

Implementasi Deteksi *Copy-move Forgery* pada Citra menggunakan Metode Histogram of Oriented Gradients (HOG)

Ni Luh Made Asri Mulyasari, Nanik Suciati, Arya Yudhi Wijaya

Departemen Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember (ITS)

Jl. Arief Rahman Hakim, Surabaya 60111 Indonesia

e-mail: nanik@if.its.ac.id

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 properti-properti seperti *noise*, warna dan tekstur akan kompatibel dengan seluruh citra dan dengan demikian menjadi lebih sulit untuk dibedakan dan deteksi secara kasat mata. Penelitian 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.

I. PENDAHULUAN

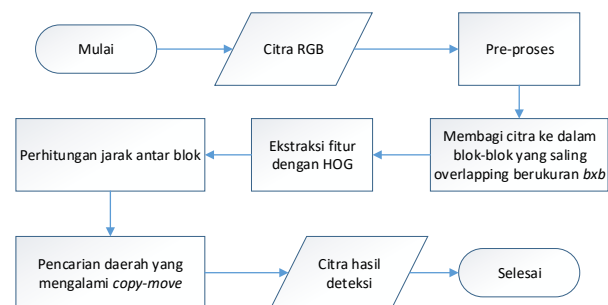
SEMAKIN berkembangnya teknologi informasi menyebabkan semakin banyaknya aplikasi yang diciptakan untuk memenuhi kegiatan baik untuk pembelajaran, pekerjaan, maupun hiburan. Salah satu aplikasi yang banyak dikembangkan yaitu aplikasi pengolahan citra. Dengan banyaknya aplikasi pengolahan citra tersebut, 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 [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.



Gambar 1. Diagram Alir Sistem

II. METODOLOGI

Metode yang digunakan dalam penelitian ini adalah *Histogram of Oriented Gradients* (HOG) untuk proses ekstraksi fitur dan *Euclidean distance* untuk perhitungan jarak. Terdapat lima proses dalam penelitian ini seperti ditunjukkan Gambar 1.

A. Pre-processing

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.

B. 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 bxb. Bila citra grayscale berukuran MxN maka akan ada sebanyak (M-b+1)x(N-b+1) blok yang saling overlap. Blok-blok ini selanjutnya diekstraksi dengan metode HOG.

C. 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. Proses ekstraksi fitur dengan HOG ditunjukkan pada Gambar 2. Ekstraksi fitur dalam studi ini menggunakan data keluaran preprocessing.

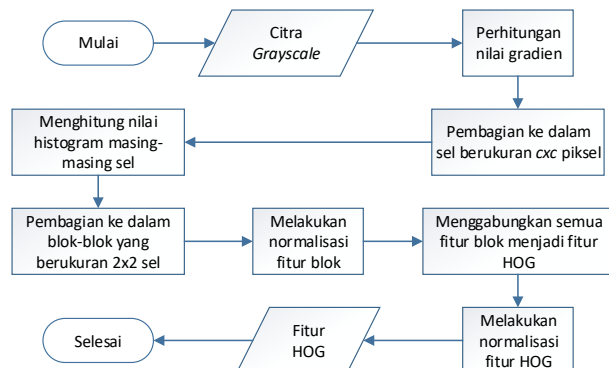
Terdapat empat tahapan dalam melakukan ekstraksi fitur dengan *Histogram of Oriented Gradients* ini yaitu pertama melakukan perhitungan gradien untuk pada citra masukan. Perhitungan gradien dilakukan dengan cara konvolusi dengan operator gradien [-1 0 1] Kemudian melakukan transformasi gradien ke dalam koordinat sumbu dengan sudut diantara 0° sampai 180° yang disebut dengan orientasi gradien.

Kedua dilakukan pembagian citra ke dalam sel yang berukuran 8x8 piksel. Kemudian citra dibagi lagi menjadi blok-blok yang berukuran 2x2 sel.

Ketiga dilakukan perhitungan nilai histogram pada setiap sel. Fitur blok didapatkan dari rangkaian 4 histogram dari 4 sel yang menyusun blok. Kemudian dilakukan normalisasi pada setiap fitur blok yang telah didapatkan.

Keempat dilakukan penggabungan dari hasil normalisasi semua fitur blok menjadi satu fitur HOG. Kemudian fitur HOG ini dinormalisasi lagi dengan *euclidean norm*.

Hasil dari proses ekstraksi fitur dengan HOG ini berupa matriks yang berukuran (M-b+1)x(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.



Gambar 2. Diagram Alir Proses HOG

D. Perhitungan Jarak menggunakan Euclidean Distance

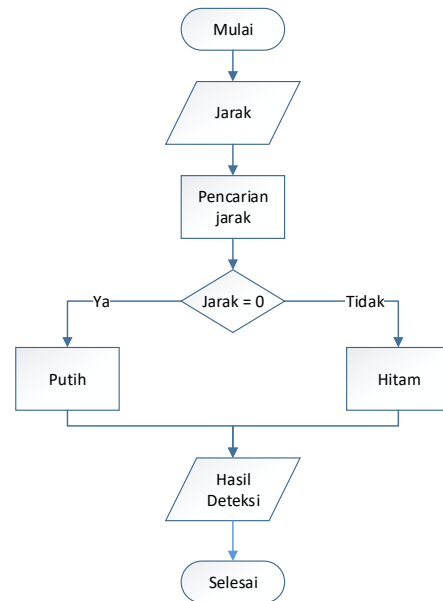
Melakukan identifikasi terhadap blok-blok yang memiliki kemiripan fitur. Terlebih dahulu dilakukan pengurutan fitur vektor secara leksikografis. Pengurutan secara leksikografis merupakan pengurutan berdasarkan urutan kamus. Hal ini dilakukan agar fitur-fitur yang memiliki kemiripan berada pada tempat 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* (1) dari fitur vektor. Dimana variabel *p* dan *q* merupakan dua titik yang akan dihitung jaraknya, sedangkan *d* menunjukkan nilai jarak dari kedua titik yang dihitung.

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

E. 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.



Gambar 3. Diagram Alir Proses Pencarian Daerah Copy-move

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 berdekatan. Jika ditemukan blok dengan fitur yang mirip dengan jarak=0, maka blok itu akan ditandai sebagai daerah *copy-move* dengan memberikan warna putih. Proses pencarian daerah *copy-move* ini ditunjukkan pada Gambar 3.

III. UJI COBA DAN AVALUASI

Skenario uji coba dilakukan untuk mengetahui persentase akurasi, *specificity* dan *sensitifity* dari citra hasil deteksi. 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*. *Sensitifity* menunjukkan persentase hasil deteksi objek *copy-move* dalam hal ini daerah yang berwarna putih dibandingkan dengan *ground truth*.

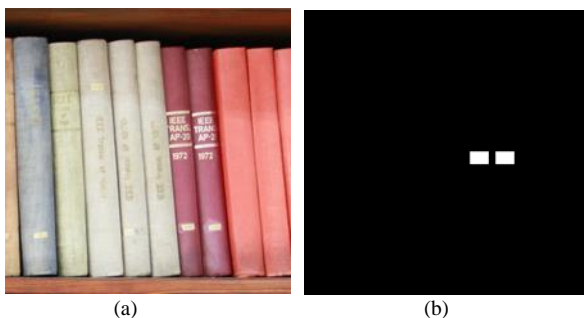
A. Skenario Uji Coba Ukuran Blok Berbeda

Dalam uji coba ini digunakan 3 ukuran blok berbeda yaitu 8x8, 16x16 dan 32x32. Tabel 1 menunjukkan hasil uji coba dengan parameter ukuran blok 8x8. Terlihat rata-rata akurasi 99,58%, rata-rata *specificity* sebesar 99,73%, dan rata-rata *sensitifity* 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*.

Tabel 1. Hasil Uji Coba Performa Dengan Ukuran Blok 8x8

Nama Citra	Akurasi (%)	Specificity (%)	Sensitifity (%)
001_F.png	99,73	99,96	87,12
002_F.png	99,80	99,98	97,81
003_F.png	99,76	99,71	100,00
004_F.png	99,91	99,91	100,00
005_F.png	99,93	99,96	96,49
006_F.png	99,92	99,92	100,00
007_F.png	99,69	99,77	98,96
008_F.png	99,76	99,77	99,71
009_F.png	99,30	99,28	100,00
010_F.png	99,77	99,80	98,26
011_F.png	99,13	99,15	98,42
012_F.png	98,52	98,11	100,00
013_F.png	99,26	99,83	88,29
014_F.png	99,40	99,97	95,21
015_F.png	99,84	99,83	100,00
016_F.png	98,86	99,93	89,66
017_F.png	99,50	99,99	98,27
018_F.png	99,96	99,96	100,00
019_F.png	99,76	99,75	100,00
020_F.png	99,73	100,00	98,46
Rata-rata	99,58	99,73	97,33

Gambar 4 menunjukkan hasil deteksi 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 4. Hasil Uji Coba dengan Ukuran Blok 8x8 (a) Citra Masukan, (b) Citra Hasil Deteksi

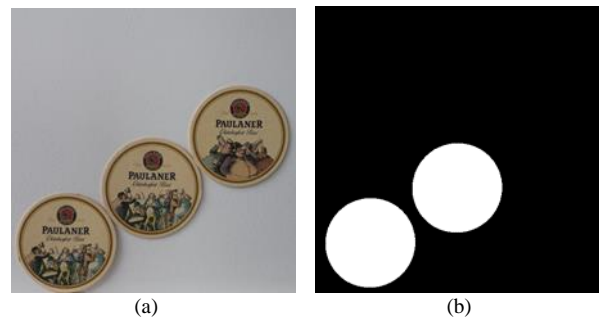
Tabel 2 menunjukkan hasil uji coba dengan parameter ukuran blok 16x16. Terlihat rata-rata akurasi 99,59%, rata-rata *specificity* sebesar 99,91%, dan rata-rata *sensitifity* sebesar 90,21%. Dilihat dari nilai persentase tersebut, hasil uji coba

parameter blok dengan ukuran 16x16 piksel menghasilkan hasil deteksi yang optimal.

Tabel 2. Hasil Uji Coba Performa dengan Ukuran Blok 16x16

Nama Citra	Akurasi (%)	Specificity (%)	Sensitifity (%)
001_F.png	99,67	99,97	83,78
002_F.png	99,68	99,98	96,13
003_F.png	99,76	99,71	100,00
004_F.png	99,91	99,91	100,00
005_F.png	99,88	99,97	90,02
006_F.png	99,00	100,00	0,00
007_F.png	99,63	99,84	97,55
008_F.png	99,72	99,81	98,95
009_F.png	99,91	99,91	99,69
010_F.png	99,80	99,93	93,49
011_F.png	99,63	99,92	87,86
012_F.png	99,72	99,65	100,00
013_F.png	99,01	100,00	80,04
014_F.png	99,00	99,98	91,72
015_F.png	99,84	99,83	100,00
016_F.png	98,81	100,00	88,64
017_F.png	99,40	99,99	97,92
018_F.png	99,96	99,96	100,00
019_F.png	99,83	99,82	100,00
020_F.png	99,73	100,00	98,46
Rata-rata	99,59	99,91	90,21

Gambar 5 menunjukkan hasil deteksi pada uji coba dengan parameter ukuran blok 16x16 piksel. Daerah yang mengalami *copy-move* adalah dua buak koin pada citra masukan.



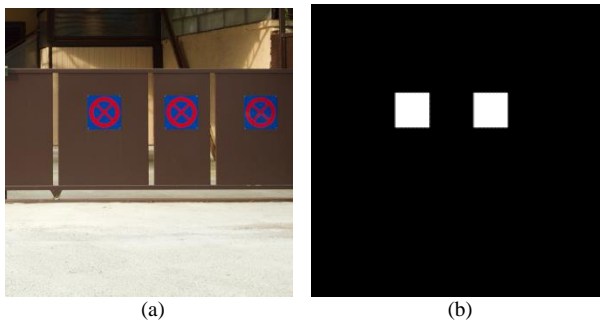
Gambar 5. Hasil Uji Coba dengan Ukuran Blok 16x16 (a) Citra Masukan, (b) Citra Hasil Deteksi

Tabel 3 menunjukkan hasil uji coba dengan parameter ukuran blok 32x32. 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.

Tabel 3. Hasil Uji Coba Performa dengan Citra Kabur (*Image Blurring*)

Nama Citra	Akurasi (%)	Specificity (%)	Sensitifity (%)
001_F_CA3	99,68	99,97	84,02
002_F_CA3	99,68	99,98	96,13
003_F_CA3	99,75	99,71	100,00
004_F_CA3	99,91	99,91	100,00
005_F_CA3	99,88	99,97	90,02
006_F_CA3	99,00	100,00	0,00
007_F_CA3	99,63	99,84	97,55
008_F_CA3	99,72	99,81	98,97
009_F_CA3	99,91	99,91	99,69
010_F_CA3	99,80	99,93	93,49
011_F_CA3	99,64	99,92	88,33
012_F_CA3	99,72	99,65	100,00
013_F_CA3	99,01	100,00	80,19
014_F_CA3	99,00	99,98	91,75
015_F_CA3	99,84	99,83	100,00
016_F_CA3	98,82	100,00	88,70
017_F_CA3	99,41	99,99	97,96
018_F_CA3	99,96	99,96	100,00
019_F_CA3	99,83	99,82	100,00
020_F_CA3	99,73	100,00	98,46
Rata-rata	99,60	99,91	90,26

Gambar 6 menunjukkan hasil deteksi pada uji coba dengan parameter ukuran blok 32x32 piksel. Daerah yang mengalami *copy-move* adalah kotak pertama dan kedua pada citra masukan.



Gambar 6. Hasil Uji Coba dengan Ukuran Blok 32x32 (a) Citra Masukan, (b) Citra Hasil Deteksi

B. Skenario Uji Coba dengan Variasi Data

Skenario uji coba pertama adalah perhitungan akurasi, *specificity* dan *sensitifity* dengan data citra yang divariasasi. Data yang digunakan adalah data citra dengan variasi penyesuaian kontras dan *blur*. Data uji coba ini diunduh dari CoMoFod *database* [6] yang sudah megalami penyesuaian kontras dan *blur* (kabur). Parameter ukuran blok yang digunakan adalah 16x16 piksel karena dalam uji coba paramer ukuran blok memberikan hasil terbaik.

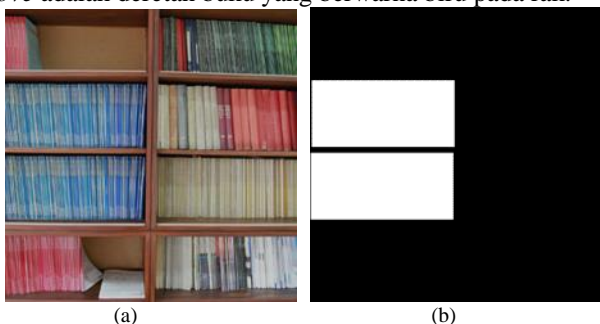
Tabel 4 menunjukkan hasil uji coba dengan data citra yang mengalami penyesuaian kontras. Terlihat rata-rata akurasi 99,60%, rata-rata *specificity* sebesar 99,91%, dan rata-rata *sensitifity* sebesar 90,26%. Dilihat dari nilai persentase tersebut, hasil uji coba dengan variasi data citra yang mengalami penyesuaian kontras menghasilkan hasil deteksi yang optimal.

Tabel 4.

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

Nama Citra	Akurasi (%)	Specificity (%)	Sensitifity (%)
001_F_CA3	99,68	99,97	84,02
002_F_CA3	99,68	99,98	96,13
003_F_CA3	99,75	99,71	100,00
004_F_CA3	99,91	99,91	100,00
005_F_CA3	99,88	99,97	90,02
006_F_CA3	99,00	100,00	0,00
007_F_CA3	99,63	99,84	97,55
008_F_CA3	99,72	99,81	98,97
009_F_CA3	99,91	99,91	99,69
010_F_CA3	99,80	99,93	93,49
011_F_CA3	99,64	99,92	88,33
012_F_CA3	99,72	99,65	100,00
013_F_CA3	99,01	100,00	80,19
014_F_CA3	99,00	99,98	91,75
015_F_CA3	99,84	99,83	100,00
016_F_CA3	98,82	100,00	88,70
017_F_CA3	99,41	99,99	97,96
018_F_CA3	99,96	99,96	100,00
019_F_CA3	99,83	99,82	100,00
020_F_CA3	99,73	100,00	98,46
Rata-rata	99,60	99,91	90,26

Gambar 7 menunjukkan hasil dari uji coba dengan variasi citra penyesuaian kontras. Daerah citra yang mengalami *copy-move* adalah deretan buku yang berwarna biru pada rak.



Gambar 7. Hasil Uji Coba dengan Citra Penyesuaian Kontras (a) Citra Masukan, (b) Citra Hasil Deteksi

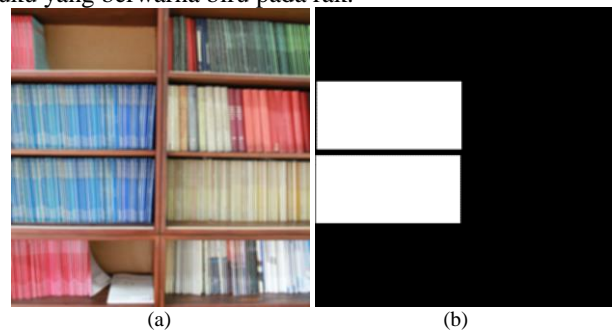
Tabel 5 menunjukkan hasil uji coba dengan data citra yang mengalami *blur*. Terlihat rata-rata akurasi 99,10%, rata-rata *specificity* sebesar 100%, dan rata-rata *sensitifity* sebesar 79,77%. Dilihat dari nilai persentase tersebut, hasil uji coba dengan variasi data citra yang mengalami *blur* menghasilkan hasil deteksi yang optimal.

Tabel 5.

Hasil Uji Coba Performa dengan Ukuran Blok 32x32

Nama Citra	Akurasi (%)	Specificity (%)	Sensitifity (%)
001_F.png	98,18	100,00	0,00
002_F.png	99,49	99,99	93,66
003_F.png	99,75	99,76	99,70
004_F.png	99,91	99,91	100,00
005_F.png	99,02	100,00	0,00
006_F.png	99,00	100,00	0,00
007_F.png	99,19	99,89	92,09
008_F.png	99,23	99,88	93,94
009_F.png	99,87	99,93	98,22
010_F.png	98,07	100,00	0,00
011_F.png	99,18	99,95	68,53
012_F.png	99,72	99,65	100,00
013_F.png	95,03	100,00	0,00
014_F.png	95,27	99,99	60,07
015_F.png	99,84	99,83	100,00
016_F.png	98,06	100,00	81,40
017_F.png	99,04	99,99	96,63
018_F.png	99,45	100,00	0,00
019_F.png	96,66	100,00	0,00
020_F.png	99,73	100,00	98,46
Rata-rata	98,68	99,94	59,13

Gambar 8 menunjukkan hasil dari uji coba dengan citra *blur*. Daerah citra yang mengalami *copy-move* adalah deretan buku yang berwarna biru pada rak.



Gambar 8. Hasil Uji Coba dengan Citra Kabur (a) Citra Masukan, (b) Citra Hasil Deteksi

IV. 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%.

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].