan rata-rata akurasi, specificity dan sensitivity adalah 99,59%, 99,91%, dan 90,21%.*

***Kata kunci: Copy-move, Euclidean Distance, Histogram of Oriented Gradients, pemalsuan citra***

*[Halaman ini sengaja dikosongkan]*

# **IMPLEMENTATION OF COPY-MOVE IMAGE FORGERY DETECTION USING HISTOGRAM OF ORIENTED GRADIENTS (HOG)**

Student Name : Ni Luh Made Asri Mulyasari  
Registration Number : 5113 100 085  
Department : Informatics Engineering, FTIf ITS  
First Supervisor : Dr. Eng. Nanik Suciati, S.Kom.,  
M.Kom.  
Second Supervisor : Arya Yudhi Wijaya, S.Kom., M.Kom.

## ***Abstract***

*Images forgery are composed of various types, one of which is duplicating a portion of the image to another part of the image. This is called copy-move forgery in the image. Because the duplicated part comes from the same image, properties such as noise, color and texture will be compatible with all images and thus become more difficult to distinguish and perceive. The purpose of copy-move on the image can vary. General is generally done for the addition of objects or to an object in an image.*

*This final project proposes a copy-move counterfeiting detection system on the image using the Histogram of Oriented Gradients (HOG) method. The first image data is divided into overlapping blocks then each block is extracted with HOG which produces the vector feature. Then copy-move detection is searched by calculating the distance between features of each block, blocks that have a zero distance value are areas experiencing copy-move.*

*Testing of 20 images from the CoMoFod database. Based on the results shows that this method can provide accurate copy-move detection with average accuracy, specificity and sensitivity is 99.59%, 99.91%, and 90.21%.*

***Keywords: Copy-move, Euclidean Distance, Histogram of Oriented Gradients, image forgery***

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

## KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa karena berkat rahmat-Nya sehingga penulis dapat menyelesaikan tugas akhir yang berjudul “**Implementasi Deteksi Copy-move Forgery pada Citra Menggunakan Metode Histogram of Oriented Gradients (HOG)**”.

Buku tugas akhir ini disusun dengan harapan dapat memberikan manfaat dalam bidang forensik terkait deteksi terhadap *copy-move* pada suatu citra. Selain itu, penulis berharap dapat memberikan kontribusi positif bagi kampus Teknik Informatika ITS.

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

1. Orang tua penulis Bapak I Wayan Sunata dan Ibu Ni Made Seriasih yang selalu memberikan dukungan moral, spiritual dan material serta senantiasa memberikan doa demi kelancaran dan kemudahan penulis dalam mengerjakan tugas akhir.
2. Ibu Dr. Eng. Nanik Suciati, S.Kom., M.Kom. selaku dosen pembimbing 1 sekaligus dosen wali penulis yang telah memberi ide, nasihat dan arahan sehingga penulis dapat menyelesaikan tugas akhir ini.
3. Bapak Arya Yudhi Wijaya, S.Kom., M.Kom. selaku dosen pembimbing 2 penulis yang telah memberi ide, nasihat dan arahan sehingga penulis dapat menyelesaikan tugas akhir ini.
4. Ibu Umi Laili Yuhana, S.Kom.,M.Sc. dan Ibu Diana Purwitasari, S.Kom.,M.Sc. selaku dosen wali penulis terdahulu yang telah memberikan banyak masukan dan nasihat terkait dengan perkuliahan penulis selama perkuliahan di Teknik Informatika ITS.
5. Alm. Kakek penulis, Nenek penulis, Kakak penulis, Kakek dan Nenek penulis di Surabaya dan seluruh keluarga besar

yang telah memberikan dukungan yang besar baik secara langsung maupun tidak langsung.

6. Tim pembimbing 3 Rizky Haqiqi, Ilham Gurat S.Kom, Lophita Y N. yang telah banyak memberikan bantuan kepada penulis selama mengerjakan tugas akhir ini.
7. Teman-teman di Lab KCV: para admin yang telah banyak membantu memfasilitasi dalam pengerjaan serta teman-teman yang selalu ada di Lab KCV, yang sudah menemani penulis.
8. Teman-teman seperjuangan yang selalu memberi motivasi kepada penulis: Lophita, Gurat, Nela, Ayu, Devira, Ekky, Haqiqi, Nuning, Nida, Ine, Sani, Novita, Didit, Reza, Ihsan. Dan teman-teman TC angkatan 2013 yang selalu memberikan pengalaman baru bagi penulis.
9. Teman-teman bermain dan seperantauan Ayu Khrisna, Desak Pratiwi, Alit Juliani.
10. Pihak-pihak lain yang tidak bisa penulis sebutkan satu-persatu.

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

Surabaya, Juli 2017

Penulis

## DAFTAR ISI

LEMBAR PENGESAHAN.....	<b>Error! Bookmark not defined.</b>
<i>Abstrak</i>	vii
<i>Abstract</i> .....	ix
KATA PENGANTAR .....	xi
DAFTAR ISI.....	xiii
DAFTAR GAMBAR .....	xvii
DAFTAR TABEL .....	xxi
DAFTAR KODE SUMBER .....	xxiii
BAB I PENDAHULUAN .....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah .....	2
1.4 Tujuan Tugas Akhir .....	3
1.5 Manfaat Tugas Akhir .....	3
1.6 Metodologi .....	3
1.7 Sistematika Laporan.....	4
BAB II DASAR TEORI.....	7
2.1 Citra Digital .....	7
2.2 Citra Biner .....	7
2.3 Citra <i>Grayscale</i> (Skala Keabuan) .....	8
2.4 Citra Berwarna .....	8
2.5 Forensik Citra Digital.....	9
2.6 <i>Copy-move</i> .....	9
2.7 Histogram of Oriented Gradients .....	10
2.8 Euclidean Distance.....	15
2.9 Matlab .....	15
2.10 Confusion Matrix.....	16
BAB III PERANCANGAN PERANGKAT LUNAK .....	19
3.1 Desain Metode Secara Umum .....	19
3.2 Data.....	20
3.2.1 Data Masukan .....	20
3.2.2 Data Keluaran .....	22

3.3	Perancangan Proses.....	22
3.3.1	<i>Preprocessing</i> .....	22
3.3.2	Membagi Citra ke dalam Blok-blok yang <i>Overlap</i> .....	23
3.3.3	Ekstraksi Fitur dengan HOG.....	23
3.3.4	Perhitungan Jarak menggunakan <i>Euclidean Distance</i> .....	25
3.3.5	Pencarian Daerah yang Mengalami <i>Copy-move</i> .....	25
BAB IV IMPLEMENTASI.....		27
4.1	Lingkungan Implementasi.....	27
4.2	Implementasi .....	27
4.2.1	Implementasi Tahap <i>Preprocessing</i> .....	28
4.2.2	Implementasi Ekstraksi Fitur dengan HOG.....	28
4.2.3	Implementasi Perhitungan Jarak menggunakan <i>Euclidean Distance</i> .....	29
4.2.4	Implementasi Pencarian Daerah yang Mengalami <i>Copy-move</i> 30	
BAB V UJI COBA DAN EVALUASI.....		33
5.1	Lingkungan Uji Coba.....	33
5.2	Data Uji Coba.....	33
5.3	Hasil Uji Coba Citra Setiap Proses .....	44
5.3.1	Hasil Uji Coba <i>Preprocessing</i> .....	45
5.3.2	Hasil Uji Coba Pembagian ke dalam Blok-blok yang Saling <i>Overlap</i> .....	46
5.3.3	Hasil Uji Coba Tahap Ekstraksi Fitur dengan HOG .....	47
5.3.4	Hasil Uji Coba Tahap Perhitungan Jarak dengan <i>Euclidean Distance</i> .....	48
5.3.5	Hasil Uji Coba Tahap Pencarian Dearah yang Mengalami <i>Copy-move</i> .....	48
5.4	Skenario Uji Coba.....	49
5.4.1	Skenario Uji Coba Ukuran Blok Berbeda.....	49
5.4.2	Skenario Uji Coba dengan Variasi Data.....	57
5.5	Evaluasi.....	62
5.5.1	Evaluasi Uji Coba dengan Ukuran Blok Berbeda .....	62
5.5.2	Evaluasi Uji Coba dengan Data Variasi .....	63



BAB VI KESIMPULAN DAN SARAN .....	65
6.1 Kesimpulan.....	65
6.2 Saran .....	65
DAFTAR PUSTAKA .....	67
LAMPIRAN .....	69
BIODATA PENULIS .....	80

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

## DAFTAR GAMBAR

Gambar 2.1 Contoh Citra Biner .....	7
Gambar 2.2 Contoh Citra Keabuan.....	8
Gambar 2.3 Contoh Citra Berwarna[6].....	9
Gambar 2.4 Contoh Citra Copy-Move[6] .....	10
Gambar 2.5 Skema Algoritma HOG[5] .....	11
Gambar 2.7 Citra Setelah Dihitung Besar Gradiennya .....	13
Gambar 2.8 Panjang Bin Histogram[14].....	14
Gambar 2.9 Logo Matlab[15].....	16
Gambar 3.1 Diagram Alir Sistem Secara Umum .....	20
Gambar 3.2 Contoh Data Masukan yang Mengalami Copy-move[6] .....	21
Gambar 3.3 Contoh Data Masukan Ground Truth dari Citra yang Mengalami Copy-move[6] .....	21
Gambar 3.4 Data Citra Keluaran yang Diharapkan .....	22
Gambar 3.5 Diagram Alir Tahap Preprocessing.....	23
Gambar 3.6 Diagram Alir Proses HOG .....	24
Gambar 3.7 Diagram Alir Proses Pencarian Daerah <i>Copy-move</i>	26
Gambar 5.1 (a) Citra Copy-move 001_F.png, (b) Ground Truth	34
Gambar 5.2 (a) Citra Copy-move 002_F.png, (b) Ground Truth	34
Gambar 5.3 (a) Citra Copy-move 003_F.png, (b) Ground Truth	35
Gambar 5.4 (a) Citra Copy-move 004_F.png, (b) Ground Truth	35
Gambar 5.5 (a) Citra Copy-move 005_F.png, (b) Ground Truth	36
Gambar 5.6 (a) Citra Copy-move 006_F.png, (b) Ground Truth	36
Gambar 5.7 (a) Citra Copy-move 007_F.png, (b) Ground Truth	37
Gambar 5.8 (a) Citra Copy-move 008_F.png, (b) Ground Truth	37
Gambar 5.9 (a) Citra Copy-move 009_F.png, (b) Ground Truth	38
Gambar 5.10 (a) Citra Copy-move 010_F.png, (b) Ground Truth .....	38
Gambar 5.11 (a) Citra Copy-move 011_F.png, (b) Ground Truth .....	39
Gambar 5.12 (a) Citra Copy-move 012_F.png, (b) Ground Truth .....	39
Gambar 5.13 (a) Citra Copy-move 013_F.png, (b) Ground Truth .....	40

Gambar 5.14 (a) Citra Copy-move 014_F.png, (b) Ground Truth .....	40
Gambar 5.15 (a) Citra Copy-move 015_F.png, (b) Ground Truth .....	41
Gambar 5.16 (a) Citra Copy-move 016_F.png, (b) Ground Truth .....	41
Gambar 5.17 (a) Citra Copy-move 017_F.png, (b) Ground Truth .....	42
Gambar 5.18 (a) Citra Copy-move 018_F.png, (b) Ground Truth .....	42
Gambar 5.19 (a) Citra Copy-move 019_F.png, (b) Ground Truth .....	43
Gambar 5.20 (a) Citra Copy-move 020_F.png, (b) Ground Truth .....	43
Gambar 5.21 Citra Copy-move Asli (RGB).....	45
Gambar 5.22 Citra Grayscale Hasil <i>Preprocessing</i> .....	45
Gambar 5.23 Citra yang Telah Dibagi Menjadi Blok-blok.....	46
Gambar 5.24 Nilai Matriks Citra untuk 1 blok berukuran 8x8....	46
Gambar 5.25 Fitur yang Tebentuk dari Satu Blok.....	47
Gambar 5.26 Visualisasi Fitur HOG untuk Satu Blok .....	47
Gambar 5.27 Matrix ysng Menyimpan Jarak serta Koordinat Blok .....	48
Gambar 5.28 Citra Hasil Deteksi.....	49
Gambar 5.29 Contoh Citra Hasil Deteksi dengan Sensitifity 100% pada Uji Coba Parameter Blok 8x8 (a) Citra Masukan (b) Citra Hasil Deteksi .....	<b>Error! Bookmark not defined.</b>
Gambar 5.30 Contoh Citra Hasil Deteksi dengan Sensitifity 87,12% pada Uji Coba Parameter Blok 8x8 (a) Citra Masukan (b) Citra Hasil Deteksi.....	52
Gambar 5.31 Contoh Citra Hasil Deteksi dengan Sensitifity 100% pada Uji Coba Parameter Blok 16x16 (a) Citra Masukan (b) Citra Hasil Deteksi .....	54
Gambar 5.32 Contoh Citra Hasil Deteksi dengan Sensitifity 0% pada Uji Coba Parameter Blok 16x16 (a) Citra Masukan (b) Citra Hasil Deteksi .....	54

Gambar 5.33 Contoh Citra Hasil Deteksi dengan Sensitivity 100% pada Uji Coba Parameter Blok 32x32 (a) Citra Masukan (b) Citra Hasil Deteksi .....	56
Gambar 5.34 Contoh Citra Hasil Deteksi dengan Sensitivity 0% pada Uji Coba Parameter Blok 32x32 (a) Citra Masukan (b) Citra Hasil Deteksi .....	57
Gambar 5.35 Contoh Citra Hasil Deteksi dengan Sensitivity 100% pada Uji Coba dengan Citra Penyesuaian Kontras (a) Citra Masukan (b) Citra Hasil Deteksi .....	59
Gambar 5.36 Contoh Citra Hasil Deteksi dengan Sensitivity 0% pada Uji Coba dengan Citra Penyesuaian Kontras (a) Citra Masukan (b) Citra Hasil Deteksi .....	59
Gambar 5.37 Contoh Citra Hasil Deteksi dengan <i>Sensitivity</i> 100% pada Uji Coba dengan <i>Image Blurring</i> (a) Citra Masukan (b) Citra Hasil Deteksi .....	61
Gambar 5.38 Contoh Citra Hasil Deteksi dengan Sensitivity 0% pada Uji Coba dengan <i>Image Blurring</i> (a) Citra Masukan (b) Citra Hasil Deteksi .....	62

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

## DAFTAR TABEL

Tabel 2.1 <i>Confusion Matrix</i> Dua Kelas.....	16
Tabel 4.1 Lingkungan Implementasi Perangkat Keras dan Perangkat Lunak .....	27
Tabel 5.1 Spesifikasi Lingkungan Uji Coba.....	33
Tabel 5.2 Detail Dataset Citra Copy-move .....	44
Tabel 5.3 Hasil Skenario Uji Coba Performa dengan Ukuran Blok 8x8 .....	51
Tabel 5.4 Hasil Skenario Uji Coba Performa dengan Ukuran Blok 16x16 .....	53
Tabel 5.5 Hasil Skenario Uji Coba Performa dengan Ukuran Blok 32x32 .....	55
Tabel 5.6 Hasil Skenario Uji Coba Performa dengan Citra Penyesuaian Kontras ( <i>Contrast Adjustments</i> ) .....	58
Tabel 5.7 Hasil Skenario Uji Coba Performa dengan Citra Penyesuaian Kontras ( <i>Image Blurring</i> ).....	60

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



## DAFTAR KODE SUMBER

Kode Sumber 4.1 Implementasi Tahap <i>Pre-processing</i> .....	28
Kode Sumber 4.2 Implementasi Proses Ekstraksi Fitur dengan HOG.....	28
Kode Sumber 4.3 Implementasi Perhitungan Jarak Menggunakan Euclidean Distance .....	30
Kode Sumber 4.4 Implementasi Pencarian Dearah yang Mengalami Copy-move.....	31

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

# BAB I

## PENDAHULUAN

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

### 1.1 Latar Belakang

Semakin berkembangnya teknologi informasi menyebabkan semakin banyaknya aplikasi yang diciptakan untuk memenuhi kegiatan baik untuk pembelajaran, pekerjaan, maupun hiburan. Salah satu aplikasi yang banyak dikembangkan yaitu aplikasi pengolahan citra. Dengan banyaknya aplikasi pengolahan citra tersebut, maka akan semakin mudah dilakukan perubahan wilayah-wilayah tertentu pada citra, sehingga banyak tersebar citra rekayasa dan hasil manipulasi/tidak asli. Jika hal ini terjadi secara terus-menerus akan menimbulkan efek yang tidak baik, terutama jika kegiatan ini dilakukan dengan tujuan yang negatif akan ada pihak tertentu yang dirugikan.

Pemalsuan citra ini terdiri dari bermacam-macam jenis yang salah satunya adalah melakukan duplikasi suatu bagian gambar kemudian ditempatkan pada bagian lain di gambar tersebut. Hal ini disebut sebagai *copy-move forgery* pada citra. *Copy-move* merupakan bentuk yang paling umum dari pemalsuan citra digital [1]. Karena bagian yang diduplikasi berasal dari citra yang sama, maka property-properti seperti *noise*, warna dan tekstur akan kompatibel dengan seluruh citra dan dengan demikian menjadi lebih sulit untuk dibedakan dan deteksi secara kasat mata [2]. Tujuan dilakukannya *copy-move* pada citra dapat bermacam-macam. Tetapi pada umumnya dilakukan untuk penambahan objek atau untuk menyembunyikan suatu objek pada suatu citra[3].

Berdasarkan hal tersebut, maka diperlukan suatu sistem yang dapat mendeteksi kepalsuan pada citra. Sistem deteksi ini dibuat dengan menggunakan metode *Histogram of Oriented Gradients*

(HOG). HOG pertama kali dikembangkan oleh Dalal dkk. [4]. sebagai deskriptor fitur yang kuat untuk mendeteksi objek dalam sistem visi komputer, yang terbukti sangat efektif dalam mendeteksi pejalan kaki [5]. Kinerja dari metode ini berkaitan dengan akurasi deteksi dan kompleksitas komputasi [1].

Pembuatan sistem deteksi ini diharapkan dapat mencapai hasil yang maksimal dan bermanfaat bagi pihak yang membutuhkan khususnya terkait dengan deteksi *copy-move forgery* pada citra.

## 1.2 Rumusan Masalah

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

1. Bagaimana tahap *preprocessing* dalam melakukan implementasi deteksi *copy-move* pada citra?
2. Bagaimana melakukan ekstraksi fitur pada citra menggunakan metode *Histogram of Oriented Gradients*?
3. Bagaimana melakukan pencarian daerah yang mengalami *copy-move* pada citra?
4. Bagaimana mengetahui performa yang dari proses deteksi *copy-move* pemalsuan citra?

## 1.3 Batasan Masalah

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

1. Aplikasi dibangun dengan menggunakan MATLAB.
2. Dataset yang digunakan adalah dataset citra dengan resolusi 512x512 piksel dari CoMoFod *database* University of Zagreb, Faculty of Electrical Engineering and Computing, Department of Wireless Communications, Zagreb, Croatia. Dataset diunduh dari *website* Video Communications Laboratory (VCL) [6].
3. Dataset citra yang digunakan sebanyak 20 citra.
4. Citra yang diuji merupakan citra yang telah mengalami pemalsuan *copy-move*.

## 1.4 Tujuan Tugas Akhir

Tujuan tugas akhir ini adalah untuk mengembangkan sistem deteksi *copy-move forgery* atau pemalsuan berupa *copy-move* pada citra dengan menggunakan metode *Histogram of Oriented Gradients* (HOG).

## 1.5 Manfaat Tugas Akhir

Manfaat dari tugas akhir ini adalah menghasilkan sistem untuk mendeteksi pemalsuan berupa *copy-move* pada citra dengan menggunakan metode *Histogram of Oriented Gradients* yang dapat memberikan kontribusi pada pengolahan citra terutama pada bidang forensik digital.

## 1.6 Metodologi

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

### 1. Penyusunan Proposal Tugas Akhir

Pada tahap ini, dilakukan penyusunan proposal tugas akhir. Proposal berisi deskripsi dan gambaran awal tugas akhir yang akan dibuat. Proposal ini juga berisi garis besar proses tahapan penyusunan tugas akhir ini.

### 2. Studi Literatur

Pada studi literatur, dilakukan pengumpulan data dan studi terhadap sejumlah referensi yang diperlukan dalam pengerjaan tugas akhir. Referensi tersebut didapatkan dari beberapa artikel yang dipublikasikan oleh jurnal. Selain dari artikel, studi literatur juga dilakukan melalui pencarian referensi dari internet yang membahas mengenai informasi yang dibutuhkan, seperti *copy-move*, metode *Histogram of Oriented Gradients* dan beberapa metode lain yang dapat mendukung pada setiap tahapannya.

### 3. Analisis dan Desain Perangkat Lunak

Pada tahap ini disusun rancang bangun dari perangkat lunak yang dibangun. Pengguna memilih citra yang telah mengalami *copy-move* dengan ukuran 512 x 512 piksel sebagai data masukan. Kemudian, sistem akan memproses citra masukan

dengan melakukan *preprocessing*, ekstraksi fitur dengan HOG deskriptor, pencocokan blok-blok yang memiliki kemiripan fitur dan melakukan perhitungan performa. Setelah proses selesai, sistem akan menampilkan hasil berupa daerah citra yang mengalami *copy-move*.

#### 4. Implementasi Perangkat Lunak

Sistem deteksi *copy-move* ini akan dibuat dengan bahasa pemrograman MATLAB menggunakan kaskas bantu IDE MATLAB 8.3 (R2014a) pada platform *desktop*. Dan digunakan kaskas bantu lain sebagai pendukung yang diantaranya adalah *Microsoft Excel* sebagai pengolah angka dan *Notepad* untuk membuat nama file data citra masukan.

#### 5. Uji Coba dan Evaluasi

Dalam tahap ini, dilakukan pengujian parameter-parameter yang dibutuhkan pada proses ekstraksi fitur dan pencocokan.

## 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 deteksi *copy-move* pada citra.

#### **Bab IV Implementasi**

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

#### **Bab V Uji Coba dan Evaluasi**

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

#### **Bab VI Kesimpulan dan Saran**

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

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



## **BAB II**

### **DASAR TEORI**

Pada bab ini diuraikan mengenai dasar-dasar teori yang digunakan dalam pengerjaan tugas akhir dengan tujuan untuk memberikan gambaran secara umum terhadap penelitian yang dikerjakan. Bab ini berisi penjelasan mengenai *copy-move*, metode *Histogram of Oriented Gradients*, dan teori-teori lain yang mendukung pembuatan tugas akhir ini.

#### **2.1 Citra Digital**

Citra digital merupakan sebuah larik (*array*) yang berisi nilai-nilai real maupun kompleks yang direpresentasikan dengan deretan tertentu. Suatu citra dapat didefinisikan sebagai fungsi  $f(x,y)$  berukuran  $M$  baris dan  $N$  kolom, dengan  $x$  dan  $y$  adalah koordinat spasial, dan amplitudo  $f$  di titik koordinat  $(x,y)$  dinamakan intensitas atau tingkat keabuan dari citra pada titik tersebut. Apabila nilai  $x$ ,  $y$ , dan nilai amplitudo  $f$  secara keseluruhan barhingga (*finite*) dan bernilai diskrit maka dapat dikatakan bahwa citra tersebut adalah citra digital [7].

#### **2.2 Citra Biner**

Citra biner adalah citra yang hanya memiliki dua nilai derajat keabuan yaitu hitam dan putih. Pikel yang bernilai 1 melambangkan warna hitam dan piksel yang bernilai 0 melambangkan warna putih[8].



**Gambar 2.1**Contoh Citra Biner

Gambar 2.1 menunjukkan contoh citra biner. Citra biner sering sekali muncul sebagai hasil dari proses pengolahan, seperti segmentasi, deteksi, morfologi ataupun *dithering*. Fungsi dari binerisasi sendiri adalah untuk mempermudah proses pengenalan pola, karena pola akan lebih mudah terdeteksi pada citra yang mengandung lebih sedikit warna.

### 2.3 Citra *Grayscale* (Skala Keabuan)

Citra grayscale merupakan citra yang terdiri dari satu layer warna dengan derajat keabuan tertentu. Kebanyakan citra digital 8-bit, maka sistem *grey-scale* diukur berdasarkan skala intensitas kecerahan yang bernilai 0 – 255, dimana 0 merupakan hitam pekat dan 255 merupakan yang terputih[8]. Gambar 2.2 merupakan contoh citra yang memiliki skala keabuan.



**Gambar 2.2 Contoh Citra Keabuan**

### 2.4 Citra Berwarna

RGB adalah suatu model warna yang terdiri dari merah, hijau, dan biru, digabungkan dalam membentuk suatu susunan warna yang luas. Setiap warna dasar, misalnya merah, dapat diberi rentang-nilai. Untuk monitor komputer, nilai rentangnya paling

kecil = 0 dan paling besar = 255. Pilihan skala 256 ini didasarkan pada cara mengungkap 8 digit bilangan biner yang digunakan oleh mesin komputer. Dengan cara ini, akan diperoleh warna campuran sebanyak  $256 \times 256 \times 256 = 16.777.216$  jenis warna [8]. Contoh citra berwarna ditunjukkan pada Gambar 2.3.



**Gambar 2.3 Contoh Citra Berwarna[6]**

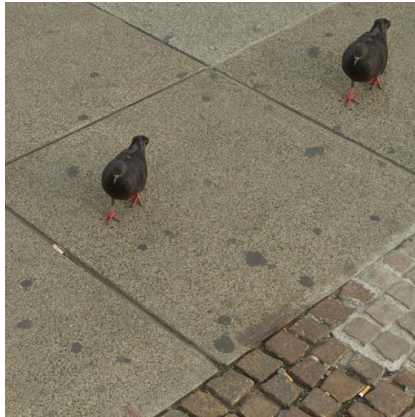
## **2.5 Forensik Citra Digital**

Forensik Citra Digital adalah bidang penelitian baru yang bertujuan untuk memvalidasi keaslian suatu gambar dengan memulihkan informasi tentang sejarahnya. Dua masalah utama yang dihadapi adalah identifikasi perangkat pencitraan yang dapat mengambil gambar dan deteksi jejak pemalsuan[9]. Teknik memvalidasi gambar dapat dikelompokkan menjadi dua metode yaitu, metode aktif dan metode pasif. Metode aktif seperti *watermarking* sedangkan metode pasif adalah kasus dimana informasi di dalam suatu citra sulit untuk dipecahkan seperti kasus *copy-move*[10].

## **2.6 Copy-move**

Teknik *copy-move* merupakan salah satu teknik pemalsuan citra dengan dengan cara mengambil mengambil objek tertentu dari

citra asli dan menambahkannya pada citra tersebut dengan tujuan untuk menambah jumlah atau merubah objek yang sama pada citra asli [11]. *Copy-move forgery* mudah dilakukan dan relatif efektif dalam manipulasi gambar. Hal ini terutama ketika kedua sumber dan daerah target berasal dari gambar yang memiliki *property* yang sama seperti warna, pencahayaan dan noise yang cocok dengan wilayah gambar yang akan dimanipulasi [12]. Gambar 2.4 menunjukkan contoh citra yang mengalami *copy-move*. Pada Gambar 2.4 daerah citra yang mengalami *copy-move* ada objek burung yang di tempelkan sebanyak satu kali.



**Gambar 2.4 Contoh Citra *Copy-Move*[6]**

## **2.7 Histogram of Oriented Gradients**

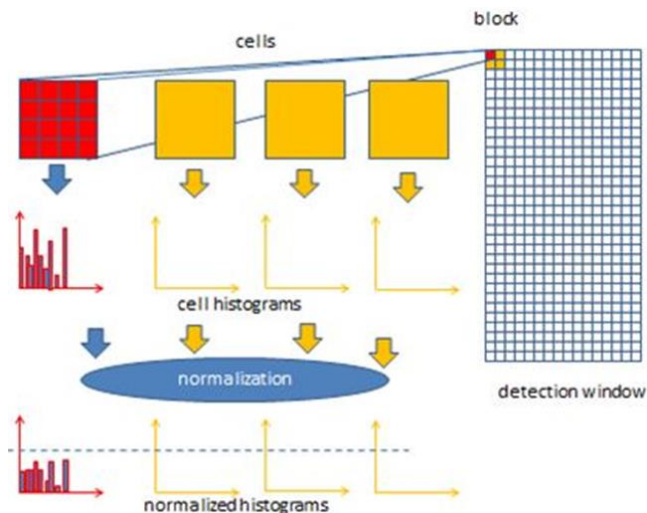
*Histogram of Oriented Gradients* (HOG) adalah sebuah fitur deskriptor yang digunakan untuk mendeteksi benda-benda dalam visi komputer dan pengolahan citra. Teknik HOG deskriptor menghitung kejadian orientasi gradien dalam porsi lokal dari *image-detection window* atau *Region of Interest* (ROI).

HOG pertama kali dikembangkan oleh Dalal dkk. sebagai deskriptor fitur yang kuat untuk mendeteksi objek dalam sistem visi komputer, yang terbukti sangat efektif dalam mendeteksi

pejalan kaki. Implementasi algoritma deskriptor HOG adalah sebagai berikut:

- Membagi citra menjadi wilayah kecil yang terhubung yang disebut sel, dan untuk setiap sel menghitung histogram dari arah gradien atau orientasi tepi untuk piksel dalam sel.
- Discretize setiap sel dalam sudut bin sesuai dengan orientasi gradien.
- Piksel setiap sel kontribusi gradien menurut pada bin sudut yang sesuai.
- Kelompok sel yang berdekatan dianggap sebagai daerah spasial yang disebut blok. Pengelompokan sel ke dalam blok adalah dasar untuk pengelompokan dan normalisasi histogram.
- Kelompok normalisasi histogram merupakan blok histogram. Set blok histogram tersebut merupakan descriptor [5].

Gambar 2.5 menunjukkan skema implementasi algoritma HOG dari pembagian sel, blok, hingga proses normalisasi histogram.



**Gambar 2.5 Skema Algoritma HOG[5]**

Metode HOG banyak digunakan pada *computer vision*. 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 ke bentuk daerah kecil yang disebut sel. Setiap sel disusun dari sebuah histogram dari gradien, kombinasi dari histogram ini dijadikan sebagai deskriptor yang mewakili sebuah obyek [13].

HOG terdiri dari pengelompokan informasi gradien (sel dan blok), maupun histogram yang berorientasi gradien. Fitur descriptor HOG dari orientasi histogram dapat dihitung sebagai berikut:

#### 1. Menghitung Gradien Citra

Operator gradien sederhana  $[-1, 0, 1]$  diterapkan untuk menghitung gradien citra setiap titik di sepanjang dua arah. Gradien pada suatu titik  $(x, y)$  di citra  $I$  dapat ditemukan dengan perhitungan konvolusi pada operator gradien dengan citra. Persamaan (2.1) menunjukkan perhitungan konvolusi untuk gradien pada arah  $x$  dan Persamaan (2.2) menunjukkan perhitungan konvolusi pada arah  $y$ .

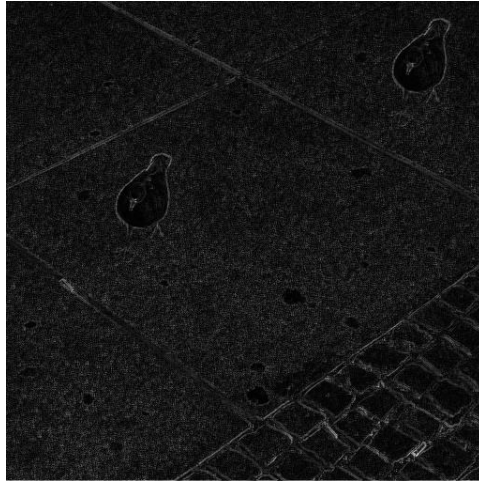
$$G_x = [-1, 0, 1] * I(x, y) \quad (2.1)$$

dan

$$G_y = [-1, 0, 1] T * I(x, y) \quad (2.2)$$

Besaran gradien pada suatu titik  $(x, y)$  ditentukan pada Persamaan (2.3). Gambar 2.6 menunjukkan gambar setelah perhitungan besaran gradien.

$$G(x, y) = \sqrt{G_x(x, y)^2 + G_y(x, y)^2} \quad (2.3)$$



**Gambar 2.6 Citra Setelah Dihitung Besar Gradiennya**

Setelah mendapatkan besaran gradient, kemudian gradient ditransformasikan ke dalam koordinat sumbu dengan sudut diantara  $0^\circ$  sampai  $180^\circ$  yang disebut dengan orientasi gradient. Orientasi dari suatu titik  $(x,y)$  ditentukan dengan Persamaan (2.4).

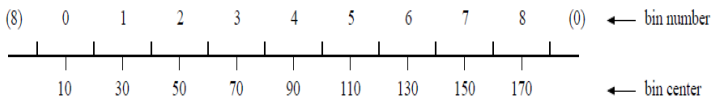
$$\theta(x, y) = \tan^{-1} \left( \frac{G_y(x, y)}{G_x(x, y)} \right) \quad (2.4)$$

2. Menghitung Histogram dari Orientasi Gradien Setiap Sel  
Citra dibagi ke dalam sel-sel yang berukuran  $8 \times 8$  piksel. Kemudian setiap sel dihitung nilai histogramnya. Bin yang dipakai dalam histogram adalah 9 bin. Penentuan nilai histogram dilakukan melalui *voting* terhadap masing-masing sel pada citra. Gambar 2.7 merupakan gambar 9 bin histogram yang digunakan pada voting untuk menentukan kontribusi nilai histogram.  
Kontribusi nilai histogram ( $v$ ) dapat diperoleh dari Persamaan (2.5) dan Persamaan (2.6) dimana  $\mu$  merupakan

besar gradien pada piksel,  $c$  merupakan nilai tengah sudut pada bin,  $\theta$  adalah sudut orientasi gradient pada piksel,  $B$  merupakan panjang bin histogram,  $w$  adalah lebar dari nilai tengah sudut dimana  $w = \frac{180}{B}$  [14].

$$v_j = \mu \frac{c_j + 1 - \theta}{w} \text{ untuk bin ke } j = \left\lfloor \frac{\theta}{w} - \frac{1}{2} \right\rfloor \bmod B \quad (2.5)$$

$$v_{j+1} = \mu \frac{\theta - c_j}{w} \text{ untuk bin ke } (j+1) \bmod B \quad (2.6)$$



**Gambar 2.7 Panjang Bin Histogram**[14]

### 3. Normalisasi Fitur Blok

Citra dibagi ke dalam blok-blok yang berukuran  $2 \times 2$  sel, satu blok menggambarkan satu fitur blok yang terdiri dari kumpulan nilai histogram sel yang membentuk blok. Persamaan (2.7) merupakan rumus untuk melakukan normalisasi pada fitur blok. Variabel  $b$  merupakan nilai fitur blok dan  $e$  merupakan bilangan positif yang bernilai kecil untuk menghindari pembagian dengan 0.

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

### 4. Normalisasi Fitur HOG

Pengumpulan deskriptor HOG dari semua blok dalam wilayah deteksi untuk digabungkan menjadi fitur vektor untuk digunakan dalam deteksi [1]. Persamaan (2.8) merupakan rumus untuk normalisasi pada fitur vektor. Variable  $h$  merupakan nilai fitur vektor dan  $e$  merupakan



bilangan positif yang bernilai kecil untuk menghindari pembagian dengan 0.

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

## 2.8 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 [13].

Nilai *Euclidean distance* didapatkan dari Persamaan (2.9) dimana variabel  $p$  dan  $q$  merupakan dua titik yang akan dihitung jaraknya, sedangkan  $d$  menunjukkan nilai jarak dari kedua titik yang dihitung.

$$d(p, q) = d(q, p) = \sqrt{\sum_{i=1}^n (q_i - p_i)^2} \quad (2.9)$$

## 2.9 Matlab

Matlab (*Matrix Laboratory*) adalah sebuah lingkungan komputasi numerikal dan bahasa pemrograman komputer generasi keempat. Dikembangkan oleh *The MathWorks*, Matlab memungkinkan manipulasi matriks, pem-plot-an fungsi dan data, implementasi algoritma, pembuatan antarmuka pengguna, dan peng-antarmuka-an dengan program dalam bahasa lainnya. Matlab digunakan untuk *machine learning*, pemrosesan sinyal, pengolahan gambar, visi komputer, komunikasi, desain kontrol, robotika, dan masih banyak lagi [15]. Gambar 2.8 merupakan logo Matlab.



Gambar 2.8 Logo Matlab[15]

## 2.10 Confusion Matrix

*Confusion matrix* adalah suatu metode yang dapat digunakan untuk mengukur performa suatu algoritma pada konsep data mining. *Confusion matrix* berisi informasi tentang klasifikasi aktual dan prediksi yang dilakukan oleh system klasifikasi[16].

Pada tugas akhir ini, perhitungan performa dilakukan dengan cara membandingkan citra *ground truth* dengan citra hasil deteksi. *Confusion matrix* dapat digambarkan berupa tabel seperti disajikan pada Tabel 2.1.

**Tabel 2.1 *Confusion Matrix* Dua Kelas**

Prediksi	1	0
Aktual		
1	TP	FN
0	FP	TN

Berikut merupakan penjelasan dari tabel Tabel 2.1:

1. Kondisi TP (*True Positive*) yaitu area *copy-move* yang terdeteksi sebagai area *copy-move*.
2. Kondisi FP (*False Positive*) yaitu area non-*copy-move* yang terdeteksi sebagai area *copy-move*.
3. Kondisi FN (*False Negative*) yaitu area *copy-move* yang terdeteksi sebagai area non-*copy-move*.
4. Kondisi TN (*True Negative*) yaitu area non-*copy-move* yang terdeteksi sebagai area non-*copy-move*.

Dari *Confusion Matrix* bisa didapatkan berbagai informasi mengenai performa, diantaranya akurasi, *specificity*, dan *sensitivity*.

Akurasi merupakan hasil bagi dari jumlah prediksi yang terklasifikasi secara benar dibagi dengan total data yang diklasifikasi. Rumus perhitungan akurasi ditunjukkan pada Persamaan (2.10)

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN} \quad (2.10)$$

*Specificity* adalah jumlah proporsi data yang prediksi kelas hasil klasifikasinya salah dari total data yang berkelas salah pada *ground truth*. Rumus perhitungan *specificity* ditunjukkan pada Persamaan (2.11)

$$Specificity = \frac{TN}{FP + TN} \quad (2.11)$$

*Sensitivity* adalah jumlah proporsi data yang prediksi kelas hasil klasifikasinya benar dari total data yang berkelas benar pada *ground truth*. Rumus perhitungan *sensitivity* ditunjukkan pada Persamaan (2.12)

$$Sensitivity = \frac{TP}{TP + FN} \quad (2.12)$$

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

## **BAB III**

### **PERANCANGAN PERANGKAT LUNAK**

Pada bab ini akan dijelaskan mengenai perancangan aplikasi yang akan dibangun. Perancangan yang dijelaskan meliputi data dan proses. Data yang dimaksud berupa data yang akan diolah dalam pembuatan Tugas Akhir ini. Dijelaskan juga desain metode secara umum.

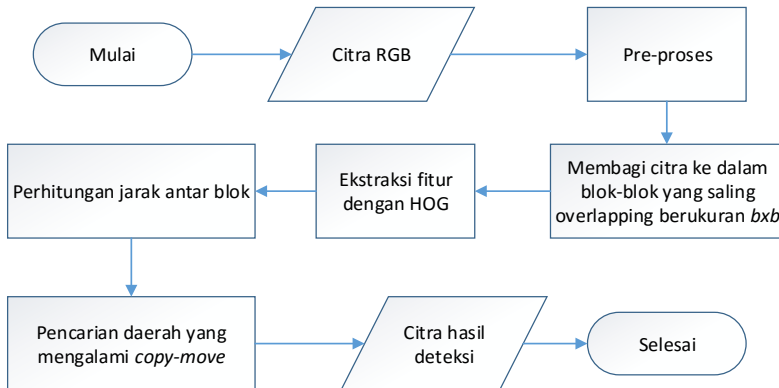
#### **3.1 Desain Metode Secara Umum**

Tugas akhir ini membangun aplikasi untuk melakukan deteksi *copy-move* pada citra. Data masukan yang digunakan adalah data citra dengan resolusi 512x512 piksel. Dataset diambil dari CoMoFod *database*. Data keluaran dari aplikasi merupakan citra biner (hitam putih) yang merepresentasikan hasil deteksi daerah yang mengalami *copy-move*. Metode yang digunakan untuk membangun aplikasi ini adalah *Histogram of Oriented Gradients* (HOG).

Data citra pertama kali dilakukan *preprocessing* dengan mengubahnya menjadi keabuan, kemudian membagi citra ke dalam blok-blok yang saling *overlap* yang berukuran *bxb*. Selanjutnya melakukan proses ekstraksi fitur pada setiap blok dengan metode HOG. Langkah pertama metode HOG yaitu melakukan perhitungan gradien dengan konvolusi dan membaginya ke dalam sel yang berukuran tetap. Kemudian melakukan perhitungan nilai histogram setiap sel. Satu sel terdiri dari satu histogram yang terdiri dari 9 bin. Selanjutnya membentuk blok-blok yang *overlap* atau tumpang tindih. Setiap blok berukuran 2x2 sel. Selanjutnya dilakukan normalisasi terhadap fitur blok. Kemudian fitur-fitur blok digabungkan menjadi satu menjadi sebuah fitur HOG. Fitur HOG ini kembali dinormalisasi untuk mendapat kecerahan citra yang sama.

Setelah proses ekstraksi fitur dengan HOG selanjutnya adalah mengurutkan fitur pada setiap blok. Pengurutan ini dilakukan untuk mengurangi waktu komputasi. Setelah semua fitur

diurutkan kemudian melakukan perhitungan jarak antar blok. Kemudian dilakukan pencarian blok-blok yang memiliki kemiripan untuk lebih mudah mendapatkan hasil deteksi. Blok yang memiliki kemiripan memiliki nilai jarak 0 (nol). Blok yang di cocokkan adalah 5 tetangga terdekat. Alur proses secara umum ditunjukkan pada Gambar 3.1.



**Gambar 3.1 Diagram Alir Sistem Secara Umum**

## 3.2 Data

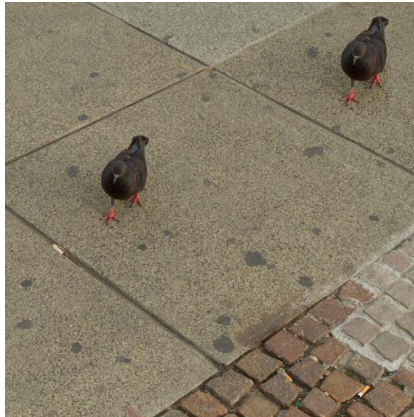
Pada sub bab ini akan dijelaskan data yang akan digunakan sebagai masukan dan selanjutnya diolah dan digunakan dalam pengujian aplikasi sehingga menghasilkan hasil yang diharapkan.

### 3.2.1 Data Masukan

Data masukan adalah suatu data yang digunakan sebagai masukan awal dari sistem. Data yang digunakan pada sistem ini berupa data citra yang mengalami *copy-move* dan citra *ground truth* dari citra yang mengalami *copy-move*. Data citra masukan yang mengalami *copy-move* ini merupakan citra RGB yang berukuran 512x512 piksel lalu dilakukan *preprocessing* yaitu konversi menjadi *grayscale* (keabuan). Kemudian dilakukan ekstraksi fitur dengan menggunakan metode *Histogram of*

*Oriented Gradients* (HOG). Sedangkan citra *ground truth* merupakan citra biner yang berukuran 512x512 piksel.

Data citra yang digunakan adalah citra yang telah mengalami *copy-move* berjumlah 20 buah citra dengan masing-masing *ground truth*-nya. Citra *copy-move* merupakan citra yang telah mengalami *copy-move* seperti ditunjukkan pada Gambar 3.2.



**Gambar 3.2 Contoh Data Masukan yang Mengalami *Copy-move*[6]**



**Gambar 3.3 Contoh Data Masukan *Ground Truth* dari Citra yang Mengalami *Copy-move*[6]**

Citra *ground truth* merupakan citra yang menggambarkan bagian daerah citra yang mengalami *copy-move*. Dimana daerah *copy-move* digambarkan berwarna putih sedangkan bagian yang tidak mengalami *copy-move* digambarkan berwarna hitam. Citra *ground truth* digambarkan pada Gambar 3.3.

### 3.2.2 Data Keluaran

Setelah data masukan diproses, maka akan menghasilkan data keluaran. Data keluaran ini berupa citra biner (hitam putih) yang menunjukkan daerah terjadinya *copy-move*. Gambar 3.4 menunjukkan citra keluaran yang diharapkan.



**Gambar 3.4 Data Citra Keluaran yang Diharapkan**

## 3.3 Perancangan Proses

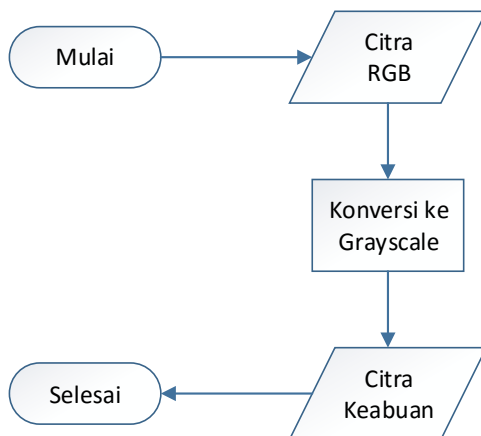
Perancangan proses dilakukan untuk memberikan gambaran mengenai setiap proses yang terdapat pada aplikasi deteksi *copy-move* pada citra.

### 3.3.1 *Preprocessing*

*Preprocessing* merupakan tahap awal dari proses deteksi *copy-move* ini. Tahap ini dilakukan agar citra masukan siap untuk diproses lebih lanjut.



Citra masukan disini berupa citra *Red Green Blue* (RGB) yang kemudian diubah menjadi *grayscale* atau keabuan. Citra RGB memiliki 3 dimensi yaitu dimensi merah, dimensi hijau, dan dimensi biru. Sedangkan setelah diubah menjadi keabuan citra hanya akan memiliki 2 dimensi. Diagram alir tahapan *preprocessing* ditunjukkan pada Gambar 3.5.



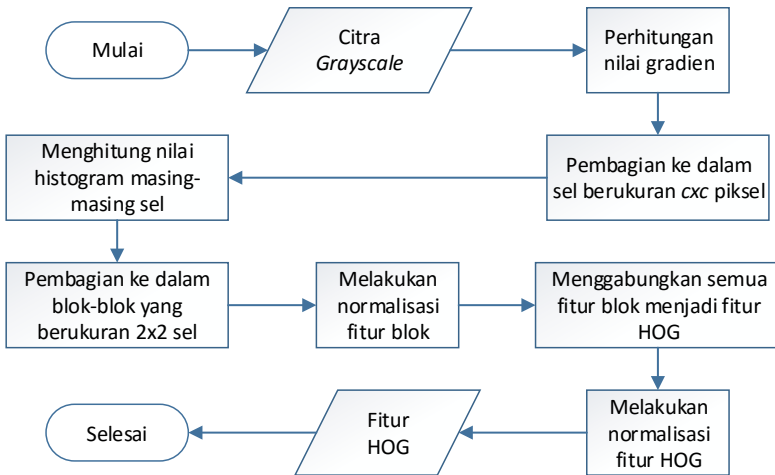
**Gambar 3.5 Diagram Alir Tahap Preprocessing**

### 3.3.2 Membagi Citra ke dalam Blok-blok yang *Overlap*

Untuk mengidentifikasi daerah *copy-move* pada citra masukan, citra hasil *preprocessing* dibagi ke dalam blok-blok yang saling *overlap* atau tumpang tindih yang berukuran  $b \times b$ . Bila citra *grayscale* berukuran  $M \times N$  maka akan ada sebanyak  $(M-b+1) \times (N-b+1)$  blok yang saling *overlap*. Blok-blok ini selanjutnya diekstraksi dengan metode HOG.

### 3.3.3 Ekstraksi Fitur dengan HOG

Ekstraksi fitur dilakukan untuk mendapatkan fitur vektor yang akan digunakan dalam proses deteksi dengan menghitung jarak kedekatan antar fitur. Kedekatan ini menandakan kemiripan antara fitur satu dengan yang lain.



**Gambar 3.6 Diagram Alir Proses HOG**

Ekstraksi fitur dalam tugas akhir ini menggunakan data keluaran *preprocessing*. Terdapat empat tahapan dalam melakukan ekstraksi fitur dengan *Histogram of Oriented Gradients* ini yaitu pertama melakukan perhitungan gradien untuk pada citra masukan. Perhitungan gradien dilakukan dengan cara konvolusi sesuai dengan Persamaan (2.1) dan Persamaan (2.2). Persamaan (2.3) untuk mendapatkan nilai gradien pada titik  $(x,y)$ . Kemudian melakukan transformasi gradien ke dalam koordinat sumbu dengan sudut diantara  $0^\circ$  sampai  $180^\circ$  yang disebut dengan orientasi gradient sesuai dengan Persamaan (2.4). Citra hasil perhitungan gradien ditunjukkan pada Gambar 2.6.

Kedua dilakukan pembagian citra ke dalam sel yang berukuran  $8 \times 8$  piksel. Kemudian citra dibagi lagi menjadi blok-blok yang berukuran  $2 \times 2$  sel. Ilustrasi sel yang membentuk blok disajikan pada Gambar 2.5.

Ketiga dilakukan perhitungan nilai histogram pada setiap sel. Proses ini dilakukan sesuai dengan Persamaan (2.5) dan

Persamaan (2.6). Fitur blok didapatkan dari rangkaian 4 histogram dari 4 sel yang menyusun blok. Kemudian dilakukan normalisasi pada setiap fitur blok yang telah didapatkan. Normalisasi dilakukan dengan *euclidean norm* sesuai dengan Persamaan (2.7).

Keempat dilakukan penggabungan dari hasil normalisasi semua fitur blok menjadi satu fitur HOG. Kemudian fitur HOG ini dinormalisasi lagi dengan *euclidean norm* sesuai dengan Persamaan (2.8). Gambaran tahap-tahap proses HOG ditunjukkan pada Gambar 3.6.

Hasil dari proses ekstraksi fitur dengan HOG ini berupa matriks yang berukuran  $(M-b+1) \times (N-b+1)$  baris dan 36 kolom. Angka 36 didapat dari jumlah fitur setiap blok. Karena satu blok terbentuk dari 4 sel, dan satu sel terbentuk dari sebuah histogram yang memiliki bin sebanyak 9. Nilai setiap bin inilah yang mewakili satu fitur.

### 3.3.4 Perhitungan Jarak menggunakan *Euclidean Distance*

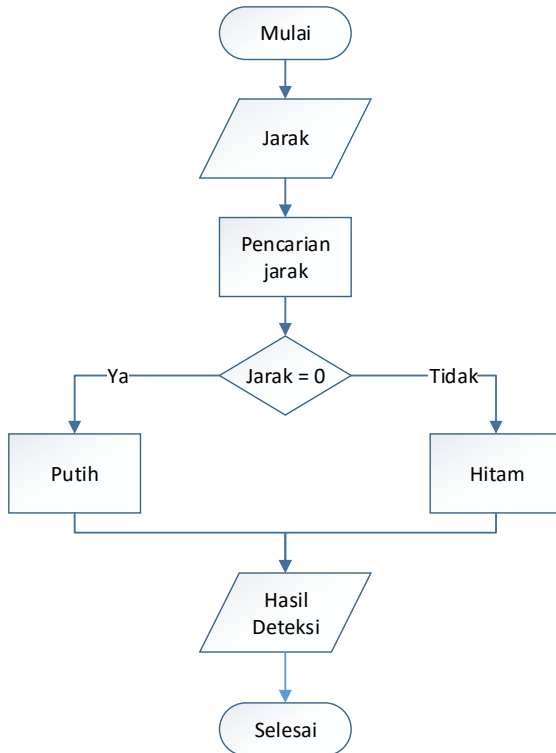
Melakukan identifikasi terhadap blok-blok yang memiliki kemiripan fitur. Terlebih dahulu dilakukan pengurutan fitur vektor secara leksikografis. Pengurutan secara leksikografis merupakan pengurutan berdasarkan urutan kamus. Hal ini dilakukan agar fitur-fitur yang memiliki kemiripan berada pada tempat yang berdekatan sehingga proses pencocokan blok yang memiliki kemiripan berlangsung lebih cepat yang berkaitan dengan kompleksitas dalam komputasi.

Dalam kasus ini, dilakukan pencocokan blok untuk mencocokkan kesesuaian blok dan mengidentifikasi daerah yang mungkin telah dipalsukan. Blok yang sesuai diidentifikasi dengan memperkirakan *Euclidean distance* dari fitur vektor. Perhitungan *Euclidean distance* dilakukan sesuai dengan Persamaan (2.9).

### 3.3.5 Pencarian Daerah yang Mengalami *Copy-move*

Pencarian daerah yang mengalami *copy-move* dilakukan dengan cara membandingkan jarak setiap blok. Daerah yang

mengalami *copy-move* dapat diketahui dari jarak dari setiap blok. Blok yang memiliki kemiripan fitur akan memiliki jarak yang kecil. Dalam hal ini daerah yang mengalami *copy-move* memiliki nilai jarak nol. Blok-blok yang dibandingkan hanya 5 blok yang terdekat. Hal ini karena sebelumnya fitur-fitur ini telah diurutkan sehingga fitur yang mempunyai kemiripan terletak bedekatan. Jika ditemukan blok dengan fitur yang mirip dengan jarak = 0, maka blok itu akan ditandai sebagai daerah *copy-move* dengan memberikan warna putih. Gambaran proses pencarian daerah deteksi ditunjukkan pada Gambar 3.7.



**Gambar 3.7 Diagram Alir Proses Pencarian Daerah *Copy-move***

## BAB IV IMPLEMENTASI

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

### 4.1 Lingkungan Implementasi

Objek citra yang akan diolah pada implementasi tugas akhir ini adalah sebuah citra yang telah mengalami *copy-move forgery*.

Dalam implementasi algoritma deteksi *copy-move* tersebut, digunakan perangkat keras dan perangkat lunak seperti ditunjukkan pada **Tabel 4.1**.

**Tabel 4.1 Lingkungan Implementasi Perangkat Keras dan Perangkat Lunak**

Perangkat	Jenis Perangkat	Spesifikasi
Perangkat Keras	Prosesor	Intel(R) Core(TM) i3-5005U CPU @ 2.00GHz (4CPUs)
	Memori	4GB
Perangkat Lunak	Sistem Operasi	Windows 10 Pro 64-bit
	Perangkat Pengembang	MATLAB R2014a

### 4.2 Implementasi

Pada tahap implementasi ini menjelaskan mengenai pembangunan perangkat lunak secara detail dan menampilkan kode sumber yang digunakan mulai dari tahap *preprocessing* hingga tahap pendeteksian citra *copy-move* menggunakan metode *Histogram of Oriented Gradients*. Pada tugas akhir ini, implementasi menggunakan bahasa pemrograman Matlab dan

dataset yang digunakan berupa citra *copy-move* yang berukuran 512x512 piksel sebanyak 20 citra.

#### 4.2.1 Implementasi Tahap *Preprocessing*

Tahap *preprocessing* pada tugas akhir ini adalah melakukan konversi citra masukan berupa citra RGB menjadi citra *grayscale* atau keabuan. Tahap ini dilakukan dengan memanggil fungsi *rgb2gray*.

```
1. I = imread(path/nama_citra.png);
2. img = rgb2gray(I);
```

#### Kode Sumber 4.1 Implementasi Tahap *Preprocessing*

#### 4.2.2 Implementasi Ekstraksi Fitur dengan HOG

Tahap selanjutnya adalah melakukan ekstraksi fitur dengan HOG. Pada tahap ini parameter yang digunakan adalah citra *grayscale* hasil *pre-processing*.

```
1. block = 16;
2. [row, col] = size(img);
3. column = (row-block+1)*(col-block+1);
4. MatrixA = zeros(column,36);
5. Mi = zeros(1,2);
6. i = 1;
7. for r1 = 1:row-block+1
8.     for c1 = 1:col-block+1
9.         B = img(r1:r1+block-1,c1:c1+block-1);
10.        [features, visualization] = extractHOGFeatures(B);
11.        plot(visualization);
12.        MatrixA(i,:) = features;
13.        Mi(i,:) = [r1 c1];
14.        i=i+1;
15.    end
16.    disp(r1);
17. End
```

#### Kode Sumber 4.2 Implementasi Proses Ekstraksi Fitur dengan HOG

Pada Kode Sumber 4.2 baris pertama merupakan inisialisasi ukuran blok yang akan digunakan. Baris kedua mendefinisikan matriks dari citra masukan. Baris ketiga merupakan rumus perhitungan jumlah blok-blok *overlap* yang akan terbentuk. Baris ke-4 merupakan inisialisasi dari *MatrixA*. *MatrixA* adalah matriks yang akan menyimpan hasil akhir dari fitur-fitur yang dihasilkan dari proses ekstraksi fitur ini. Baris ke-5 merupakan inisialisasi matriks untuk menyimpan index dari fitur pada *MatrixA*. Selanjutnya melakukan perulangan untuk melakukan ekstraksi HOG pada setiap blok yang diimplementasikan pada baris 6-11, kemudian melakukan pemanggilan fungsi *extractHOGFeatures* pada baris 12. Proses ekstraksi dengan HOG ini dilakukan dengan menggunakan fungsi *extractHOGFeatures* dari Matlab yang sesuai dengan uraian pada subbab 2.7.

### **4.2.3 Implementasi Perhitungan Jarak menggunakan *Euclidean Distance***

Setelah mendapat fitur HOG, kemudian melakukan perhitungan jarak antar blok. Namun sebelum perhitungan jarak, blok-blok penyusun fitur HOG diurutkan terlebih dahulu secara leksikografis. Pengurutan secara leksikografis merupakan pengurutan berdasarkan urutan kamus. Implementasi proses pengurutan secara leksikografis ditunjukkan pada Kode Sumber 4.3 baris pertama. Variabel *Beta* merupakan matriks yang menampung fitur HOG setelah diurutkan. Sedangkan *MatrixA* menunjukkan fitur HOG yang belum diurutkan. Proses pengurutan ini menggunakan fungsi matlab *sortrows*. Fungsi *sortrows* ini mengurutkan elemen matriks secara *ascending* berdasarkan elemen pada kolom pertama.

Baris 2-11 merupakan deklarasi perhitungan jarak untuk 5 fitur diatas dan 5 fitur dibawah fitur yang dihitung nilai jaraknya. Baris 13 merupakan fungsi perhitungan jarak menggunakan *Euclidean distance* sesuai dengan Persamaan (2.9). Baris 13-19 merupakan fungsi untuk menyimpan jarak beserta koordinat dari 2 blok yang dihitung jaraknya.

```

1. Beta = sortrows(MatrixA);
2. L = length(Beta);
3. Jarakmin = zeros(L,5);
4. for i=1:L
5.     Jarakbesar = 999999;
6.     Jarak = zeros(10,5);
7.     k=1;
8.     for j=i-5:i+5
9.         if(j<1 || j>L || j==i)
10.            k=k+1;
11.            continue;
12.        end
13.        selisih = pdist([Beta(i,1:36);Beta(j,1:36)]
14.            , 'euclidean');
15.        if(selisih<Jarakbesar)
16.            Jarakbesar = selisih;
17.            Jarakmin(i,1) = selisih;
18.            Jarakmin(i,2) = Beta(i,37);
19.            Jarakmin(i,3) = Beta(i,38);
20.            Jarakmin(i,4) = Beta(j,37);
21.            Jarakmin(i,5) = Beta(j,38);
22.        end
23.    end
24. end

```

### Kode Sumber 4.3 Implementasi Perhitungan Jarak Menggunakan Euclidean Distance

#### 4.2.4 Implementasi Pencarian Daerah yang Mengalami *Copy-move*

Daerah yang mengalami *copy-move* dicari dengan mencocokkan jarak antar blok. Blok-blok yang memiliki fitur yang sama akan memiliki nilai jarak nol. Baris pertama dari Kode Sumber 4.4 merupakan inisialisasi dari matriks yang berisi nilai jarak, matriks *Jarakmin* menyimpan nilai jarak antara 2 blok yang dibandingkan beserta nilai koordinat kedua blok yang dibandingkan. Baris 2-13 merupakan perbandingan 2 blok pada koordinat  $x$  dan  $y$ .



ketika ditemukan jarak yang bernilai nol maka blok tersebut akan diberi warna putih.

```
1. for i=1:size(Jarakmin,1)
2.     if Jarakmin(i,1) == 0
3.         for hitam1=Jarakmin(i,2):Jarakmin(i,2)+b
4.             lock
5.                 for hitam2=Jarakmin(i,3):Jarakmin(i,
6.                 3)+block
7.                     imhasil(hitam1,hitam2)=255;
8.                 end
9.             end
10.            for hitam1=Jarakmin(i,4):Jarakmin(i,4)+b
11.                lock
12.                for hitam2=Jarakmin(i,5):Jarakmin(i,
13.                5)+block
14.                    imhasil(hitam1,hitam2)=255;
15.                end
16.            end
17.        end
18.    end
```

**Kode Sumber 4.4 Implementasi Pencarian Dearah yang Mengalami *Copy-move***

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

## BAB V

### UJI COBA DAN EVALUASI

Dalam bab ini dibahas mengenai skenario uji coba dan evaluasi hasil uji coba sistem deteksi *copy-move* pada citra 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 untuk pendeteksian *copy-move* pada citra dengan *Histogram of Oriented Gradients* pada tugas akhir ini adalah dengan menggunakan spesifikasi perangkat keras dan perangkat lunak yang ditunjukkan pada Tabel 5.1.

**Tabel 5.1 Spesifikasi Lingkungan Uji Coba**

Perangkat	Jenis Perangkat	Spesifikasi
Perangkat Keras	Prosesor	Intel(R) Core(TM) i3-5005U CPU @ 2.00GHz (4CPUs)
	Memori	4GB
Perangkat Lunak	Sistem Operasi	Windows 10 Pro 64-bit
	Perangkat Pengembang	MATLAB R2014a

#### 5.2 Data Uji Coba

Data uji coba yang digunakan sebagai masukan adalah citra yang telah mengalami *copy-move* beserta *ground truth* dari masing-masing citra *copy-move*. Data uji coba ini didapatkan dari CoMoFod *database* sebanyak 20 citra yang berukuran 512x512 piksel.

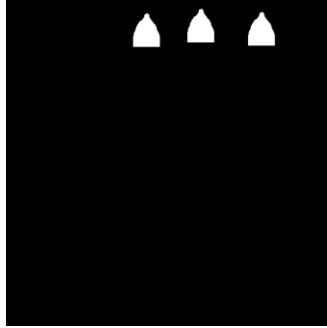
Berikut merupakan citra *copy-move* yang digunakan sebagai data uji coba beserta *ground truth*-nya. Uji coba ini bertujuan untuk mengetahui daerah yang mengalami pemalsuan *copy-move* pada citra masukan.

## 1. 001\_F.png

Pada Gambar 5.1 daerah yang mengalami *copy-move* adalah 3 kubah.



(a)



(b)

**Gambar 5.1 (a) Citra *Copy-move* 001\_F.png, (b) *Ground Truth***

## 2. 002\_F.png

Pada Gambar 5.2 daerah yang mengalami *copy-move* adalah beberapa bagian di tengah gambar.



(a)

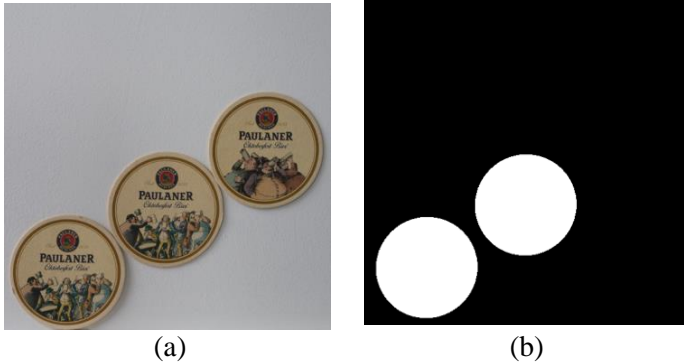


(b)

**Gambar 5.2 (a) Citra *Copy-move* 002\_F.png, (b) *Ground Truth***

## 3. 003\_F.png

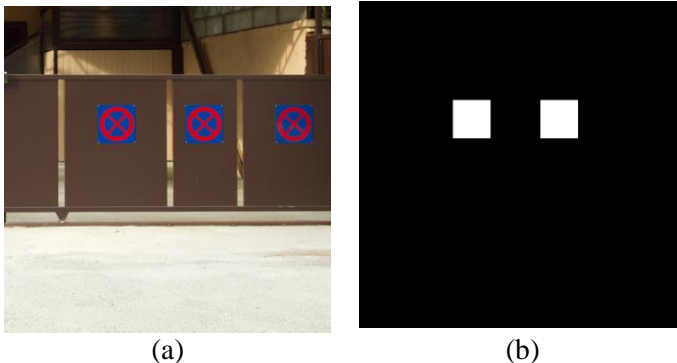
Pada Gambar 5.3 daerah yang mengalami *copy-move* adalah dua koin di kiri bawah.



**Gambar 5.3 (a) Citra *Copy-move* 003\_F.png, (b) *Ground Truth***

## 4. 004\_F.png

Pada Gambar 5.4 daerah yang mengalami *copy-move* adalah kotak pertama dan kedua dari kiri.



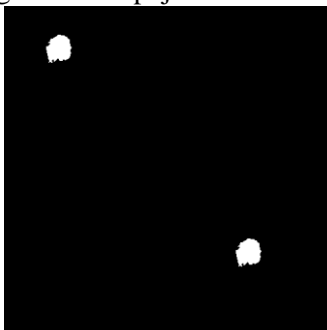
**Gambar 5.4 (a) Citra *Copy-move* 004\_F.png, (b) *Ground Truth***

## 5. 005\_F.png

Pada Gambar 5.5 daerah yang mengalami *copy-move* adalah bunga kecil yang berada di pojok.



(a)



(b)

**Gambar 5.5 (a) Citra *Copy-move* 005\_F.png, (b) *Ground Truth***

## 6. 006\_F.png

Pada Gambar 5.6 daerah yang mengalami *copy-move* adalah bagian dari tembok.



(a)



(b)

**Gambar 5.6 (a) Citra *Copy-move* 006\_F.png, (b) *Ground Truth***

## 7. 007\_F.png

Pada Gambar 5.7 daerah yang mengalami *copy-move* adalah pucuk dalam satu pohon.



(a)

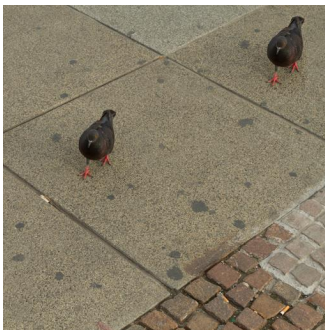


(b)

**Gambar 5.7 (a) Citra *Copy-move* 007\_F.png, (b) *Ground Truth***

## 8. 008\_F.png

Pada Gambar 5.8 daerah yang mengalami *copy-move* adalah objek burung.



(a)

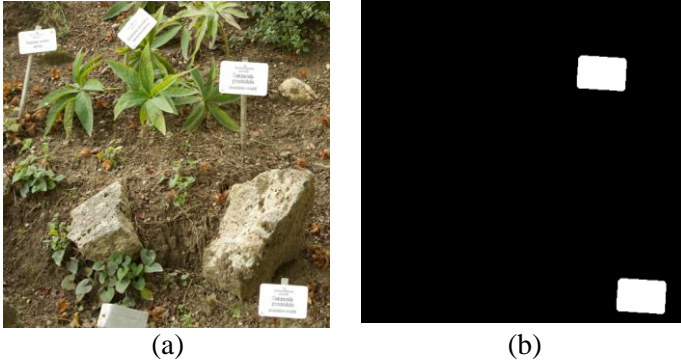


(b)

**Gambar 5.8 (a) Citra *Copy-move* 008\_F.png, (b) *Ground Truth***

9. 009\_F.png

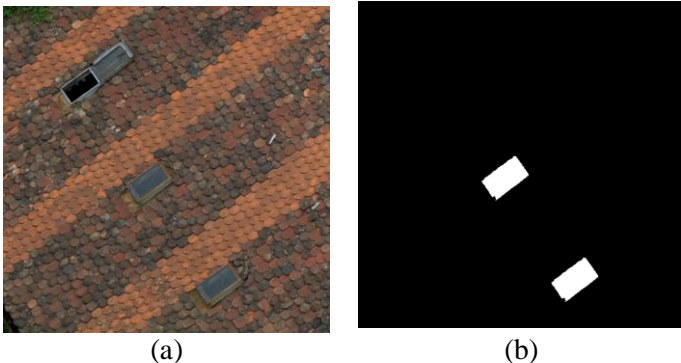
Pada Gambar 5.9 daerah yang mengalami *copy-move* adalah dua papan di bagian kanan.



**Gambar 5.9 (a) Citra *Copy-move* 009\_F.png, (b) *Ground Truth***

10. 010\_F.png

Pada Gambar 5.10 daerah yang mengalami *copy-move* adalah dua kotak yang tertutup.

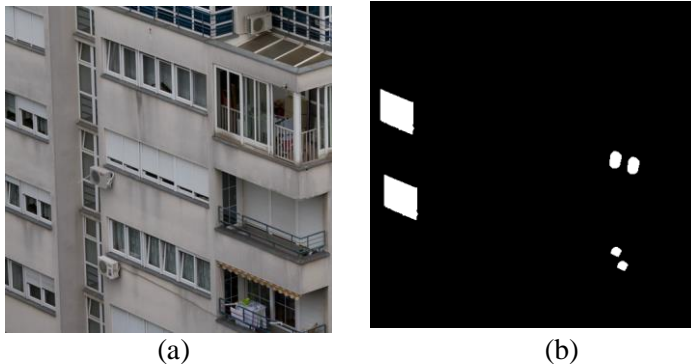


**Gambar 5.10 (a) Citra *Copy-move* 010\_F.png, (b) *Ground Truth***



## 11. 011\_F.png

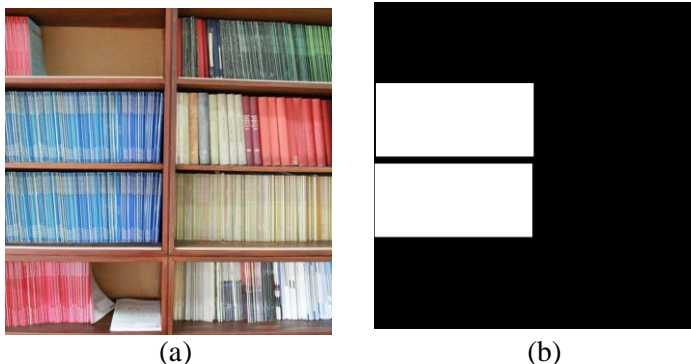
Pada Gambar 5.11 daerah yang mengalami *copy-move* adalah jendela di bagian kiri dan beberapa titik di bagian kanan.



**Gambar 5.11 (a) Citra *Copy-move* 011\_F.png, (b) *Ground Truth***

## 12. 012\_F.png

Pada Gambar 5.12 daerah yang mengalami *copy-move* adalah susunan buku berwarna biru di sebelah kiri.



**Gambar 5.12 (a) Citra *Copy-move* 012\_F.png, (b) *Ground Truth***

## 13. 013\_F.png

Pada Gambar 5.13 daerah yang mengalami *copy-move* adalah tiga bunga.



(a)



(b)

**Gambar 5.13 (a) Citra *Copy-move* 013\_F.png, (b) *Ground Truth***

## 14. 014\_F.png

Pada Gambar 5.14 daerah yang mengalami *copy-move* adalah serangkaian daun.



(a)



(b)

**Gambar 5.14 (a) Citra *Copy-move* 014\_F.png, (b) *Ground Truth***

15. 015\_F.png

Pada Gambar 5.15 daerah yang mengalami *copy-move* adalah bagian dinding di atas jendela.



(a)

(b)

**Gambar 5.15 (a) Citra *Copy-move* 015\_F.png, (b) *Ground Truth***

16. 016\_F.png

Pada Gambar 5.16 daerah yang mengalami *copy-move* adalah ukiran di atas jendela.



(a)

(b)

**Gambar 5.16 (a) Citra *Copy-move* 016\_F.png, (b) *Ground Truth***

17. 017\_F.png

Pada Gambar 5.17 daerah yang mengalami *copy-move* adalah bagian balkon.



(a)



(b)

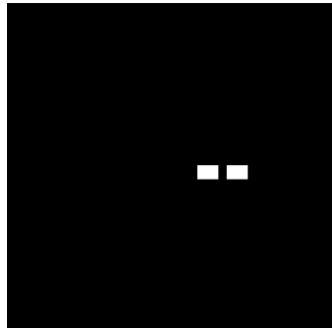
**Gambar 5.17 (a) Citra *Copy-move* 017\_F.png, (b) *Ground Truth***

18. 018\_F.png

Pada Gambar 5.18 daerah yang mengalami *copy-move* adalah angka tahun pada buku.



(a)



(b)

**Gambar 5.18 (a) Citra *Copy-move* 018\_F.png, (b) *Ground Truth***

19. 019\_F.png

Pada Gambar 5.19 daerah yang mengalami *copy-move* adalah bagian jendela.



(a)



(b)

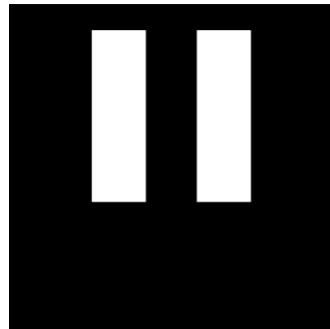
**Gambar 5.19** (a) Citra *Copy-move* 019\_F.png, (b) *Ground Truth*

20. 020\_F.png

Pada Gambar 5.20 daerah yang mengalami *copy-move* adalah patok.



(a)



(b)

**Gambar 5.20** (a) Citra *Copy-move* 020\_F.png, (b) *Ground Truth*

Detail dari ke-20 citra data uji coba tersebut disajikan pada Tabel 5.2.

**Tabel 5.2 Detail Dataset Citra *Copy-move***

No.	Nama Citra	Dimensi Citra	Ukuran Citra
1	001_F.png	512 x 512	481 KB
2	002_F.png	512 x 512	679 KB
3	003_F.png	512 x 512	419 KB
4	004_F.png	512 x 512	360 KB
5	005_F.png	512 x 512	572 KB
6	006_F.png	512 x 512	652 KB
7	007_F.png	512 x 512	563 KB
8	008_F.png	512 x 512	546 KB
9	009_F.png	512 x 512	598 KB
10	010_F.png	512 x 512	510 KB
11	011_F.png	512 x 512	428 KB
12	012_F.png	512 x 512	453 KB
13	013_F.png	512 x 512	611 KB
14	014_F.png	512 x 512	539 KB
15	015_F.png	512 x 512	406 KB
16	016_F.png	512 x 512	341 KB
17	017_F.png	512 x 512	439 KB
18	018_F.png	512 x 512	377 KB
19	019_F.png	512 x 512	433 KB
20	020_F.png	512 x 512	557 KB

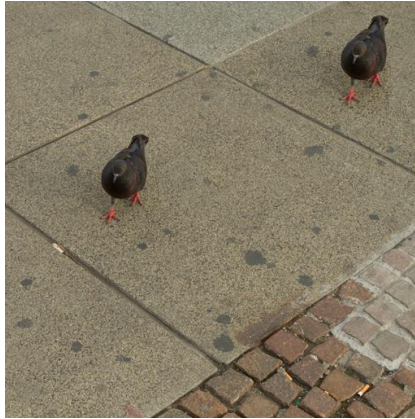
### 5.3 Hasil Uji Coba Citra Setiap Proses

Sub bab ini menjelaskan hasil dari uji coba dari setiap tahapan yang dilakukan untuk mendeteksi pemalsuan citra berupa *copy-*

*move*. Dimulai dari tahap preprocessing hingga akhir yang menghasilkan citra hasil deteksi.

### **5.3.1 Hasil Uji Coba *Preprocessing***

Tahap *preprocessing* adalah melakukan konversi citra RGB menjadi *grayscale* atau keabuan. Citra *copy-move* asli dalam RGB ditunjukkan pada Gambar 5.21 dan citra *grayscale* hasil preprocessing ditunjukkan pada Gambar 5.22.



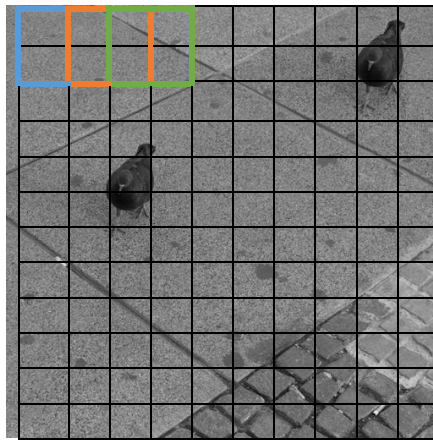
**Gambar 5.21 Citra *Copy-move* Asli (RGB)**



**Gambar 5.22 Citra *Grayscale* Hasil *Preprocessing***

### 5.3.2 Hasil Uji Coba Pembagian ke dalam Blok-blok yang Saling *Overlap*

Setelah tahap *preprocessing*, selanjutnya citra dibagi ke dalam blok-blok yang saling *overlap* yang berukuran sama setiap bloknya. Gambar 5.23 menunjukkan citra yang telah dibagi ke dalam blok-blok yang saling *overlap*. Gambar 5.24 menunjukkan nilai matriks citra untuk satu blok yang berukuran 8x8 piksel.



**Gambar 5.23** Citra yang Telah Dibagi Menjadi Blok-blok

104	85	75	69	89	77	63	67
77	66	74	70	66	75	78	100
82	72	56	63	62	75	95	122
85	78	75	88	102	114	116	117
70	76	113	130	119	121	133	124
96	127	104	112	126	128	127	118
128	138	118	122	123	122	135	127
133	125	128	119	108	123	135	137

**Gambar 5.24** Nilai Matriks Citra untuk 1 blok berukuran 8x8

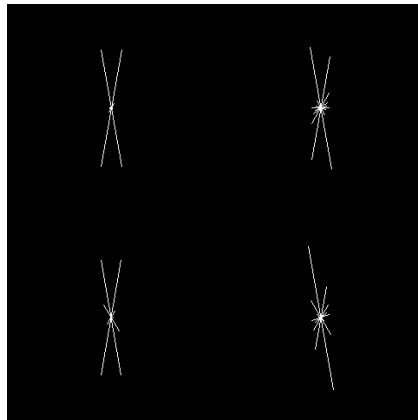


### 5.3.3 Hasil Uji Coba Tahap Ekstraksi Fitur dengan HOG

Tahap ekstraksi fitur dengan HOG menghasilkan kumpulan fitur vektor sebanyak  $(M-b+1) \times (N-b+1)$  baris dan 36 kolom.  $M$  dan  $N$  menunjukkan dimensi citra masukan,  $b$  menunjukkan ukuran blok, dan 36 didapat dari jumlah fitur setiap blok. Karena satu blok terbentuk dari 4 sel, dan satu sel terbentuk dari sebuah histogram yang memiliki bin sebanyak 9. Nilai setiap bin inilah yang mewakili satu fitur. Jika dimensi citra masukan  $512 \times 512$  dan ukuran blok yang digunakan  $8 \times 8$  piksel, maka akan menghasilkan 255025 blok yang saling *overlap*. Dan fitur vektor yang akan terbentuk sebanyak 255025 baris 36 kolom. Gambar 5.25 menunjukkan fitur yang terbentuk dari satu blok, Gambar 5.26 menunjukkan visualisasi fitur HOG untuk satu blok.

0.2 113	0.2 113	0.2 113	0.2 113	0.2 060	0.1 210	0.1 254	0.1 016	0.1 737	0.1 736	0.2 084	0.1 626	0.1 958	0.1 622	0.1 684	0.1 815	0.1 114	0.1 317
0.2 076	0.1 338	0.1 520	0.1 656	0.1 107	0.1 481	0.1 362	0.1 529	0.1 591	0.1 137	0.1 725	0.1 453	0.1 517	0.2 113	0.1 802	0.1 774	0.1 328	0.1 627

**Gambar 5.25** Fitur yang Terbentuk dari Satu Blok



**Gambar 5.26** Visualisasi Fitur HOG untuk Satu Blok

### 5.3.4 Hasil Uji Coba Tahap Perhitungan Jarak dengan *Euclidean Distance*

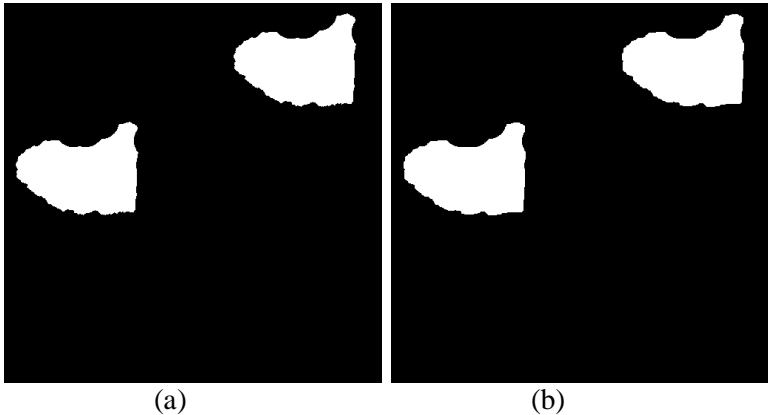
Perhitungan jarak dengan *Euclidean distance* dilakukan dengan menghitung jarak antara dua blok. Hasil dari proses ini adalah sebuah matriks yang terdiri dari 5 kolom. Kolom pertama berisi nilai jarak antara dua blok yang dibandingkan, kolom kedua sampai kolom kelima menyimpan koordinat (x,y) dari dua blok yang dihitung jaraknya. Matriks hasil perhitungan jarak ditunjukkan pada Gambar 5.27.

0,3021	480	217	464	234
0,1400	403	330	403	331
0	88	429	234	134
0	234	134	88	429
0	88	428	234	133
0	234	133	88	428
0,2648	324	497	464	236
0,2648	464	236	324	497

**Gambar 5.27 Matrix yang Menyimpan Jarak serta Koordinat Blok**

### 5.3.5 Hasil Uji Coba Tahap Pencarian Daerah yang Mengalami *Copy-move*

Setelah nilai jarak didapatkan kemudian dilakukan pencarian jarak yang bernilai nol. Kemudian dua blok yang berjarak nol tersebut ditandai sebagai daerah yang mengalami *copy-move* dan diberi warna putih. Setelah mendapatkan citra hasil deteksi, kemudian dilakukan perhitungan performa dengan membandingkan citra hasil deteksi dengan citra *ground truth*. Gambar 5.28 menunjukkan citra hasil deteksi daerah yang mengalami *copy-move*.



**Gambar 5.28** Citra Hasil Deteksi (a) Citra *Ground Truth*,  
(b) Citra Hasil Deteksi

## 5.4 Skenario Uji Coba

Skenario Uji coba diperlukam untuk menguji kebenaran yang diusulkan. Sebelum melakukan uji coba perlu ditentukan skenario yang akan dilakukan dalam uji coba. Melalui skenario ini, program akan diuji apakah sudah berjalan dengan benar dan bagaimana performa pada masing-masing skenario dan membandingkan skenario manakah yang memiliki hasil yang lebih baik.

Skenario pengujian terdiri dari dua macam yaitu:

1. Perhitungan performa dengan mengubah ukuran blok.
2. Perhitungan performa dengan mengubah variasi data citra.

### 5.4.1 Skenario Uji Coba Ukuran Blok Berbeda

Skenario uji coba pertama adalah perhitungan akurasi, *specificity* dan *sensitiftiy* dengan mengubah ukuran blok. Nilai akurasi menunjukkan persentase hasil deteksi secara keseluruhan. *Specificity* menunjukkan persentase hasil deteksi objek non *copy-move* dalam hal ini daerah yang berwarna hitam dibandingkan dengan *ground truth*. *Sensitiftiy* menunjukkan persentase hasil deteksi objek *copy-move* dalam hal ini daerah yang berwarna putih dibandingkan dengan *ground truth*. Jadi dalam uji coba ini nilai

performa yang akan dijadikan acuan untuk hasil deteksi *copy-move* adalah nilai *sensitivity*. Dalam uji coba ini digunakan 3 ukuran blok berbeda yaitu 8x8, 16x16 dan 32x32.

#### 5.4.1.1 Skenario Uji Coba dengan Ukuran Blok 8 x 8

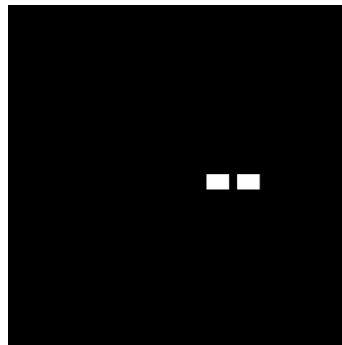
Skenario uji coba dengan ukuran blok 8x8 adalah menghitung akurasi, *specificity* dan *sensitivity* pada data uji coba citra dengan parameter ukuran blok 8x8 piksel dimana hasilnya disajikan pada Tabel 5.3. Terlihat rata-rata akurasi 99,58%, rata-rata *specificity* sebesar 99,73%, dan rata-rata *sensitivity* sebesar 97,33%. Dilihat dari nilai persentase tersebut, hasil uji coba parameter blok dengan ukuran 8x8 piksel menghasilkan hasil deteksi yang optimal namun masih ada beberapa daerah non-*copy-move* yang terdeteksi sebagai *copy-move*.

**Error! Reference source not found.** menunjukkan hasil deteksi dengan *sensitivity* tertinggi pada uji coba dengan parameter ukuran blok 8x8 piksel. Daerah yang mengalami *copy-move* adalah angka tahun pada buku yang berwarna merah pada citra masukan.

Gambar 5.30 menunjukkan hasil deteksi dengan *sensitivity* terendah pada uji coba dengan parameter ukuran blok 8x8 piksel. Daerah yang mengalami *copy-move* adalah tiga buah kubah pada citra masukan.



(a)

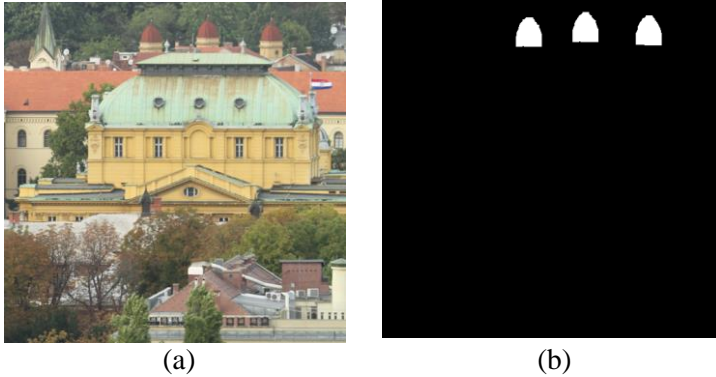


(b)

**Gambar 5.29 Contoh Citra Hasil Deteksi dengan *Sensitivity* 100% pada Uji Coba Parameter Blok 8x8 (a) Citra Masukan (b) Citra Hasil Deteksi**

**Tabel 5.3 Hasil Skenario Uji Coba Performa dengan  
Ukuran Blok 8x8**

Nama Citra	FN	FP	TN	TP	Aku-rasi (%)	Speci-ficity (%)	Sensiti-fity (%)
001_F.png	614	102	257275	4153	99,73	99,96	87,12
002_F.png	454	58	241334	20298	99,80	99,98	97,81
003_F.png	0	642	221934	39568	99,76	99,71	100,00
004_F.png	0	240	254824	7080	99,91	99,91	100,00
005_F.png	90	104	259476	2474	99,93	99,96	96,49
006_F.png	0	218	259294	2632	99,92	99,92	100,00
007_F.png	246	558	237840	23500	99,69	99,77	98,96
008_F.png	82	540	233232	28290	99,76	99,77	99,71
009_F.png	0	1825	252465	7854	99,30	99,28	100,00
010_F.png	88	519	256553	4984	99,77	99,80	98,26
011_F.png	102	2182	253518	6342	99,13	99,15	98,42
012_F.png	0	3878	201456	56810	98,52	98,11	100,00
013_F.png	1525	424	248700	11495	99,26	99,83	88,29
014_F.png	1484	76	231112	29472	99,40	99,97	95,21
015_F.png	0	426	246934	14784	99,84	99,83	100,00
016_F.png	2830	162	234616	24536	98,86	99,93	89,66
017_F.png	1290	14	187714	73126	99,50	99,99	98,27
018_F.png	0	112	260580	1452	99,96	99,96	100,00
019_F.png	0	624	252757	8763	99,76	99,75	100,00
020_F.png	706	0	216414	45024	99,73	100,00	98,46
Rata-rata					99,58	99,73	97,33



**Gambar 5.30 Contoh Citra Hasil Deteksi dengan *Sensitivity* 87,12% pada Uji Coba Parameter Blok 8x8 (a) Citra Masukan (b) Citra Hasil Deteksi**

#### 5.4.1.2 Skenario Uji Coba dengan Ukuran Blok 16x16

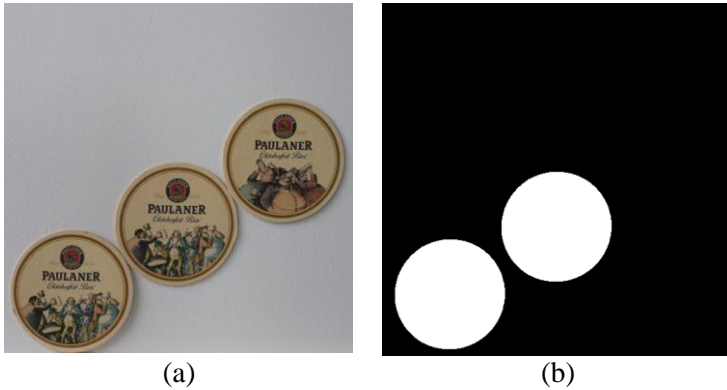
Skenario uji coba dengan ukuran blok 16x16 adalah menghitung akurasi, *specificity* dan *sensitivity* pada data uji coba citra dengan parameter ukuran blok 16 x 16 piksel dimana hasilnya disajikan pada Tabel 5.4. Terlihat rata-rata akurasi 99,59%, rata-rata *specificity* sebesar 99,91%, dan rata-rata *sensitivity* sebesar 90,21%. Dilihat dari nilai persentase tersebut, hasil uji coba parameter blok dengan ukuran 16x16 piksel menghasilkan hasil deteksi yang optimal.

Gambar 5.31 menunjukkan hasil deteksi dengan *sensitivity* tertinggi pada uji coba dengan parameter ukuran blok 16x16 piksel. Daerah yang mengalami *copy-move* adalah dua buah koin pada citra masukan.

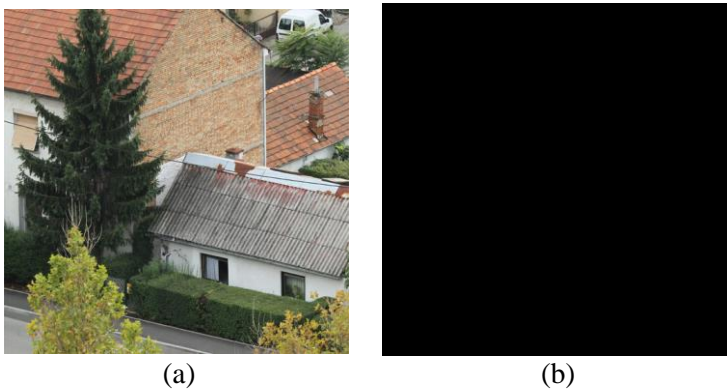
Gambar 5.32 menunjukkan hasil deteksi dengan *sensitivity* terendah pada uji coba dengan parameter ukuran blok 16x16 piksel. Daerah yang mengalami *copy-move* adalah bagian dari objek dinding pada citra masukan.

**Tabel 5.4 Hasil Skenario Uji Coba Performa dengan Ukuran Blok 16x16**

Nama Citra	FN	FP	TN	TP	Aku-rasi (%)	Speci-ficity (%)	Sensiti-fity (%)
001_F.png	773	82	257295	3994	99,67	99,97	83,78
002_F.png	804	40	241352	19948	99,68	99,98	96,13
003_F.png	0	642	221934	39568	99,76	99,71	100,00
004_F.png	0	240	254824	7080	99,91	99,91	100,00
005_F.png	256	70	259510	2308	99,88	99,97	90,02
006_F.png	2632	0	259512	0	99,00	100,00	0,00
007_F.png	582	380	238018	23164	99,63	99,84	97,55
008_F.png	298	434	233338	28074	99,72	99,81	98,95
009_F.png	24	220	254070	7830	99,91	99,91	99,69
010_F.png	330	186	256886	4742	99,80	99,93	93,49
011_F.png	782	198	255502	5662	99,63	99,92	87,86
012_F.png	0	726	204608	56810	99,72	99,65	100,00
013_F.png	2599	7	249117	10421	99,01	100,00	80,04
014_F.png	2564	56	231132	28392	99,00	99,98	91,72
015_F.png	0	426	246934	14784	99,84	99,83	100,00
016_F.png	3108	0	234778	24258	98,81	100,00	88,64
017_F.png	1546	14	187714	72870	99,40	99,99	97,92
018_F.png	0	112	260580	1452	99,96	99,96	100,00
019_F.png	0	453	252928	8763	99,83	99,82	100,00
020_F.png	706	0	216414	45024	99,73	100,00	98,46
Rata-rata					99,59	99,91	90,21



**Gambar 5.31** Contoh Citra Hasil Deteksi dengan *Sensitivity* 100% pada Uji Coba Parameter Blok 16x16 (a) Citra Masukan (b) Citra Hasil Deteksi



**Gambar 5.32** Contoh Citra Hasil Deteksi dengan *Sensitivity* 0% pada Uji Coba Parameter Blok 16x16 (a) Citra Masukan (b) Citra Hasil Deteksi

### 5.4.1.3 Skenario Uji Coba dengan Ukuran Blok 32x32

Skenario uji coba dengan ukuran blok 32x32 adalah menghitung akurasi, *specificity* dan *sensitivity* pada data uji coba

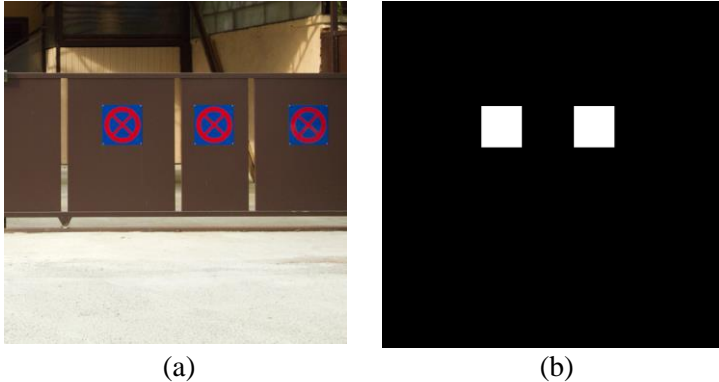


citra dengan parameter ukuran blok 32 x 32 piksel dimana hasilnya disajikan pada Tabel 5.5.

**Tabel 5.5 Hasil Skenario Uji Coba Performa dengan Ukuran Blok 32x32**

Nama Citra	FN	FP	TN	TP	Aku-rasi (%)	Speci-ficity (%)	Sensiti-fity (%)
001_F.png	4767	0	257377	0	98,18	100,00	0,00
002_F.png	1316	32	241360	19436	99,49	99,99	93,66
003_F.png	120	532	222044	39448	99,75	99,76	99,70
004_F.png	0	240	254824	7080	99,91	99,91	100,00
005_F.png	2564	0	259580	0	99,02	100,00	0,00
006_F.png	2632	0	259512	0	99,00	100,00	0,00
007_F.png	1878	258	238140	21868	99,19	99,89	92,09
008_F.png	1718	292	233480	26654	99,23	99,88	93,94
009_F.png	140	188	254102	7714	99,87	99,93	98,22
010_F.png	5072	0	257072	0	98,07	100,00	0,00
011_F.png	2028	134	255566	4416	99,18	99,95	68,53
012_F.png	0	726	204608	56810	99,72	99,65	100,00
013_F.png	13020	0	249124	0	95,03	100,00	0,00
014_F.png	12362	30	231158	18594	95,27	99,99	60,07
015_F.png	0	426	246934	14784	99,84	99,83	100,00
016_F.png	5090	0	234778	22276	98,06	100,00	81,40
017_F.png	2506	14	187714	71910	99,04	99,99	96,63
018_F.png	1452	0	260692	0	99,45	100,00	0,00
019_F.png	8763	0	253381	0	96,66	100,00	0,00
020_F.png	706	0	216414	45024	99,73	100,00	98,46
Rata-rata					98,68	99,94	59,13

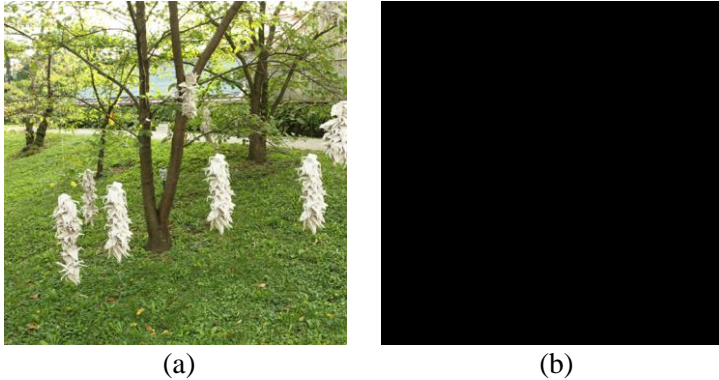
Dari Tabel 5.5 terlihat rata-rata akurasi 98,68%, rata-rata *specificity* sebesar 99,94%, dan rata-rata *sensitifity* sebesar 59,13%. Dilihat dari nilai persentase tersebut, hasil uji coba parameter blok dengan ukuran 32x32 piksel menghasilkan hasil deteksi yang cukup optimal. Namun masih menghasilkan hasil yang tidak tepat pada citra dengan dengan daerah *copy-move* yang kecil.



**Gambar 5.33 Contoh Citra Hasil Deteksi dengan *Sensitifity* 100% pada Uji Coba Parameter Blok 32x32 (a) Citra Masukan (b) Citra Hasil Deteksi**

Gambar 5.33 menunjukkan hasil deteksi dengan *sensitifity* tertinggi pada uji coba dengan parameter ukuran blok 32x32 piksel. Daerah yang mengalami *copy-move* adalah kotak pertama dan kedua pada citra masukan.

Gambar 5.34 menunjukkan hasil deteksi dengan *sensitifity* terendah pada uji coba dengan parameter ukuran blok 32x32 piksel karena program tidak berhasil mendeteksi daerah yang mengalami *copy-move*. Daerah yang mengalami *copy-move* adalah 3 bunga yang berwarna putih pada citra masukan.



**Gambar 5.34 Contoh Citra Hasil Deteksi dengan Sensitivity 0% pada Uji Coba Parameter Blok 32x32 (a) Citra Masukan (b) Citra Hasil Deteksi**

## 5.4.2 Skenario Uji Coba dengan Variasi Data

Skenario uji coba pertama adalah perhitungan akurasi, *specificity* dan *sensitivity* dengan data citra yang divariasikan. Data yang digunakan adalah data citra dengan variasi penyesuaian kontras dan *blurr*. Data uji coba ini diunduh dari CoMoFod *database* yang sudah mengalami penyesuaian kontras dan *blurr* (kabur). Parameter ukuran blok yang digunakan adalah 16x16 piksel karena dalam uji coba parameter ukuran blok memberikan hasil terbaik.

### 5.4.2.1 Skenario Uji Coba dengan Citra Penyesuaian Kontras (*Contrast Adjustments*)

Skenario uji coba dengan penyesuaian kontras adalah menghitung akurasi, *specificity* dan *sensitivity* pada data uji coba citra dengan variasi data citra yang mengalami penyesuaian kontras dimana hasilnya disajikan pada

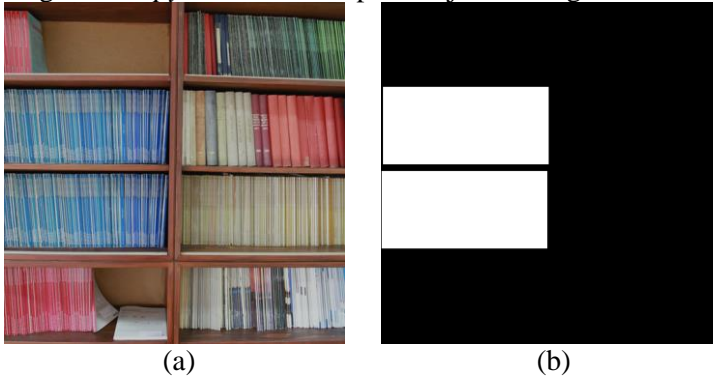
Tabel 5.6. Terlihat rata-rata akurasi 99,60%, rata-rata *specificity* sebesar 99,91%, dan rata-rata *sensitivity* sebesar 90,26%. Dilihat dari nilai persentase tersebut, hasil uji coba dengan variasi data citra yang mengalami *penyesuaian kontras* menghasilkan hasil deteksi yang optimal.

**Tabel 5.6 Hasil Skenario Uji Coba Performa dengan Citra Penyesuaian Kontras (*Contrast Adjustments*)**

Nama Citra	FN	FP	TN	TP	Aku-rasi (%)	Speci-ficity (%)	Sensiti-fity (%)
001_F_CA3	762	86	257291	4005	99,68	99,97	84,02
002_F_CA3	804	44	241348	19948	99,68	99,98	96,13
003_F_CA3	0	646	221930	39568	99,75	99,71	100,00
004_F_CA3	0	240	254824	7080	99,91	99,91	100,00
005_F_CA3	256	68	259512	2308	99,88	99,97	90,02
006_F_CA3	2632	0	259512	0	99,00	100,00	0,00
007_F_CA3	582	380	238018	23164	99,63	99,84	97,55
008_F_CA3	292	440	233332	28080	99,72	99,81	98,97
009_F_CA3	24	220	254070	7830	99,91	99,91	99,69
010_F_CA3	330	188	256884	4742	99,80	99,93	93,49
011_F_CA3	752	200	255500	5692	99,64	99,92	88,33
012_F_CA3	0	726	204608	56810	99,72	99,65	100,00
013_F_CA3	2579	7	249117	10441	99,01	100,00	80,19
014_F_CA3	2554	56	231132	28402	99,00	99,98	91,75
015_F_CA3	0	426	246934	14784	99,84	99,83	100,00
016_F_CA3	3092	0	234778	24274	98,82	100,00	88,70
017_F_CA3	1518	18	187710	72898	99,41	99,99	97,96
018_F_CA3	0	112	260580	1452	99,96	99,96	100,00
019_F_CA3	0	453	252928	8763	99,83	99,82	100,00
020_F_CA3	706	0	216414	45024	99,73	100,00	98,46
Rata-rata					99,60	99,91	90,26

Gambar 5.35 menunjukkan hasil dari uji coba dengan citra penyesuaian kontras yang memiliki nilai *sensitifity* tertinggi pada uji coba citra dengan variasi penyesuaian kontras. Daerah citra yang mengalami *copy-move* adalah deretan buku yang berwarna biru pada rak.

Gambar 5.36 menunjukkan hasil dari uji coba dengan citra penyesuaian kontras yang memiliki nilai *sensitivity* terendah pada uji coba citra dengan variasi penyesuaian kontras. Daerah citra yang mengalami *copy-move* terletak pada objek dinding.



**Gambar 5.35 Contoh Citra Hasil Deteksi dengan *Sensitivity* 100% pada Uji Coba dengan Citra Penyesuaian Kontras (a) Citra Masukan (b) Citra Hasil Deteksi**



**Gambar 5.36 Contoh Citra Hasil Deteksi dengan *Sensitivity* 0% pada Uji Coba dengan Citra Penyesuaian Kontras (a) Citra Masukan (b) Citra Hasil Deteksi**

### 5.4.2.2 Skenario Uji Coba dengan Citra Kabur

Skenario uji coba dengan citra kabur adalah menghitung akurasi, *specificity* dan *sensitivity* pada uji coba citra dengan variasi data citra yang dikaburkan dan hasilnya disajikan pada Tabel 5.7.

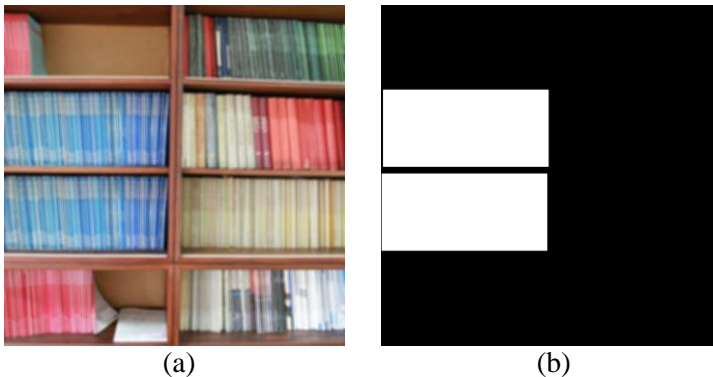
**Tabel 5.7 Hasil Skenario Uji Coba Performa dengan Citra Kabur (*Image Blurring*)**

Nama Citra	FN	FP	TN	TP	Aku- rasi (%)	Speci- ficity (%)	Sensiti- fity (%)
001_F_IB2	1504	0	257377	3263	99,43	100	68,45
002_F_IB2	2640	0	241392	18112	98,99	100	87,28
003_F_IB2	1650	0	222576	37918	99,37	100	95,83
004_F_IB2	696	0	255064	6384	99,73	100	90,17
005_F_IB2	752	0	259580	1812	99,71	100	70,67
006_F_IB2	2632	0	259512	0	99,00	100	0,00
007_F_IB2	2558	0	238398	21188	99,02	100	89,23
008_F_IB2	2068	0	233772	26304	99,21	100	92,71
009_F_IB2	844	0	254290	7010	99,68	100	89,25
010_F_IB2	1172	0	257072	3900	99,55	100	76,89
011_F_IB2	2130	0	255700	4314	99,19	100	66,95
012_F_IB2	2154	0	205334	54656	99,18	100	96,21
013_F_IB2	4076	0	249124	8944	98,45	100	68,69
014_F_IB2	6304	0	231188	24652	97,60	100	79,64
015_F_IB2	1072	0	247360	13712	99,59	100	92,75
016_F_IB2	5028	0	234778	22338	98,08	100	81,63
017_F_IB2	4976	0	187728	69440	98,10	100	93,31
018_F_IB2	312	0	260692	1140	99,88	100	78,51
019_F_IB2	1323	0	253381	7440	99,50	100	84,90
020_F_IB2	3490	0	216414	42240	98,67	100	92,37
Rata-rata					99,10	100	79,77

Dari Tabel 5.7 terlihat rata-rata akurasi 99,10%, rata-rata *specificity* sebesar 100%, dan rata-rata *sensitifity* sebesar 79,77%. Dilihat dari nilai persentase tersebut, hasil uji coba dengan variasi data citra yang mengalami *blurr* menghasilkan hasil deteksi yang optimal.

Gambar 5.37 menunjukkan hasil dari uji coba dengan citra *blurr* yang memiliki nilai *sensitifity* tertinggi pada uji coba citra dengan *blurr*. Daerah citra yang mengalami *copy-move* adalah deretan buku yang berwarna biru pada rak.

Gambar 5.38 menunjukkan hasil dari uji coba dengan citra *blurr* yang memiliki nilai *sensitifity* terendah pada uji coba citra dengan variasi *blurr*. Daerah citra yang mengalami *copy-move* terletak pada objek dinding.



**Gambar 5.37 Contoh Citra Hasil Deteksi dengan *Sensitifity* 100% pada Uji Coba dengan *Image Blurring* (a) Citra Masukan (b) Citra Hasil Deteksi**



**Gambar 5.38** Contoh Citra Hasil Deteksi dengan *Sensitifity* 0% pada Uji Coba dengan Image Blurring (a) Citra Masukan (b) Citra Hasil Deteksi

## 5.5 Evaluasi

Subbab ini menjelaskan hasil dari serangkaian uji coba yang dilakukan dan kendala yang dihadapi selama proses pengerjaan.

### 5.5.1 Evaluasi Uji Coba dengan Ukuran Blok Berbeda

Evaluasi dari uji coba ukuran blok berbeda ini yaitu, ukuran blok sangat berpengaruh dalam proses deteksi *copy-move*. Jika ukuran blok yang digunakan lebih besar maka objek *copy-move* yang kecil akan susah terdeteksi, sebaliknya jika ukuran blok yang digunakan lebih kecil, maka besar kemungkinan akan terdeteksinya objek non-*copy-move*.

Pada uji coba dengan ukuran blok 8x8, secara keseluruhan sistem dapat mendeteksi daerah yang mengalami *copy-move* dengan baik. Namun pada beberapa data terdapat beberapa daerah non-*copy-move* yang terdeteksi sebagai *copy-move*. Dari hasil uji coba ini didapatkan rata-rata akurasi, *specificity* dan *sensitifity* masing-masing 99,58%, 99,73% dan 97,33%.

Pada uji coba dengan ukuran blok 16x16, secara keseluruhan sistem dapat mendeteksi daerah yang mengalami *copy-move* dengan baik. Tidak ada daerah non-*copy-move* yang terdeteksi. Namun masih ada daerah *copy-move* pada data uji coba yang tidak



terdeteksi. Dari hasil uji coba ini didapatkan rata-rata akurasi, *specificity* dan *sensitiftiy* masing-masing 99,59%, 99,91% dan 90,21%.

Pada uji coba dengan ukuran blok 32x32, secara keseluruhan sistem dapat mendeteksi daerah yang mengalami *copy-move* dengan baik. Tidak ada daerah *non-copy-move* yang terdeteksi. Namun banyak daerah *copy-move* yang berukuran kecil tidak terdeteksi sebagai daerah *copy-move*. Dari hasil uji coba ini didapatkan rata-rata akurasi, *specificity* dan *sensitiftiy* masing-masing 98,68%, 99,94% dan 59,13%.

Dari hasil uji coba dengan tiga ukuran blok yang berbeda yaitu 8x8, 16x16 dan 32x32. Didapatkan hasil terbaik yaitu pada uji coba dengan ukuran blok 16x16.

### 5.5.2 Evaluasi Uji Coba dengan Data Variasi

Evaluasi dari uji coba dengan variasi data memberikan hasil yang optimal. Uji coba dilakukan pada 2 variasi data yaitu citra dengan penyesuaian kontras dan citra dengan *blurr*. Dalam uji coba ini parameter ukuran blok yang digunakan adalah yang menghasilkan hasil terbaik yaitu 16x16.

Pada uji coba citra dengan penyesuaian kontras, secara keseluruhan sistem dapat mendeteksi daerah yang mengalami *copy-move* dengan baik. Dari hasil uji coba ini didapatkan rata-rata akurasi, *specificity* dan *sensitiftiy* masing-masing 99,60%, 99,91% dan 90,26%.

Pada uji coba citra dengan *blurr*, secara keseluruhan sistem dapat mendeteksi daerah yang mengalami *copy-move* dengan baik. Dari hasil uji coba ini didapatkan rata-rata akurasi, *specificity* dan *sensitiftiy* masing-masing 99,10%, 100,00% dan 79,77%.

***[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. Metode *Histogram of Oriented Gradients* (HOG) dapat digunakan untuk mendeteksi daerah pada citra yang mengalami *copy-move*.
2. Hasil terbaik yang didapat dari proses deteksi adalah ketika blok yang digunakan berukuran 16x16 piksel.
3. Sistem deteksi *copy-move* dengan metode *Histogram of Oriented Gradients* (HOG) dapat mendeteksi citra *copy-move* yang mengalami variasi penyesuaian kontras dan *blur*.
4. Dari uji coba yang dilakukan terhadap 20 citra menunjukkan bahwa metode ini dapat memberikan hasil deteksi *copy-move* yang akurat dengan rata-rata akurasi, *specificity* dan *sensitivity* adalah 99,59%, 99,91%, dan 90,21%.

#### **6.2 Saran**

Saran yang dapat diberikan dalam pengujian metode deteksi *copy-move* pada citra adalah sebagai berikut:

1. Melakukan uji coba dengan ekstensi citra yang berbeda.
2. Melakukan uji coba dengan lebih banyak variasi data citra.
3. Sebaiknya sistem dapat melakukan variasi data, seperti menambah *filter* atau *noise*.

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

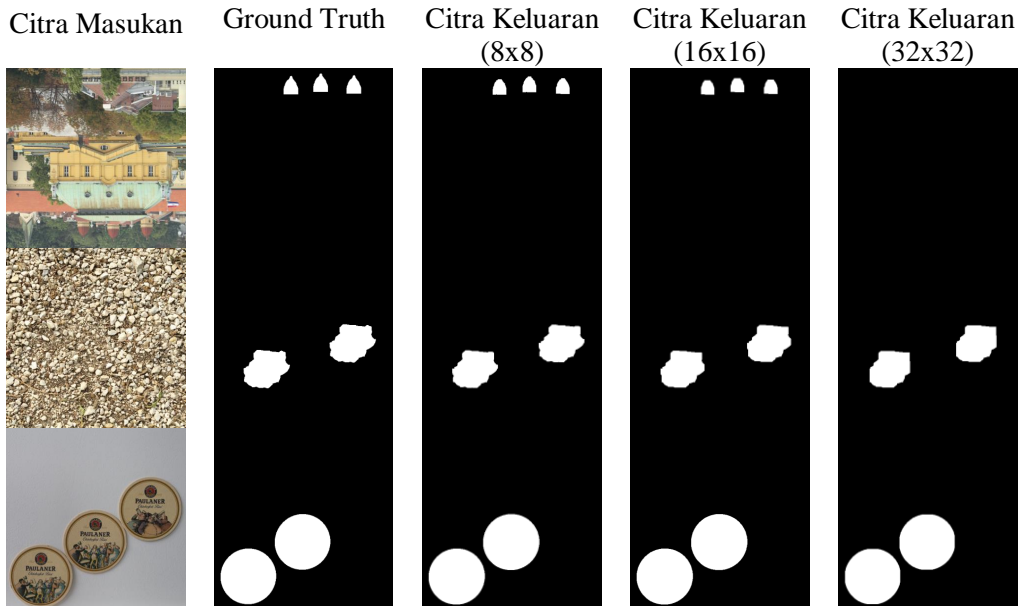
## DAFTAR PUSTAKA

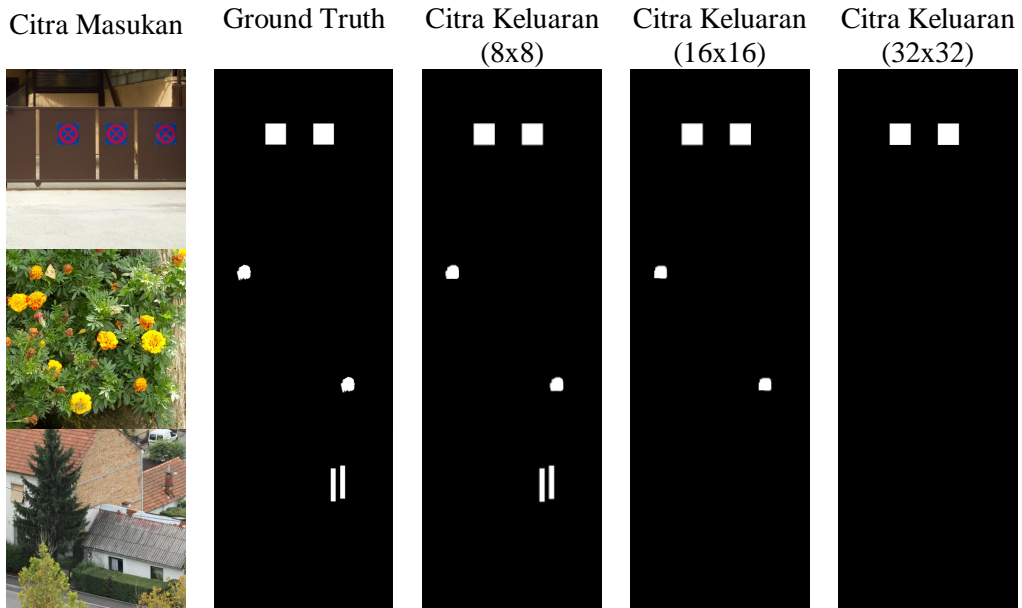
- [1] J.-C. Lee, C.-P. Chang, dan W.-K. Chen, "Detection of copy-move image forgery using histogram of orientated gradients," *Inf. Sci.*, vol. 321, hal. 250–262, Nov 2015.
- [2] D. R. Sina dan A. Harjoko, "Deteksi Copy Move Forgery Pada Citra Menggunakan Exact Match, DWT Haar dan Daubechies," *IJEIS - Indones. J. Electron. Instrum. Syst.*, vol. 6, no. 1, hal. 25–36, Apr 2016.
- [3] P. E. Kresnha, E. Susilowati, dan Y. Adharani, "Pendeteksian Manipulasi Citra Berbasis Copy-move Forgery Menggunakan Euclidean Distance dengan Single Value Decomposition," *Semin. Nas. Teknol. Inf. Dan Multimed. 2016*, Feb 2016.
- [4] N. Dalal dan B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05)*, 2005, vol. 1, hal. 886–893.
- [5] "Histogram of Oriented Gradients (HOG) Descriptor | Intel® Software." [Daring]. Tersedia pada: <https://software.intel.com/en-us/node/529070>. [Diakses: 12-Des-2016].
- [6] "CoMoFoD." [Daring]. Tersedia pada: <http://www.vcl.fer.hr/comofod/>. [Diakses: 16-Des-2016].
- [7] D. Putra, *Pengolahan Citra Digital*. Penerbit Andi.
- [8] "Implementasi Metode Histogram Of Oriented Gradients Dengan Optima | 123dok." [Daring]. Tersedia pada: <https://text-id.123dok.com/document/ky6mgm4q-implementasi-metode-histogram-of-oriented-gradients-dengan-optimalisasi-algoritma-frei-chen-untuk-deteksi-citra-manusia.html>. [Diakses: 22-Jun-2017].
- [9] J. Kazemier dan M. Heijkoop, "Digital Image Forensics," *SC RUG 2011 Proc.*, hal. 91.
- [10] G. Lynch, F. Y. Shih, dan Hong-Yuan Mark Liao, "An efficient expanding block algorithm for image copy-move forgery detection," *Inf. Sci.*, vol. 239, hal. 253–265, Mar 2013.

- [11]Zulfan, F. Arnia, dan R. Muharar, “Deteksi Pemalsuan Citra dengan Teknik Copy-Move Menggunakan Metode Ordinal Measure dari Koefisien Discrete Cosine Transform,” *J. Nas. Tek. ELEKTRO*, vol. 5, no. 2, Mar 2016.
- [12]N. B. A. Warif *et al.*, “Copy-move forgery detection: Survey, challenges and future directions,” *J. Netw. Comput. Appl.*, vol. 75, hal. 259–278, Nov 2016.
- [13]A. F. Randa, N. Suciati, dan D. A. Navastara, “Implementasi Metode Kombinasi Histogram Of Oriented Gradients Dan Hierarchical Centroid Untuk Sketch Based Image Retrieval,” vol. Vol. 5, No. 2, (2016), hal. A3111–A316.
- [14]C. Tomasi, “Histogram of Oriented Gradients.”
- [15]“MATLAB - MathWorks.” [Daring]. Tersedia pada: <https://www.mathworks.com/products/matlab.html>. [Diakses: 28-Apr-2017].
- [16]“Confusion Matrix.” [Daring]. Tersedia pada: [http://www2.cs.uregina.ca/~dbd/cs831/notes/confusion\\_matrix/confusion\\_matrix.html](http://www2.cs.uregina.ca/~dbd/cs831/notes/confusion_matrix/confusion_matrix.html). [Diakses: 30-Mei-2017].

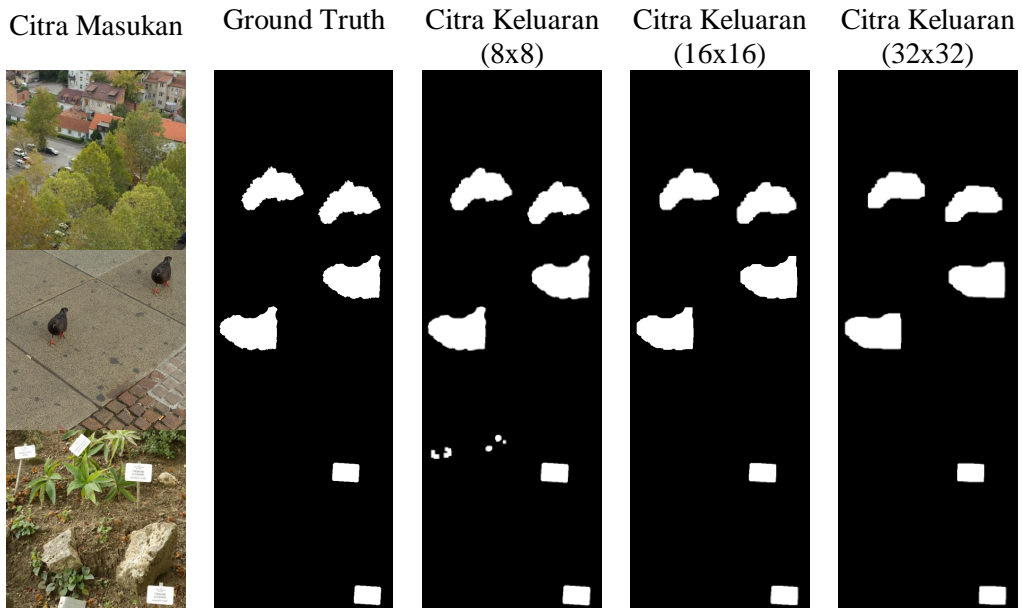
## LAMPIRAN

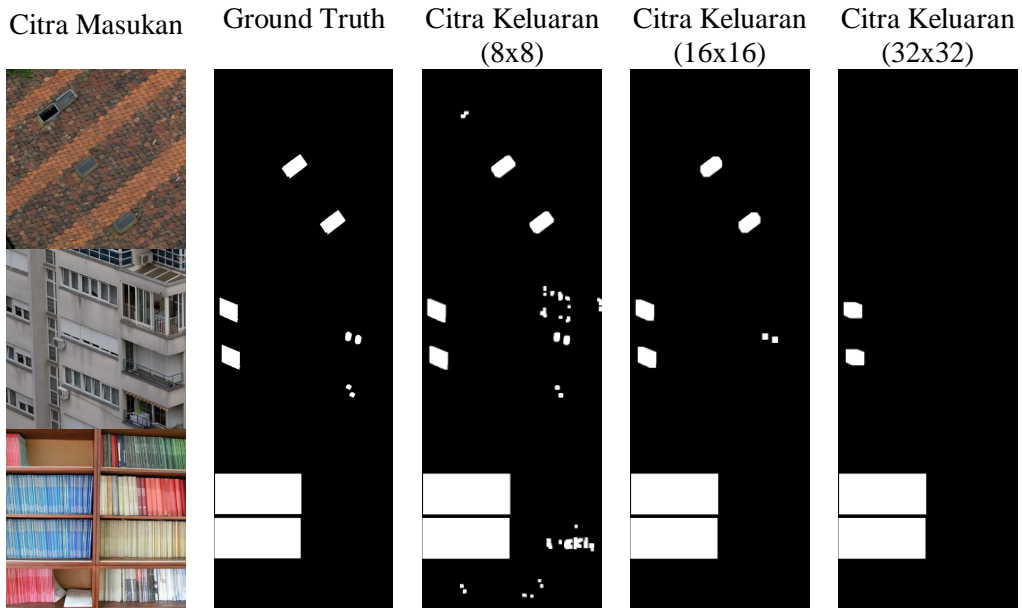
### A. Hasil Uji Coba Parameter Ukuran Blok

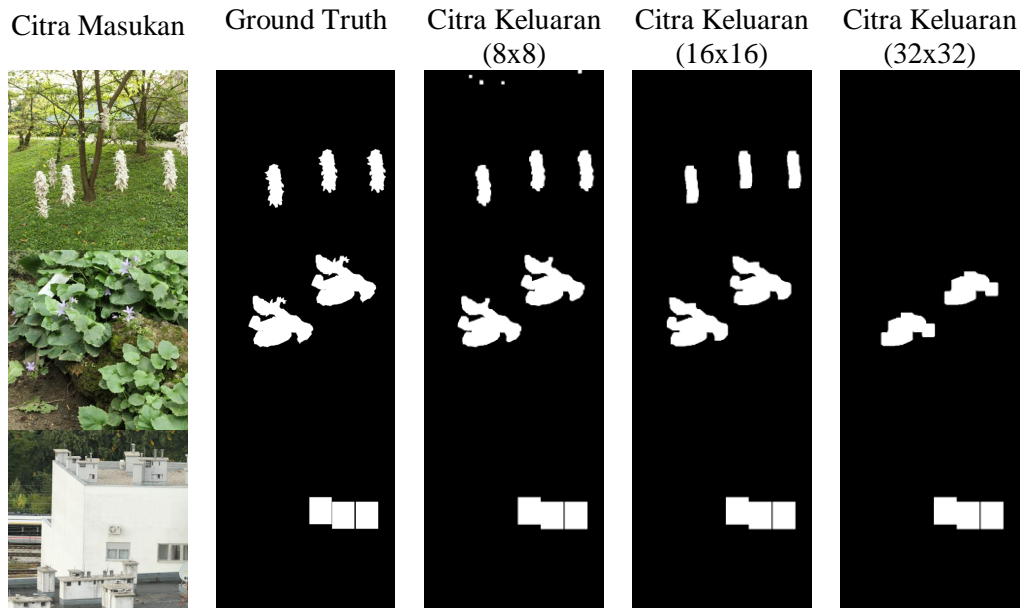


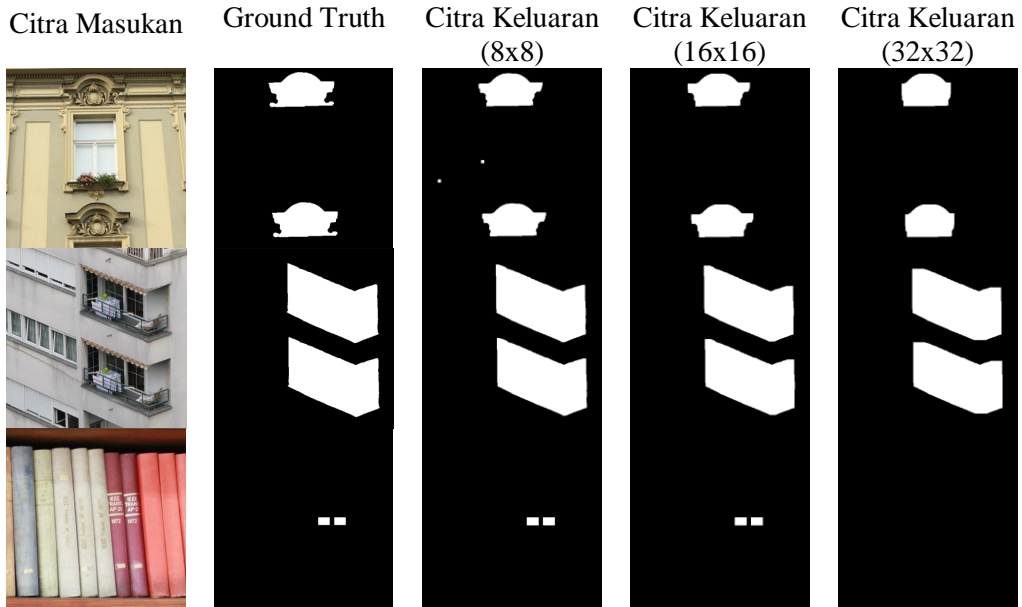








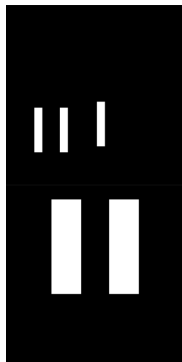
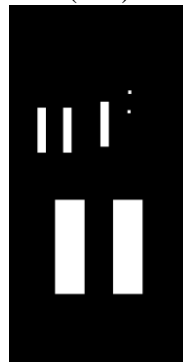










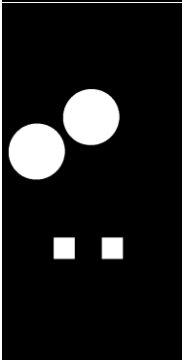
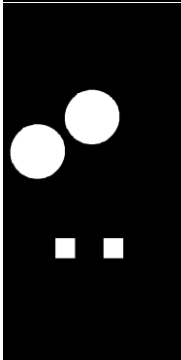


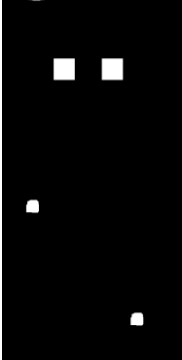
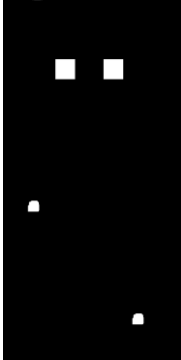


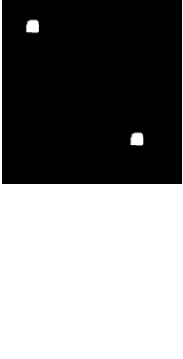

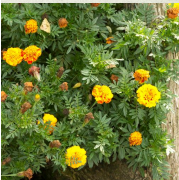
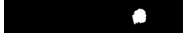

Citra Masukan



Ground Truth

Citra Keluaran  
(8x8)Citra Keluaran  
(16x16)Citra Keluaran  
(32x32)

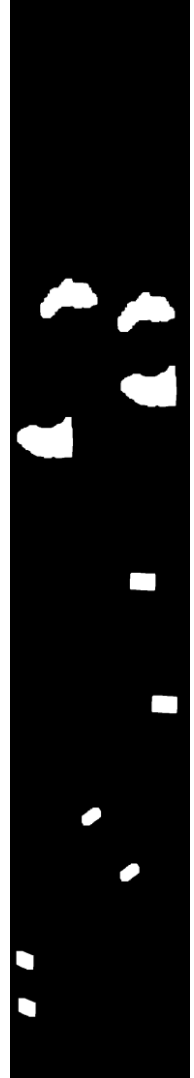
**B. Hasil Uji Coba Parameter Variasi Data Citra**

Citra Masukan	Ground Truth	Citra Keluaran ( <i>Contrast Adjustments</i> )	Citra Keluaran ( <i>Image Blurring</i> )			
						
						
						
						
						

Citra Masukan



Ground Truth

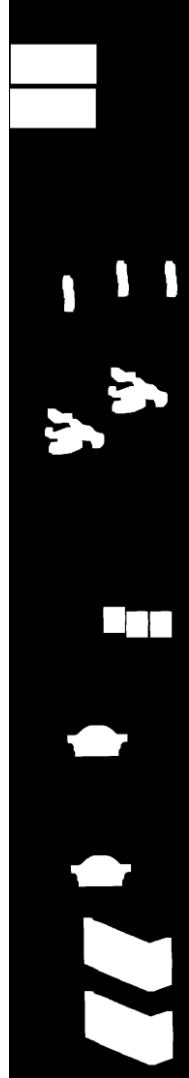
Citra Keluaran  
(*Contrast Adjustments*)Citra Keluaran  
(*Image Blurring*)

Citra Masukan

Ground Truth

Citra Keluaran  
(*Contrast Adjustments*)

Citra Keluaran  
(*Image Blurring*)

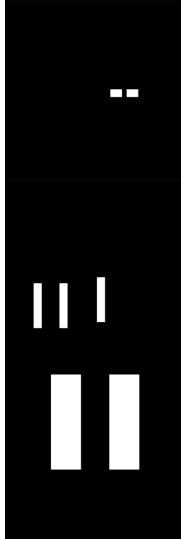




Citra Masukan



Ground Truth

Citra Keluaran  
(*Contrast  
Adjustments*)Citra Keluaran  
(*Image  
Blurring*)

## BIODATA PENULIS



Ni Luh Made Asri Mulyasari, merupakan anak ke-2 dari pasangan Bapak I Wayan Sunata dan Ibu Ni Made Seriesih yang lahir di Tabanan, pada tanggal 28 April 1995. Penulis menempuh pendidikan mula dari TK Berdikari Gadungan (2000), SD Negeri 4 Gadungan (2001-2007), SMP Negeri 3 Selemadeg Timur (2007-2010), SMA Negeri 1 Tabanan (2010-2013) dan S1 Jurusan Teknik Informatika di Institut Teknologi Sepuluh Nopember Surabaya, Fakultas Teknologi Informasi angkatan tahun 2013.

Penulis memiliki pengalaman organisasi sebagai Staf Departemen Pengembangan Profesi Himpunan Mahasiswa Teknik Computer-Informatika ITS 2014-2015, Staf Departemen Seni Tim Pembina Kerohanian Hindu ITS 2014-2015, Sekretaris Internal Badan Eksekutif Mahasiswa Fakultas Teknologi Informasi ITS 2015-2016 dan Wakil Kepala Departemen Pengabdian Masyarakat Tim Pembina Kerohanian Hindu ITS 2015-2016. Penulis juga memiliki pengalaman kepanitiaan sebagai Sie Kerohanian TPKH Games 2014, Staf Dana Usaha Schematics 2014 dan 2015, Sie Publikasi dan Dokumentasi Bakti Sosial TPKH-ITS 2015, Kesekretariatan Seminar *How To Build Start Up As Global Business* HMTC-ITS 2015.

Penulis mengambil bidang minat Komputasi Cerdas dan Visi (KCV). Komunikasi dengan penulis dapat melalui email ke **[madeasrimulyasari@gmail.com](mailto:madeasrimulyasari@gmail.com)**.