

DETEKSI PEMALSUAN CITRA *COPY MOVE* MENGUNAKAN *DYADIC WAVELET* DAN *SCALE INVARIANT FEATURE TRANSFORM*

Wahyu Restuti Tresnaningsih¹, Endina Putri Purwandari², Desi Andreswari³

^{1,2,3}Program Studi Teknik Informatika, Fakultas Teknik, Universitas Bengkulu.
Jl. WR. Supratman Kandang Limun Bengkulu 38371A INDONESIA
(telp: 0736-341022; fax: 0736-341022)

¹uyuy_hebat@yahoo.co.id

²endinaputrip@gmail.com

³dezieandrez@yahoo.co.id

Abstrak: Pada penelitian ini dibangun sebuah aplikasi yang bertujuan untuk mendeteksi pemalsuan *copy-move* pada citra digital. Pertama-tama, citra digital akan didekomposisi menggunakan metode *dyadic wavelet transform* (DyWT) dan diambil sub-citra LL, lalu mengekstraksi fitur lokal dengan metode *scale invariant feature transform* (SIFT). Implementasi kode aplikasi ini dilakukan menggunakan bahasa pemrograman MATLAB dan metode pengembangan *prototyping*. Analisis perancangan aplikasi ini dilakukan melalui pendekatan pemrograman terstruktur menggunakan *Data Flow Diagram* (DFD). Hasil pengujian menunjukkan metode DyWT dan SIFT mampu mendeteksi pemalsuan *copy-move* pada area citra berbeda yang telah mengalami beberapa perubahan pemrosesan citra, seperti rotasi dan skala (diperbesar atau diperkecil).

Kata kunci: pemalsuan citra, *copy-move*, DyWT, SIFT, MATLAB

Abstract: *In this research, an application is develop as a mean to detect copy-move forgery on digital image. First, digital images will be decomposed using dyadic wavelet transform (DyWT) and it is gotten subband LL from decomposed image, and then extracting the local features with scale invariant feature transform (SIFT). Implement-tation of the application codes is done using MATLAB programming language and prototyping development method. This application design analysis uses Data Flow Diagram (DFD) through structured programming approach. The test results showed DyWT and SIFT method capables to detect copy-move forgery in different area of the image that has undergone several changes in image processing, such as rotation and scale (enlarged or reduced).*

Keywords: *image forgery, copy-move, DyWT, SIFT, MATLAB.*

I. PENDAHULUAN

Teknologi pengolahan citra digital yang berkembang pesat semakin memudahkan pengguna dalam memanipulasi berbagai citra digital, mulai dari tingkat pemula hingga

kebutuhan profesional, dengan memanfaatkan beragam apli-kasi pengolah citra digital yang dapat diperoleh dengan mudah dan murah, bahkan gratis. Namun, penggunaan citra hasil manipulasi dengan tujuan yang tidak baik dapat berujung pada tindakan pemalsuan citra.

Pemalsuan citra digital saat ini semakin marak terjadi dalam publikasi. Contoh nyata yang terjadi di Indonesia adalah foto akte pernikahan Presiden Joko Widodo yang beredar saat kampanye pemilihan presiden tahun lalu dan foto yang diduga mantan ketua KPK Abraham Samad bersama Putri Indonesia 2013 atau beredarnya foto-foto yang melibatkan salah satu pejabat di Provinsi Bengkulu beberapa waktu yang lalu. Publi-kasi-publikasi yang melibatkan figur publik seperti yang telah disebutkan merupakan isu yang sensitif di masyarakat dan telah menimbulkan kehebohan. Masyarakat menjadi bingung dan

merasa perlu untuk mengetahui kebenaran dan keaslian dari foto-foto tersebut. Bagi masyarakat awam, untuk mengetahui apakah foto-foto itu asli atau palsu tanpa adanya bukti atau referensi foto asli adalah hal yang sulit, sehingga diperlukan suatu alat bantu untuk menguji keaslian foto tersebut, seperti aplikasi komputer.

Salah satu jenis pemalsuan citra yang paling populer dan sering dilakukan adalah pemalsuan *copy-move* karena tekniknya yang mudah dilakukan oleh banyak orang. Teknik pemalsuan ini didasarkan pada penyalinan suatu bagian dari sebuah citra, lalu bagian yang disalin ini, ditempelkan ke bagian lain dari citra tersebut. Pemalsuan *Copy-move* ini biasa digunakan sebagai studi awal dalam pembedahan citra yang dilakukan dalam bidang keilmuan pengolahan citra digital. Oleh karena itu, teknik pemalsuan *copy-move* dijadikan sebagai topik dalam penelitian ini.

Berbagai metode telah dikembangkan untuk mendeteksi pemalsuan citra *copy-move*. Purwandari [1] menggunakan metode DWT dan SVD terhadap citra untuk mengenali citra palsu dengan metode pemalsuan duplikasi, pencerminan, transpose, rotasi, skala dan *blurring*. Bhullar dan kawan-kawan [2] menampilkan pemalsuan citra digital dengan menerapkan metode DWT dan SIFT. Muhammad dan kawan-kawan [3] menggunakan metode *dyadic wavelet* dan DCT untuk mendeteksi daerah pemalsuan dengan metode pemalsuan rotasi dan kompresi JPEG. Amerini dan kawan-kawan [4] mengenalkan skema deteksi citra berbasis SIFT dengan mempertimbangkan transformasi geometris untuk mendeteksi pemalsuan *copy-move* terhadap citra.

Skema deteksi yang diajukan oleh Amerini dan kawan-kawan [4] direimplementasikan dalam penelitian ini bersama dengan metode *dyadic wavelet* yang diterapkan dalam [3] menjadi

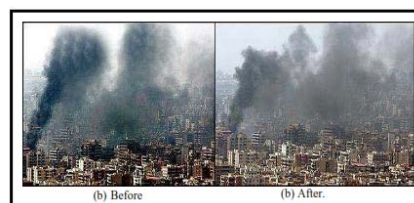
sebuah aplikasi deteksi pemalsuan citra *copy-move* menggunakan *dyadic wavelet* dan *scale invariant feature transforms*. Pengujian dilakukan terhadap beberapa kondisi pemalsuan untuk melihat kinerja aplikasi.

II. LANDASAN TEORI

A. Pemalsuan Citra Copy-Move

Ada beberapa tipe pemalsuan digital. Setiap tipe pemalsuan merupakan bagian dari tiga kategori utama, tergantung pada proses yang digunakan dalam pengkreasian citra. Kategori pemalsuan terdiri dari *Image Retouching*, *Image Splicing* dan *Copy-Move* [5].

Copy move hampir sama dengan kategori *image splicing*. Namun berbeda dengan *image splicing*, daerah yang disalin dari sebuah citra ditempelkan ke citra yang sama. Perbedaan dasarnya terletak pada penggunaan citra itu sendiri sebagai sumber sekaligus penerima dari bagian yang disalin [5].



Gambar 1. Contoh pemalsuan *copy-move* [5]

B. Dyadic Wavelet Transform (DyWT)

Untuk mengatasi kekurangan DWT, Mallat and Zhong [6] memperkenalkan DyWT, yang bersifat *shift-invariant*. Dalam hal ini, transformasi wavelet tidak melibatkan *downsampling* dan ukuran koefisien-koefisien waveletnya tidak disusutkan diantara skala-skalanya seperti DWT. Karena karakteristiknya ini, DyWT disebut *undecimated*. Stack dan kawan-kawan [7] membuktikan bahwa DyWT memiliki analisis tekstur dan kinerja deteksi yang lebih baik daripada DWT. Pergeseran kecil dalam citra masukan dapat menyebabkan

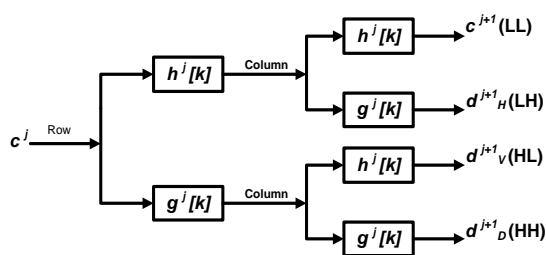
perbedaan besar pada koefisien-koefisien DWT di skala yang berbeda, yang dapat menghasilkan vektor fitur yang berbeda untuk objek yang disalin dan disisipkan dengan pergeseran spasial yang kecil [8].

Misalkan \mathbf{M} citra yang didekomposisi, dan $h[k]$ dan $g[k]$ adalah skala (*low pass*) dan filter wavelet (*high pass*). DyWT suatu citra dapat dihitung dengan menggunakan algoritma *atrous*. Algoritma ini dimulai dengan skala $j = 0$, dan gunakan $\mathbf{M}^0 = \mathbf{M}$. Lalu hitung skala dan koefisien wavelet dengan skala $j = 1, 2, \dots, J$ menggunakan persamaan (1) dan (2) :

$$c^{j+1}[n] = \sum_k h[k]c^j[n + 2^j k] \quad (1)$$

$$d^{j+1}[n] = \sum_k g[k]c^j[n + 2^j k] \quad (2)$$

Sampai tahap ini, serupa dengan DWT. Selanjutnya $h^j[k]$ dan $g^j[k]$ menjadi *filter* yang dihasilkan dari penyisipan nol pada $2^j - 1$ di $h[k]$ dan $g[k]$. Kemudian kita jalankan DyWT menggunakan *filtering* dimulai dengan Citra \mathbf{M} . Untuk menghasilkan koefisien-koefisien penskalaan dan wavelet \mathbf{M}^j dan \mathbf{D}^j pada skala $j = 1, 2, \dots, J$ melalui filter \mathbf{M}^{j-1} dengan $h^{j-1}[k]$ dan filter \mathbf{M}^{j-1} dengan $g^{j-1}[k]$.



Gambar 2. Dekomposisi satu level DyWT pada citra 2D [3]

Gambar 2 menunjukkan ilustrasi algoritma dekomposisi satu level dari suatu citra dua dimensi pada baris dan kolom. LL adalah komponen dengan frekuensi rendah dan disebut perkiraan, sedangkan HH adalah komponen dengan frekuensi tinggi [3].

C. Scale Invariant Feature Transform (SIFT)

Scale Invariant Feature Transform (SIFT) adalah sebuah algoritma dalam *computer vision* untuk mendeteksi dan mendeskripsikan fitur lokal dalam gambar. Algoritma ini dipublikasikan oleh David Lowe pada tahun 1999. Dengan menggunakan SIFT, suatu citra akan diubah menjadi vektor fitur lokal yang kemudian digunakan sebagai pendekatan dalam mendeteksi maupun mengenali objek yang dimaksud melalui titik-titik *point* atau *keypoint* [9].

Algoritma yang digunakan pada metode SIFT terdiri dari empat langkah utama [2] sebagai berikut:

a. Deteksi Ruang Skala Ekstrema

Pada algoritma SIFT digunakan fungsi *Gaussian blur* untuk membangun ruang skala sehingga disebut ruang skala *Gaussian*.

$$L(x,y,\sigma) = G(x,y,\sigma) * I(x,y) \quad (3)$$

Keterangan:

- $L(x,y,\sigma)$ adalah ruang skala hasil konvolusi dari citra asli $I(x,y)$ dengan *Gaussian filter* $G(x,y,\sigma)$ pada skala σ
- $G(x,y,\sigma)$ adalah *Gaussian filter* diperoleh dari $\frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$
- $I(x,y)$ adalah citra masukan dengan koordinat piksel
- $*$ adalah operasi konvolusi
- σ adalah faktor ruang skala atau perbedaan distribusi normal Gaussian

Setelah mendapatkan ruang skala *Gaussian* langkah selanjutnya adalah membentuk *Difference of Gaussian* (DOG) yang dihitung dengan cara:

$$D(x,y,\sigma) = L(x,y,k\sigma) - L(x,y,\sigma) \quad (4)$$

Keterangan:

- $L(x,y,k\sigma)$ adalah konvolusi dari citra asli $I(x,y)$ dengan *Gaussian filter* $G(x,y,k\sigma)$ pada skala σ dengan nilai $k = \sqrt{2}$

- $L(x,y,\sigma)$ adalah konvolusi dari citra asli $I(x,y)$ dengan *Gaussian filter* $G(x,y,\sigma)$ pada skala σ

b. Lokalisasi *Keypoint*

Pemilihan *keypoint* dari ekstrema dilakukan dengan menghapus titik-titik disepanjang tepi citra atau titik-titik berkontras rendah yang tidak stabil terhadap variasi citra. Dengan menggunakan ekspansi Taylor fungsi ruang skala, $D(x,y,\sigma)$, yang digeser sehingga titik sampel berasal dari rumus berikut:

$$D(x) = D + \frac{\partial D^T}{\partial x} x + \frac{1}{2} x^T \frac{\partial^2 D}{\partial x^2} x \quad (5)$$

Keterangan:

- $x = (x,y,\sigma)^T$ adalah titik tengah dari titik sampel

Nilai fungsi pada ekstrema $D(\hat{x})$ sangat baik untuk menghapus nilai ekstrema yang tidak stabil dan berkontras rendah, yang dirumuskan berikut:

$$D(\hat{x}) = D + \frac{1}{2} \frac{\partial D^T}{\partial x} \hat{x} \quad (6)$$

Keterangan:

- \hat{x} adalah turunan yang diambil dari fungsi 2.5 terhadap x , dimana x adalah nol, $\hat{x} = -\frac{\partial^2 D^{-1}}{\partial x^2} \frac{\partial D}{\partial x}$

Selanjutnya untuk menghapus respon tepi dari operator DOG digunakan matriks *Hessian*, \mathbf{H} , orde dua.

$$\mathbf{H} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix} \quad (7)$$

Dimana D diperkirakan dengan mengambil perbedaan titik sampel ketetanggaan.

Kita juga dapat menghitung jumlah *trace* dan *determinant* dari matriks *Hessian* dengan cara:

$$\begin{aligned} Tr(\mathbf{H}) &= D_{xx} + D_{yy} = \alpha + \beta \\ Det(\mathbf{H}) &= D_{xx}D_{yy} - (D_{xy})^2 = \alpha\beta \\ \frac{Tr(\mathbf{H})^2}{Det(\mathbf{H})} &= \frac{(\alpha+\beta)^2}{\alpha\beta} = \frac{(r\beta+r\beta)^2}{r\beta^2} = \frac{(r+1)^2}{r} \end{aligned} \quad (8)$$

Keterangan:

- $\alpha = small\ eigenvalue$
- $\beta = large\ eigenvalue$
- $r = rasio\ antara\ \beta\ dan\ \alpha\ (r = \alpha/\beta)$

Titik fitur yang dihasilkan harus memenuhi persamaan di 9, jika tidak maka akan dieliminasi.

$$\frac{Tr(\mathbf{H})^2}{Det(\mathbf{H})} < \frac{(r+1)^2}{r} \quad (9)$$

c. Penempatan Orientasi

Untuk mendapatkan rotasi yang tidak berubah, setiap *keypoint* ditempatkan dalam suatu orientasi. Untuk setiap *Gaussian* sampel citra yang dihaluskan, $L(x,y)$, besarnya gradient, $m(x,y)$, dan orientasi, $\theta(x,y)$, dihitung menggunakan perbedaan piksel :

$$\begin{aligned} m(x,y) &= \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2} \\ \theta(x,y) &= \tan^{-1} \frac{L(x,y+1) - L(x,y-1)}{L(x+1,y) - L(x-1,y)} \end{aligned} \quad (10)$$

Arah Gradien dari titik-titik fitur dihitung menggunakan histogram gradien berorientasi. Puncak histogram orientasi mewakili arah dominan dari gradien lokal.

d. Pembangkitan Deskriptor *Keypoint*

Nilai histogram orientasi pada bidang citra dan ruang skala membentuk des-kriptor. Dengan histogram larik 4x4 dan masing-masing 8 tempat orientasi, hasil 4x4x8 = 128 elemen vektor fitur.

D. Clustering

Pengelompokkan menggunakan *Hierarchical Agglomerative Clustering* diselesaikan dengan menggunakan salah satu dari banyak metode *linkage*, seperti *median*, *centroid* atau *ward*. Penelitian ini menggunakan metode *linkage ward*. Untuk setiap kluster C berasal dari kluster A dan B, yang memiliki sejumlah titik n_C . Jika x_{Ci} mewakili titik ke- i dalam kelompok, selanjutnya metode *linkage ward* bekerja dengan penambahan jumlah kuadrat jarak diantara seluruh titik dalam

kelompok dan *centroid* kelompok menggunakan [2]

$$dist(C, D) = \sqrt{\frac{2n_C n_D}{n_C + n_D} \|\bar{x}_C - \bar{x}_D\|_2} \quad (11)$$

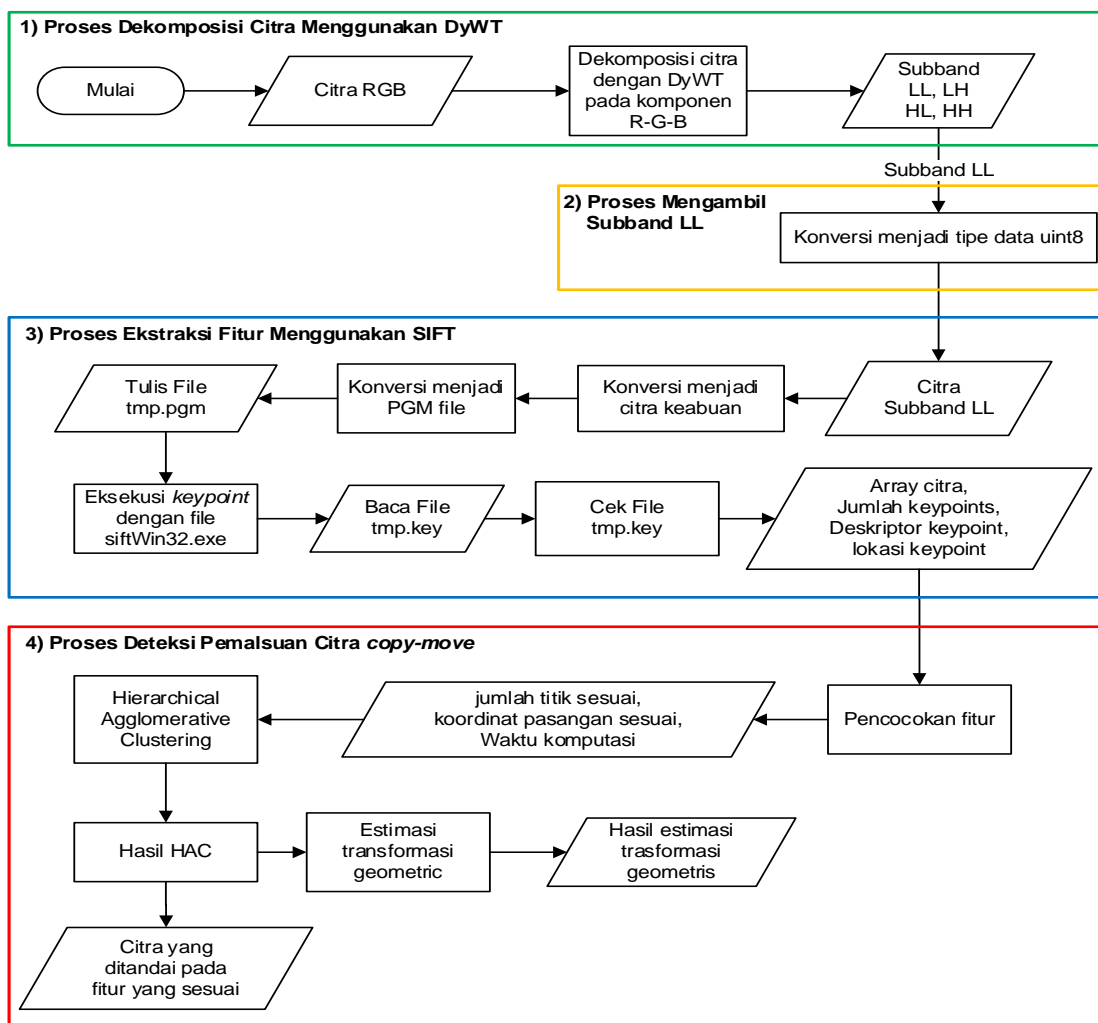
E. RANdom Sample Consensuc (RANSAC)

RANSAC (*Random Sample Consensus*) merupakan metode yang digunakan untuk mencari *Homography Matrix* dan sekaligus berfungsi untuk menghilangkan *outliers* dari *feature-feature* yang telah ditemukan. *Outliers* sendiri merupakan *feature* yang memiliki nilai menyimpang dari kebanyakan *feature* lain. *Homography Matrix* merupakan *matrix* transformasi yang berguna untuk memproyeksikan gambar satu pada gambar lain sesuai dengan *feature match* yang ditemukan.

Dengan mengalikan gambar dengan *Homography Matrix*, gambar akan mengalami transformasi geometri seperti translasi, rotasi, *scaling*, *skew*, *shear*, dan lain – lain.

$$F. \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = H \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

Dimana merupakan $(u \ v \ 1)^T$ merupakan x' , $(u \ v \ 1)^T$ merupakan x dan H merupakan *Homography Matrix* yang berupa matriks 3 x 3. *Homography Matrix* inilah yang digunakan untuk menyatukan kedua gambar yang saling berhubungan. *Output* dari metode ini adalah sebuah *Homography Matrix*. Untuk menyatukan kedua gambar, gambar pertama hanya perlu dikalikan oleh *matrix* tersebut [10].



Gambar 3. Alur Kerja Aplikasi

III. ANALISIS DAN PERANCANGAN

A. Alur Kerja Aplikasi

Analisis alur kerja aplikasi merupakan analisis yang ditujukan untuk menguraikan cara kerja dari suatu aplikasi yang dibangun secara runtun dimulai dari aktivitas masukan hingga menghasilkan suatu keluaran melalui proses-proses yang dijalankan aplikasi Secara garis besar, alur kerja aplikasi yang dibangun ditampilkan pada gambar 3. Pengguna akan diminta untuk memasukkan citra yang ingin dideteksi ke dalam aplikasi, lalu citra akan diproses hingga menghasilkan keluaran hasil deteksi. Untuk mempermudah pembuatan aplikasi, alur kerja dibagi kedalam 4 proses umum yaitu proses dekomposisi citra, mengambil *subband* LL, ekstraksi fitur dan deteksi.

IV. PEMBAHASAN

Aplikasi deteksi ini telah diimplementasikan menggunakan *software* MATLAB R2013b versi 8.2.0.701 dengan *notebook* berspesifikasi RAM 3 GB dan *processor* AMD A6-3420 APU. Pengujian dilakukan pada beragam citra dengan ukuran dan dimensi citra berbeda. Pada citra uji diperlakukan kondisi atau serangan dengan ukuran area bagian pemalsuan berbeda dan manipulasi geometris pemrosesan citra berbeda. Dalam pengujian ini, citra yang digunakan adalah citra warna dengan format *.JPEG.

A. Tampilan Antarmuka untuk deteksi citra

Saat pertama kali halaman menu *Detection* muncul hanya tombol *Open Image* yang aktif pada *Button Group*. Tampilan halaman deteksi dapat dilihat pada gambar 10.

Untuk mendeteksi citra digital, pengguna terlebih dahulu harus memilih citra masukan dari komputer. Untuk memilih citra masukan maka

pengguna memilih tombol *Open Image* pada halaman menu *Detection*.



Gambar 10. Halaman menu *Detection*

Setelah pengguna berhasil memilih citra masukan yang ingin diuji, maka aplikasi akan menampilkannya di kotak *input* dan menampilkan informasi citra, serta mengaktifkan tombol *Apply DyWT* seperti pada gambar 11 berikut.



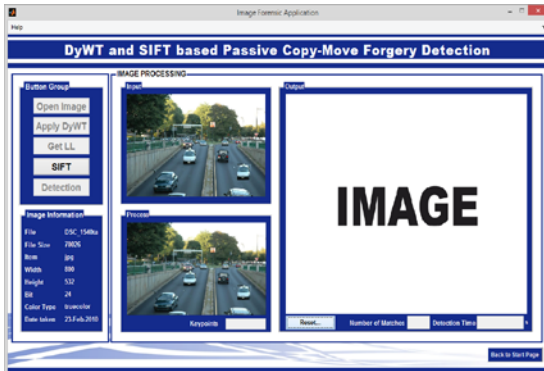
Gambar 11. Aplikasi menampilkan citra masukan

Proses selanjutnya yang dilakukan pengguna adalah memilih tombol *Apply DyWT*. Ketika pengguna telah memilih tombol ini, citra masukan akan didekomposisi dengan metode DyWT dan hasilnya akan ditampilkan pada kotak *process*, tombol *Get LL* juga diaktifkan seperti pada gambar 12.



Gambar 12. Aplikasi menampilkan hasil dekomposisi citra

Setelah tombol *Get LL* aktif, pengguna dapat mengam-bil *subband LL* dari hasil dekomposisi citra masukan. Ketika pengguna telah memilih tombol ini, maka aplikasi akan menampilkan citra *subband LL* pada kotak *process* dan meng-aktifkan tombol *SIFT* seperti pada gambar 13.



Gambar 13. Aplikasi menampilkan citra subband LL

Ketika pengguna memilih tombol *SIFT* tidak hanya hasil ekstraksi fitur yang muncul, namun juga mengaktifkan tombol *Detection*. Aplikasi juga akan menampilkan jumlah fitur yang berhasil diekstraksi dalam bentuk *keypoint*, seperti pada gambar 14.



Gambar 14. Aplikasi menampilkan hasil ekstraksi fitur

Proses selanjutnya adalah memilih tombol *Detection* yang telah aktif untuk melakukan deteksi dari hasil ekstraksi fitur. Setelah memilih tombol ini, maka aplikasi akan menampilkan hasil deteksi pada kotak *output*. Aplikasi juga akan menampilkan jumlah titik-titik yang bersesuaian dan lamanya waktu deteksi yang dilakukan, seperti pada gambar 15.



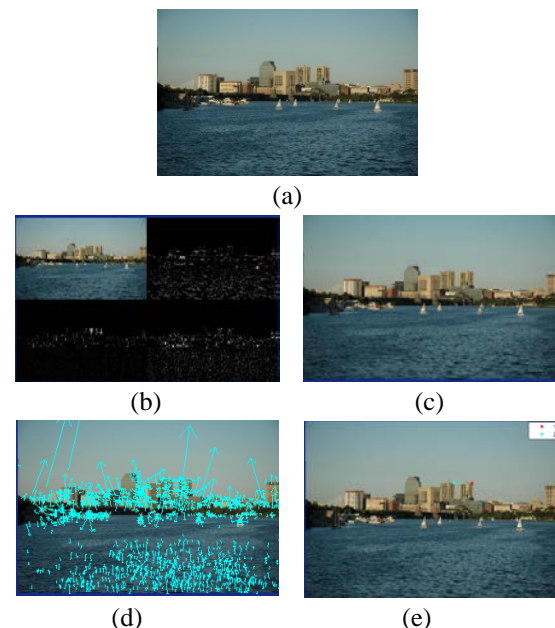
Gambar 15. Aplikasi menampilkan hasil deteksi citra

B. Pengujian Aplikasi

Pengujian aplikasi dilakukan terhadap citra asli dan citra dengan pemalsuan *copy-move*. Berikut pengujian yang dilakukan dalam penelitian ini.

1. Hasil Pengujian Terhadap Citra Masukan Tanpa Pemalsuan *Copy-Move* dari dataset MICC-F220

Pada pengujian terhadap citra asli dilakukan dengan mem-berikan masukan berupa citra original. Citra uji yang diguna-kan sebanyak 10 citra original diambil dari dataset MICC-F220 [4].



Gambar 16. (a) citra masukan. (b) citra hasil DyWT. (c) citra subband LL (d) hasil ekstraksi fitur SIFT. (e) hasil deteksi citra

Gambar 16 menunjukkan berbagai proses yang terjadi pada pengujian aplikasi terhadap citra tanpa pemalsuan *copy-move*. Gambar 16(a) adalah gambar citra masukan yang diproses oleh aplikasi. Citra masukan didekomposisi menggunakan

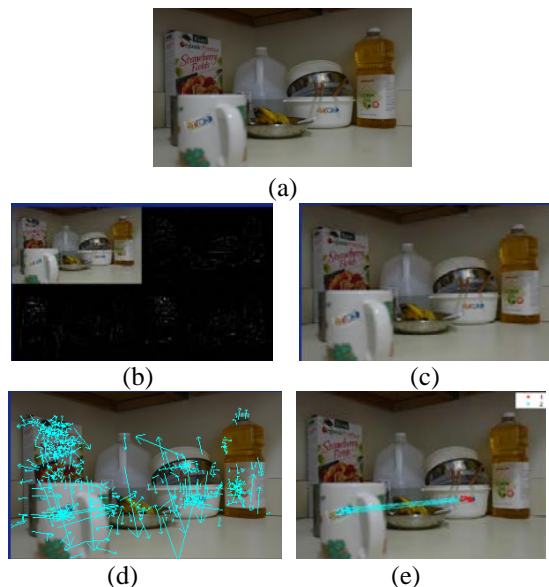
metode DyWT, sehingga menghasilkan gambar 16(b). Dari hasil dekomposisi, lalu diambil *subband* LL sebagai citra yang akan diproses seperti pada gambar 16(c). Citra *subband* LL kemudian diekstrak fitur nya dengan metode SIFT sehingga menghasilkan tampilan seperti pada gambar 16(d). Setelah diperoleh fitur-fitur dari *subband* LL, dilakukan proses deteksi hingga memperoleh hasil deteksi seperti gambar 16(e).

Tabel 1. Hasil pengujian terhadap citra original

No	Data	Keypoint	Waktu Deteksi (detik)	Hasil Deteksi	Ground Truth
1	Citra 1	7824	67.827	Original	Original
2	Citra 2	617	0.977748	Original	Original
3	Citra 3	1731	3.31083	Original	Original
4	Citra 4	2754	8.27356	Original	Original
5	Citra 5	565	0.318681	Original	Original
6	Citra 6	6661	46.686	Original	Original
7	Citra 7	2996	9.19268	Original	Original
8	Citra 8	1090	1.74726	Original	Original
9	Citra 9	2620	6.89405	Original	Original
10	Citra 10	307	0.119891	Original	Original

2. Hasil Pengujian Terhadap Citra Masukan Dengan Pemalsuan *Copy-Move* dari Dataset MICC-F220

Pada pengujian terhadap citra palsu dilakukan dengan memberikan masukan berupa citra palsu *copy-move*. Citra uji yang digunakan sebanyak 22 citra palsu diambil dari dataset MICC-F220 [4].



Gambar 17. (a) citra palsu. (b) citra hasil DyWT. (c) citra *subband* LL (d) hasil ekstraksi fitur SIFT. (e) hasil deteksi citra

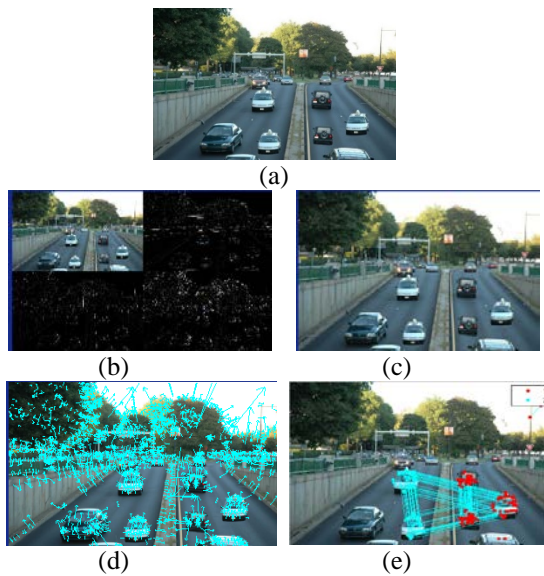
Gambar 17 menunjukkan proses-proses yang terjadi dalam mendeteksi pemalsuan citra pada aplikasi. Citra masukan yang diuji berupa citra palsu ditunjukkan pada gambar 17(a). Aplikasi berhasil mendeteksi adanya pemalsuan *copy-move* yang ditunjukkan pada gambar 17(e).

Tabel 2. Hasil pengujian terhadap citra palsu

No	Data	Serangan	Keypoint	Waktu Deteksi (detik)	Hasil Deteksi
1	Citra 1	Tanpa Serangan	689	0.725915	Palsu
2	Citra 2	Skala $x = 1.2$ $y = 1.2$	712	0.808356	Palsu
3	Citra 3	Skala $x = 1.3$ $y = 1.3$	716	0.739492	Palsu
4	Citra 4	Skala $x = 1.4$ $y = 1.2$	712	0.726412	Palsu
5	Citra 5	Rotasi 10°	690	0.734432	Palsu
6	Citra 6	Rotasi 20°	684	0.798321	Palsu
7	Citra 7	Rotasi 30°	679	0.712513	Palsu
8	Citra 8	Rotasi 40°	693	0.726581	Palsu
9	Citra 9	Rotasi 10° $x = 1.2$ $y = 1.2$	720	0.744736	Palsu
10	Citra 10	Rotasi 20° Skala $x = 1.4$ $y = 1.2$	718	0.734582	Palsu
11	Citra 11	Tanpa Serangan	2824	8.84357	Palsu
12	Citra 12	Skala $x = 1.2$ $y = 1.2$	2861	8.69566	Palsu
13	Citra 13	Skala $x = 1.3$ $y = 1.3$	2854	8.92094	Palsu
14	Citra 14	Skala $x = 1.4$ $y = 1.2$	2867	8.76033	Palsu
15	Citra 15	Rotasi 10°	2835	8.79285	Palsu
16	Citra 16	Rotasi 20°	2836	8.57539	Palsu
17	Citra 17	Rotasi 30°	2836	8.72604	Palsu
18	Citra 18	Rotasi 40°	2844	8.4786	Palsu
19	Citra 19	Rotasi 10° Skala $x = 1.2$ $y = 1.2$	2882	8.91516	Palsu
20	Citra 20	Rotasi 20° Skala $x = 1.4$ $y = 1.2$	2878	8.91288	Palsu
21	Citra 21	Rotasi 40°	586	0.606765	Palsu
22	Citra 22	Rotasi 20° Skala $x = 1.4$ $y = 1.2$	600	0.614721	Palsu

3. Hasil Pengujian terhadap Citra Masukan dengan Pemalsuan *Multiple Copy-Move* dari dataset MICC-F8multi

Pada pengujian terhadap citra *multiple copy-move* dilaku-kan dengan menggunakan citra uji sebanyak 8 citra palsu yang diambil dari dataset MICC-F8multi [4].



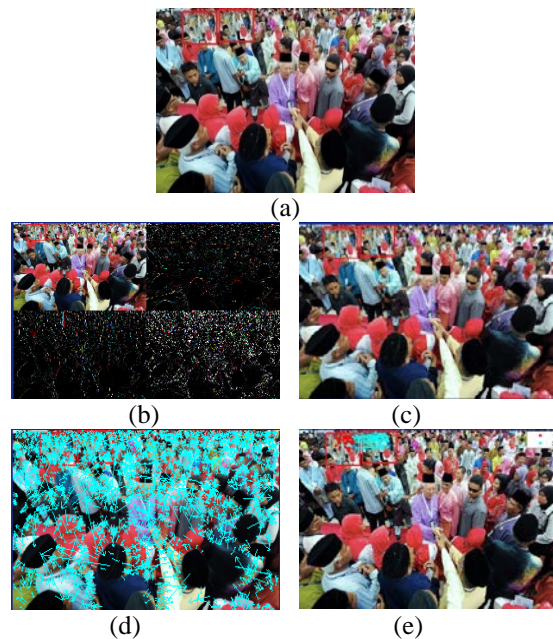
Gambar 18. (a) citra palsu. (b) citra hasil DyWT. (c) citra subband LL (d) hasil ekstraksi fitur SIFT. (e) hasil deteksi citra

Tabel 3 Hasil pengujian terhadap citra *multiple copy-move*

No	Data	Ukuran Citra	Serangan	Keypoint	Waktu Deteksi (detik)	Hasil Deteksi
1	Citra 1	800x532 (114 kb)	Skala diperkecil $x = 0.9$ $y = 0.9$	2978	26.265	Palsu
2	Citra 2	800x532 (56.4kb)	Skala diperbesar $x = 1.4$ $y = 1.2$	893	4.09742	Palsu
3	Citra 3	2048x1536 (581 kb)	Tanpa Serangan	2411	15.0641	Palsu
4	Citra 4	2048x1536 (960 kb)	Skala diperbesar $x = 2.0$ $y = 1.6$	28929	701.848	Palsu
5	Citra 5	2048x1536 (820 kb)	Tanpa Serangan	10812	144.161	Palsu
6	Citra 6	2048x1536 (913 kb)	Skala diperkecil $x = 1.2$ $y = 1.2$	20808	352.481	Palsu
7	Citra 7	800x532 (202 kb)	Rotasi 10° Skala diperbesar $x = 1.2$ $y = 1.5$	4028	33.6468	Palsu
8	Citra 8	947 x 683 (378 kb)	Skala diperkecil $x = 1.2$ $y = 1.2$	4819	48.8736	Palsu

4. Hasil Pengujian terhadap Citra Masukan dengan Pemalsuan *Copy-Move* dari Dokumentasi pribadi dan internet

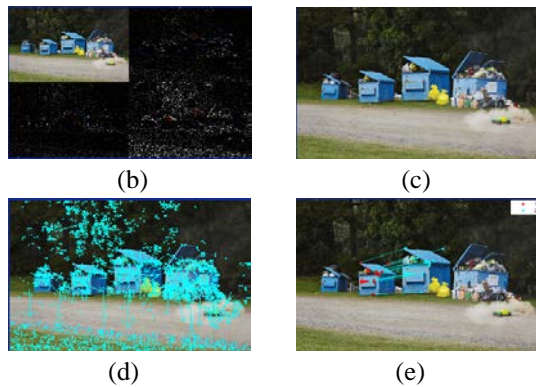
Selain pengujian menggunakan dataset, pengujian juga dilakukan dengan citra uji palsu sebanyak 10 citra berasal dari internet dan dokumentasi pribadi.



Gambar 19. (a) citra palsu dari internet. (b) citra hasil DyWT. (c) citra subband LL (d) hasil ekstraksi fitur SIFT. (e) hasil deteksi citra

Gambar 19 adalah pengujian yang dilakukan terhadap citra dengan pemalsuan *copy-move* yang diperoleh dari internet. Gambar 19(a) menunjukkan citra uji pada aplikasi. Citra lalu didekomposisi menggunakan metode DyWT pada gambar 19(b).Kemudian diambil *subband* LL yang ditunjukkan gambar 19(c). Ekstraksi fitur lalu diterapkan pada *subband* LL sehingga menghasilkan gambar 19(d). Hasil deteksi pemalsuan ditunjukkan pada gambar 19(e).





Gambar 20. (a) citra palsu dari dokumentasi. (b) citra hasil DyWT. (c) citra subband LL (d) hasil ekstraksi fitur SIFT. (e) hasil deteksi citra

Pengujian pada gambar 20 dilakukan menggunakan citra yang berasal dari dokumentasi. Citra ini kemudian diolah menjadi citra palsu dengan *software* Adobe Photoshop 7.0.

Tabel 4 Hasil pengujian terhadap citra palsu dokumentasi

No	Data	Ukuran Citra	Serangan	Key-point	Waktu Deteksi (detik)	Hasil Deteksi
1	Citra 1	1000x 641 (241 kb)	Tanpa Serangan	663	2.04872	Palsu
2	Citra 2	638x 379 (166 kb)	Tanpa Serangan	3079	20.8412	Palsu
3	Citra 3	1280x 960 (534 kb)	Tanpa Serangan	14390	626.051	Palsu
4	Citra 4	1280x 960 (526 kb)	Skala x = 1.3 y = 1.4	13925	376.093	Palsu
5	Citra 5	1280x 960 (535 kb)	Rotasi 45°	14444	769.406	Palsu
6	Citra 6	1280x 960 (528 kb)	Rotasi 45° Skala x = 1.3 y = 1.4	13987	621.485	Palsu
7	Citra 7	699x 365 (90.9 kb)	Tanpa Serangan	2562	10.5462	Palsu
8	Citra 8	900x 600 (193 kb)	Skala x = 1.3 y = 1.4	3013	14.2172	Palsu
9	Citra 9	640x 360 (117 kb)	Tanpa Serangan	2314	14.984	Palsu
10	Citra 10	699x 2365 (171 kb)	Skala x = 0.8 y = 0.8	1376	3.71993	Palsu

V. KESIMPULAN

1. Penelitian ini telah menghasilkan aplikasi yang mampu mendeteksi adanya pemalsuan *copy move* pada citra digital dengan menggunakan

metode *Dyadic Wavelet* dan *Scale Invariant Feature Transform*.

2. Metode DyWT dan SIFT yang diimplementasikan pada aplikasi ini menghasilkan tingkat akurasi sebagai berikut:

a. Pengujian dengan menggunakan 10 citra masukan tanpa pemalsuan *copy-move* berhasil mendeteksi dengan benar citra masukan sebagai citra asli.

b. Pengujian dengan menggunakan 22 citra pemalsuan *copy-move* dengan serangan rotasi sebesar 10°, 20°, 30° dan 40° derajat, serta serangan skala dengan faktor skala horizontal sebesar 1.2, 1.3, dan 1.4, dan faktor skala vertikal sebesar 1.2 dan 1.3, berhasil mendeteksi dengan benar citra masukan sebagai citra palsu.

c. Pengujian dengan menggunakan 8 citra pemalsuan *multiple copy-move* dengan serangan rotasi sebesar 10° dan serangan skala dengan faktor skala horizontal sebesar 0.9, 1.2, 1.4 dan 2, dan faktor skala vertikal sebesar 0.9, 1.2, 1.5 dan 1.6, berhasil mendeteksi dengan benar citra masukan sebagai citra palsu.

d. Pengujian dengan menggunakan 10 citra pemalsuan *copy-move* yang berasal dari internet dan dokumentasi pribadi dengan serangan rotasi sebesar 45° dan serangan skala dengan faktor skala horizontal dan vertical sebesar 0.8 dan 1.3, berhasil mendeteksi dengan benar citra masukan sebagai citra palsu.

3. Berdasarkan hasil pengujian yang telah dilakukan diketahui bahwa penerapan metode transformasi *dyadic wavelet* dan transformasi *scale invariant feature* mampu mendeteksi serangan *post-processing* dengan metode

pemalsuan rotasi dan skala (diperbesar dan diperkecil) yang diterapkan pada citra uji.

VI. SARAN

1. Kekurangan hasil deteksi dari penggunaan metode DyWT dan SIFT yang telah diimplementasikan dalam aplikasi ini adalah adanya *false matches* yaitu titik-titik yang terdeteksi sebagai titik yang sesuai, namun tidak tepat dengan titik-titik *copy-move* yang sebenarnya, sehingga kedepannya pengembangan aplikasi dapat difokuskan untuk meminimalisir hasil deteksi tersebut, misalnya dengan pengujian *threshold*.
2. Aplikasi ini dapat terus dikembangkan lebih lanjut dalam hal metode yang digunakan, misalnya dengan menggunakan dua *subband* hasil citra dekomposisi untuk diproses lebih lanjut. Dalam hal ini, penggunaan *subband* LL yang menyimpan banyak nilai-nilai kesamaan dapat dibandingkan dengan *subband* HH yang menyimpan nilai-nilai berbeda, sehingga titik-titik yang tidak sesuai dapat diminimalisir. Pemilihan nilai ambang juga perlu dilakukan pengujian lebih lanjut agar diperoleh nilai ambang yang lebih akurat dalam menemukan titik-titik yang sesuai. Pengujian metode *clustering* juga dapat dikembangkan untuk aplikasi ini agar citra hasil deteksi menjadi lebih baik.

Image Forgery Detection," *IEEE Journals*, 2012.

- [[4] I. Amerini, L. Ballan, R. Caldelli, A. D. Bimbo and G. Serra, "A Sift-based forensic method for copy-move attack detection and transformation recovery," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 3, pp. 1099-1110, 2011.
- [[5] A. A. A. Ahmadi, "IMAGE FORGERY DETECTION EXPLOITING IMAGE RANDOMNESS," King Saud University, Arab Saudi, 2013-2014.
- [[6] S. Mallat and S. Zhong, "Characterization of Signal from Multiscale Edges," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 6, pp. 710-732, July 1992.
- [[7] J.-L. Starck, J. Fadili and F. Murtagh, "The undecimated wavelet decomposition and its reconstruction," *IEEE Transactions on Image Processing*, vol. 16, no. 2, pp. 297-309, 2007.
- [[8] G. Muhammad, M. Hussain and G. Bebis, "Passive Copy Move Image Forgery Detection Using Undecimated Dyadic Wavelet Transform," *Digital Investigations*, vol. 9, no. 1, pp. 49-57, 2012.
- [[9] A. Setiawan and R. S. Basuki, "Pencocokan Citra Berbasis Scale Invariant Feature Transform (SIFT) menggunakan Arc Cosinus," *Jurnal Teknik Informatika-S1 Udinus*, 2013.
- [[10] E. R. F. A. Putra, Liliana and K. Gunadi, "Aplikasi Automatic Image Stitching pada Kumpulan Gambar dalam Satu Scene," *Student Journals Universitas Kristen Petra*, 2014.

REFERENSI

- [[1] Purwandari, Endina Putri., "Deteksi Pemalsuan Copy-Move Duplicated Region Pada Citra Digital dengan Komputasi Numerik," *Pseudocode*, vol. 1, no. 1, pp. 22-29, 2014.
- [[2] L. K. Bhullar, S. Bhudiraja and A. Dhindsa, "DWT and SIFT based Passive Copy-Move Forgery Detection," *International Journal of Computer Applications*, vol. 95, June 2012.
- [[3] G. Muhammad, M. Hussain, A. M. Mirza and G. Bebis, "Dyadic Wavelets and DCT Based Blind Copy-Move