

Purwarupa Sistem Pendeteksi Garis Landasan Pacu pada Pesawat Terbang

Muhammad Shahid Ardi*¹, Agus Harjoko², Raden Sumiharto³

¹Program studi Elektronika dan Instrumentasi FMIPA UGM

^{2,3}Lab. Elektronika dan Instrumentasi, JIKE, FMIPA UGM, Yogyakarta

e-mail: *¹shahid_ardi@mail.ugm.ac.id, ²aharjoko@ugm.ac.id, ³r_sumiharto@ugm.ac.id

Abstrak

Teknologi UAV (*unmanned aerial vehicle*) pada masa sekarang mengalami perkembangan yang pesat. Salah satu bentuk perkembangannya adalah sistem autopilot. Sistem ini bekerja secara otomatis untuk mengendalikan proses terbang pesawat baik dari tinggal landas sampai proses landing. Akan tetapi dalam proses mendarat dibutuhkan sistem yang dapat memandu pesawat untuk dapat mendarat pada jalur landasan dengan baik, salah satunya dengan teknologi citra digital.

Dalam penelitian ini masalah tersebut coba diselesaikan dengan membuat purwarupa sistem identifikasi garis untuk proses pendaratan UAV. Pembuatan sistem menggunakan perangkat Mini PC Zotac Zbox nano AD10 dan OpenCV sebagai pustaka pengolah citra dengan pertimbangan sistem ini dapat dikembangkan lebih lanjut menjadi bagian dari sistem autopilot yang siap digunakan..

Kata kunci : UAV, autopilot.

Abstract

UAV's technology at present is experiencing rapid growth. One form of its development is the autopilot system. This system works automatically to control the aircraft flying process from take-off to landing. However, in the process of take-off or landing, a system is needed to guide the aircraft. Such a system could be developed based a digital image processing technology.

In this research, a prototype of a system for the identification of the line for UAV landing is developed. A MiniPC Zotac ZBOX nano AD10 is selected as the hardware platform. The OpenCV image processing library is used to implement the prototype. The prototype is tested using the recorded video. The prototype works well for speed up to 30 km/hour.

Keywords UAV, autopilot

1. PENDAHULUAN

Pesawat tanpa awak atau yang biasa disebut dengan UAV (*unmanned aerial vehicle*) merupakan perkembangan dari pesawat *aeromodelling* yang memiliki fungsi atau tujuan khusus. UAV dapat berfungsi untuk pengawasan dari udara terhadap suatu daerah tertentu. UAV dapat pula dipergunakan untuk melakukan foto udara dan melakukan misi pengintaian atau mata-mata untuk keperluan militer.

Ada berbagai bentuk, ukuran, konfigurasi, dan karakteristik UAV. Secara historis UAV *drone* pada awalnya dibuat dengan sangat sederhana dan akhirnya dikontrol secara otonom (*autonomous*) semakin banyak produksi dan digunakan. Saat ini, UAV telah digunakan untuk

melakukan misi inteljen, pemantauan (*surveillance*), pengintaian (*reconnaissance*) serta misi serangan (*attack*). Banyak dilaporkan bahwa UAV telah berhasil dengan tingkat akurasi tinggi dalam melakukan misi intelijen, pemantauan, pengintaian dan serangan dengan menggunakan roket, rudal dan bom [1].

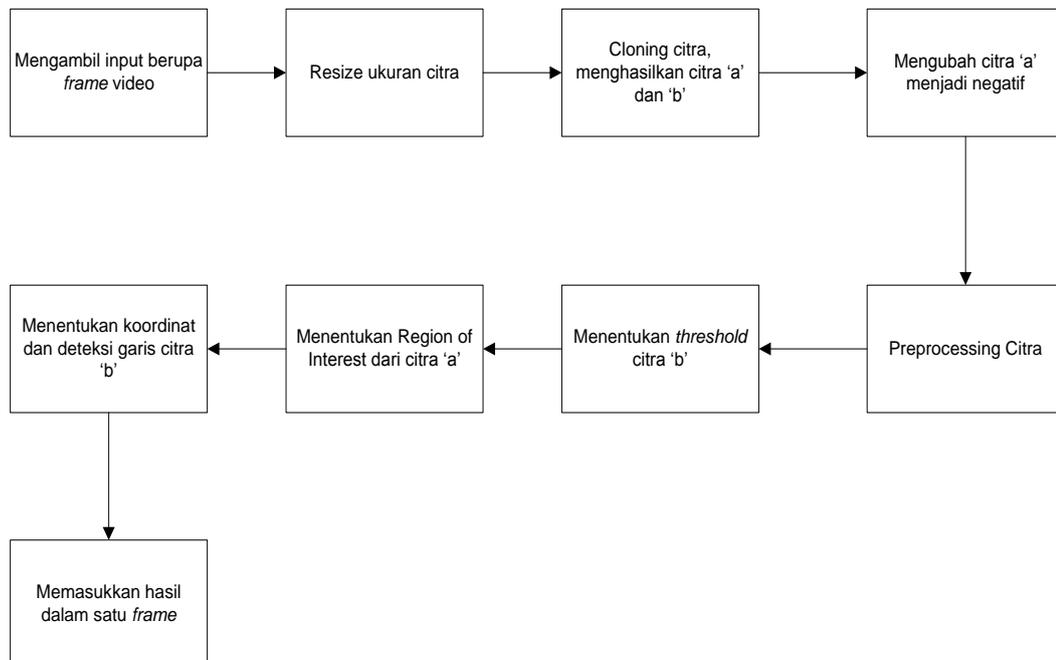
Proses pendaratan atau landing pada suatu UAV merupakan hal yang kompleks karena membutuhkan energi potensial dan kinetic yang cukup banyak dari pesawat terbang disamping adanya kendala yang dinamis seperti perubahan angin yang mendadak, berat muatan, serta tinggi dan kecepatan disetiap arah [2].

Pada pendaratan otomatis UAV diperlukan sebuah sistem sedemikian rupa yang mampu mengatur *aileron*, *rudder*, dan *elevator* serta tenaga dari motor sehingga mampu menempatkan UAV pada posisi yang sesuai untuk mendarat. Pada bagian sistem pendaratan otomatis banyak metode yang digunakan dalam membuat pesawat dapat mendarat secara otomatis. Salah satunya adalah dengan menggunakan citra dari landasan yang kemudian diproses untuk diambil bagian yang penting dari hasil pengolahan tersebut, untuk selanjutnya didapatkan data yang mempengaruhi pergerakan pesawat. Citra yang didapat merupakan hasil dari pengambilan oleh kamera yang dipasang pada pesawat untuk selanjutnya diolah menggunakan perangkat pengolah citra baik computer atau microprocessor yang khusus.

2. RANCANGAN PROTOTIPE

Sistem yang dibangun dalam penelitian ini merupakan sistem yang bekerja sebagai pengolah citra digital. Sistem ini memiliki peran sebagai pendeteksi garis pada landasan pacu yang tertangkap oleh kamera. Garis yang tertangkap selanjutnya diolah menggunakan pustaka pengolah citra, dalam hal ini OpenCV 2.3.1.

Proses pengolahan citra video dilakukan dalam setiap *frame* yang menghasilkan keluaran berupa koordinat sumbu x dari garis yang terdeteksi. Diagram proses pengolahan ditunjukkan pada Gambar 1.



Gambar 1. Blok diagram proses

Sesuai dengan blok diagram pada Gambar 1 proses pengolahan citra dilakukan secara *sequencial*. Berikut dijelaskan proses dari kerja sistem tersebut.

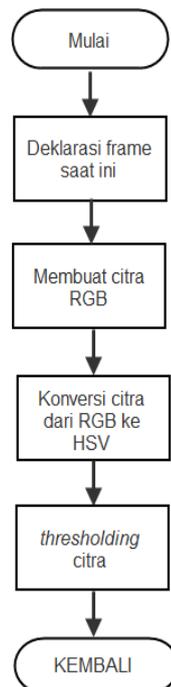
- *Input frame*
Masukan sistem berupa video yang terdiri dari banyak *frame*.
- *Resize* ukuran
Proses yang kedua adalah melakukan pengecilan ukuran atau *resize* dari setiap *frame* video yang masuk. Proses ini diperlukan untuk mendapatkan resolusi citra yang sesuai dengan literatur yang sudah ada dimana dalam literature tersebut nilai resolusi dari citra yang digunakan adalah 640x480 piksel sehingga dalam pengolahannya dapat maksimal.
- *Cloning citra*
Proses yang ketiga adalah penggandaan citra atau disebut *cloning* sehingga didapatkan dua buah citra yang sama bentuk dan ukurannya (selanjutnya disebut citra 'a' dan 'b').
- *Invert Citra*
Proses inversi dilakukan terhadap citra 'a' hasil penggandaan dari proses sebelumnya untuk mengubah citra tersebut menjadi citra negatif, sehingga akan tampak jelas perubahan warna yang terjadi dari garis yang berwarna putih pada citra RGB menjadi warna hitam pada citra negatif.
- *Preprocessing citra*
Proses ini berfungsi untuk mengubah citra RGB 'a' menjadi *IplImage* sekaligus mendeklarasikan pointer untuk citra. Kemudian citra RGB diubah menjadi citra HSV agar garis lebih mudah dideteksi.
- *Thresholding*
Proses dilakukan terhadap citra 'b', sehingga dihasilkan citra hasil *threshold* dengan batas yang telah ditentukan.
- *Menentukan Region of Interest*
Proses ini berfungsi untuk menentukan area seleksi dari citra yang akan di deteksi. Proses dilakukan terhadap citra 'a' dengan membuat area kotak berwarna merah, area didalam kotak tersebut yang siap untuk diseleksi.
- *Menentukan koordinat dan deteksi garis citra*
Pada proses ini dilakukan penentuan koordinat tengah dari citra hasil *thresholding*.
- *Memasukkan hasil dalam satu frame*
Proses ini merupakan proses terakhir, citra hasil deteksi dan dari proses ROI digabungkan kedalam satu *frame*. Dari *frame* tersebut diperoleh gambar hasil deteksi dan nilai dari koordinat tengah garis.

2.1 Rancangan perangkat lunak

Rancangan perangkat lunak dalam sistem ini terdapat dalam proses pengolahan citra. Perangkat lunak menerima masukan berupa gambar dari kamera dan memproses nya. Dalam penelitian ini rancangan perangkat lunak dibagi menjadi tiga bagian yaitu bagian *preprocessing image*, deteksi titik tengah dan tepi dengan *blob*, serta *Region of Interest*.

2.1.1 Preprocessing Image

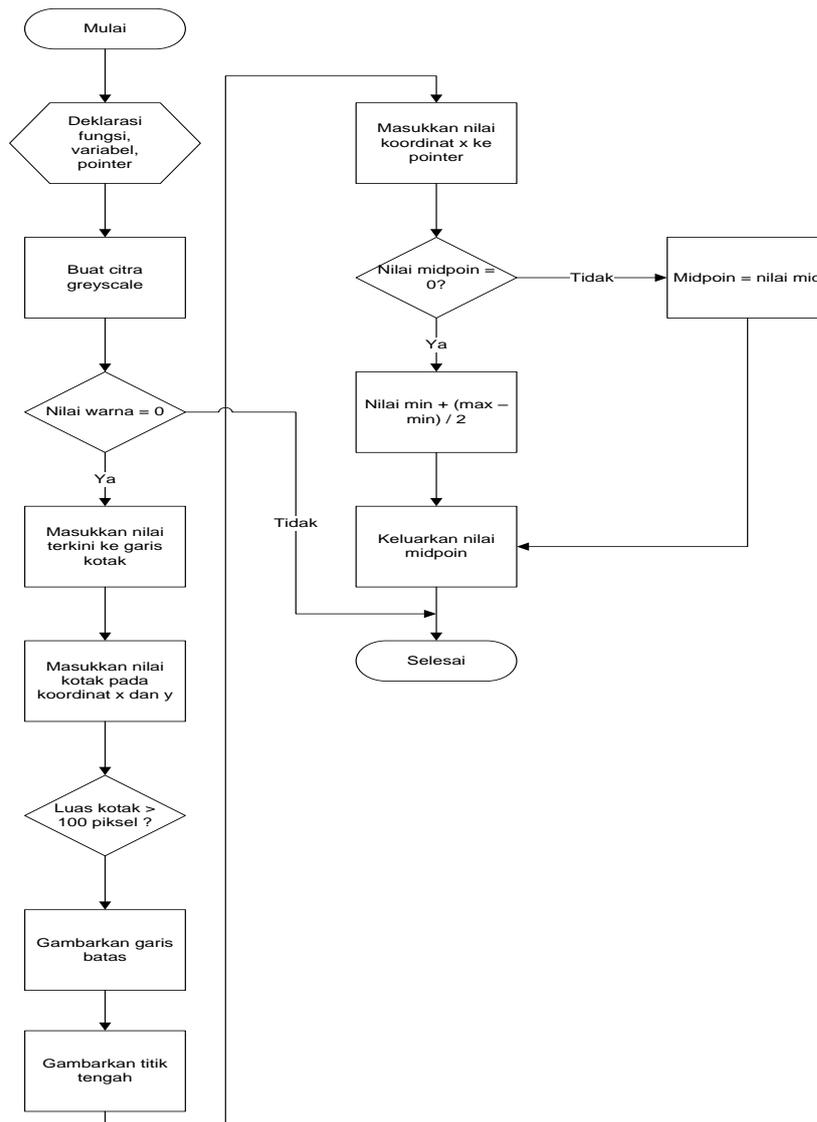
Proses awal yang dilakukan adalah pengambilan citra oleh handycam sebagai masukan. Kemudian mengubah tipe masukan pada kamera menjadi *IplImage* sekaligus deklarasasi pointer untuk citra. Proses berikutnya adalah membuat citra RGB. Setelah itu proses berikutnya adalah melakukan konversi dari citra RGB ke HSV. Pada proses ini dicari nilai batas atas dan batas bawah nilai HSV untuk membedakan warna yang akan di *threshold*.. Setelah itu dilakukan proses *thresholding* untuk mendapatkan citra biner, sehingga terlihat daerah yang termasuk objek dan *background*. Objek dalam hal ini adalah garis putih pada landasan sedangkan *background* adalah landasan itu sendiri. Diagram alir pada *preprocessing image* ini ditunjukkan pada Gambar 2.



Gambar 2. Diagram alir *preprocessing image*

2.1.2 Deteksi garis

Proses pada deteksi tepi setelah citra dari landasan berhasil dipisah antara garis putih dengan latar belakang nya adalah pencarian titik tengah pada garis landasan. Diagram alir dari proses ditunjukkan pada Gambar 3. Proses dimulai dengan deklarasi variable, fungsi, dan pointer. Kemudian dibuat citra 8 bit 1 channel atau *grayscale*, selanjutnya proses memasukkan nilai terkini dalam garis. Kemudian dibuat variable untuk mengisi nilai tersebut, variable yang dibuat mewakili dua titik koordinat x dan y serta lebar (*width*) dan tinggi (*height*). Proses selanjtnya mencari nilai piksel dari citra. Apabila terdeteksi dan memiliki luas area melebihi 100 piksel proses akan dilanjutkan.



Gambar 3 Diagram alir deteksi garis

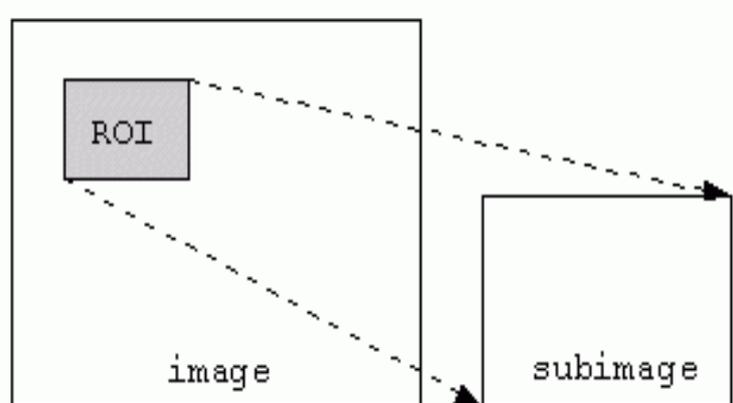
Proses pencarian nilai tengah dimulai dengan membuat batas area paling luar yang terdeteksi dengan garis kotak. Setelah proses membuat garis batas, selanjutnya ditentukan titik tengah atau koordinat tengah dari kotak tersebut. Berdasarkan dengan informasi yang didapat dari lebar dan tinggi kotak dapat dihitung koordinat tengah. Dimisalkan dari dua titik koordinat $A(x,y)$ koordinat pojok kiri atas kota kemudian $C(x_2,y_2)$ dan $D(x_3,y_3)$ merupakan dua koordinat tengah. Untuk menentukan titik tengah penghitungannya sebagai berikut :

- Untuk nilai $x_2 = x + (\text{lebar kotak} / 2) - 1$
- Untuk nilai $y_2 = y + (\text{tinggi kotak} / 2) - 1$
- Untuk nilai $x_3 = x + (\text{lebar kotak} / 2) + 1$
- Untuk nilai $y_3 = y + (\text{tinggi kotak} / 2) + 1$

2.1.3 Region of Interest

Sebuah citra dapat dianggap mengandung sub citra yang kadang kala disebut *region of interest* (ROI), atau daerah. Daerah yang dimaksud adalah bagian dari sebuah citra dimana

memungkinkan diproses untuk menekan blur sementara bagian lain diproses untuk meningkatkan penampilan warna. Pada system ini fungsi ROI untuk mendeteksi garis yang masuk pada area tersebut. Sehingga apabila terdapat citra garis yang berada diluar ROI maka proses pendeteksi tidak dilakukan.



Gambar 3. Citra dengan RoI

Pada citra yang ditunjukkan pada Gambar 3 sebuah ROI didefinisikan di bagian sudut kanan atas dari sebuah citra. Metode ini berguna salah satunya untuk memotong sebuah objek dari citra, atau jika diinginkan untuk proses *template matching* dengan sub-citra [3].

2.2 Landasan pacu

Perancangan landasan pacu dibuat menyerupai landasan pesawat terbang yang memiliki Precision Approach Instrument [4]. Oleh sebab itu pola pembuatan garis pada landasan mengacu pada standard FAA tersebut, seperti yang ditunjukkan pada Gambar 4. Garis landasan dibuat berwarna putih dengan ukuran dan panjang lintasan yang telah ditentukan menyerupai landasan pesawat terbang.



Gambar 4. Pola garis landasan Precision Approach Instrument

Penggunaan landasan pacu pada penelitian ini sebagai objek dalam pengambilan data video terbang. Landasan pacu (*runway*) dibuat menyerupai landasan asli untuk pesawat terbang seperti yang ditunjukkan Gambar 5. Pembuatan garis landasan menggunakan kertas berwarna putih dengan ukuran panjang dan lebar $\pm 29,5 \times 5$ centimeter. Lokasi landasan pacu berada di arah timur laut Sekolah Tinggi Teknologi Adisutjipto (STTA) berada di daerah Maguwoharjo, Sleman.



Gambar 5. Pola garis pada landasan

Panjang dan lebar lintasan beraspal kurang lebih 68 meter dan 5,25 meter. Panjang garis pada landasan dibuat sepanjang kurang lebih 20 meter. Pola garis disusun menyerupai landasan pada pesawat terbang sesuai dengan standar dari FAA (Federal Aviation Administration) dengan skala 1:40.

2. 3 Implementasi perangkat keras

Perangkat keras yang digunakan dalam penelitian ini adalah sebuah mini-pc ZOTAC ZBOX nano AD10 dan SONY handycam yang ditunjukkan Gambar 6. *Mini Personal Computer (Mini-PC)* merupakan sebuah komputer dengan ukuran yang kecil yang terdiri dari CPU, memori, media penyimpanan serta modul antarmuka lain yang dirakit dalam satu board. Komputer ini sangat luas penggunaannya, mulai dari bidang kedokteran, pengendalian dan monitor pesawat, radar, telekomunikasi, database, sistem kendali di industri, pencitraan, peralatan bergerak, pengendalian robot, finger scanning dan proses-proses lain yang membutuhkan perhitungan dan algoritma sederhana sampai kompleks.



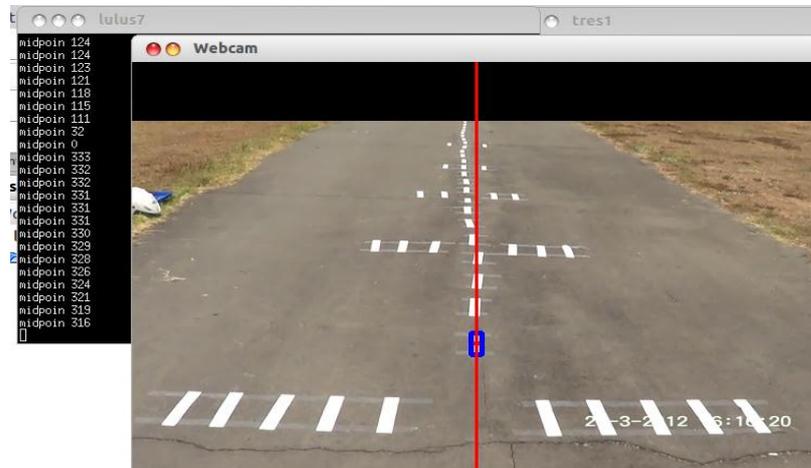
Gambar 6. ZOTAC ZBOX nano AD10

3. HASIL DAN PEMBAHASAN

3. 1 Pengujian jarak deteksi

Pada pengujian ini digunakan webcam dengan resolusi 640x480 pixel. Pada pengujian jarak deteksi memfokuskan pada jarak maksimal dari kamera untuk dapat mendeteksi garis. Proses pengambilan data dilakukan dengan camera perekam dengan melakukan variasi ketinggian yang dikukur dari permukaan tanah. Variasi ketinggian dimulai dari 130 cm yang

merupakan ketinggian dari tripod dari permukaan tanah. Gambar 7 menunjukkan proses deteksi garis landasan dari video yang dijalankan pada mini-pc.



Gambar 7 Proses deteksi garis

Hasil pengujian seperti pada Tabel 1 menunjukkan bahwa pada jarak 3 meter dari permukaan tanah, garis tidak dapat terdeteksi. Hal ini karena nilai piksel dari garis yang terdeteksi kurang dari 100 piksel. Sistem mensyaratkan agar nilai dari lebar x panjang area blob harus melebihi 100 piksel. Jarak tiga meter menunjukkan bahwa proses deteksi sudah tidak dapat dilakukan.

Tabel 1 Hasil pengujian ketinggian deteksi

Tinggi kamera	Keterangan
300cm	Tidak terdeteksi
240cm	Terdeteksi dengan baik
180cm	Terdeteksi dengan baik
160cm	Terdeteksi dengan baik
130cm	Terdeteksi dengan baik
100cm	Terdeteksi dengan baik
77cm	Terdeteksi dengan baik
60cm	Terdeteksi dengan baik
40cm	Terdeteksi dengan baik
30cm	Terdeteksi dengan baik
20cm	Terdeteksi dengan baik

3. 2 Pengujian fungsional deteksi

Pengujian ini dilakukan untuk mengetahui seberapa tepat sistem dapat mendeteksi garis pada landasan. Sistem dapat mengenali garis yang masuk dalam ROI. Pengujian ketepatan

deteksi garis dilakukan dengan menganalisis kecepatan pengambilan citra video kemudian dicari jumlah frame total dari citra video tersebut. Frame yang didapat kemudian diujikan pada sistem sehingga dapat diketahui jumlah frame yang terdeteksi dan yang tidak. Pengambilan citra video dilakukan menggunakan sepeda motor dan oleh dua orang, satu orang sebagai pengemudi dan satunya bertugas merekam. Pada pengujian ini diambil beberapa citra video dengan variasi tingkat kecepatan. Hasil dari pengujian ditunjukkan seperti pada Tabel 2.

Tabel 2 Hasil pengujian fungsional deteksi

No Percobaan	Kecepatan sepeda motor	Jumlah <i>frame</i> total	Jumlah <i>frame</i> yang garisnya terdeteksi	Persentase deteksi
1	10km/jam \pm 1	140	140	100%
2	10km/jam \pm 1	142	142	100%
3	10km/jam \pm 1	129	129	100%
4	20km/jam \pm 1	71	71	100%
5	20km/jam \pm 1	71	71	100%
6	20km/jam \pm 1	66	66	100%
7	30km/jam \pm 1	54	50	92.6%
8	30km/jam \pm 1	48	46	95.83%
9	30km/jam \pm 1	50	48	96 %

Berdasarkan hasil yang tertera pada Tabel 6.3 sistem yang dibuat mampu mendeteksi garis dengan baik pada kondisi kecepatan 10km/jam sampai 20km/jam. Sistem mampu memberikan data koordinat sumbu x dari garis yang masuk ke daerah ROI. Ukuran frame yang dibuat yakni 640x480. Angka yang keluar dari jendela merupakan posisi dalam sumbu x saat garis terdeteksi.

Pada percobaan ke 7,8,9 proses deteksi garis tidak berhasil dengan baik, karena dari jumlah total frame terdapat beberapa frame yang tidak terlacak. Setelah dilakukan analisis terjadi *miss* pada beberapa *frame*, hal ini dikarenakan kecepatan frame yang masuk dalam ROI lebih cepat daripada waktu proses deteksi.

4. KESIMPULAN

1. Sistem dapat mendeteksi garis landasan dengan keluaran berupa nilai koordinat sumbu x pada kecepatan pengambilan video sampai dengan 30km/jam.
2. Ketepatan dari pendeteksian terhadap garis dapat mencapai 100% jika garis yang terdeteksi tidak lebih dari 1 pada kecepatan pengambilan video sampai dengan 20km/jam.
3. Sistem dapat mendeteksi garis landasan pada ketinggian dibawah tiga meter dari permukaan tanah.

DAFTAR PUSTAKA

- [1] Pramadi, L. N., 2010, Perlukah NKRI membentuk Skadron UAV(UAS)?, <http://www.tandef.net/perlukah-nkri-membentuk-skadron-uav-uas> diakses tanggal 10 Maret 2012.
- [2] Huh, S. dan Shim, D.H., 2010, *A Vision-based automatic landing method for fixed-winguavs*, J. Intell. Robotics System, 2010: 217-231
- [3] Nasrudin, 2009, OpenCV Region of Interest (ROI), <http://nashruddin.com/OpenCV Region of Interest %28ROI%29s>. Diakses pada tanggal 1 Juni 2012
- [4] FAA., 2011, *A Quick reference: airfield standards*. Second Edition