

DETEKSI PEMALSUAN *COPY-MOVE* PADA CITRA DIGITAL MENGUNAKAN *SCALE INVARIANT FEATURE TRANSFORM (SIFT)* DAN *GENERALIZED 2 NEAREST-NEIGHBOR (G2NN)*

Muhammad Salman Lubis^{*)}, Ajub Ajulian Zahra, Imam Santoso

Departemen Teknik Elektro, Universitas Diponegoro Semarang
Jl. Prof. Sudharto, SH, Kampus UNDIP Tembalang, Semarang 50275, Indonesia

^{*)}E-mail: lubissalman@gmail.com

Abstrak

Copy-move adalah jenis pemalsuan citra digital dengan cara menyalin suatu wilayah atau segmen pada citra lalu meletakkannya pada daerah lain pada citra yang sama. Oleh karena pentingnya citra sebagai media komunikasi bagi manusia, diperlukan metode deteksi pemalsuan copy-move untuk mengetahui tingkat keotentikan sebuah citra digital. Pada penelitian ini dirancang perangkat lunak simulasi sistem pendeteksi pemalsuan copy-move pada citra digital. Proses deteksi diawali dengan tahap prapengolahan dilanjutkan tahap ekstraksi ciri dari citra uji menggunakan Scale Invariant Feature Transform (SIFT). Tahap terakhir adalah proses deteksi pemalsuan dengan melakukan pencocokan ciri menggunakan Generalized 2 Nearest-Neighbor dengan 3 variasi nilai ambang T, yaitu 0,3, 0,4, dan 0,5. Berdasarkan pengujian dengan variasi jumlah titik kunci minimal yang cocok sebanyak 2, 5, dan 10, tingkat akurasi tertinggi pengujian tanpa serangan derau Gaussian adalah 100% dengan nilai ambang T sebesar 0,4 dan titik kunci minimal yang cocok sebanyak 2 titik kunci. Pada pengujian dengan derau Gaussian didapat nilai persentase akurasi tertinggi sebesar 65% dengan nilai ambang T sebesar 0,4 dan 0,5 serta titik kunci minimal yang cocok sebanyak 2 titik kunci.

Kata Kunci: Copy-Move, Derau Gaussian, SIFT, G2NN

Abstract

Copy-move is type of digital image forgery by copying an area or segment on the image and then placing it in another area in the same image. Because of the importance of image as communication medium for humans, a method is needed to detect copy-move forgery to determine the authenticity of a digital image. In this research, a copy-move forgery detection system simulation software on digital image is designed. The detection process begins with the pre-processing stage and then the feature extraction stage from the test image using Scale Invariant Feature Transform (SIFT). The last step is the forgery detection process by matching the features obtained with Generalized 2 Nearest-Neighbor with 3 variations of threshold value T 0,3, 0,4, and 0,5. Based on the testing with variations in number of minimum matched keypoints as many as 2, 5 and 10, the highest accuracy percentage of testing without Gaussian noise attack obtained is 100% with threshold value T 0,4 and minimum number of matched keypoint as many as 2 keypoints. In testing with Gaussian noise, the highest accuracy percentage obtained is 65% with threshold values T 0,4 and 0,5 and minimum number of matched keypoint as many as 2 keypoints.

Keywords: Copy-Move, Gaussian Noise, SIFT, G2NN

1. Pendahuluan

Cepatnya perkembangan teknologi membuat manipulasi informasi digital semakin mudah. Bentuk informasi digital yang sering dimanipulasi adalah citra digital. Untuk mendeteksi adanya manipulasi pada citra digital, diperlukan suatu metode forensik digital [1]. Salah satu bentuk manipulasi citra digital yang sering dilakukan adalah manipulasi *copy-move*.

Manipulasi *copy-move* adalah metode manipulasi citra digital dengan cara melakukan penggandaan (*copy*) terhadap sebuah objek pada citra digital dan menyimpannya pada objek yang ingin ditutupi (*move*) pada citra yang sama [2].

Beberapa metode telah digunakan sebagai pendeteksi pemalsuan *copy-move* pada citra. Metode tersebut antara lain dengan metode *Discrete Cosine Transform* [3], *Principal Component Analysis* [4], *Robust Detection* [5], dan lain sebagainya.

Pada tahun 1999, David Lowe [6] memperkenalkan metode pendeteksi dan pendeskripsi ciri pada citra digital yang disebut *Scale Invariant Feature Transform* (SIFT). Ciri yang dihasilkan dari metode ini tidak berubah terhadap perlakuan rotasi, translasi, dan penskalaan. Ciri yang didapatkan dari metode SIFT kemudian dapat dicocokkan dengan ciri dari citra lain yang juga didapat dengan metode SIFT untuk mendeteksi adanya kesamaan antara citra-citra tersebut. Proses ini disebut dengan proses pencocokan atau *matching*. Pada tahun 2011, Amerini dkk [7] menyarankan metode pencocokan ciri baru yang disebut metode *Generalized 2 Nearest-Neighbor*.

Metode SIFT dapat digunakan untuk mendapatkan ciri dari suatu citra yang dicurigai mengalami proses manipulasi. Ciri yang didapatkan dari citra tersebut dicocokkan antara satu segmen atau wilayah dengan wilayah lain pada citra yang sama dengan metode *Generalized 2 Nearest-Neighbor* (G2NN). Dengan menggunakan metode ini diharapkan akan menghasilkan sebuah algoritma deteksi pemalsuan *copy-move* yang akurat.

2. Metode

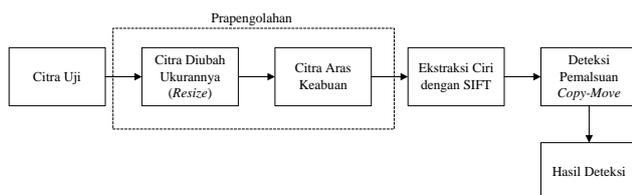
2.1. Perangkat Pendukung

Perangkat lunak yang digunakan untuk membuat simulasi perangkat lunak pendeteksi pemalsuan *copy-move* pada citra digital ini adalah Matlab R2014a, sedangkan perangkat keras yang digunakan adalah *personal computer* (PC) dengan spesifikasi sebagai berikut:

- Laptop Lenovo g470 dengan prosesor Intel Pentium Dual Core b950 (2.1 GHz).
- RAM 4GB DDR3.
- Sistem Operasi Microsoft Windows 7 Ultimate 32 bit.

2.2. Perancangan Perangkat Lunak

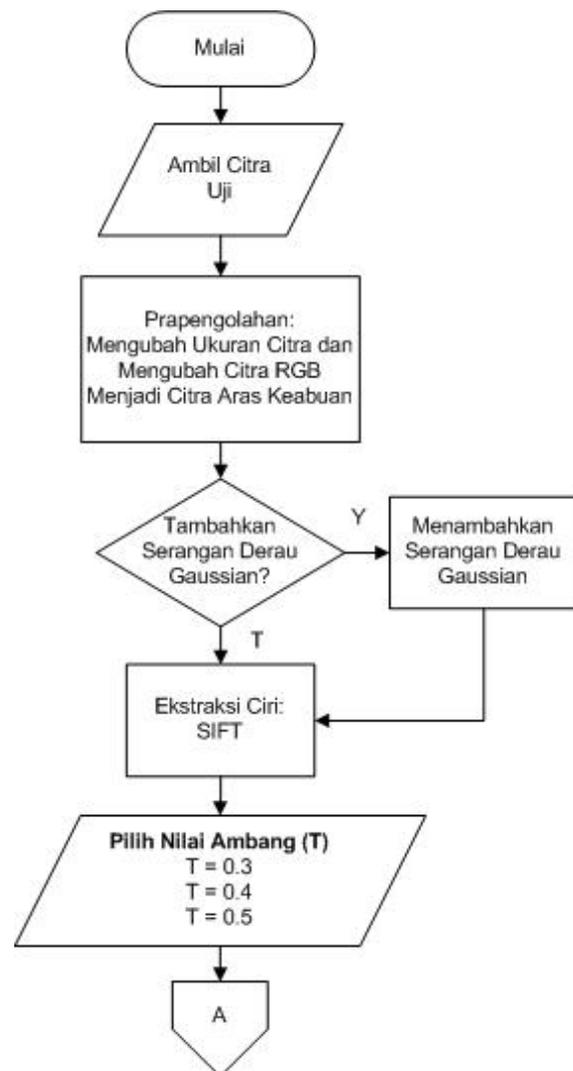
Secara garis besar, proses deteksi pemalsuan *copy-move* pada citra digital dalam simulasi ini terdiri atas 3 tahap, yaitu tahap prapengolahan, ekstraksi ciri menggunakan *Scale Invariant Feature Transform* (SIFT), dan deteksi pemalsuan *copy-move* menggunakan *Generalized 2 Nearest-Neighbor*. Gambar 1 menunjukkan diagram blok tahapan-tahapan ini.

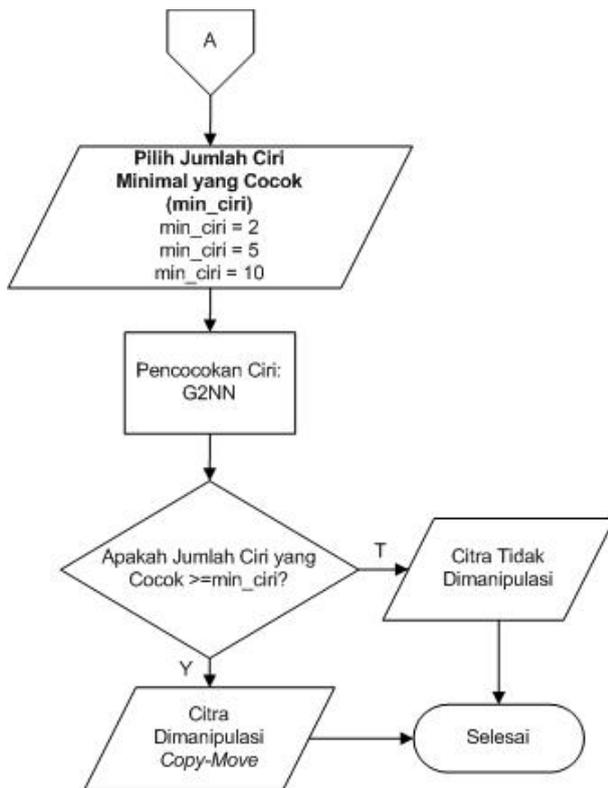


Gambar 1. Diagram blok perancangan simulasi

Pada proses prapengolahan, citra uji diubah ukurannya menjadi 400x300 piksel. Hal ini bertujuan untuk mempercepat waktu komputasi karena ukuran citra masukan yang bervariasi dan relatif besar. Setelah itu, citra diubah dari citra RGB menjadi citra aras keabuan agar proses ekstraksi ciri dengan algoritma SIFT dapat dilakukan. Hal ini dikarenakan algoritma SIFT hanya mampu dilakukan pada citra beraras keabuan.

Proses-proses yang terjadi pada simulasi perangkat lunak ini ditunjukkan secara lebih rinci pada Gambar 2.





Gambar 2. Diagram alir perancangan simulasi

Pertama-tama, citra uji melalui proses prapengolahan berupa perubahan ukuran citra dan perubahan aras citra menjadi keabuan. Terdapat pilihan untuk menambahkan gangguan berupa derau Gaussian sebelum citra melalui proses ekstraksi ciri. Penambahan derau Gaussian ini bertujuan untuk menguji pengaruh gangguan derau Gaussian terhadap akurasi pendeteksian pemalsuan *copy-move*.

Tahap selanjutnya adalah tahap ekstraksi ciri dengan menggunakan metode SIFT. Setelah itu, dilakukan proses pemilihan masukan atau input nilai ambang T. Nilai ambang T yang tersedia dalam perangkat lunak ini adalah 0,3, 0,4, dan 0,5.

Setelah masukan nilai ambang dipilih, langkah selanjutnya adalah memilih jumlah ciri minimal yang cocok. Hasil dari pemilihan jumlah ciri minimal yang cocok tersebut disimpan dalam variabel *min_ciri*. Ada tiga pilihan *min_ciri* yang tersedia yaitu *min_ciri*=2, *min_ciri*=5, *min_ciri*=10.

Setelah jumlah ciri minimal yang cocok dipilih, maka tahap selanjutnya adalah proses pencocokan ciri menggunakan G2NN. Jumlah ciri yang cocok akan menentukan apakah citra masukan mengalami pemalsuan *copy-move* atau tidak.

Pada tahap terakhir, yaitu tahap penentuan status citra, jumlah ciri yang cocok dibandingkan dengan variabel

min_ciri. Jika jumlah ciri yang cocok lebih besar atau sama dengan nilai *min_ciri*, maka citra berstatus mengalami pemalsuan *copy-move*, sedangkan jika jumlah ciri yang cocok lebih kecil dari nilai *min_ciri*, maka citra tersebut berstatus tidak mengalami pemalsuan *copy-move*.

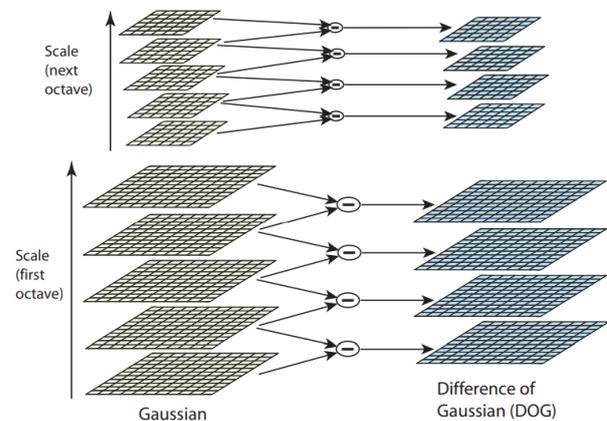
2.3. Scale Invariant Feature Transform (SIFT)

Proses yang berjalan pada metode SIFT terbagi menjadi empat tahapan, yaitu pendeteksian nilai ekstrem pada skala ruang, penentuan ciri atau titik kunci (*keypoint*), penentuan orientasi, dan pembangkitan *keypoint descriptor*.

Tahap pertama dari algoritma SIFT adalah Pendeteksian nilai ekstrem pada skala ruang. Tahap ini diperlukan untuk menentukan titik kunci dari suatu citra. Nilai ekstrem ini merupakan kandidat titik kunci (*keypoint*) dari SIFT. Skala ruang dari suatu citra didefinisikan sebagai fungsi $L(x,y,\sigma)$ yang merupakan hasil dari konvolusi antara fungsi Gaussian $G(x,y,\sigma)$ dengan citra masukan $I(x,y,\sigma)$. Setelah skala ruang dari citra didapatkan, langkah selanjutnya adalah menghitung *Difference of Gaussian* (DoG) dari citra tersebut menggunakan persamaan 1.

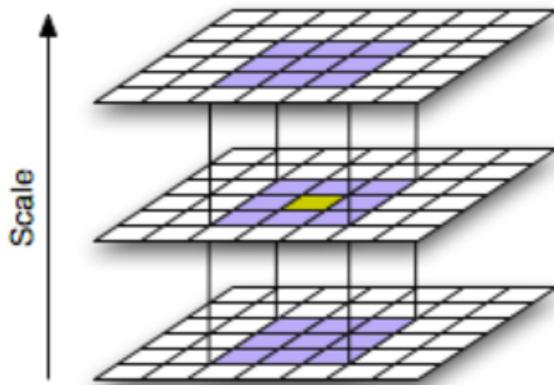
$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \quad (1)$$

Ilustrasi dari pembentukan skala ruang ditunjukkan oleh gambar 3.



Gambar 3. Ilustrasi pembentukan DoG (*Difference of Gaussian*)

Untuk mendeteksi nilai ekstrem pada citra DoG, nilai tiap piksel pada skala ruang DoG dibandingkan dengan delapan piksel yang berada di sekelilingnya dan sembilan piksel yang bersesuaian pada citra DoG sebelumnya dan setelahnya. Proses ini dapat dilihat pada Gambar 4.

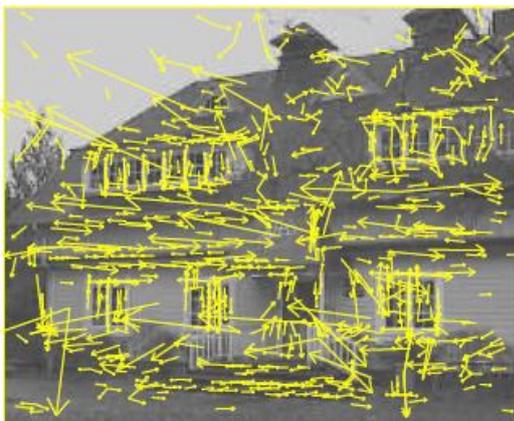


Gambar 4. Ilustrasi proses pencarian nilai ekstrem

Tahap ke dua dari metode SIFT adalah penentuan titik kunci atau *keypoint*. Lokasi titik kunci didapat dengan persamaan 2.

$$\hat{x} = -\frac{\partial^2 D^{-1}}{\partial x^2} \frac{\partial D}{\partial x} \quad (2)$$

Gambar 5 menunjukkan lokasi titik kunci awal yang merupakan nilai ekstrem pada citra.



Gambar 5. Lokasi titik kunci awal

Nilai titik kunci didapat menggunakan persamaan 3.

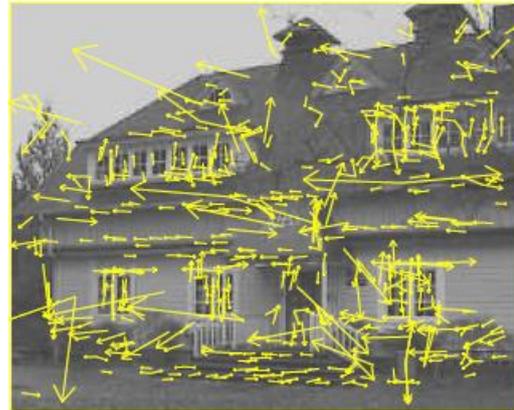
$$D(\hat{x}) = D + \frac{1}{2} \frac{\partial D^T}{\partial x} \hat{x} \quad (3)$$

Untuk menghilangkan titik kunci dengan kontras rendah, nilai ekstrem $|D(\hat{x})|$ yang lebih rendah dari 0.03 dihilangkan. Untuk menghilangkan kandidat titik kunci yang kurang jelas dan terletak sepanjang tepi, maka digunakan matriks Hessian H orde dua, seperti ditunjukkan pada persamaan 4.

$$H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix}$$

$$\frac{D_{xx}+D_{yy}}{D_{xx}D_{yy}-(D_{xy})^2} < \frac{(r+1)^2}{r} \quad (4)$$

dengan r merupakan ambang batas dari *principal curvature* yang diizinkan. Titik kunci yang mempunyai nilai *principal curvature* lebih besar dari $r = 10$ akan dihilangkan. Gambar 6 menunjukkan titik kunci setelah kandidat titik kunci dengan kontras rendah, kurang jelas, dan terletak sepanjang tepi dihilangkan.



Gambar 6. Lokasi titik kunci akhir

Tahap ke tiga dari metode SIFT adalah penentuan orientasi yang bertujuan untuk mendapatkan sifat tak berubah atau *invariant* terhadap perlakuan rotasi. Penentuan orientasi dilakukan dengan persamaan 5 dan 6.

$$m(x, y) = \sqrt{L(x+1, y) - L(x-1, y) + L(x, y+1) - L(x, y-1)} \quad (5)$$

$$\theta(x, y) = \tan^{-1} \left(\frac{L(x, y+1) - L(x, y-1)}{L(x+1, y) - L(x-1, y)} \right) \quad (6)$$

Tahap terakhir dari metode SIFT adalah pembangkitan *keypoint descriptor*. Pada tahap ini, masing-masing titik kunci atau *keypoint* yang telah diberikan orientasi diberikan ciri khusus (*descriptor*). Proses ini bertujuan untuk mendapatkan titik kunci yang tak berubah atau *invariant* terhadap perubahan intensitas cahaya atau perubahan sudut pandang tiga dimensi. *Descriptor* merupakan histogram orientasi pada piksel dengan ukuran 4x4. Histogram ini dihitung dari *magnitude* dan nilai orientasi dari sampel dalam wilayah 16x16 di sekitar titik kunci. *Magnitude* dihitung dengan fungsi Gaussian dengan σ sama dengan satu setengah lebar *descriptor*. Hasil akhir dari tahapan ini adalah sebuah vektor dengan jumlah elemen sebanyak $4 \times 4 \times 8 = 128$ elemen.

2.4. Generalized 2 Nearest-Neighbor

Setelah proses ekstraksi ciri dengan menggunakan metode SIFT selesai, maka akan didapatkan sejumlah n-titik kunci dan *descriptor*-nya masing-masing. Langkah selanjutnya adalah melakukan proses pencocokan antar

descriptor pada tiap titik kunci untuk mengidentifikasi segmen yang sama pada citra uji.

Dimisalkan metode SIFT diaplikasikan pada sebuah citra uji sehingga menghasilkan kumpulan titik kunci atau *keypoint* $X = \{x_1, \dots, x_n\}$ dengan masing-masing *descriptor* $\{f_1, \dots, f_n\}$. Metode 2NN diaplikasikan pada tiap f_i dari masing-masing titik kunci. Cara untuk mencari kecocokan pada tiap titik kunci dengan metode 2NN adalah dengan mencari tetangga terdekat antara titik kunci x_i dengan semua $(n-1)$ titik kunci lain yang ditemukan pada citra uji. Tetangga terdekat yang dimaksud adalah titik kunci dengan jarak Euclidean terkecil. Misalkan D didefinisikan sebagai vektor kemiripan (*similarity vector*) dari *descriptor* suatu titik kunci yang berisi jarak-jarak Euclidean yang diurutkan dari terkecil hingga terbesar dengan *descriptor* dari titik kunci lain. Fungsi untuk menghitung jarak Euclidean dapat dilihat pada persamaan berikut [5]:

$$D = \{d_1, d_2, d_3, \dots, d_{n-1}\}$$

dengan

$$d = \sqrt{(f_a - f_b)^2} \quad (7)$$

Keterangan:

D = vektor kemiripan (*similarity vector*) dari suatu titik kunci

d = Jarak Euclidean *descriptor*

f_a = vektor *descriptor* dari titik kunci tersebut

f_b = vektor *descriptor* dari titik kunci lain

n = panjang vektor *descriptor* citra uji

Setelah jarak Euclidian didapat, langkah selanjutnya adalah menghitung operasi pembagian antara jarak tetangga terdekat pertama (*first closest-neighbor*) dengan jarak tetangga terdekat kedua (*second closest-neighbor*). Hasil dari pembagian tersebut dibandingkan dengan batas (*threshold*) T . Dengan cara ini, *keypoint* akan dikategorikan sebagai cocok jika memenuhi persyaratan [5]:

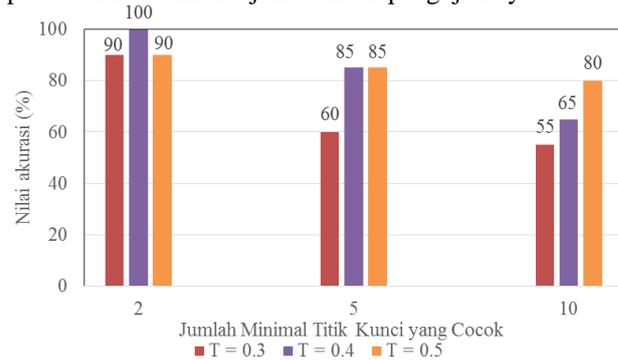
$$d_1/d_2 < T \text{ dengan } T \in (0,1). \quad (8)$$

Metode pencocokan 2NN kurang cocok untuk mendeteksi pemalsuan pada citra dengan banyak daerah yang digandakan karena metode ini hanya mengevaluasi dua tetangga terdekat pada tiap titik kunci. Untuk mengatasi persoalan ini, Amerini [5] mengusulkan metode baru yang merupakan pengembangan dari metode 2NN. Metode ini disebut *Generalized 2 Nearest-Neighbor* (G2NN). Pada metode ini dilakukan perulangan perhitungan d_1/d_2 sampai hasil yang didapatkan lebih besar dari batas (*threshold*) T . Dimisalkan k adalah suatu nilai dimana proses perulangan berhenti, maka setiap titik kunci dengan jarak $\{d_1, \dots, d_k\}$ dengan $1 \leq k < n$ dikelompokkan sebagai 'cocok'.

3. Hasil Pengujian dan Analisis

3.1. Pengujian tanpa Derau Gaussian

Pengujian tanpa derau Gaussian adalah pengujian citra dengan variasi penggunaan nilai ambang T dan jumlah titik kunci minimal pada pengujian citra uji tanpa derau Gaussian. Sebanyak 20 citra diujikan dengan variasi jumlah titik kunci minimal yang cocok yaitu 2, 5 dan 10 dan variasi nilai ambang T yaitu 0,3, 0,4, dan 0,5. Grafik pada Gambar 7 menunjukkan hasil pengujiannya.



Gambar 7. Grafik nilai persentase akurasi keberhasilan pendeteksian citra uji tanpa serangan derau Gaussian

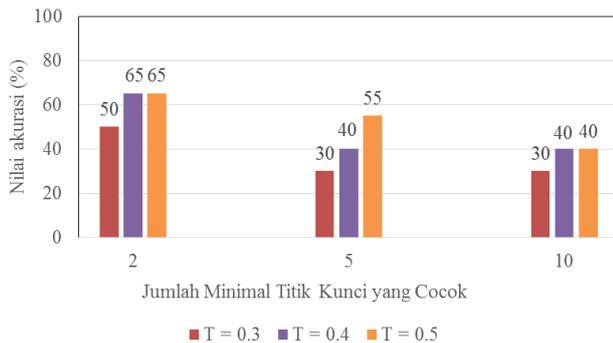
Dari grafik pada Gambar 7 dapat diketahui bahwa hasil pengujian citra dengan jumlah minimal ciri cocok sebanyak 2 ciri dengan variasi ambang $T=0,4$ memiliki nilai akurasi yang paling tinggi yaitu sebesar 100%. Variasi ambang $T=0,3$ dan $T=0,5$ memiliki nilai akurasi sama yaitu 90%. Hasil pengujian citra dengan jumlah minimal ciri cocok sebanyak 5 ciri dengan variasi ambang $T=0,4$ dan $T=0,5$ memiliki nilai akurasi 85%. Variasi ambang $T=0,3$ memiliki nilai akurasi terendah yaitu 60%. Hasil pengujian citra dengan jumlah minimal ciri cocok sebanyak 10 ciri dengan variasi ambang $T=0,5$ memiliki nilai akurasi yang paling tinggi yaitu sebesar 80%. Variasi ambang $T=0,3$ memiliki nilai akurasi 55% dan variasi ambang $T=0,4$ memiliki nilai akurasi 65%. Gambar 8 merupakan contoh hasil deteksi pemalsuan pada citra tanpa serangan derau Gaussian.



Gambar 8. Hasil deteksi pada citra uji tanpa serangan derau Gaussian

3.2. Pengujian dengan Derau Gaussian

Pengujian tanpa derau Gaussian adalah pengujian citra dengan variasi penggunaan nilai ambang T dan jumlah titik kunci minimal pada pengujian citra uji dengan derau Gaussian. Sebanyak 20 citra diujikan dengan variasi jumlah titik kunci minimal yang cocok yaitu 2, 5 dan 10 dan variasi nilai ambang T yaitu 0,3, 0,4, dan 0,5. Grafik pada Gambar 9 di bawah ini menunjukkan hasil pengujiannya.



Gambar 9. Grafik nilai persentase akurasi keberhasilan pendeteksian citra uji dengan serangan derau Gaussian

Dari grafik pada Gambar 9, dapat diketahui bahwa hasil pengujian citra dengan jumlah minimal ciri cocok sebanyak 2 ciri dengan variasi ambang $T=0,3$ memiliki nilai akurasi yang paling rendah yaitu sebesar 50%. Variasi ambang $T=0,4$ dan $T=0,5$ memiliki nilai akurasi 65%. Hasil pengujian citra dengan jumlah minimal ciri cocok sebanyak 5 ciri dengan variasi ambang $T=0,3$ memiliki nilai akurasi 30%. Variasi $T=0,4$ memiliki nilai akurasi 40% dan variasi ambang $T=0,5$ memiliki nilai akurasi terbesar yaitu 55%. Hasil pengujian citra dengan jumlah minimal ciri cocok sebanyak 10 ciri dengan variasi ambang $T=0,4$ dan $T=0,5$ memiliki nilai akurasi yang sama yaitu sebesar 40%. Variasi ambang $T=0,3$ memiliki nilai akurasi terkecil yaitu 30% .

Dari grafik di atas terlihat bahwa nilai akurasi pengujian dengan serangan derau Gaussian ini relatif lebih rendah dari pada nilai akurasi pengujian tanpa derau gaussian. Hal ini disebabkan karena serangan derau gaussian membuat algoritma SIFT yang merupakan algoritma berbasis titik kunci (*keypoint*) menjadi bekerja tidak optimal karena titik-titik kunci atau *keypoint* yang seharusnya dapat terdeteksi menjadi tidak terdeteksi sehingga mengakibatkan jumlah ciri yang didapatkan dari citra dengan serangan derau Gaussian lebih sedikit dibandingkan dengan citra uji yang tidak mendapatkan serangan derau Gaussian. Gambar 10 merupakan contoh citra yang mengalami serangan derau Gaussian dan Gambar 11 merupakan contoh hasil deteksinya.



Gambar 10. Citra yang mengalami serangan derau Gaussian



Gambar 11. Hasil deteksi citra yang mengalami serangan derau Gaussian

4. Kesimpulan

Dari percobaan dapat diketahui bahwa akurasi pendeteksian menurun setelah citra ditambahkan serangan derau Gaussian. Simulasi pendeteksian pemalsuan *copy-move* tanpa serangan derau Gaussian mencapai hasil akurasi tertinggi dengan nilai 100% pada nilai $T=0,4$ dan jumlah minimal ciri atau titik kunci bernilai 2. Simulasi pendeteksian pemalsuan *copy-move* dengan serangan derau Gaussian mencapai hasil akurasi tertinggi dengan nilai 65% pada nilai $T=0,4$ dan $T=0,5$ dengan jumlah minimal ciri atau titik kunci bernilai 2. Semakin tinggi jumlah titik kunci minimal membuat akurasi deteksi cenderung berkurang. Nilai ambang $T=0,3$ menghasilkan nilai akurasi yang relatif rendah dibanding nilai T lainnya di semua skenario pengujian. Untuk penelitian lebih lanjut, simulasi ini dapat ditambah dengan metode *clustering* seperti J-Linkage atau *agglomerative hierarchical clustering* setelah metode pencocokan ciri untuk menambah akurasi pendeteksian.

Referensi

- [1]. G. Palmer, "A Road Map for Digital Forensic Research," Laporan Teknis (DTR-T001-01) untuk *Digital Forensic Research Workshop*, New York, 2001.
- [2]. Malti Puri, Dr. Vinay Chopra, "A Survey: Copy-Move Forgery Detection Methods", *International Journal of Computer Systems (IJCS)*, vol. 3, no. 9, hal: 582-586, September 2016.
- [3]. J. Fridrich, D. Soukal, dan J. Lukas, "Detection of copy-move forgery in digital images," dalam Proc. *Digital Forensic Research Workshop*, Cleveland, OH, 2003.
- [4]. Popescu dan H. Farid, "Exposing digital forgeries by detecting duplicated image regions," Dartmouth College, Computer Science, Laporan Teknis (TR2004-515), 2004.
- [5]. W. Luo dan J. Huang, "Robust Detection of Region-Duplication Forgery in Digital Image," dipresentasikan di IEEE The 18th International Conference on Pattern Recognition (ICPR'06), 2006.
- [6]. D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. 'l Journal of Computer Vision*, vol. 60, no. 2, hal. 91-110, Januari 2004.
- [7]. I. Amerini, L. Ballan, R. Caldelli, A. Del Bimbo, dan G. Serra, "A SIFT-based forensic method for copy-move attack detection and transformation recovery," *IEEE Trans. on Information Forensics and Security*, vol. 6, no. 3, hal. 1099-1110, September 2011.