

# Optimasi Algoritma Semut untuk Deteksi Tepi pada Foto Udara

Febri Liantoni<sup>1</sup>, Rifki Indra Perwira<sup>2</sup>

<sup>1</sup>Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Adhi Tama Surabaya  
E-mail: febri.liantoni@gmail.com

<sup>2</sup>Teknik Informatika, Fakultas Teknologi Industri, UPN "Veteran" Yogyakarta  
E-mail: rifki@upnyk.ac.id

## ABSTRACT

*The photographic image is a digital image that is useful to obtain information from an area. This information obtained based on the data analysis object using media or tool. The information is processed by using digital image processing techniques. One of the techniques to obtain information is based on digital image processing with the edge detection. Edge detection is the process of extracting information from an image. One of detection technique that can be used is ant algorithm. Ant algorithm is optimal algorithm inspired by foraging behavior of ants. In general, the initial distribution process ants on an ant algorithm is by randomly. This allows an imbalance of distribution of ants which then affects the path discovery process. The Optimization is done by spreading the ants based on the level gradient in the image area. Steps taken by dividing the image into 16 areas of equal size. The part that has a high gradient will get of ants more. Ant algorithm results will be compared with conventional edge detection. The comparison process is performed to determine differences in edge detection results obtained. Based on the experiment, the optimization of ant algorithms capable of generating image edge detection more detail and have thicker margins than using conventional edge detection.*

**Keywords:** *photographic image, digital image processing, ant algorithm, gradient.*

## INTISARI

Foto udara merupakan citra digital yang berguna untuk memperoleh informasi dari suatu daerah. Informasi diperoleh berdasarkan analisis data obyek menggunakan media atau alat. Informasi ini diproses dengan memanfaatkan teknik pengolahan citra digital. Salah satu cara memperoleh informasi berdasarkan teknik pengolahan citra digital melalui proses deteksi tepi. Deteksi tepi adalah penggalian informasi dari suatu citra. Salah satu teknik deteksi tepi yang bisa dipakai yaitu menggunakan algoritma semut. Algoritma semut merupakan algoritma optimasi yang terinspirasi dari perilaku semut dalam mencari makan. Pada umumnya penyebaran semut awal pada algoritma semut dilakukan secara acak. Hal ini memungkinkan terjadi ketidakseimbangan distribusi semut yang kemudian mempengaruhi penemuan jalur. Optimasi dilakukan dengan cara menyebarkan semut berdasarkan tingkat *gradient* pada area citra. Langkah yang dilakukan dengan membagi citra menjadi 16 area dengan ukuran sama besar. Bagian yang memiliki gradient tinggi akan mendapatkan pembagian semut yang lebih banyak. Hasil algoritma semut akan dibandingkan dengan deteksi tepi konvensional. Perbandingan dilakukan untuk mengetahui perbedaan hasil deteksi tepi yang diperoleh. Berdasarkan percobaan yang dilakukan, optimasi algoritma semut mampu menghasilkan deteksi tepi citra yang lebih detail dan memiliki garis tepi yang lebih tebal dibandingkan menggunakan deteksi tepi konvensional.

**Kata kunci:** *foto udara, pengolahan citra digital, algoritma semut, gradient.*

## PENDAHULUAN

Foto udara bisa dikatakan sebagai produk dari penginderaan jauh. Foto udara diperoleh dengan bantuan alat seperti wahana balon udara, pesawat udara, maupun gantole dan tanpa kotak langsung dengan obyek yang direkam. Foto udara digunakan untuk memperoleh informasi suatu daerah atau obyek yang diinginkan dengan analisis data yang diperoleh dengan menggunakan media atau alat. Metode untuk melakukan pemrosesan terhadap citra masukan berupa foto udara dengan pemanfaatan teknik pengolahan citra digital. Seringkali citra yang digunakan tidak dalam kondisi yang ideal untuk dikaji dikarenakan banyaknya gangguan, dapat berupa bayangan, foto atau gambar kabur,

kurang jelasnya kenampakan obyek sehingga dapat menimbulkan masalah dan mempengaruhi hasil interpolasi serta akan mempengaruhi analisa dan perencanaan yang akan dilakukan, maka diperlukan berbagai teknik pengolahan citra untuk memperoleh citra yang ideal. Salah satu cara untuk memperoleh informasi dari foto udara yaitu dengan pengolahan citra digital melalui proses deteksi tepi.

Deteksi tepi merupakan penggalian informasi tepi dari sebuah gambar. Hal ini dianggap sebagai langkah dasar yang digunakan dalam sebagian besar aplikasi pengolahan citra (Verma dkk, 2011). Tepi dalam sebuah gambar dapat dianggap sebagai batas antara dua daerah yang berbeda. Banyak pendekatan telah

digunakan untuk melakukan deteksi tepi pada sebuah gambar. Beberapa metode yang umum digunakan adalah prewitt, roberts, dan sobel (Dorigo dkk, 2006). Proses deteksi tepi pola foto udara dapat dilakukan dengan mengenali karakteristik bentuk dari hasil foto. Beberapa pendekatan yang dilakukan pada proses deteksi tepi gambar yang menghasilkan tepi yang baik misalnya menggunakan algoritma semut. Pendekatan algoritma semut atau biasa disebut *Ant Colony* merupakan metode *heuristic* yang meniru perilaku semut untuk memecahkan masalah optimasi *discrete* (Dorigo dkk, 2006). Algoritma ini menggambarkan bagaimana semut menggunakan senyawa kimia khusus yang disebut feromon untuk menandai jalur antara sumber makanan dan koloni mereka. Jalur feromon akan digunakan oleh semut berikutnya sebagai panduan untuk mencari makanan karena feromon meningkatkan kemungkinan jalan untuk dipilih. Algoritma semut biasa digunakan untuk menyelesaikan berbagai permasalahan Non Polinomial seperti *Traveling Salesman Problem* (TSP), deteksi tepi, *network packet routing*, *vehicular routing*, *quadratic assignment problem* (Dorigo dkk, 2006).

Algoritma semut pada umumnya, proses penyebaran semut awal dilakukan secara acak. Kondisi ini dapat menyebabkan ketidakseimbangan distribusi semut yang kemudian mempengaruhi penemuan jalur. Beberapa penelitian dilakukan mengatasi permasalahan ini. Pada tahun 2010, Rahebi dkk melakukan penelitian dengan menggabungkan algoritma semut dan algoritma genetika dalam memperbaiki penyebaran semut untuk meningkatkan konvergensi. Pada tahun 2008, Tian dkk menggunakan *Ant Colony System* (ACS) dengan menghitung jumlah semut untuk mendapatkan hasil deteksi tepi yang lebih baik.

Pada penelitian ini akan dilakukan deteksi tepi pada foto udara dengan menggunakan optimasi algoritma semut. Optimasi algoritma semut diusulkan dengan fokus untuk penyebaran semut berdasarkan tingkat *gradient*. Langkah yang dilakukan dengan membagi citra menjadi 16 area dengan ukuran sama besar. Kemudian semut akan disebar berdasarkan nilai *gradient*. Bagian yang memiliki *gradient* tinggi akan mendapatkan pembagian semut yang lebih banyak. Semut tidak disebar secara acak, akan tetapi ditempatkan di *gradient* tertinggi di tiap area.

Pada penelitian sebelumnya di tahun 2014, peneliti telah melakukan penelitian algoritma semut dengan melakukan penyebaran semut berdasarkan nilai *gradient*, proses yang dilakukan dengan membagi citra menjadi 4 area, kemudian semut disebar secara acak. Hasil yang

didapatkan kurang optimal, karena citra masih menghasilkan tepi yang terkumpul pada area tertentu (Liantoni dkk, 2014).

Berdasarkan latar belakang tersebut maka rumusan masalahnya adalah bagaimana mengatasi kekurangan dari deteksi tepi metode sobel dengan menggunakan deteksi tepi foto udara?

Adapun tujuan dari penelitian ini adalah mendeteksi tepi foto udara menggunakan *Ant Colony Optimization* (ACO).

Penelitian sejenis lainnya adalah pengenalan objek foto udara dengan metode deteksi dilakukan oleh Helen Anderson dengan meneliti batas tepi citra, tekstur citra dan *noise* citra (Helen, 1987).

Pada tahun 2010, Iqbal dan Sani, melakukan penelitian tentang pengekstraksian jalan dari foto udara dengan menggunakan *threshold*. Pada penelitian tersebut hasil jalan tidak terekstraksi dengan baik dan memiliki banyak derau sehingga berubahnya nilai derajat keabuan pada jalan dan menyebabkan gambar hasil tidak sempurna.

Verma dkk (2011) melakukan penelitian deteksi tepi menggunakan optimasi algoritma semut dan *adaptive threshold*. Pada penelitian ini algoritma semut terdapat aturan perpindahan dengan faktor probabilitas pada delapan piksel ketetanggaan. Proses ini dihitung dengan menggunakan matrik perpindahan probabilitas. Piksel dengan faktor probabilitas maksimum dalam mendeteksi ketetanggaan memiliki piksel tepi. Untuk mengurangi perpindahan yang berulang pada semut dilakukan aturan *stopping criteria*, yaitu perpindahan semut akan berhenti jika melewati jalur yang sudah dilewati semut yang lain dan ketika semua piksel ketetanggaan (8 piksel) sudah dilewati semua oleh semut maka perpindahan akan berhenti (Verma dkk, 2011).

Perbedaan penelitian ini dengan penelitian sebelumnya adalah fokus penyebaran semut berdasarkan tingkat *gradient* dengan membagi citra menjadi 16 area sama besar. Semut tidak disebar secara acak namun ditempatkan di *gradient* tertinggi di tiap area sehingga hasil foto tepi udara akan lebih optimal.

#### Algoritma Semut

Algoritma semut merupakan metode *heuristic* yang meniru perilaku semut untuk memecahkan masalah optimasi *discrete*. Semut menggunakan senyawa kimia khusus yang disebut feromon untuk menandai jalur antara sumber makanan dan koloni mereka. Jalur feromon digunakan oleh semut

berikutnya sebagai referensi untuk mencari makanan karena feromon meningkatkan kemungkinan jalan untuk dipilih. Optimasi algoritma semut biasa disebut dengan metode *Ant Colony Optimization* (ACO).

#### Deteksi Berdasarkan *Gradient*

Deteksi tepi merupakan pencarian informasi tepi dari sebuah gambar. Deteksi tepi memiliki tujuan untuk menandai bagian yang menjadi detail dari gambar. Deteksi tepi juga digunakan untuk memperbaiki detail dari gambar yang kabur karena kesalahan atau adanya efek dari perolehan gambar. Deteksi tepi hasil pengembangan dari metode *gradient* citra. Implementasi penggunaan *magnitude* dari *gradient* merupakan langkah awal pada pengolahan citra digital (Gonzalez dan Woods, 1992). Nilai *gradient* pada suatu area menunjukkan adanya keberadaan tepi pada area tersebut. *Gradient* terbentuk atas transisi atau perubahan warna secara gradual. Untuk fungsi  $f(x,y)$  pada koordinat *gradient*  $(x,y)$  didefinisikan sebagai *two-dimensional column vector* dengan Persamaan 1. Untuk *magnitude* dari vektor ditunjukkan oleh Persamaan 2.

$$\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \quad (1)$$

$$\nabla f = \left[ \left( \frac{\partial f}{\partial x} \right)^2 + \left( \frac{\partial f}{\partial y} \right)^2 \right]^{1/2} \quad (2)$$

Komponen dari *gradient* vektor itu sendiri adalah operator linear, tetapi *magnitude* dari vektor adalah bukan merupakan operator linear karena merupakan operasi kuadrat dan akar kuadrat (Gonzalez dan Woods dkk, 1992). Dalam pengolahan citra, operator berbasis turunan menjadi dasar dalam penghitungan. Beberapa contoh yaitu menggunakan operator sobel, prewit, dan robert (Gonzalez dan Woods, 1992). Gambar 1 merupakan *pseudo-convolution kernels* yang digunakan untuk menghitung *gradient magnitude* dengan cepat.

$a_1$	$a_2$	$a_3$
$a_4$	$a_5$	$a_6$
$a_7$	$a_8$	$a_9$

Gambar 1. *Pseudo-Convolution Kernels* Ukuran 3 x 3

Salah satu metode penentuan *gradient* citra yaitu operator sobel. Langkah pertama yang dilakukan operator sobel adalah memperkirakan nilai *gradient* horizontal ( $G_x$ )

dan *gradient* vertikal ( $G_y$ ). Cara ini dilakukan dengan operasi kernel terhadap matrik citra seperti yang ditunjukkan pada Gambar 2. Matrik kernel untuk  $G_x$ , didefinisikan sebagai operator  $K_x$ , sedangkan matrik kernel untuk  $G_y$  didefinisikan sebagai  $K_y$ , seperti pada Persamaan 3.

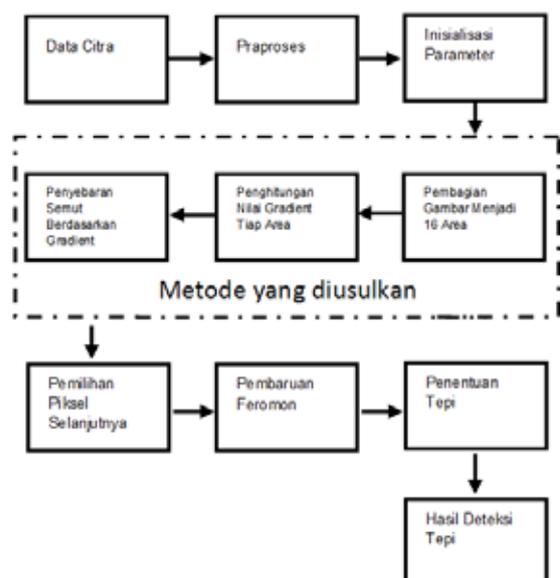
$$K_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad K_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad (3)$$

Setelah nilai *gradient* pada arah vertikal dan horizontal didapatkan, kemudian dicari nilai penghitungan matrik yang menghasilkan nilai *gradient* citra. Nilai *gradient* pada arah vertikal dan horizontal juga digunakan untuk menentukan arah sudut dari tepi ( $\theta$ ). Nilai dari  $\theta$  akan mempengaruhi perlakuan suatu piksel terhadap piksel tetangganya. Penghitungan nilai arah tepi dihitung dengan Persamaan 4.

$$\theta = \tan^{-1} \left( \frac{G_y}{G_x} \right) \quad (4)$$

#### Deteksi Tepi dengan Algoritma Semut

Ada beberapa tahapan yang dilakukan pada algoritma semut. Pada penelitian ini diusulkan deteksi tepi dengan algoritma semut pada penempatan semut awal berdasarkan nilai *gradient*. Gambar 2 adalah blok diagram proses deteksi tepi yang akan dilakukan pada algoritma semut.



Gambar 2. Diagram Metode Optimasi Algoritma Semut

#### Praproses

Data masukan yang digunakan dalam sistem adalah citra RGB. Dari data citra RGB

tersebut diubah menjadi citra keabuan. Perubahan citra warna menjadi citra keabuan menggunakan metode penghitungan *lightness* dan *average*. Selain metode itu ada juga metode *luminosity* yang merupakan pengembangan dari metode *lightness* dan *average*. Metode ini menghitung nilai setiap elemen warna dari citra, yaitu *red*, *green* dan *blue*. Penghitungan ini dengan cara menambahkan bobot sesuai persepsi penglihatan manusia. Metode *luminosity* dilakukan dengan menghitung nilai setiap elemen warna *R*, *G*, dan *B* dengan menambahkan bobot sesuai persepsi penglihatan manusia. Penglihatan manusia lebih sensitif terhadap warna hijau, oleh karena itu elemen *G* memiliki bobot yang paling tinggi dari elemen warna lainnya. Penghitungan untuk konversi citra warna menjadi citra keabuan dapat dirumuskan secara matematis seperti pada Persamaan 5.

$$Luminosity = \frac{R*299+G*587+B*114}{1000} \quad (5)$$

Pada penelitian ini metode *luminosity* digunakan untuk melakukan konversi citra warna menjadi citra keabuan karena sifat metode *luminosity* yang lebih dekat dengan persepsi penglihatan manusia. Setelah diperoleh citra keabuan kemudian dilakukan deteksi tepi menggunakan algoritma semut.

#### Inisialisasi Parameter

Tahapan awal dari penelitian yang dilakukan adalah inisialisasi parameter masukan. Parameter terdiri dari jumlah semut (*K*), jumlah step konstruksi (*L*), iterasi (*N*), *feromon evaporation rate* ( $\rho$ ), *feromon decay* ( $\varphi$ ), faktor bobot feromon ( $\alpha$ ), dan faktor pembobot informasi *heuristic* ( $\beta$ ). Matrik informasi *heuristic* berupa matrik intensitas warna setiap piksel. Matrik feromon awal diinisialisasi 1 atau 0. Jumlah semut pada optimasi algoritma semut berdasarkan nilai *gradient* deteksi tepi. Jumlah semut yang digunakan sebagai parameter masukan berdasarkan Persamaan 6.

$$K = \text{Jumlah gradient deteksi tepi} \geq \text{threshold} \quad (6)$$

#### Pembagian Gambar

Dalam penelitian ini, akan mengusulkan pendekatan baru untuk optimalisasi penyebaran semut dalam algoritma semut untuk deteksi tepi. Pada metode standar penyebaran semut dilakukan secara acak. Hal ini memungkinkan terjadi ketidakseimbangan penyebaran semut yang kemudian berakibat pada penemuan jalur yang dihasilkan. Metode

yang diusulkan dengan membagi gambar menjadi 16 area sama besar.

#### Penghitungan Nilai Gradient

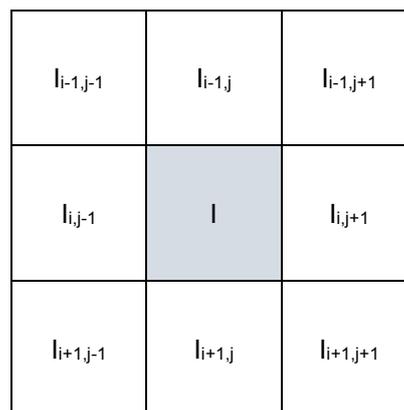
Pada proses ini bertujuan untuk mengetahui potensi nilai tepi dari area gambar. Operator *gradient* dapat direpresentasikan oleh dua buah kernel konvolusi (*Gx*) dan (*Gy*), yang masing-masing mendefinisikan operasi penghitungan *gradient* dalam arah sumbu *x* dan sumbu *y* yang saling tegak lurus. Pada penelitian ini akan digunakan operator sobel untuk menghitung nilai *gradient*.

#### Penyebaran Semut

Penyebaran semut dilakukan dengan menyesuaikan potensi tepi di tiap area berdasarkan nilai *gradient*. Area yang memiliki *gradient* besar akan memiliki jumlah penyebaran semut yang lebih banyak. Semut tidak disebar secara acak, akan tetapi ditempatkan di *gradient* tertinggi di tiap area. Cara ini diharapkan dapat digunakan untuk optimasi penemuan jalur dalam proses optimasi algoritma semut.

#### Pemilihan Piksel

Pada proses deteksi tepi, semut akan berpindah ke piksel tetangga. Piksel tetangga yang digunakan adalah 8 ketetanggaan. Gambar 3 menunjukkan hubungan ketetanggaan pada tiap piksel.



Gambar 3. Hubungan Piksel Ketetanggaan

Perpindahan piksel berdasarkan derajat eksplorasi (*q*). Algoritma ACS digunakan karena memiliki aturan *proportional pseudorandom* untuk mengoptimalkan gerakan semut. Aturan *proportional pseudorandom* menggunakan ambang batas yang ditetapkan ( $q_0$ ), yang nilainya antara 0 hingga 1.

Pada gerakan setiap semut yang dilakukan secara acak menghasilkan ( $q$ ) yang nilainya disebarkan antara 0 dan 1. Jika ( $q$ ) lebih besar dari ( $q_0$ ), maka aturan *proportional pseudorandom* digunakan untuk menentukan gerakan semut. Namun, jika ( $q$ ) lebih kecil dari ( $q_0$ ), semut harus bergerak sesuai dengan transisi yang memaksimalkan nilai *update* feromon ( $\tau_{ij}^\alpha$ ) dan informasi *heuristic* ( $\eta_{ij}^\beta$ ). Aturan *proportional pseudorandom* diberikan dalam Persamaan 7.

$$P_{i,j}^{(n)} = \frac{(\tau_{ij}^{n-1})^\alpha (\eta_{ij})^\beta}{\sum_{j \in \Omega_i} (\tau_{ij}^{n-1})^\alpha (\eta_{ij})^\beta} \text{ if } j \in \Omega_i \quad (7)$$

$\Omega_i$  adalah *node* ketetanggaan dari semut yang diberikan pada *node* ke- $i$ . Pergerakan semut pada piksel baris  $i$  dan kolom  $j$  ( $P_{ij}$ ) sama dengan perkalian feromon ( $\tau$ ) dan informasi *heuristic* ( $\eta$ ) yang dibagi dengan total perkalian lingkungan 8-ketetanggaan. Kedua feromon dan informasi *heuristic* dilengkapi dengan feromon bobot faktor ( $\alpha$ ) dan informasi *heuristic* bobot faktor ( $\beta$ ).

#### Update Feromon

Terdapat dua mekanisme pembaruan feromon ( $\tau$ ), yaitu pembaruan feromon lokal dan pembaruan feromon global. Setiap kali *step* konstruksi, matrik feromon lokal akan diperbarui karena feromon akan mengalami kerusakan (*feromon decay*). Rumus untuk feromon lokal dan pembaruan feromon global didefinisikan dalam Persamaan 8 dan Persamaan 9.

$$\tau_{ij} = (1 - \phi) \cdot \tau_{ij} + \phi \cdot \tau_0 \quad (8)$$

$$\tau_{ij} = (1 - \rho) \cdot \tau_{ij} + \rho \cdot \Delta\tau_{ij} \quad (9)$$

Pembaruan feromon lokal bertujuan untuk memverifikasi pencarian yang dilakukan oleh semut berikutnya selama iterasi (Dorigo dkk, 2006). Implementasinya dengan menggunakan feromon peluruhan ( $\phi$ ) dan feromon inis ( $\tau_0$ ) untuk menurunkan konsentrasi feromon di tepi yang dilalui. Oleh karena itu, semut berikutnya dapat menghasilkan solusi yang berbeda. Sedangkan penguapan feromon ( $\rho$ ) dan total deposit feromon ( $\Delta\tau_{ij}$ ) digunakan untuk menerapkan pembaruan feromon global.

#### Penentuan Tepi

Pada penentuan tepi, sebuah keputusan diambil pada setiap piksel untuk menentukan tepi atau bukan. Cara ini dilakukan dengan

mengimplementasikan sebuah *threshold* ( $T$ ) pada matrik feromon akhir ( $\tau^{(N)}$ ) (Tian dkk, 2008). Pada penelitian yang diusulkan ini nilai *threshold* ( $T$ ) diadaptasi berdasarkan metode otsu (Otsu, 1979). *Otsu thresholding* diimplementasikan untuk menentukan solusi terbaik berdasarkan jumlah feromon yang disimpan dalam setiap piksel. Pendekatan otsu dapat menentukan suatu variabel yang berguna membedakan antara dua atau lebih kelompok yang muncul secara alami. Penentuan tepi berdasarkan nilai matrik feromon akhir. Teknik *otsu threshold* akan mengurangi hasil citra keabuan menjadi citra *binary* dengan dua kemungkinan nilai untuk tiap piksel. Dari nilai matrik feromon akhir akan menunjukkan sebuah piksel dikatakan tepi atau bukan tepi.

#### Parameter Uji Coba

Penelitian ini menggunakan beberapa parameter masukan sebagai nilai penghitungan algoritma semut. Tabel 1 merupakan parameter uji coba yang digunakan pada metode yang diusulkan.

Tabel 1. Parameter Uji Coba

Parameter	Nilai
$K$ (jumlah semut)	$K = \text{Jumlah gradient} \geq \text{threshold}$
$L$ (jumlah konstruksi step)	10
$\alpha$ (faktor pembobot feromon)	0.5
$\beta$ (faktor pembobot informasi heuristic)	1
$\rho$ (feromon penguapan)	0.1
$\phi$ (feromon peluruhan)	0.05
$\tau_{init}$ (feromon inis)	0.0001
$\Omega$ (piksel ketetanggaan)	8
$N$ (jumlah iterasi)	4

## HASIL DAN PEMBAHASAN

Pengujian dilakukan untuk mengetahui tingkat keberhasilan sistem dalam menghasilkan deteksi tepi citra. Pada tahap *praproses* akan didapatkan citra keabuan. Hasil dari *praproses* digunakan sebagai data pada tahap deteksi tepi citra. Pengujian yang dilakukan dengan membandingkan hasil deteksi tepi citra antara optimasi algoritma semut dengan algoritma deteksi tepi konvensional. Citra masukan berupa citra RGB. Salah satu contoh pengambilan citra ditunjukkan pada Gambar 4.

Gambar 4 menunjukkan foto udara suatu wilayah. Foto udara ini akan digunakan sebagai citra masukan pada *praproses*. Citra masukan yang berupa citra RGB dikonversi menjadi citra abu-abu. Hasil konversi citra abu-abu ditunjukkan pada Gambar 5.



Gambar 4. Contoh Foto Udara

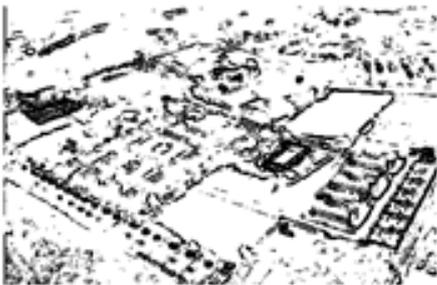


Gambar 5. Hasil Konversi Citra Abu-Abu

Dari data citra abu-abu kemudian dilakukan proses deteksi tepi. Uji coba deteksi tepi citra dilakukan dengan menggunakan optimasi algoritma semut dan dengan menggunakan deteksi tepi konvensional. Hasil citra deteksi tepi sobel dan algoritma semut seperti ditunjukkan pada Gambar 6 dan Gambar 7.



Gambar 6. Hasil Tepi Sobel



Gambar 7. Hasil Citra Algoritma Semut

Gambar 6 dan Gambar 7 menunjukkan optimasi algoritma semut menghasilkan citra tepi yang lebih detail dan memiliki garis tepi

yang lebih tebal. Hal ini menunjukkan hasil deteksi tepi foto (citra) menggunakan optimasi algoritma semut lebih baik dari hasil deteksi sobel.

## KESIMPULAN

Kinerja optimasi algoritma semut untuk proses deteksi tepi mampu menghasilkan fitur yang baik dibandingkan menggunakan metode deteksi tepi sobel. Kondisi ini ditunjukkan dari hasil optimasi algoritma semut yang menghasilkan citra dengan tepi yang lebih detail dan memiliki garis tepi yang lebih tebal.

## SARAN

Pergerakan untuk menentukan tepi piksel ketetanggaan secara acak perlu dilakukan perbaikan misalnya dengan menggunakan *gradient* arah dalam menentukan piksel berikutnya yang dipilih sebagai tepi.

## DAFTAR PUSTAKA

- Charu, G. dan Sunanda, G. 2013. *Edge Detection of an Image based on Ant Colony Optimization Technique*. International Journal of Science and Research. Vol. 2. Issue 6. 114-120.
- Dorigo, M., Birattari, M. dan Stutzle, T. 2006. *Ant Colony Optimization: Artificial Ants as a Computational Intelligence Technique*. IEEE Computational Intelligence Magazine. 28-39.
- Gonzalez, R. dan Woods, R. 1992. *Digital Image Processing*. Addison Wesley. hal 414-428. University of Pennsylvania Department of Computer and Information Science.
- Helen, A. 1987. *Edge Detection for Object Recognition in Aerial Photographs*.
- Iqbal, M. dan Sani, A. 2010. *Analisis Threshold Untuk Mengekstraksi Jalan dari Foto Udara*. Universitas Sumatera Utara.
- Liantoni, F., Kirana, K.C., dan Muliawati, T.H. 2014. *Adaptive Ant Colony Optimization based Gradient for Edge Detection*. Journal of Computer Science. Vol. 7. Issue 2. 78-84.
- Otsu, N. 1979. *A Threshold Selection Method from Gray Level Histograms*. IEEE Trans. Syst. Man. Cybern. Vol. 9. 62-66.
- Rahebi, J., Elmi, Z., Nia, A.F., dan Shayan, K. 2010. *Digital Image Processing using an Ant Colony Optimization based on Genetic Algorithm*. IEEE. 145-149.
- Tian, Jiang, Yu, Weiyu, Xie Shengli. 2008. *An Ant Colony Optimization Algorithm for*

*Image Edge Detection*. IEEE Congress on Evolutionary Computation. 751-756.  
Verma, O.P., Singhal, P., Garg, S., dan Deepthi, S.C. 2011. *Edge Detection using Adaptive Thresholding and Ant*

*Colony Optimization, Information and Communication Technologies (WICT)*. 2011 World Congress on IEEE. 113-118.