

IDENTIFIKASI PERPINDAHAN POSISI UAV BERBASIS PEMROSESAN CITRA DIGITAL

Ariel Valentino

Fakultas Teknik Elektro Universitas Surabaya
valentinoariel14@gmail.com

Abstrak – Penentuan posisi pada pesawat tanpa awak atau UAV sangat penting untuk menjalankan sistem kendali pesawat tersebut. Tetapi sensor GPS yang digunakan pada UAV tidak selalu dapat memberikan informasi posisi sensor tersebut. Alasan tersebut mendasari penelitian ini untuk dikembangkan topik terkait visual odometry pada daerah yang sensor GPS tidak dapat lakukan. Fokus dari penelitian ini adalah mengembangkan algoritma yang memiliki kompleksitas rendah. Algoritma tersebut akan dikaji dengan berdasarkan tingkat akurasi, kompleksitas notasi big-O, dan waktu eksekusi. Algoritma dikatakan berhasil dengan harapan mampu mengidentifikasi perpindahan posisi dengan akurasi lebih dari 80%. Adapun algoritma yang dikembangkan pada penelitian ini yaitu: (1)cross-correlation, (2)Harris-Stephen corner detection, dan (3)penyesuaian intensitas. Keandalan algoritma akan diuji dengan melakukan beberapa variasi pengujian. Berdasarkan pengujian yang telah dilakukan, algoritma terbaik untuk mengidentifikasi pergerakan dihasilkan oleh algoritma Harris-Stephen corner detection. Algoritma ini dapat mengidentifikasi pergerakan dengan tingkat akurasi di atas 80% untuk berbagai variasi gerakan, ketinggian, dan variasi jumlah objek pada gambar.

Kata kunci : *Visual odometry, cross-correlation, Harris-Stephens corner detection, penyesuaian intensitas*

Abstract – *The position estimation of unmanned aerial vehicle (UAV) is very important in order to control it autonomously. But, the GPS sensor used on UAVs does not always provide information about its position. Because of that, this final project aims to develop visual odometry in unreachable GPS signal areas. The focus of this research is to develop algorithm that have a low complexity. The algorithm will be assessed based on accuracy level, big-O notation, and execution time. The algorithm will identify as successful if it is able to identify an accuracy in position displacement of more than 80%. The algorithm that is developed in the final project i.e. (1)cross-correlation, (2)Harris-Stephen corner detection, and (3)intensity matching. The reliability of the algorithm will be tested by conducting several test variations. Based on the algorithm implemented in the testing, the best algorithm to identify movement change is Harris-Stephen corner detection algorithm. That algorithm can identify movement with an accuracy of above 80% for a wide variety of movements, heights, and amounts of objects in the image.*

Keywords: *Visual odometry, cross-correlation, Harris-Stephens corner detection, intensity macthcng*

PENDAHULUAN

UAV atau *Unmanned Aerial Vehicle* merupakan suatu teknologi yang sedang populer dan berkembang pesat. UAV alias pesawat tanpa awak ini memiliki fitur penting yaitu UAV dapat menentukan lokasi dan mengetahui pergerakan [1]. Dengan fitur tersebut suatu UAV dapat bergerak ke suatu lokasi yang ditentukan tanpa harus ada seorang pilot yang mengendalikannya UAV tersebut. Metode autonomous UAV ini dikembangkan dengan menggunakan beberapa sensor yang terpasang pada UAV, salah satunya adalah sensor GPS. GPS (*Global Positioning System*) merupakan sensor yang dapat memberikan informasi berupa koordinat posisi absolut dari lokasi sensor tersebut berada. Selanjutnya data sensor akan diolah sehingga suatu UAV dapat bergerak dan berpindah posisi sesuai dengan yang diperintahkan pada koordinat yang ingin dituju terutama di area terbuka[1].

Namun dalam kondisi tertentu sistem navigasi GPS tidak lagi dapat diandalkan[1]. Prinsipnya sensor GPS akan menerima sinyal dari beberapa satelit di sekitar bumi. Sinyal tersebut akan menghasilkan daerah *intersect*/gabungan yang terbentuk dari daerah jangkauan antar satelit yang dapat menerima sinyal GPS, yang selanjutnya diolah agar mendapatkan titik koordinat dari sensor GPS tersebut. Dengan kata lain sensor GPS sangat bergantung pada sinyal yang diperoleh. Namun pada kenyataannya sinyal dari satelit ini akan susah menembus awan ataupun gedung, sehingga dapat memberikan informasi yang tidak valid bahkan tidak dapat lagi menentukan lokasi. Selain masalah tersebut, tingkat keakurasian sensor GPS ini juga bergantung pada beberapa faktor lain misalnya faktor alat dan faktor kalkulasi.

Oleh sebab itu, penggunaan sensor GPS pada ruangan *indoor* atau yang melintasi gedung tidak disarankan. Solusi lain yang ingin dikembangkan adalah dengan memanfaatkan kamera dari UAV. Kamera UAV umumnya digunakan untuk merekam objek yang berada di bawah UAV. Hasil rekaman tersebut dapat dimanfaatkan untuk mengetahui pergerakan suatu UAV. Metode dengan disebut juga dengan *Visual Odometry* (VO).

Topik terkait penentuan posisi UAV dengan menggunakan rekaman kamera sudah pernah dipublikasikan N.D. Pah dan Henry H. [2] dengan judul: "*The Development of Image-based Algorithm to Identify Altitude Change of a*

Quadcopter”, pada tahun 2015, dan penelitian yang kedua N.D. Pah [3] dengan judul "*Image-based Distance Change Identification by Segment Correlation*", pada tahun 2015. Kedua penelitian tersebut mengembangkan algoritma VO untuk mengetahui perubahan ketinggian suatu UAV. Hasil kedua penelitian tersebut menghasilkan keluaran yang cukup baik yaitu dengan keakurasian mencapai 96% dan 97%.

Penelitian terkait penentuan pergerakan suatu UAV dengan berbasis kamera juga pernah dilakukan oleh Chaira T and et. al [4] dengan judul, "*Low Computational-complexity Algorithms for Vision-aided Inertial Navigation of Micro Aerial Vehicles*", pada tahun 2014. Pada penelitian ini mengembangkan algoritma VO yang memiliki kompleksitas yang rendah untuk mengidentifikasi 6-DOF dari pergerakan suatu UAV dengan menggunakan metode *2-point algorithm* dan *3-point algorithm*. Penelitian tersebut difokuskan untuk mencari 6 pergerakan dari suatu MAV/*Micro Aerial Vehicles* diantaranya: *pitch*, *roll*, *yaw*, *heave*, *sway*, dan *surge*.

Dari ketiga penelitian di atas memiliki kesamaan yaitu mengembangkan algoritma yang memiliki kompleksitas yang rendah untuk mengidentifikasi posisi UAV. Hal tersebut menjadi hal yang harus dipertimbangkan dalam mendesain algoritma VO. Alasannya adalah identifikasi perpindahan posisi sering kali dilakukan secara *real time*, sehingga algoritma VO yang digunakan harus dapat diproses dengan cepat.

Dari latar belakang tersebut, penulis ingin mengembangkan lebih lanjut terkait topik tentang VO. Penelitian yang akan dibuat dikembangkan dengan berdasarkan perubahan objek yang terekam oleh kamera untuk dapat mengidentifikasi perubahan posisi UAV. Topik ini yang akan dikembangkan juga akan mempertimbangkan kompleksitas algoritma yang akan dibuat. Hasil keluaran dari penelitian ini adalah kajian terkait algoritma yang akan dikembangkan.

METODE PENELITIAN

Pada penelitian ini, dirancang algoritma *visual odometry* (VO). Algoritma yang dikembangkan terdiri dari 3 yaitu: algoritma *cross-correlation*, algoritma Harris-Stephens *corner detection*, dan algoritma penyesuaian intensitas. Hasil penelitian ini merupakan hasil kajian keandalan dari ketiga algoritma tersebut. Target dari algoritma yang dikembangkan yaitu mencapai lebih dari 80% tingkat akurasi identifikasi posisi.

Langkah awal dalam melakukan penelitian ini adalah merancang suatu *platform* untuk mengumpulkan data pergerakan. *Platform* ini tersusun dengan menggunakan dua buah *slider*. *Platform* didesain agar dapat kamera dapat merekam gerakan pada sumbu x dan sumbu y. Gerakan tersebut dihasilkan oleh motor yang kemudian gerakan direkam dengan menggunakan sensor *rotary encoder* KY-040. Pada tahap selanjutnya data gerakan tersebut dilakukan pengujian akurasi. Hasil pengujian menunjukkan bahwa data sensor *rotary encoder* memiliki *error* sebesar 0,823185% untuk sumbu x dan 0,87465% untuk sumbu y. Dari pengujian tersebut menunjukkan data pergerakan memiliki akurasi yang tinggi, sehingga data dapat digunakan sebagai data referensi pada algoritma VO.

Tahap selanjutnya adalah perancangan desain ketiga algoritma VO. Algoritma yang pertama adalah algoritma *cross-correlation*. Secara matematis, algoritma ini ditulis dengan persamaan 1:

$$(f * g)[n] = \sum_{m=-\infty}^{\infty} f[m] * g[m + (n - D)] \quad (1)$$

dimana

f = *raw* data suatu sinyal referensi f

g = *raw* data suatu sinyal pembanding g

n = nomor pergeseran yang dilakukan

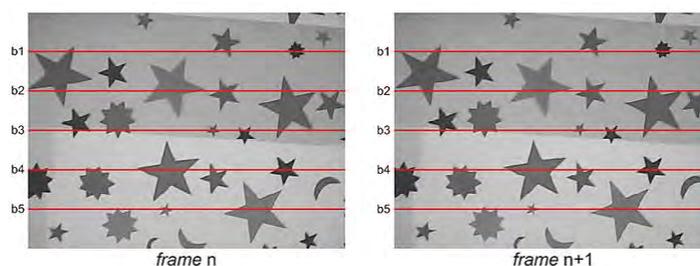
m = nomor urut data

D = panjang data pada sinyal f atau g

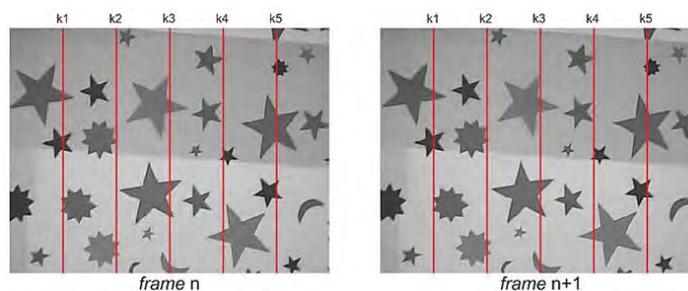
Pada prinsipnya persamaan 1 digunakan untuk mencari jumlahan dari hasil *dot product* sinyal f terhadap sinyal g. Namun secara khusus algoritma ini akan melakukan pergeseran pada sinyal f dan akan mencari kembali hasil *dot product* sinyal tersebut dengan sinyal g. Metode ini dapat disebut juga dengan *sliding dot*

product. Tujuan dari *cross-correlation* adalah untuk mencari korelasi sinyal *f* terhadap sinyal *g*.

Algoritma VO yang pertama ini merupakan hasil implementasi persamaan 1. Langkah – langkah untuk mengimplementasikan metode tersebut diawali dengan menuntukan sinyal yang akan dicari korelasinya. Sinyal yang akan dibandingkan merupakan sinyal baris dan kolom antar dua *frame*. Sinyal baris dan kolom yang terpilih ditunjukkan dengan gambar 1 dan 2.



Gambar 1: Sinyal baris yang terpilih



Gambar 2: Sinyal kolom yang terpilih

Algoritma ini menggunakan 5 sinyal referensi baris dan 5 sinyal referensi kolom pada *frame n*. Kedua sinyal tersebut akan dicari nilai simpangan datanya sesuai dengan persamaan 2 dan 3. Tujuannya adalah agar nilai yang dihasilkan memiliki nilai bertanda negatif.

$$b(i) = b(i) - \text{mean}(b) \quad (2)$$

$$k(i) = k(i) - \text{mean}(k) \quad (3)$$

dimana

$b(i)$ = data sinyal baris ke - i

$k(i)$ = data sinyal baris ke - i

$\text{mean}(b)$ = rata -rata data sinyal b

$\text{mean}(k)$ = rata -rata data sinyal k

Setelah mencari nilai simpangan pada setiap sinyal referensi, nilai simpangan juga dicari pada setiap sinyal yang akan dibandingkan pada *frame* $n+1$. Proses selanjutnya adalah mencari korelasi sinyal yang dilakukan sebanyak 2 kali. Korelasi pertama antara simpangan setiap sinyal baris dan kolom pada *frame* n terhadap dirinya sendiri, sedangkan korelasi kedua antara simpangan setiap sinyal baris dan kolom pada *frame* n dengan simpangan setiap sinyal baris dan kolom di lokasi yang sama pada *frame* $n+1$.

Kedua hasil korelasi belum dapat menyatakan perpindahan posisi yang terjadi antar kedua *frame*. Untuk menyatakan perpindahan posisi pada sumbu x , maka perpindahan dinyatakan dengan mengurangkan hasil korelasi sinyal baris pada *frame* n terhadap *frame* $n+1$ dengan hasil korelasi sinyal baris pada *frame* n terhadap dirinya sendiri. Sedangkan untuk menyatakan perpindahan posisi pada sumbu y , maka perpindahan dinyatakan dengan cara yang serupa yaitu mengurangkan hasil korelasi sinyal kolom pada *frame* n terhadap *frame* $n+1$ dengan hasil korelasi sinyal kolom pada *frame* n terhadap dirinya sendiri.

Perancangan algoritma yang kedua yaitu algoritma Harris-Stephens *corner detection*. Secara matematis, algoritma tersebut ditulis seperti persamaan 4:

$$E(\mathbf{u}, \mathbf{v}) = \sum_{x,y} w(x, y) [(I(x + \mathbf{u}, y + \mathbf{v}) - I(x, y))]^2 \quad (4)$$

dimana

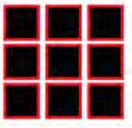
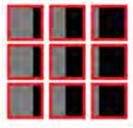
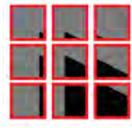
$E(\mathbf{u}, \mathbf{v})$ = variasi intensitas pada window yang tergeser \mathbf{u}, \mathbf{v}

$w(x, y)$ = *window* biner / gaussian

$I(x + \mathbf{u}, y + \mathbf{v})$ =Intensitas yang bergeser sejauh $x + \mathbf{u}, y + \mathbf{v}$

$I(x, y)$ =Intensitas yang awal pada posisi (x,y)

Algoritma ini digunakan untuk mencari fitur khusus pada gambar. Fitur yang dimaksud adalah titik sudut suatu objek. Untuk menentukan titik sudut atau bukan dilakukan dengan meninjau perbedaan intensitas pada suatu *window*. *Window* tersebut adalah suatu jendela kecil berukuran tertentu yang digunakan untuk mengidentifikasi objek. Untuk memahami suatu objek disebut sebagai titik sudut atau bukan dijelaskan melalui gambar 3.

| | <i>flat</i> | <i>edge</i> | <i>corner</i> |
|---------------------------|---|---|--|
| Objek |  |  |  |
| Window yang sudah digeser |  |  |  |

Gambar 3: Cara kerja pencari titik sudut[5]

Pada gambar 3, dapat terlihat bahwa algoritma Harris-Stephens *corner detection* membagi objek menjadi 3 kategori yaitu *flat*, *edge*, dan *corner*. Suatu objek disebut sebagai *flat* jika pada setiap *window* yang digeser di sekitar objek menunjukkan perbedaan intensitas yang kecil. Sedangkan objek disebut sebagai *edge* jika pada setiap *window* yang digeser disekitar objek hanya menunjukkan perbedaan yang signifikan pada pergeseran yang searah. Selain itu, objek disebut sebagai *corner* jika pada setiap *window* yang digeser di sekitar objek menunjukkan perbedaan yang signifikan pada setiap arah.

Selanjutnya adalah merancang algoritma VO yang kedua dengan mengimplementasikan persamaan 4. Pada MATLAB algoritma ini tersedia dalam fungsi **corner**, fungsi tersebut hanya terbatas untuk 200 titik sudut saja. Contoh hasil dari pencarian titik sudut yang terdeteksi pada *frame* n disajikan pada gambar 4. Titik sudut yang berhasil teridentifikasi ditandai dengan tanda "*" warna merah.



Gambar 4: Hasil deteksi titik sudut pada gambar

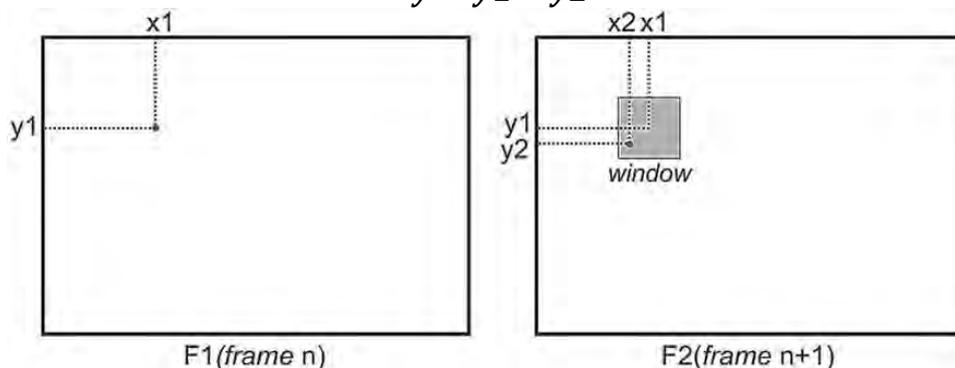
Data titik sudut yang teridentifikasi merupakan data koordinat titik sudut tersebut. Dari semua titik sudut yang berhasil teridentifikasi hanya akan dipilih sejumlah titik sudut sebagai data yang akan dibandingkan pada *frame* berikutnya. Banyaknya data titik sudut yang dipilih memenuhi syarat sebagai berikut: 1 data jika titik sudut terdeteksi kurang dari 3 titik, 3 data jika titik sudut yang terdeteksi

lebih dari 2 titik dan kurang dari 5 titik, dan 5 data jika titik sudut yang terdeteksi lebih dari 4 titik.

Data yang dipilih tersebut akan digunakan sebagai titik tengah untuk membentuk suatu *window* pada *frame* berikutnya. Gambar 5 menjelaskan cara kerja dari algoritma Harris-Stephen *corner detection* yang dikembangkan untuk mendeteksi pergerakan. Data koordinat titik sudut yang terpilih ditunjukkan dengan nama x_1, y_1 , sedangkan x_2, y_2 merupakan contoh data koordinat titik sudut yang teridentifikasi pada *frame* berikutnya. Dengan menggunakan persamaan 5 dan 6, kedua data koordinat tersebut akan diketahui perpindahan posisi yang telah terjadi.

$$x = x_1 - x_2 \tag{5}$$

$$y = y_1 - y_2 \tag{6}$$



Gambar 5: Ilustrasi dari metode yang digunakan

Perancangan algoritma yang ketiga merupakan hasil adopsi dari algoritma Harris-Stephen *corner detection*. Pada prinsipnya kedua algoritma ini memanfaatkan perbedaan intensitas pada suatu *window*. Namun perbedaannya adalah metode ini tidak mementingkan suatu titik sudut. Metode ini dijelaskan secara matematis pada persamaan 7.

$$E(u, v) = \sum_{x,y} w(x, y) [abs(Ib(x + u, y + v) - Ia(x, y))] \tag{7}$$

dimana

$E(u, v)$ = variasi intensitas pada *window* yang tergeser u, v

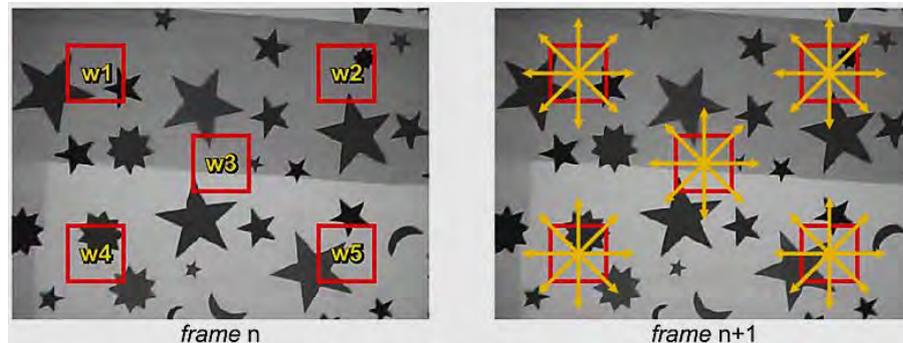
$w(x, y)$ = *window* biner

$Ib(x + u, y + v)$ =Intensitas yang bergeser sejauh $x + u, y + v$ frame n+1

$Ia(x, y)$ =Intensitas yang awal pada posisi (x,y) frame n

Persamaan 7 menjelaskan cara kerja algoritma ini untuk mencari suatu perpindahan. Persamaan tersebut membutuhkan *window* pada *frame* n, lalu *window* ini akan dijadikan *window* pembanding untuk *window* yang akan digeser pada

frame n+1. Algoritma ini akan menggunakan 5 buah *window* untuk mendeteksi perpindahan yang terjadi. Pada Gambar 7 menjelaskan tentang letak dan pergeseran yang dilakukan pada *frame* n dan *frame* n+1.



Gambar 6. Penentuan letak dan pergeseran *window* pada dua *frame*

Pada *frame* n+1, *window* akan digeser sejauh 6 (enam) kali ke kiri, ke kanan, ke atas, dan ke bawah. Setiap kali *window* tersebut digeser dilakukan juga perhitungan seperti yang dijelaskan pada persamaan 7. Dari perhitungan tersebut akan didapatkan variasi intensitas $E(u, v)$ pada daerah pergeseran yang dilakukan pada *frame* n+1. Suatu perpindahan yang terjadi berarti ditunjukkan oleh nilai variasi intensitas $E(u, v)$ paling minimum.

Namun nilai dari pencarian nilai minimum $E(u, v)$ merupakan nomor urut pergeseran yang telah dilakukan. Nomor tersebut tidak mewakili perpindahan yang terjadi. Untuk mengetahui perpindahan yang terjadi perlu dikonversi terlebih dahulu menjadi nilai perpindahan x dan y yang dijelaskan pada Gambar 7. Proses konversi nilai tersebut dapat dilakukan dengan memanfaatkan prinsip hasil bagi dan sisa hasil bagi yang memenuhi syarat algoritma berikut:

```

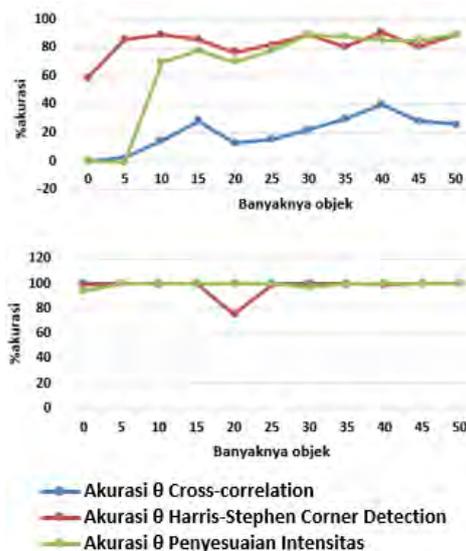
m=nilai minimum( $E(u, v)$ );
a=m/13;
if ((a=1)or(a=2)or(a=3)or(a=4)or(a=5)or(a=6)or(a=7)
or(a=8)or(a=9)or(a=10)or(a=11)or(a=12)or(a=13))
    {
        y= (m div 13)-7;
        x=6;
    } else
    {
        y=((m div 13)+1)-7;
        x=(m mod 13)-7;
    }
end

```


Langkah awal dalam pengujian yang telah dilakukan adalah menguji alat yang digunakan untuk merekam data pergerakan. Data hasil pengujian ini sangat penting yang berguna sebagai data pembanding dari data gerakan yang akan diidentifikasi oleh algoritma. Hasil pengujian data sensor *rotary encoder* KY-040 menunjukkan *error* sebesar 3,741% yang dicapai setelah menambahkan rangkaian *hardware debouncer*. Pengujian terhadap alat lainnya adalah pengujian jarak tempuh sensor *rotary encoder* KY-040 yang menunjukkan *error* kurang dari 1% untuk sumbu x dan y. Dari hasil pengujian alat ini dapat disimpulkan bahwa data gerakan yang berhasil teridentifikasi oleh sensor *rotary encoder* KY-040 memiliki akurasi yang tinggi.

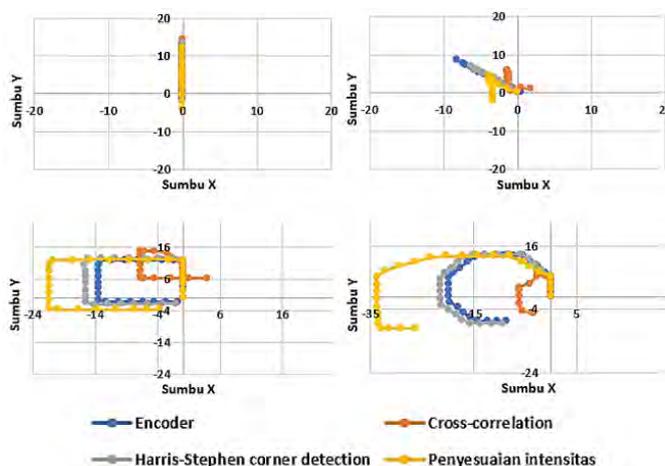
Pengujian selanjutnya adalah pengujian keandalan algoritma yang dilakukan terhadap variasi objek, gerakan, dan ketinggian. Pengujian algoritma terhadap variasi objek bertujuan untuk mengetahui keandalan akurasi dari variasi banyaknya objek yang direkam. Pengujian algoritma terhadap variasi gerakan bertujuan untuk mengetahui keandalan algoritma untuk mengidentifikasi perubahan gerakan ditinjau dari arah geraknya. Sedangkan pengujian algoritma terhadap ketinggian bertujuan untuk mengetahui keandalan akurasi terhadap perubahan ketinggian antara kamera dengan bidang objek.

Data akurasi hasil pengujian ketiga algoritma terhadap variasi objek disajikan pada gambar 9. Data tersebut merupakan hasil perbandingan dari data kecepatan dan arah yang teridentifikasi oleh ketiga algoritma dengan data hasil bacaan *sensor rotary encoder*. Dari pengujian ini dapat disimpulkan bahwa akurasi algoritma Harris-Stephen *corner detection* dan penyesuaian intensitas memiliki akurasi lebih dari 80% untuk hasil *capture* lebih dari 30 objek, sedangkan algoritma *cross-correlation* hanya memiliki tingkat akurasi kurang dari 39,83% pada setiap variasi objek.



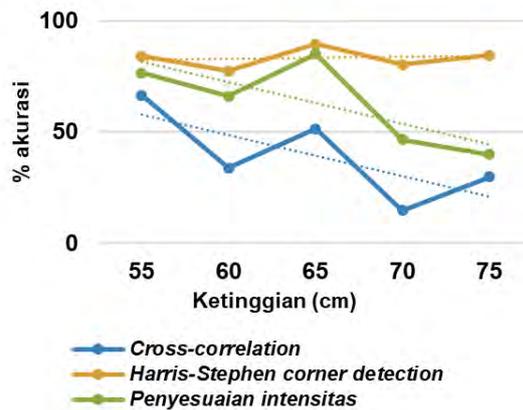
Gambar 9. Grafik perbandingan akurasi kecepatan dan arah terhadap variasi objek

Data posisi dari keempat gerakan disajikan pada sistem koordinat kartesius pada gambar 10. Data tersebut diperoleh dari hasil identifikasi posisi pada sumbu x dan y oleh ketiga algoritma, dan disertakan juga data perbandingan dari hasil bacaan sensor *rotary encoder*. Dari hasil pengujian ini dapat disimpulkan bahwa algoritma Harris-Stephen *corner detection* menunjukkan hasil yang baik, yang ditunjukkan dengan hasil simpangan kurang dari 2,509 cm pada keempat gerakan. Sedangkan algoritma *cross-correlation* dan penyesuaian intensitas menunjukkan hasil simpangan yang jauh dari data perbandingan seiring dengan bertambah kompleksnya pergerakan yang diujikan. Hasil simpangan yang terjadi mencapai 15,84 cm untuk algoritma *cross-correlation* dan 20,6 cm untuk algoritma penyesuaian intensitas.



Gambar 10. Plot hasil posisi ketiga algoritma pada keempat gerakan

Data akurasi algoritma terhadap variasi ketinggian disajikan dalam grafik pada gambar 11. Data tersebut merupakan data akurasi hasil identifikasi algoritma terhadap perubahan ketinggian. Dari hasil pengujian ini dapat disimpulkan bahwa algoritma Harris-Stephen *corner detection* tidak memiliki pengaruh signifikan terhadap variasi ketinggian. Sedangkan kedua algoritma lainnya, menunjukkan hasil akurasi yang cenderung menurun ketika ketinggian yang diujikan bertambah.



Gambar 11. Plot hasil akurasi ketiga algoritma variasi ketinggian

Pengujian yang terakhir adalah pengujian performa algoritma untuk mengidentifikasi pergerakan. Data hasil pengujian ini merupakan data waktu eksekusi yang dibutuhkan suatu algoritma untuk mengidentifikasi pergerakan yang disajikan pada tabel 1. Dari hasil pengujian ini terlihat bahwa seiring dengan bertambahnya *frame* maka lama waktu eksekusi algoritma akan semakin bertambah. Data tersebut akan dianalisis untuk melihat waktu yang dibutuhkan oleh algoritma untuk mengidentifikasi pergerakan per satu identifikasi pergerakan. Untuk mengidentifikasi satu pergerakan dibutuhkan 2 *frame* yaitu *frame* n dan *frame* n+1, maka waktu eksekusi per satu identifikasi pergerakan diperoleh dari total lama waktu dibagi dengan banyaknya frame dikali 2 (dua).

Tabel 1. Hasil pengujian waktu eksekusi pada keempat gerakan

| Gerakan | Banyaknya <i>frame</i> | Total waktu eksekusi algoritma | | |
|---------|------------------------|--------------------------------|--|------------------------|
| | | <i>cross-correlation</i> | Harris-Stephen <i>corner detection</i> | penyesuaian intensitas |
| 1 | 446 | 18,164 detik | 44,217 detik | 45,842 detik |
| 2 | 468 | 18,905 detik | 46,657 detik | 48,679 detik |
| 3 | 748 | 28,691 detik | 69,085 detik | 74,218 detik |
| 4 | 889 | 30,974 detik | 81,772 detik | 90,805 detik |

Hasil pengujian performa algoritma untuk mengidentifikasi pergerakan yang tercepat dihasilkan oleh algoritma *cross-correlation* dengan 0,07716 detik per satu identifikasi pergerakan. Sedangkan waktu eksekusi untuk kedua algoritma lainnya menunjukkan hasil lebih dari 2 (dua) kali waktu eksekusi algoritma *cross-correlation*. Waktu eksekusi algoritma Harris-Stephen *corner detection* adalah sebesar 0,19159 detik per satu identifikasi pergerakan, dan waktu paling lambat terjadi dengan menggunakan algoritma penyesuaian intensitas yaitu sebesar 0,20408 detik per satu identifikasi pergerakan.

KESIMPULAN DAN SARAN

Pada penelitian ini dikembangkan 3 (tiga) algoritma *visual odometry* untuk mengidentifikasi pergerakan UAV dalam arah dan kecepatan. Ketiga algoritma tersebut antara lain algoritma *cross-correlation*, Harris-Stephen *corner detection*, dan penyesuaian intensitas. Hasil identifikasi ketiga algoritma diharapkan mampu mencapai tingkat akurasi lebih dari 80%. Dari pengujian yang telah dilakukan dapat disimpulkan bahwa algoritma Harris-Stephen *corner detection* merupakan algoritma terbaik pada penelitian ini. Algoritma tersebut mampu mengidentifikasi pergerakan dengan tingkat akurasi lebih dari 80% untuk berbagai variasi gerakan, ketinggian, dan variasi jumlah objek pada gambar.

Saran pengembangan terkait algoritma Harris-Stephen *corner detection* yang digunakan untuk mengidentifikasi pergerakan pada penelitian mendatang, penulis menyarankan untuk membatasi pencarian titik sudut pada *frame* n. Metode pencarian titik sudut pada *frame* n disarankan untuk memilih hanya pada titik sudut yang dianggap perlu saja. Pembatasan pencarian titik sudut pada *frame* n dapat mengurangi beban perhitungan algoritma, sehingga proses eksekusi algoritma dapat berjalan lebih cepat.

DAFTAR PUSTAKA

- [1] Tomas K. *and et. al.*, “A Simple Visual Navigation system for an UAV”, in: 9th IEEE International Multi-Conference on System, Signal, and Devices (SSD), 2012.
- [2] N.D. Pah dan Henry H., “The Development of Image-based Algorithm to Identify Altitude Change of a Quadcopter”, in: the 7th International Conference on Information Technology and Electrical Engineering (ICITEE), 2015.
- [3] N. D. Pah, “Image-based Distance Change Identification by Segment Correlation”, in: *Proceedings of Second International Conference on Electrical System, Technology and Information (ICESTI)*, 2015
- [4] Chaira T. *and et. al.*, “Low Computational-complexity Algorithms for Vision-aided Inertial Navigation of Micro Aerial Vehicles”, *Readings in Sciencedirect: Robotics and Autonomous Systems*, vol 69, page 80-97, 2015.
- [5] Yung-Yu C., *lecture topic: Digital Visual Effects, Readings in Spring 2006*