

Sistem Pentautan Citra Udara Menggunakan Algoritme SURF dan Metode Reduksi Data

Zaki Hamizan*¹, Raden Sumiharto²

¹Program Studi Elektronika dan Instrumentasi, FMIPA UGM, Yogyakarta, Indonesia

²Departemen Ilmu Komputer dan Elektronika, FMIPA UGM, Yogyakarta, Indonesia

e-mail: *¹zaki_hamizan@yahoo.co.id, ²r_sumiharto@ugm.ac.id

Abstrak

Salah satu cara pentautan citra udara dapat menggunakan algoritme SURF (*Speeded Up Robust Features*). Algoritme ini memiliki ketahanan terhadap perubahan skala, rotasi, blurring, pencahayaan, dan perubahan bentuk. Namun sebagian hasil dari deteksi keypoint SURF tidak selalu dianggap sebagai keypoint yang menarik sehingga perlu dieliminasi.

Sistem yang dibuat menggunakan detektor SURF pada proses pendeteksian. Metode reduksi data mengeliminasi keypoint yang dianggap saling berdekatan. Kemudian dilanjutkan dengan proses deskripsi keypoint dengan deskriptor SURF. Hasil deskripsi selanjutnya dicocokkan menggunakan FLANN. Tahap selanjutnya adalah pencarian pola matriks homograf dengan RANSAC dan pentautan citra dengan menumpuk keypoint menggunakan *warpPerspective*.

Pengujian sistem pentautan dilakukan dengan beberapa variasi citra masukan, yaitu variasi skala, rotasi, dan overlap. Berdasarkan hasil pengujian, metode Reduksi data yang diterapkan memiliki nilai radius minimal optimum pada rentang 40 piksel hingga 100 piksel. Proses komputasi tetap dapat dikerjakan dengan reduksi jumlah keypoint hingga 90% dari jumlah awal keypoint. Rata-rata waktu komputasi dengan menggunakan metode reduksi data 39,41% lebih cepat daripada tanpa metode reduksi data.

Kata kunci— FLANN, RANSAC, eliminasi keypoint, radius minimal

Abstract

One of the algorithm for aerial image stitching system is SURF (*Speeded Up Robust Features*). It is a robust algorithm which is invariant to image scale, rotation, blurring, illumination, and affine transformation. Although SURF has good performance, some of the detected keypoints are not always considered as necessary keypoints. As a result, these unnecessary keypoints are needed to be eliminated to decrease computation time.

The proposed system uses SURF detector in the detection process. The data reduction method will eliminate couple of keypoints which have near distance each other. Next, the keypoints will be described by SURF descriptor. The description Results further matched using FLANN. The next step is the search pattern with RANSAC homography matrix and stitch the picture to accumulate keypoints using *warpPerspective*.

Stitching system are tested with some variations, such as scale variations, rotation variations, and overlap variations on the image. Based on the result, the proposed Data Reduction method has optimum value of minimal radius from 40 pixels to 100 pixels. The stitching process is still working with up to 90% keypoint number reduction. Average computation time using data reduction method are 39,41% faster than without data reduction method.

Keywords— FLANN, RANSAC, keypoint elimination, minimal radius

1. PENDAHULUAN

Dewasa ini perkembangan teknologi telah membawa kemajuan dalam berbagai aspek kehidupan manusia. Salah satu teknologi yang menarik perhatian dalam perkembangannya adalah teknologi pengolahan citra. Pengolahan citra dapat dikerjakan menggunakan alat, seperti kamera yang diinstalasikan pada UAV (*Unmanned Aerial Vehicle*), yang digunakan untuk mengambil citra pada area uji untuk kebutuhan analisis sehingga memungkinkan peneliti mendapatkan informasi-informasi yang diperlukan [1].

Proses pengambilan citra udara harus dilakukan berulang untuk mendapatkan citra dengan cakupan area yang luas [1]. Hasil dari pengambilan citra udara tersebut berupa citra yang terpisah antara satu citra dengan citra yang lain. Citra yang sudah diambil dapat ditautkan dengan menggunakan algoritme SURF (*Speeded Up Robust Features*) [2]. Algoritme ini bekerja dengan menentukan titik acuan (*keypoint*) dari sebuah citra yang nilainya tetap ketika mengalami perubahan skala, rotasi, *blurring*, pencahayaan, dan perubahan bentuk [3,4,5].

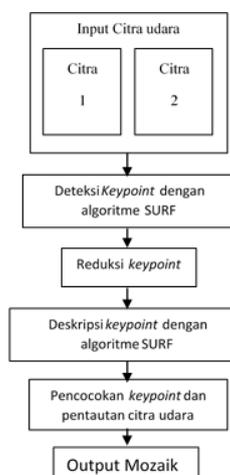
Walaupun algoritme SURF memiliki kemampuan yang baik dalam proses deteksi *keypoint*, sebagian hasil dari deteksi tersebut tidak selalu dianggap sebagai *keypoint* yang menarik pada citra yang diujikan [6]. Oleh sebab itu, *keypoint* tersebut hanya akan menambah waktu komputasi pada algoritma SURF sementara dalam prosesnya, setidaknya hanya dibutuhkan 4 pasang dari hasil pencocokan *keypoint* antar citra supaya proses pentautan citra dapat dilakukan. Dengan demikian *keypoint* yang tidak menarik perlu dihilangkan [6].

2. METODE PENELITIAN

2.1 Analisis Sistem

Pada penelitian ini dibuat sistem yang mampu mentautkan citra udara dengan tampilan citra terbatas, menjadi sebuah keluaran yang menghasilkan tampilan citra dengan cakupan area yang lebih luas.

Sistem yang dirancang membutuhkan dua buah masukan citra pada setiap prosesnya. Citra pertama merupakan citra uji yang diatur sedemikian sehingga dapat dilakukan pengujian sedangkan citra kedua merupakan citra acuan yang ukuran dan bentuknya tidak dirubah. Kedua citra udara yang selesai diakuisisi akan ditemukan *keypoint* dari masing-masing citra dengan algoritme SURF. Jumlah *keypoint* yang terdeteksi akan dieliminasi sebagian dengan metode reduksi data. Setelah jumlah *keypoint* citra sudah tereduksi, dilakukan proses deskripsi *keypoint* untuk dicocokkan dengan kedua citra udara. Setelah proses tersebut selesai maka hasil pencocokkan akan menjadi acuan untuk melakukan pentautan kedua citra. Gambar 1 menunjukkan blok diagram rancangan sistem pentautan citra udara.



Gambar 1 Diagram blok sistem pentautan citra udara

2.2 Peralatan

Peralatan sistem yang digunakan untuk mendukung proses penelitian ini terdiri dari perangkat keras dan perangkat lunak yang masing-masing memiliki spesifikasi sebagai berikut.

2. 2.1 Spesifikasi Perangkat Keras

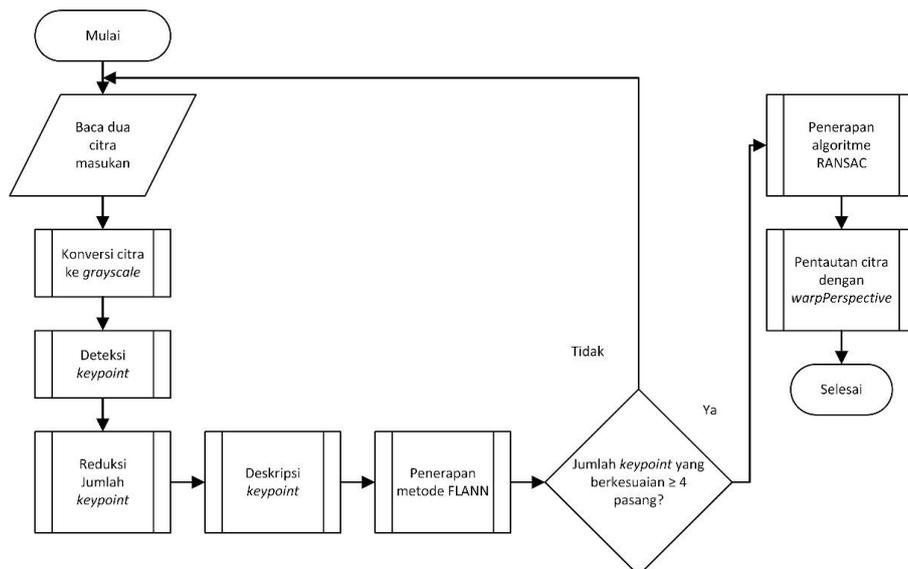
1. Prosesor AMD Vision A4 Dual-Core 1,9GHz
2. RAM 4GB DDR3
3. GPU AMD Radeon™ HD 6480G

2. 2.2 Sistem Perangkat Lunak

1. Sistem operasi Linux Ubuntu 14.04 LTS sebagai sarana pengoperasian sistem
2. IDE Geany dengan pustaka OpenCV 3.1.0 yang telah terintegrasi di dalamnya sebagai *compiler* program

2.3 Rancangan Perangkat Lunak

Sistem yang dirancang merupakan sistem yang dapat mentautkan dua buah masukan citra udara pada setiap prosesnya sehingga akan diperoleh hasil pentautan berupa citra dengan cakupan yang lebih luas. Berikut merupakan diagram alir sistem pentautan citra udara yang ditunjukkan pada Gambar 2.



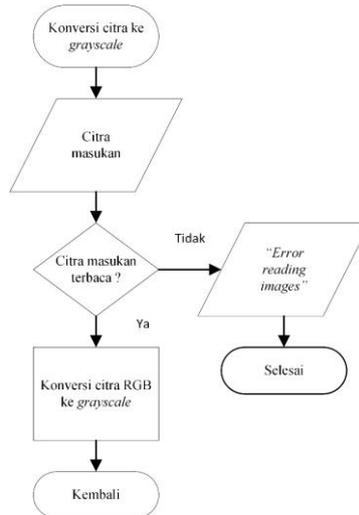
Gambar 2 Diagram alir sistem secara keseluruhan

Citra udara yang selesai diakuisisi akan ditemukan *keypoint* dari masing-masing citra dengan algoritme SURF. Kemudian *keypoint* yang terdeteksi akan dieliminasi sebagian dengan metode reduksi data. Setelah jumlah *keypoint* pada citra sudah tereduksi, dilakukan proses deskripsi *keypoint* dengan deskriptor SURF. *Keypoint* pada kedua citra masukan yang sudah dideskripsi kemudian dicocokkan menggunakan FLANN. Hasil pencocokkan tersebut akan dijadikan acuan untuk mencari matriks homograf menggunakan RANSAC dimana matriks tersebut akan digunakan dalam proses *warpPerspective*. Hasil dari proses *warpPerspective* berupa hasil pentautan kedua citra masukan.

2. 3.1 Konversi Citra ke Grayscale

Tahap awal sebelum proses pengolahan citra adalah mengonversikan citra dari RGB ke *grayscale*. Tahap ini diawali dengan pembacaan citra masukan. Sistem akan berjalan apabila terdeteksi dua buah citra masukan pada direktori. Kedua citra tersebut adalah citra uji dan citra acuan. Masing-masing citra masukan merupakan citra RGB, ketika kedua citra terdeteksi, maka

dilakukan proses konversi RGB ke *grayscale*. Keadaan awal citra terdiri dari 3 lapisan matrik, yaitu *R-layer*, *G-layer* dan *B-layer*. Kondisi citra dengan 3 lapisan matrik RGB ini membuat proses-proses dalam pengolahan selanjutnya harus memperhatikan ketiga lapisan matriks tersebut. Bila setiap proses perhitungan harus memperhatikan 3 layer, maka harus dilakukan 3 perhitungan yang sama hal ini memerlukan waktu yang relatif lama, sehingga untuk meminimalisir proses perhitungan pengolahan citra yakni dengan penggunaan citra dalam kondisi *grayscale*. Citra *grayscale* hanya memiliki derajat keabuan. Derajat keabuan pada citra *grayscale* merupakan nilai rata-rata dari *R-layer*, *G-layer* dan *B-layer*. Program akan dihentikan jika tidak terdeteksi citra masukan. Gambar 3 menunjukkan diagram alir konversi citra ke *grayscale*.

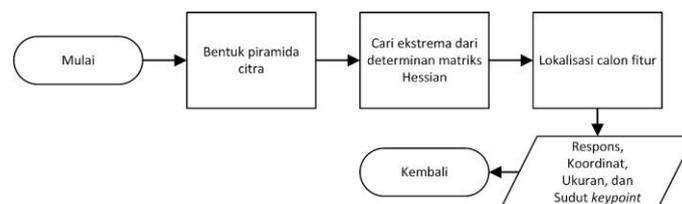


Gambar 3 Diagram alir konversi citra ke *grayscale*

2. 3.2 Deteksi Keypoint

Hasil konversi RGB ke *grayscale* kemudian dilanjutkan dengan pendeteksian *keypoint*. Fitur gelembung (*blob-like feature*) digunakan untuk mendeteksi *keypoint*. Langkah yang dilakukan adalah membentuk piramida citra dengan menggunakan *box filter* sebagai aproksimasi dari turunan parsial kedua dari *Gaussian*. Ketika membentuk *scale space*, citra asli dikonvolusikan pada kotak filter dengan mengubah ukuran kotak filter sesuai dengan citra dan membentuk *scale space image* [4,7].

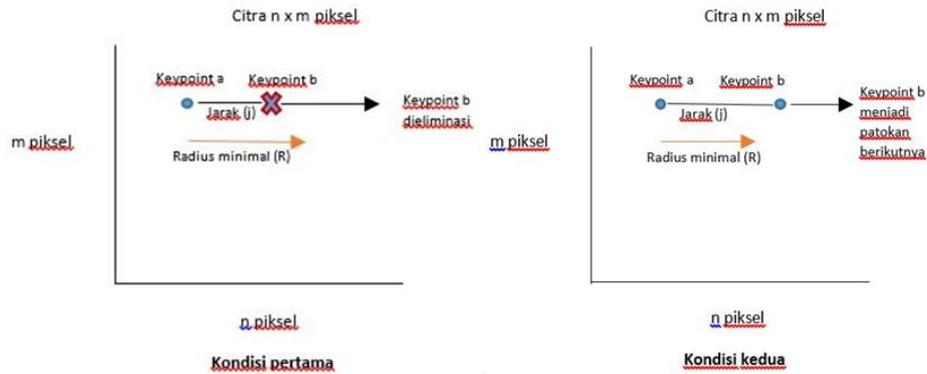
Langkah berikutnya adalah mencari ekstrema dari determinan matriks *Hessian* dibandingkan dengan tetangga-tetangganya. Lokalisasi calon fitur kemudian dilakukan pada setiap ruang skala (*scale space*) dengan menggunakan *non-maximum suppression* terhadap eksterma dari determinan matriks *Hessian*. Hasil akhir dari tahap ini adalah titik-titik acuan atau *keypoint* yang menunjukkan adanya fitur gelembung (*blob-like feature*) [4,7]. Gambar 4 menunjukkan proses pendeteksian fitur dengan SURF.



Gambar 4 Diagram alir deteksi *keypoint*

2. 3.3 Reduksi Jumlah Keypoint

Hasil dari deteksi *keypoint* pada citra kemudian akan direduksi jumlahnya dengan cara mencari nilai jarak antara *keypoint* pertama yang berperan sebagai *keypoint* acuan dengan *keypoint* berikutnya yang berperan sebagai *keypoint* tetangga. Kemudian jarak tersebut akan dibandingkan dengan nilai radius minimal yang akan divariasikan.

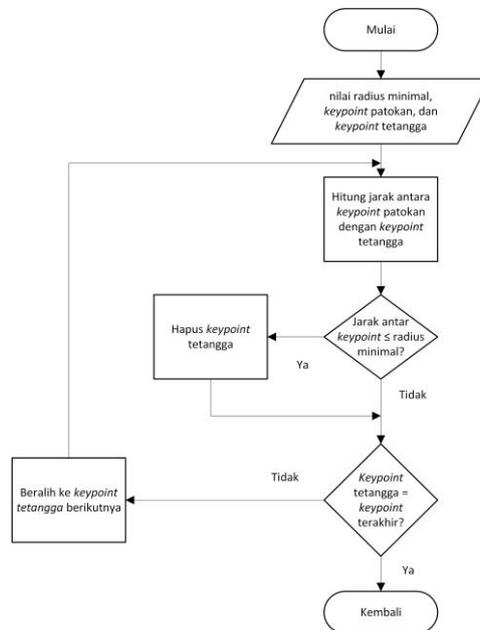


Gambar 5 Proses eliminasi *keypoint*

Berdasarkan Gambar 5 :

- Jika jarak antara *keypoint* a dengan b lebih kecil sama dengan radius minimal ($j \leq R$), maka *keypoint* b dieliminasi.
- Sebaliknya, Jika jarak antara *keypoint* a dengan b lebih besar dari radius minimal ($j > R$), maka *keypoint* b tidak dieliminasi dan menjadi *keypoint* patokan berikutnya.

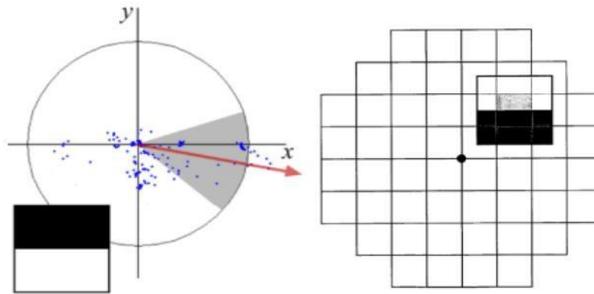
Gambar 6 menunjukkan diagram alir proses reduksi jumlah *keypoint*.



Gambar 6 Diagram alir reduksi jumlah *keypoint*

2. 3.4 Deskripsi Keypoint

Pendeskripsian *keypoint* diawali dengan menghitung respon *Haar-wavelet* terhadap sumbu-x dan sumbu-y dengan titik-titik di lingkungan tetangganya pada radius $6s$ di sekitar titik-titik fitur, dengan s adalah skala pada titik fitur yang terdeteksi seperti pada Gambar 7. Langkah *sampling* tiap skala adalah s dari masing-masing skala begitu juga perhitungan respon *wavelet*-nya sesuai dengan skalanya sehingga pada skala yang besar ukuran dari *wavelet* akan besar pula. Perhitungan citra integral digunakan untuk penapisan yang cepat. Dengan demikian hanya dibutuhkan enam operasi untuk menghitung respon pada sumbu-x dan sumbu-y di setiap skala [4].

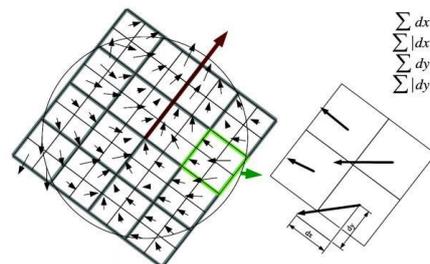


Gambar 7 Pemberian orientasi [4]

Langkah pertama untuk mengekstraksi deskriptor adalah membuat daerah berbentuk bujur sangkar yang berpusat di sekitar titik fitur dan orientasinya mengarah ke orientasi yang sudah ditentukan sebelumnya. Ukuran dari jendela bujur sangkar tersebut adalah $20s$. Setiap wilayah kemudian dibagi lagi ke dalam 4×4 sub-area. Pada bagian ini disimpan informasi spasial. Perhitungan *Haar Wavelet* kemudian dilakukan pada setiap sub-area dengan membaginya ke dalam 5×5 ruang sampel. Pada umumnya, dx disebut sebagai respon *Haar* pada arah horizontal dan dy sebagai *Haar wavelet* pada arah vertikal. Peningkatan *robustness* ketika terjadi transformasi geometrik dan kesalahan lokal, respon di dx dan dy difokuskan pada pusat *Gaussian* di *keypoint* [4]. Persamaan berikut menunjukkan vektor 4D yang diperoleh:

$$V = (\Sigma d_x, \Sigma |d_x|, \Sigma d_y, \Sigma |d_y|) \quad (1)$$

Bila ditempatkan pada masing-masing 4×4 sub-area akan didapatkan vektor deskriptor dengan panjang 64. Respon *wavelet* bervariasi terhadap perubahan iluminasi [4]. Faktor skala didapatkan dengan merubah deskriptor ke dalam vektor unit seperti pada Gambar 8.



Gambar 8 Komposisi deskriptor *keypoint* [4]

2. 3.5 Metode FLANN

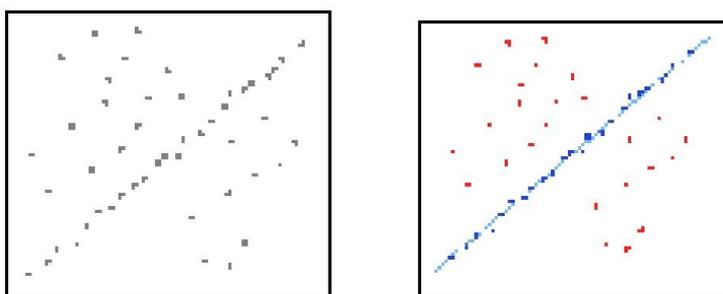
Metode *Fast Library Approximated Nearest Neighbor* (FLANN) adalah sebuah pustaka untuk melakukan pencarian cepat, perkiraan *neighborhood*, yang terdapat pada *space* dimensi yang tinggi. Pustaka ini merupakan kumpulan algoritme yang bekerja dengan baik untuk menemukan nilai tetangga terdekat, sementara untuk hasil parameter yang optimal, tergantung pada kumpulan data yang digunakan. Fitur SURF terdiri dari *keypoint*, deskriptor, dan berupa vektor. Satu citra di basis data memiliki banyak kluster untuk fitur SURF. Kluster ini dibuat

otomatis dengan menggunakan KNN (*K-Nearest Neighbor*) dengan tipe indeks KD tree, dimana KNN ini akan mencari jarak yang paling kecil antara vektor fitur dengan vektor pada kluster [8].

Proses pencocokkan fitur pada citra *query* dan fitur pada citra dalam basis data, vektor *keypoint*, dan deskriptor pada gambar *query* akan dicocokkan nilainya menggunakan KNN *search*. KNN *search* akan mencari kluster pada basis data yang nilai vektor deskriptornya paling dekat jaraknya dengan vektor deskriptor pada citra *query*. Setelah kluster didapat, kemudian akan dicari nilai vektor deskriptor pada kluster tersebut yang sama atau paling dekat dengan vektor deskriptor pada citra *query*. Jika ada yang sama maka ada satu *keypoint* yang cocok antara kedua citra tersebut. Semakin banyak jumlah *keypoint* yang cocok, maka dianggap paling baik dalam mendeteksi titik fitur [8].

2. 3.6 Algoritme RANSAC

Metode RANSAC (*Random Sample Consensus*) digunakan untuk *keypoint* yang telah dicocokkan pada *matching keypoint*. Metode ini juga dapat menentukan *inlier* dan *outlier*. *Inlier* adalah titik yang kira-kira disambungkan menjadi garis, sedangkan *outlier* adalah titik yang tidak disambungkan menjadi garis. Titik-titik *inlier* dari sampel poin acak kemudian dipilih oleh RANSAC untuk digabungkan menjadi sebuah garis. Algoritme RANSAC ini sangat bergantung dengan hasil dari tahap sebelumnya. Hal ini bertujuan untuk memperoleh pasangan kecocokan *keypoint* yang benar-benar sama [9]. Penggambaran algoritme RANSAC diberikan pada Gambar 9:



Gambar 9 Gambaran tentang RANSAC. Kiri : kumpulan dengan banyak outlier. Kanan : kumpulan data setelah proses RANSAC [9]

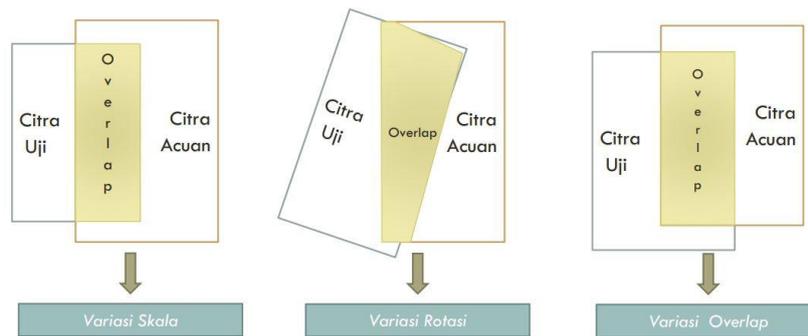
2. 3.7 Pentautan Citra dengan *warpPerspective*

Proses *warping* citra merupakan tahapan terakhir pada sistem pentautan citra yang diolah. Pada awal tahapan ini dilakukan pencarian pola kesamaan kedua citra atau matriks homograf menggunakan RANSAC. Penggunaan RANSAC dilakukan dengan menghilangkan bagian yang tidak cocok hingga didapatkan pasangan yang benar-benar sama yang membentuk suatu pola. RANSAC memiliki kemampuan untuk menghilangkan *outliers* yang ditemukan sehingga *inliers* yang diperoleh dapat digunakan sebagai penentuan pola kesamaan yang maksimal. Dari tahapan ini kemudian dilakukan pentautan kedua citra masukan menggunakan *warpPerspective* berdasarkan pola kesamaannya. Bagian *keypoints* atau pasangan *keypoints* yang sama akan saling bertautan sehingga bagian yang memiliki kesamaan akan ditumpuk. Hasil dari proses inilah yang berupa citra hasil pentautan atau citra mozaik [10].

2.4 Rencana Pengujian

Pengujian dilakukan untuk mengetahui nilai radius minimal yang dapat digunakan untuk mereduksi jumlah *keypoint* citra. Pada penelitian ini terdapat tiga variasi pengujian, yaitu variasi rotasi, skala, dan *overlap* citra. Citra yang diuji memiliki ukuran 1100 x 1500 piksel.

Pada setiap pengujian, jumlah *keypoint* setelah direduksi, lama waktu komputasi, dan kesuksesan pentautan citra dicatat untuk dianalisis hasilnya. Gambar 10 menunjukkan ilustrasi pengujian variasi citra.



Gambar 10 Ilustrasi pengujian variasi citra

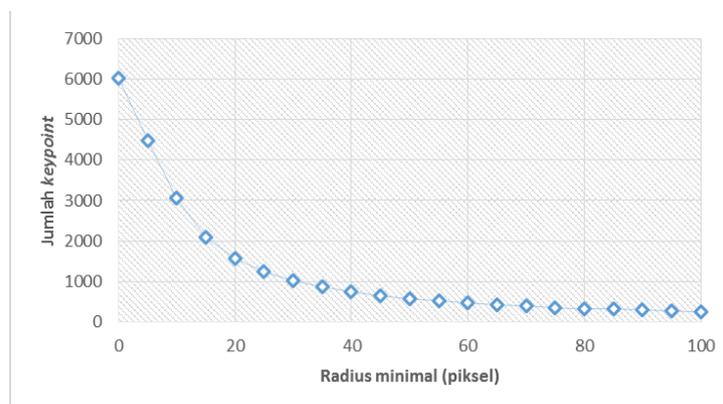
3. HASIL DAN PEMBAHASAN

Pada bagian ini dipaparkan hasil pengujian serta pembahasannya yang telah dijabarkan pada sub bab 2.4.

3.1 Hasil Pengujian Variasi Radius Minimal

Variasi nilai radius minimal yang digunakan diperoleh dari proses *trial and error*, yaitu dimulai dari 5 piksel hingga 100 piksel dengan interval 5 piksel per kenaikan nilainya. Nilai-nilai tersebut diperoleh dan ditetapkan berdasarkan kriteria berikut:

1. Jumlah *keypoint* yang direduksi sudah tidak signifikan.
2. Jumlah *keypoint* yang cocok.
3. Keberhasilan pentautan citra.

Gambar 11 Grafik jumlah *keypoint* citra

Berdasarkan pada Gambar 11, metode reduksi data mampu mereduksi jumlah *keypoint* sekitar 25% dari jumlah awal pada nilai radius minimal 5 piksel. Kemudian jumlah *keypoint* tereduksi sekitar 50% hingga 75% dari jumlah awal pada nilai radius minimal 10 piksel hingga 20 piksel. Jumlah *keypoint* tereduksi sekitar 90% pada nilai radius minimal 40 piksel hingga 50 piksel. Sementara jumlah *keypoint* yang tereduksi mulai tidak signifikan pada nilai radius minimal 55 piksel hingga 100 piksel, yaitu sekitar 1% per kenaikan nilai radius minimalnya.

3.2 Hasil Pengujian Skala

Pengujian dengan variasi skala ini bertujuan untuk mengetahui pengaruh nilai radius minimal terhadap waktu komputasi dan hasil pentautan citra masukan. Perubahan skala didapat dari perubahan yang dilakukan oleh perangkat lunak pengolah citra. Pengujian ini dilakukan dengan memasukkan citra uji yang memiliki ukuran skala bervariasi dari 50% hingga 350%

dengan kenaikan setiap 10% dan citra acuan dengan skala sebesar 100%. Tabel 2 menunjukkan sebagian hasil pengujian skala.

Tabel 1 Hasil pengujian skala

Skala (%)	Radius Minimal Optimum (piksel)	Keypoint Citra Acuan		Keypoint Citra Uji		Good Matches		Waktu (s)	
		SURF	SURF & RD	SURF	SURF & RD	SURF	SURF & RD	SURF	SURF & RD
50	40	6020	749	2998	316	31	4	4,71	2,72
100	75	6020	359	7848	412	1774	15	6,30	3,48
150	70	6020	397	6656	412	24	9	5,57	3,19
200	60	6020	467	5117	449	8	7	5,29	2,90
250	60	6020	467	3942	411	14	4	4,96	2,71
300	40	6020	749	3092	594	16	8	4,81	2,80
350	35	6020	865	2565	659	8	5	3,50	2,89

Hasil dari pengujian skala menunjukkan kisaran nilai optimum berada pada 25 piksel hingga 100 piksel dengan rata-rata 53,71 piksel. Proses reduksi jumlah *keypoint* dapat menghasilkan selisih waktu komputasi pentautan citra dengan rata-rata 2,13s

3.3 Hasil Pengujian Rotasi

Pengujian dengan variasi rotasi ini bertujuan untuk mengetahui pengaruh nilai radius minimal terhadap waktu komputasi dan hasil pentautan citra masukan. Perubahan rotasi didapat dari perubahan yang dilakukan oleh perangkat lunak pengolah citra. Pengujian ini dilakukan dengan memasukkan citra uji yang sudutnya diputar 10^0 hingga 350^0 searah jarum jam dengan kenaikan setiap 10^0 serta sudut kritis 1^0 hingga 9^0 dan 351^0 hingga 359^0 sementara citra acuan memiliki putaran sebesar 0^0 . Pentautan citra dikatakan berhasil apabila citra yang dihasilkan mampu mentautkan potongan dengan benar, walaupun tidak keseluruhan citra yang tertaut, Tabel 2 menunjukkan hasil sebagian pengujian rotasi.

Tabel 2 Hasil pengujian rotasi

Rotasi (0)	Radius Minimal Optimum (piksel)	Keypoint Citra Acuan		Keypoint Citra Uji		Good Matches		Waktu (s)	
		SURF	SURF & RD	SURF	SURF & RD	SURF	SURF & RD	SURF	SURF & RD
0	75	6020	359	7848	412	1774	15	6,3	3,48
30	70	6020	397	6534	379	4	16	5,87	3,65
60	100	6020	259	6365	247	7	6	5,81	3,12
90	75	6020	359	6059	315	1305	4	5,81	3,44
120	100	6020	259	6447	244	18	7	5,63	3,58
150	45	6020	643	6743	653	16	6	5,29	3,72
180	95	6020	266	7840	306	1899	5	6,29	3,78
210	40	6020	749	6461	755	10	8	5,97	3,93
240	40	6020	749	6363	715	5	9	5,72	3,91
270	95	6020	266	6045	235	1323	5	5,68	3,31
300	65	6020	433	6429	407	10	29	5,61	3,72
330	55	6020	525	6784	516	6	15	5,72	3,66

Hasil dari pengujian rotasi menunjukkan kisaran nilai optimum berada pada 40 piksel hingga 100 piksel dengan rata-rata 70,74 piksel. Proses reduksi jumlah *keypoint* dapat menghasilkan selisih waktu komputasi pentautan citra dengan rata-rata 2,19s.

3.4 Hasil Pengujian Overlap

Pengujian dengan variasi *overlap* ini bertujuan untuk mengetahui pengaruh nilai radius minimal terhadap waktu komputasi dan hasil pentautan citra masukan. Perubahan *overlap* didapat dari perubahan yang dilakukan oleh perangkat lunak pengolah citra. Pengujian ini

dilakukan dengan memasukkan citra uji yang memiliki ukuran *overlap* bervariasi dari 10% hingga 90% secara horizontal dan 10% hingga 100% secara vertikal dengan kenaikan setiap 10% dan citra acuan dengan skala sebesar 100%. Tabel 3 dan Tabel 4 menunjukkan sebagian hasil pengujian *overlap*.

Tabel 3 Hasil pengujian *overlap* horizontal

Overlap (%)		Radius Minimal Optimum (piksel)	Keypoint Citra Acuan		Keypoint Citra Uji		Good Matches		Waktu (s)	
H	V		SURF	SURF & RD	SUR F	SUR F & RD	SUR F	SURF & RD	SURF	SURF & RD
10	10	15	6020	2086	913	286	25	6	3,52	2,99
20	20	40	6020	749	1681	180	133	8	3,83	2,54
30	30	85	6020	314	2708	117	306	5	4,21	2,51
40	40	70	6020	397	3586	194	616	5	4,46	2,70
50	50	75	6020	359	4264	218	974	5	4,80	2,82
60	60	95	6020	266	4614	193	1586	8	4,96	2,87
70	70	95	6020	266	4744	219	1792	4	5,29	2,90
80	80	90	6020	296	5047	250	2525	9	6,09	3,37
90	90	95	6020	266	5874	261	3014	13	5,50	3,16

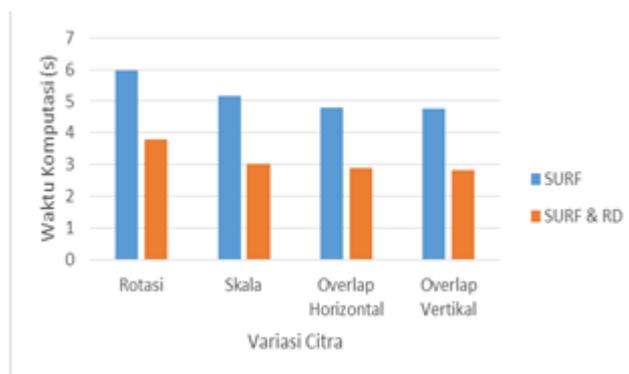
Tabel 4 Hasil pengujian *overlap* vertikal

Overlap (%)		Radius Minimal Optimum (piksel)	Keypoint Citra Acuan		Keypoint Citra Uji		Good Matches		Waktu (s)	
H	V		SURF	SURF & RD	SUR F	SUR F & RD	SUR F	SURF & RD	SURF	SURF & RD
10	10	15	6020	2086	858	306	31	6	3,87	2,8
20	20	85	6020	314	1729	83	205	5	3,92	2,39
30	30	95	6020	266	2358	99	237	5	4,14	2,45
40	40	95	6020	266	3139	126	460	4	4,36	2,56
50	50	95	6020	266	3928	161	808	5	4,63	2,7
60	60	100	6020	259	4544	177	2036	12	4,91	2,87
70	70	95	6020	266	5219	219	1848	11	5,16	3
80	80	100	6020	259	5806	230	2656	12	5,67	3,17
90	90	95	6020	266	6150	268	3034	19	5,57	3,26

Secara keseluruhan, hasil dari pengujian *overlap* horizontal citra menunjukkan kisaran nilai optimum berada pada 15 piksel hingga 100 piksel dengan rata-rata 77,53 piksel dan hasil dari pengujian *overlap* vertikal citra menunjukkan kisaran nilai optimum berada pada 15 piksel hingga 100 piksel dengan rata-rata 83,77 piksel. Proses reduksi jumlah *keypoint* dapat menghasilkan selisih waktu komputasi pentautan citra dengan rata-rata 2,19s pada uji *overlap* horizontal citra dan rata-rata 1,92s pada uji *overlap* vertikal citra.

3.5 Hasil Perbandingan Waktu Komputasi

Berdasarkan hasil keseluruhan pengujian terhadap variasi skala, rotasi, dan *overlap* citra, diperoleh perbandingan rata-rata waktu komputasi algoritme SURF dengan menggunakan metode reduksi data dan tanpa menggunakan metode tersebut yang ditunjukkan pada Gambar 12.



Gambar 12 Grafik perbandingan waktu komputasi menggunakan dan tanpa menggunakan metode reduksi data

Berdasarkan Gambar 12, diketahui metode reduksi data yang diterapkan dapat menghasilkan waktu komputasi pentautan citra dengan rata-rata 39,41% lebih cepat daripada waktu komputasi pentautan citra tanpa menggunakan metode reduksi data.

4. KESIMPULAN

Berdasarkan hasil pengamatan, pengujian, dan analisis pada hasil-hasil yang diperoleh, kesimpulan yang dapat diambil pada penelitian ini adalah:

1. Telah berhasil dibuat implementasi sistem pentautan citra udara menggunakan algoritme SURF dan metode reduksi data dengan melakukan pengujian terhadap variasi skala, rotasi, dan *overlap* citra.
2. Proses pentautan citra tetap dapat dikerjakan dengan reduksi jumlah *keypoint* citra hingga 90% dari jumlah awal *keypoint*.
3. Metode reduksi data dapat mengurangi waktu komputasi dengan rata-rata 39,41% dari waktu komputasi tanpa metode reduksi data.
4. Nilai radius minimal yang optimum berada pada rentang 40 hingga 100 piksel.

5. SARAN

Diperlukan penelitian lebih lanjut terhadap sistem pentautan citra udara menggunakan algoritme SURF dan metode reduksi data dengan menggunakan lebih dari dua citra masukan dan mengurangi peluang kegagalan pentautan citra dalam penerapan metode reduksi data.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada semua pihak yang telah memberi bimbingan, dukungan, dan saran selama kegiatan penelitian berjalan.

DAFTAR PUSTAKA

- [1]Wirawan, S., 2014, Sistem Citra Udara Pesawat Tanpa Awak dengan Pentautan Citra, *Skripsi*, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Gadjah Mada, Yogyakarta.
- [2]Dewanti, F., 2015, Purwarupa Sistem Pentautan Citra Udara Pada UAV Menggunakan Algoritme SURF (*Speeded-Up Robust Features*), *Skripsi*, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Gadjah Mada, Yogyakarta.

-
- [3]Sardi, I. L., 2012, *Analisis Implementasi Fast Corner Detector pada Image Stitching dalam Pembentuk Citra*, Institut Teknologi Telkom, Bandung.
- [4]Bay, H., Ess, A., Tuytelaars, T., dan Gool, L. V., 2006, SURF : Speeded-UP Robust Features. *Proceeding of 9th European Conference on Computer Vision*, Graz, Austria, May 7-13.
- [5]Pedersen, J. ,2011, *Study Group SURF : Feature Detection & Description*, Aarhus University, Denmark.
- [6]Wang, Z.-l., Yan, F.-h., dan Zheng, Y.-y., 2013, An Adaptive Uniform Distribution SURF for Image Stitching, *Proceeding of 6th International Congress on Image and Signal Processing (CISP 2013)*, Hangzhou, December 16-18.
- [7]Sukrawan, I., 2008, *Pengembangan DSMAC menggunakan Metoda SURF pada Sistem Peluru Kendali dan Komputer*, Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung, Bandung.
- [8]Tania, K. D. dan Arymurthy, A. M., 2010, *Tattoo Recognition Based on Speeded Up Robust Features (SURF)*, University of Sriwijaya, Palembang.
- [9]Darajati, Aisah., 2012, *Implementasi dan Analisis Citra Mosaik Berbasis Fitur dengan Metode Global Alignment untuk Pembentukan Citra Panorama pada Android*, Universitas Telkom, Bandung.
- [10]Astuti, R.D., 2011, Sistem Pentautan Foto Udara Menggunakan Deteksi Fitur Algoritma Oriented FAST And Rotated BRIEF, *Skripsi*, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Gadjah Mada, Yogyakarta.