

Perancangan Sistem *Tracking Quadrotor* untuk Sebuah Target Bergerak di Darat Menggunakan Sistem *Fuzzy*

Mochamad Rama Ramadhan, Rusdhianto Effendi A.K.

Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember (ITS)

Jl. Arief Rahman Hakim, Surabaya 60111 Indonesia

e-mail: ditto@ee.its.ac.id

Abstrak—Dalam dunia militer, pesawat tanpa awak dapat digunakan sebagai pesawat pengintai. Pesawat pengintai digunakan untuk mengumpulkan data intelijen, penyadapan atau pemantauan posisi taktis. Salah satu misi pesawat tanpa awak yaitu melakukan *tracking* kendaraan tempur atau target bergerak lain yang berada di darat. Pada Tugas Akhir ini dibahas tentang *quadrotor* yang ditugaskan untuk melakukan *tracking* terhadap target yang bergerak di darat. Pengenalan posisi *quadrotor* dan target menggunakan sistem kamera Optitrack. Dirancang algoritma program untuk dapat membedakan di mana posisi *quadrotor* dan posisi target berada. Digunakan metode kontrol logika *fuzzy* model Mamdani untuk mengatur keseimbangan sudut *roll* dan *pitch quadrotor* saat terbang. Selain pada pengaturan keseimbangan, kontrol logika *fuzzy* digunakan pula pada pengaturan posisi *quadrotor* pada sumbu X dan Y. Nilai parameter metode kontrol logika *fuzzy* diperoleh secara eksperimental. Hasil implementasi menunjukkan bahwa kontrol logika *fuzzy* dapat diterapkan pada *quadrotor* untuk menjaga kestabilan karena mampu mempertahankan sudut *roll* dan *pitch quadrotor* di sekitar sudut $\pm 5^\circ$ dengan simpangan terjauh sebesar $4,45^\circ$ dan $-7,45^\circ$ serta pada pengaturan posisi, *quadrotor* dapat bergerak menuju posisi yang dikehendaki dengan *error* posisi maksimal sebesar 0,29 meter dan -0,28 meter pada sumbu X dan Y. Saat implementasi *tracking* target, *quadrotor* mampu bergerak mengikuti target yang bergerak dengan *error* posisi maksimal saat *tracking* sebesar 0,83 meter dan 1,51 meter.

Kata Kunci—Kontrol Logika Fuzzy, *Quadrotor*, *Tracking*

I. PENDAHULUAN

Seiring dengan kemajuan teknologi, berkembang pula kegunaan pesawat tanpa awak. Pesawat tanpa awak dapat difungsikan untuk berbagai tujuan. Dengan karakteristik yang bermacam – macam, pesawat tanpa awak dapat digunakan untuk berbagai macam aplikasi, seperti pengambilan gambar dari udara, pemantauan lahan yang luas dan pengamanan daerah perbatasan negara.

Dalam dunia militer, pesawat tanpa awak dapat digunakan sebagai pesawat pengintai. Pesawat pengintai dapat digunakan untuk mengumpulkan data intelijen, penyadapan atau pemantauan medan perang. Adapun salah satu misinya yaitu melakukan *tracking* kendaraan atau target bergerak yang berada di darat.

Salah satu jenis pesawat tanpa awak adalah *quadrotor*. *Quadrotor* memiliki empat buah motor penggerak dan baling - baling di tiap ujung kerangka utama yang menghadap searah.

Dengan konfigurasi yang demikian, *quadrotor* dapat lepas landas dan mendarat secara vertikal (VTOL), bergerak ke segala arah serta dapat melayang di udara (*hover*). Dengan kemampuan tersebut, *quadrotor* dapat digunakan untuk melakukan *tracking* terhadap objek yang bergerak di darat.

Dalam penelitian ini akan dibahas penggunaan metode kontrol logika *fuzzy* sebagai pengendali kestabilan sudut *roll* dan *pitch* serta sebagai pengendali posisi *quadrotor* pada sumbu X dan Y.

Pada makalah ini terbagi menjadi lima bagian. Bagian I berisi pendahuluan. Bagian II adalah teori dasar yang digunakan pada penelitian. Bagian III menjelaskan perancangan sistem. Bagian IV membahas mengenai hasil simulasi dan implementasi. Bagian V berisikan kesimpulan.

II. TEORI QUADROTOR

A. *Quadrotor*

Quadrotor merupakan salah satu pesawat tanpa awak yang menggunakan 4 buah motor penggerak yang dipasang pada ujung - ujung dari tiang simetris berbentuk “+”. Pada setiap motor terpasang baling – baling yang bertujuan untuk menghasilkan daya angkat sehingga *quadrotor* dapat melayang.

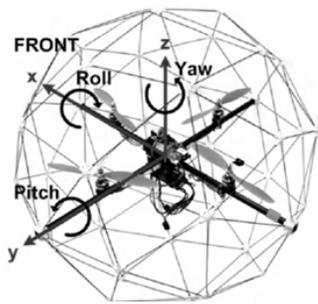
Secara umum *quadrotor* memiliki pengaturan motor depan dan motor belakang yang berputar searah jarum jam, sedangkan motor kiri dan motor kanan berputar berlawanan arah jarum jam. Motor – motor tersebut bersifat independen, yaitu kecepatannya dapat berbeda untuk satu motor dengan motor yang lain.

Dengan mengatur kecepatan dari setiap motor, maka pergerakan dari *quadrotor* dapat berubah. *Quadrotor* akan terbang dengan syarat gaya angkat dari *quadrotor* lebih besar dari gaya gravitasi. Pada kondisi titik berat yang seimbang dan karakteristik motor yang sama, kondisi *hover* tercapai saat semua motor memiliki kecepatan yang besarnya sama.

Quadrotor memiliki kelebihan untuk dapat bergerak ke segala arah. Untuk dapat menuju posisi tertentu, *quadrotor* memiliki 4 pergerakan dasar yang digunakan berdasarkan pengaturan kecepatan keempat motor, yaitu:

1) *Throttle (U1)*

Gerakan *throttle* yaitu gerakan menaikkan dan menurunkan posisi *quadrotor* pada sumbu vertikal. Pergerakan ini terjadi ketika kecepatan dari keempat motor dipercepat atau diperlambat dengan nilai penambahan yang sama.



Gambar 1. Gerak *Roll*, *Pitch* dan *Yaw* Pada *Quadrotor*[1]

2) *Roll* (U_2)

Gerakan *roll* yaitu gerakan perputaran *quadrotor* pada sumbu X. Besar sudut *roll* disimbolkan dengan simbol ϕ . Pergerakan ini terjadi ketika kecepatan motor kiri, dipercepat atau diperlambat, dan kecepatan motor kanan diubah sebaliknya.

3) *Pitch* (U_3)

Gerakan *pitch* yaitu gerakan perputaran *quadrotor* pada sumbu Y. Besar sudut *pitch* disimbolkan dengan simbol θ . Pergerakan yang terjadi ketika kecepatan motor depan, dipercepat atau diperlambat, dan kecepatan motor belakang diubah sebaliknya.

4) *Yaw* (U_4)

Gerakan *yaw* yaitu gerakan perputaran *quadrotor* pada sumbu Z. besar sudut *yaw* disimbolkan dengan simbol ψ . Pergerakan ini terjadi dengan cara mempercepat atau memperlambat motor depan dan belakang sedangkan kecepatan motor kiri dan kanan diubah sebaliknya.

Gerak *Roll*, *Pitch* dan *Yaw* ditunjukkan pada Gambar 1.

Dari pengaturan perubahan kecepatan untuk setiap motor, dapat diatur sinyal kontrol yang bekerja[2], yaitu:

$$U_1 = \frac{F_{T1} + F_{T2} + F_{T3} + F_{T4}}{m} \tag{1}$$

$$U_2 = F_{T2} - F_{T4} \tag{2}$$

$$U_3 = F_{T1} - F_{T3} \tag{3}$$

$$U_4 = (F_{T1} + F_{T3} - F_{T2} - F_{T4})d \tag{4}$$

dengan F_{Ti} adalah gaya *thrust* yang dihasilkan untuk setiap motor yang berputar, d adalah koefisien faktor *drag* dan m adalah massa total dari *quadrotor*.

B. Model Matematika *Quadrotor*[2]

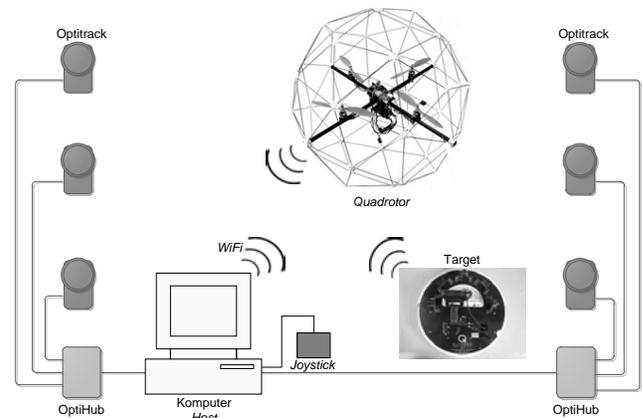
Model matematika *quadrotor* diperoleh dari hasil identifikasi fisik. Dengan menggunakan pendekatan Newton – Euler diketahui hubungan antar gaya yang bekerja pada sistem.

Persamaan dinamika *quadrotor* pada gerak rotasi dan translasi dapat dituliskan sebagai berikut:

$$\dot{p} = \frac{L}{J_{xx}} - \frac{qr}{J_{xx}}(J_{zz} - J_{yy}) \tag{5}$$

$$\dot{q} = \frac{M}{J_{yy}} - \frac{pr}{J_{yy}}(J_{xx} - J_{zz}) \tag{6}$$

(7)



Gambar 2. Arsitektur Sistem *Tracking Target* (8)

$$\dot{r} = \frac{N}{J_{zz}} - \frac{pq}{J_{zz}}(J_{yy} - J_{xx}) \tag{9}$$

$$\ddot{x} = \frac{U_1}{m}(-\sin \phi \sin \psi + \cos \phi \sin \theta \cos \psi) \tag{10}$$

$$\ddot{y} = \frac{U_1}{m}(-\sin \phi \cos \psi + \cos \phi \sin \theta \sin \psi)$$

$$\ddot{z} = \frac{U_1}{m}(\cos \phi \cos \theta - g)$$

dengan p, q, r adalah kecepatan sudut *quadrotor* untuk setiap sumbu pada *frame* bodi, sedangkan x, y, z adalah posisi *quadrotor* pada *frame* bumi. L, M, N adalah momen *roll, pitch* dan *yaw* pada *frame* bodi. J_{xx}, J_{yy}, J_{zz} adalah momen inersia pada sumbu x, y dan z . U_1 adalah gerak *throttle*, m adalah massa *quadrotor* dan g adalah percepatan gravitasi.

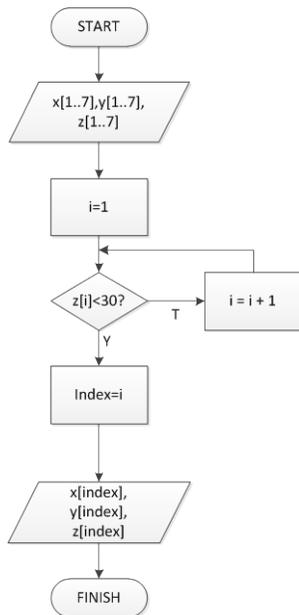
III. PERANCANGAN SISTEM *TRACKING*

A. Sistem *Tracking Target*

Arsitektur sistem *tracking target* terdiri dari komputer *host, quadrotor, sistem* kamera *Optitrack, robot target* dan *joystick* seperti yang ditunjukkan pada Gambar 2.

Pada Gambar 2, komputer *host* mengirimkan program pergerakan *quadrotor* melalui komunikasi *wireless*. Digunakan *joystick* USB sebagai pemicu terbang atau sakelar pengaktifan *quadrotor*. *Joystick* berguna sebagai sakelar pengaman ketika *quadrotor* tiba – tiba berada dalam keadaan tidak stabil dan harus dihentikan. Kemudian target dikendalikan melalui komputer *host* menggunakan komunikasi *wireless*. Pergerakan *quadrotor* dan target dibaca oleh kamera *Optitrack*. Data posisi kedua objek dikirimkan menuju komputer *host* melalui komunikasi USB. Data yang diperoleh dikirimkan kembali oleh komputer *host* menuju *quadrotor* untuk melakukan aksi koreksi posisi, sehingga *quadrotor* dapat melakukan *tracking* terhadap target.

Untuk membaca posisi target dan *quadrotor* digunakan sistem kamera *Optitrack*. Pada sistem yang dirancang, terdiri dari enam buah kamera yang dipasang di dalam ruangan. Kamera – kamera tersebut diletakkan di dinding bagian atas ruang pada sudut yang berbeda. Setiap kamera dihubungkan dengan *hub* yang dihubungkan ke komputer *host* dengan menggunakan komunikasi USB. Pada bodi *quadrotor* dan target ditempelkan *marker* reflektif, yaitu benda berbentuk



Gambar 3. Diagram Alir Penentuan Data Posisi Target

bola yang terbuat dari bahan khusus yang dapat dideteksi oleh kamera Optitrack. Sehingga posisi *quadrotor* dan target pada sumbu X, Y dan Z dapat diketahui.

Data yang dikirim Optitrack kepada komputer *host*, berupa data posisi X, Y, Z untuk setiap *marker* yang terbaca. Namun data tersebut belum dapat dipastikan apakah data posisi *quadrotor* atau data posisi robot target karena *quadrotor* dan robot target menggunakan jenis *marker* yang sama. Oleh karena itu, dirancang algoritma program, yang digunakan untuk mengenali data posisi apakah data posisi tersebut adalah data posisi *quadrotor* atau data posisi robot target.

Data yang dikirim merupakan data untuk setiap *marker*, dengan format data posisi X, Y, Z secara berurutan. Untuk menyeleksi data posisi dari robot target, digunakan perintah perbandingan *less than*. Pada sumbu Z, data posisi *quadrotor* dan data posisi robot target dapat dipastikan berbeda karena ketinggian *quadrotor* dan robot target tidak akan sama. Bila terdapat data untuk sumbu Z kurang dari 30 cm, ditentukan data tersebut merupakan data posisi untuk target. Sebaliknya bila data posisi pada sumbu Z lebih dari 30 cm maka data tersebut merupakan data posisi untuk *quadrotor*. Setelah diketahui data posisi pada sumbu Z, dapat diketahui data posisi sumbu X dan Y untuk *quadrotor* dan robot target.

B. Perancangan Kontroler

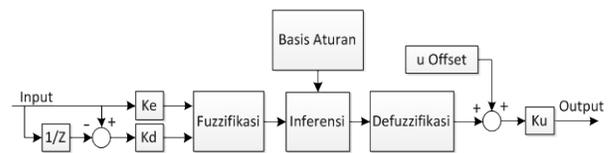
Kontroler dirancang untuk mengatasi masalah kestabilan dan sebagai pengendali posisi *quadrotor*. Metode yang dipilih adalah metode kontrol logika *fuzzy*.

Pada penelitian ini, metode kontrol logika *fuzzy* digunakan sebagai kontroler gerak rotasi sudut *roll* dan *pitch* dan kontroler gerak translasi pada sumbu X dan Y. Kontroler yang dirancang yaitu kontroler logika *fuzzy* PD dengan masukan berupa sinyal *error* dan delta *error*, yaitu

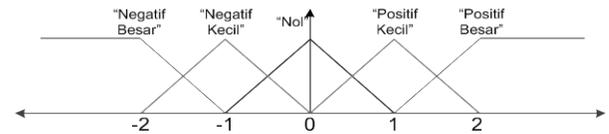
$$e(k) = r(k) - y(k) \tag{11}$$

dan

$$\tag{12}$$



Gambar 4. Kontroler Logika *Fuzzy*
 $de(k) = e(k) - e(k - 1)$



Gambar 5. Fungsi Keanggotaan *Error* dan Delta *Error*

Tabel 1.
Basis aturan hubungan *error* dengan delta *error*

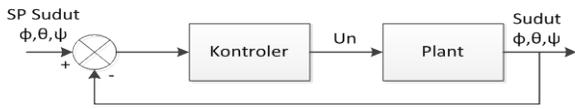
ΔE \ E	E				
	NB	NK	N	PK	PB
NB	1	1	1	2	3
NK	1	1	2	3	4
N	1	2	3	4	5
PK	2	3	4	5	5
PB	3	4	5	5	5

Dengan *error* merupakan selisih antara nilai *set point* r dengan nilai respons keluaran y pada waktu k. Delta *error* merupakan nilai *error* saat k dikurangi *error* pada saat k - 1.

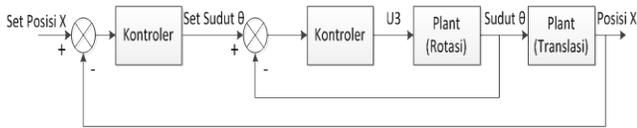
Pada kontroler logika *fuzzy* yang dirancang, digunakan lima *sub-himpunan fuzzy* berbentuk segitiga dengan jarak yang sama. Berdasarkan logika bahasa, kelima *sub-himpunan fuzzy* tersebut dibagi menjadi “negatif besar”, “negatif kecil”, “noI”, “positif kecil” dan “positif besar”.

Agar daerah kerja untuk fungsi keanggotaan tidak perlu diubah, maka ditambahkan komponen pengali. pengali yang digunakan yaitu Ke dan Kd. Ke merupakan parameter kontroler yang digunakan sebagai pengali nilai *error*. Kd merupakan parameter kontroler yang digunakan sebagai pengali nilai delta *error*. Untuk mendapatkan sinyal kontrol yang dapat memperbaiki respons sistem, diperlukan pengaturan parameter Ke dan Kd. Pencarian parameter Ke dan Kd dilakukan dengan cara *tuning* eksperimental. Proses *tuning* dilakukan dengan metode *trial and error* yaitu membandingkan respons keluaran untuk setiap perubahan nilai Ke dan Kd hingga diperoleh respons sistem yang sesuai.

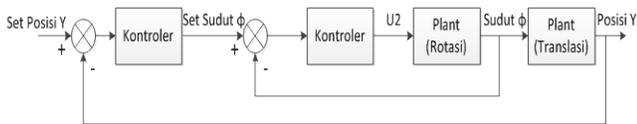
Pengambilan keputusan dilakukan dengan menggunakan aturan *If - Then* pada kedua variabel masukan untuk mendapatkan nilai keluaran. Kombinasi aturan *If - Then* menggunakan lima masukan dari fungsi keanggotaan *error* dan lima masukan dari fungsi keanggotaan delta *error* sehingga kombinasi aturan berjumlah 25 buah. Untuk menyederhanakan seluruh aturan *If - Then* tersebut, maka



Gambar 6. Diagram Blok Untuk Sistem Pengaturan Gerak Rotasi



Gambar 7. Diagram Blok Untuk Gerak Translasi Sumbu X



Gambar 8. Diagram Blok Untuk Gerak Translasi Sumbu Y.

aturan dibuat dalam bentuk tabel yang ditunjukkan pada Tabel 1. Hasil dari proses inferensi berupa himpunan fuzzy diubah menjadi nilai tegas pada proses defuzzifikasi. Teknik yang dipilih adalah menggunakan metode *center of gravity* berikut

$$y = \frac{\sum_{i=1}^r y_i \int \mu_i(y) dy}{\sum_{i=1}^r \int \mu_i(y) dy} \quad (13)$$

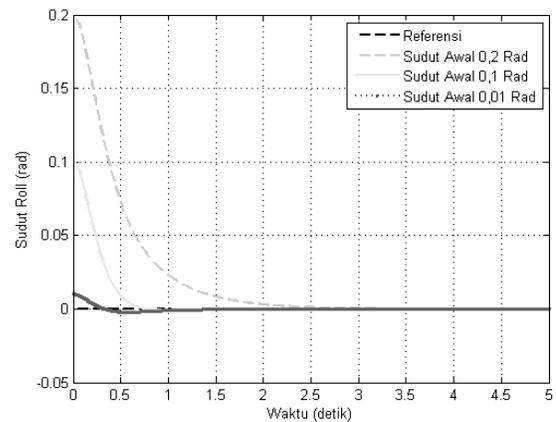
Quadrotor dapat dikatakan stabil ketika pada saat diterbangkan dapat melayang di udara (*hover*). Untuk dapat melakukan *hover*, *Quadrotor* harus seimbang, yaitu sudut *roll* dan *pitch* harus dibuat mendekati nol. Sinyal kontrol yang tidak sesuai menyebabkan *quadrotor* tidak stabil dan sulit untuk dikendalikan. Oleh karena itu, kontroler diperlukan untuk mengatur kecepatan setiap motor sedemikian rupa sehingga sudut *roll*, *pitch* dan *yaw* yang dihasilkan selalu mendekati nol.

Berikut merupakan diagram blok untuk sistem pengaturan gerak rotasi pada *quadrotor* untuk $n=2,3,4$ adalah gerak *roll*, *pitch* dan *yaw* berturut - turut.

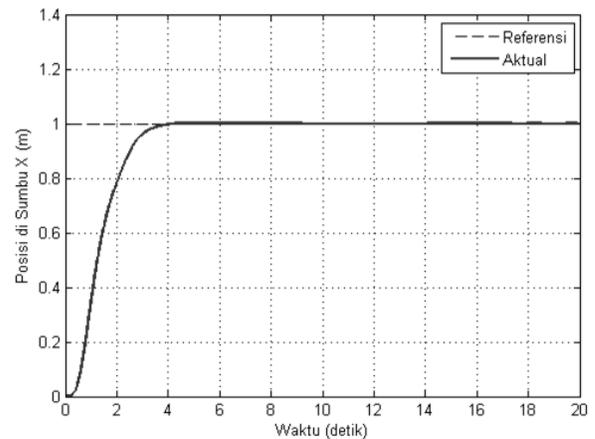
Pada *quadrotor*, terdapat pergerakan yang dapat mempengaruhi pergerakan lain. Gerakan rotasi sudut *roll* mempengaruhi posisi *quadrotor* pada sumbu Y dan gerakan rotasi sudut *pitch* mempengaruhi posisi *quadrotor* pada sumbu X. Oleh karena itu, untuk mengatur posisi *quadrotor* pada sumbu X dan Y, kontroler posisi dirancang secara *cascade* dengan kontroler sudut.

IV. PENGUJIAN PENGATURAN KESTABILAN DAN POSISI QUADROTOR

Pengujian dilakukan untuk mengetahui apakah sistem yang dirancang sesuai dengan harapan serta menguji performa kontroler fuzzy sebagai pengendali kestabilan sudut *roll* dan *pitch* serta pengendali gerak translasi pada sumbu X dan Y untuk *tracking target* pada *quadrotor*.



Gambar 9. Respons Simulasi Sudut Roll



Gambar 10. Respons Simulasi Posisi di Sumbu X

A. Hasil Simulasi

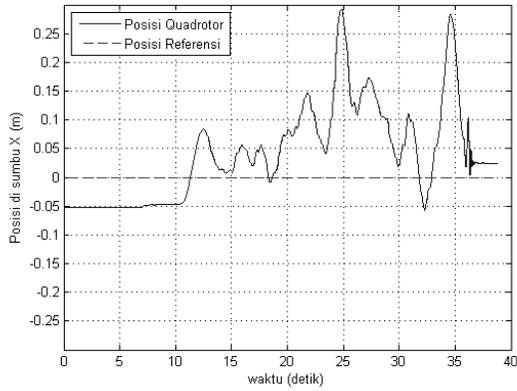
Simulasi pengaturan sudut *roll* dan sudut *pitch* dilakukan dengan menggunakan berbagai kondisi sudut awal yang berbeda. Respons yang diperoleh untuk pengaturan sudut *roll* ditunjukkan pada Gambar 9.

Dari hasil simulasi yang didapat, walau dengan pengujian beberapa kondisi sudut awal yang berbeda, respons sudut *roll* dapat kembali ke sudut 0°. Pada pengujian dengan sudut awal 10°, dapat kembali menuju 0° dalam waktu kurang dari 3 detik tanpa terjadi osilasi. Respons sudut *pitch* yang diperoleh tidak berbeda dengan respons sudut *roll* karena parameter kontroler yang digunakan sama dengan kontroler sudut *roll*.

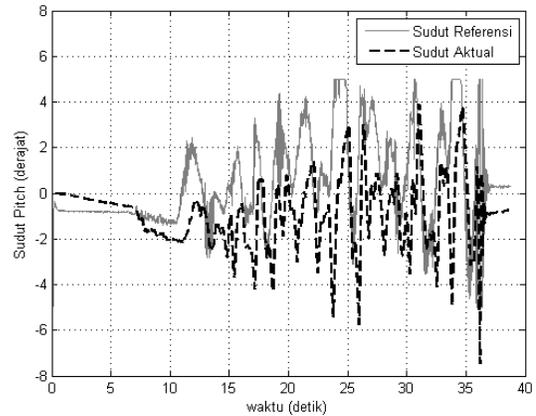
Pada simulasi pengaturan posisi, *quadrotor* diperintahkan bergerak ke posisi satu meter dari titik 0 pada sumbu X dengan kondisi awal sudah berada pada ketinggian 2 meter. Respons posisi *quadrotor* di sumbu X dapat dilihat pada Gambar 10.

Terlihat pada Gambar 10, *quadrotor* bergerak dari posisi nol menuju posisi satu meter selama empat detik. Kontroler posisi dapat mengendalikan posisi *quadrotor* sesuai dengan referensi tanpa terjadi osilasi.

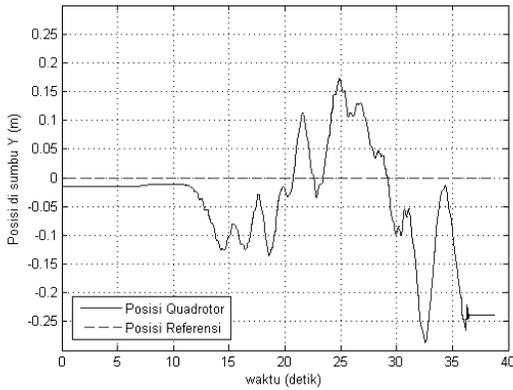
Pada kontroler posisi pada sumbu Y dilakukan pengujian yang serupa, diperoleh respons posisi yang sama dengan kontroler posisi pada sumbu X. Hal tersebut terjadi karena parameter kontroler yang digunakan sama dengan parameter pada kontroler posisi pada sumbu X.



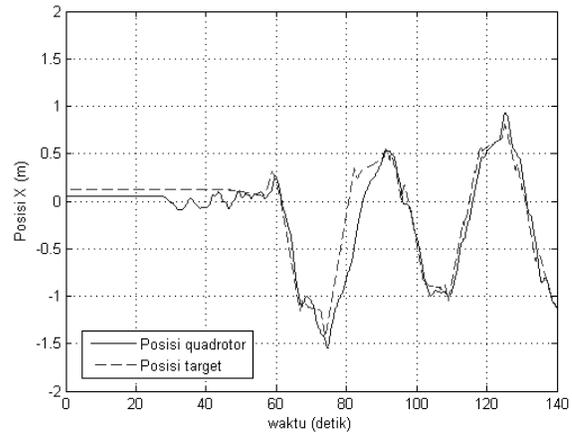
Gambar 11. Respons Posisi *Quadrotor* di Sumbu X



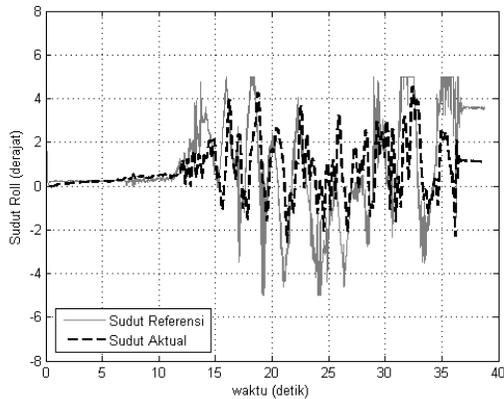
Gambar 14. Respons Sudut *Pitch* Saat *Hover*



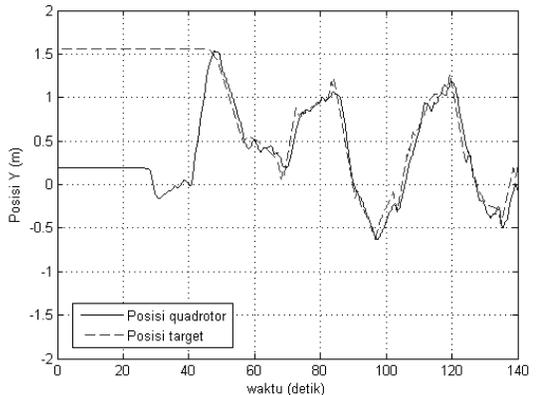
Gambar 12. Respons Posisi *Quadrotor* di Sumbu Y



Gambar 15. Posisi *Quadrotor* dan Target Pada Sumbu X



Gambar 13. Respons Sudut *Roll* Saat *Hover*



Gambar 16. Posisi *Quadrotor* dan Target Pada Sumbu Y

B. Hasil Implementasi

Pada implementasi, *quadrotor* diperintahkan untuk melakukan *hover* pada posisi (0,0) untuk sumbu X dan Y.

Respons posisi *quadrotor* pada sumbu X dan sumbu Y ditunjukkan pada Gambar 11 dan Gambar 12.

Quadrotor berusaha mempertahankan posisi di (0,0). Masih terdapat osilasi posisi di sumbu X dan Y dengan *error* posisi maksimal sebesar 0,29 meter dan 0,28 meter. Walau demikian, osilasi yang terjadi kurang dari ±0,3 meter dan simpangan yang terjadi tidak membesar.

Sinyal kontrol yang berasal dari kontroler posisi pada sumbu X dan Y dijadikan sinyal referensi untuk kontroler sudut *pitch* dan *roll*.

Gambar 13 menunjukkan referensi sudut *roll* yang bergerak dalam *range* ±5°. Referensi sudut *roll* dibatasi untuk tidak melebihi 5° agar sudut *roll quadrotor* yang dihasilkan tidak ekstrem. Pergerakan *quadrotor* dengan sudut *roll* yang ekstrem dapat menyebabkan *quadrotor* menjadi tidak stabil.

Terlihat respons memiliki simpangan sudut *roll* kurang dari ±5° dengan simpangan terjauh sebesar 4,54°.

Hal yang sama terjadi pada pengaturan posisi pada sumbu X. Untuk menjaga posisi *quadrotor* pada sumbu X agar tetap berada pada posisi 0, maka sinyal keluaran kontroler posisi X dijadikan sinyal referensi untuk kontroler sudut *pitch*. Pada

Gambar 14 terlihat sinyal referensi berubah menyesuaikan posisi *quadrotor* pada sumbu X. Kontroler sudut *pitch* akan memberikan sinyal kontrol kepada *quadrotor* agar respons sudut *pitch* sesuai dengan referensi yang diberikan. Terlihat respons sudut *pitch* memiliki simpangan di sekitar $\pm 5^\circ$ dengan simpangan terjauh yang terjadi sebesar $-7,45^\circ$.

Dengan menggunakan kontroler logika *fuzzy* yang sebelumnya telah diuji, dilakukan proses *tracking* target bergerak. Pergerakan target bergerak dikendalikan secara manual melalui komputer *host* menggunakan komunikasi *wireless*. Posisi target diikuti oleh *quadrotor* dengan ketinggian dan orientasi *quadrotor* yang tetap. Posisi *quadrotor* dan target saat proses *tracking* terlihat pada Gambar 15 dan Gambar 16.

Ketika program mulai dijalankan, *quadrotor* masih berada permukaan tanah. *Quadrotor* baru terbang pada detik ke-30. Setelah detik ke-40, *quadrotor* mulai melakukan *tracking* terhadap target.

Dari Gambar 15 dan Gambar 16 terlihat target bergerak dengan kecepatan yang berubah sehingga masih terdapat *error* posisi antara *quadrotor* dengan target. *Error* posisi maksimal pada sumbu X dan sumbu Y sebesar 0,83 meter dan 1,51 meter.

V. KESIMPULAN

Dari hasil pengujian rancangan sistem kontrol pada simulasi dan implementasi dapat diambil kesimpulan bahwa :

- Kontroler logika *fuzzy* dapat digunakan pada *quadrotor* untuk mengendalikan kestabilan sudut *roll* dan *pitch* saat *hover* di sekitar sudut $\pm 5^\circ$ dengan simpangan terjauh sudut *roll* dan *pitch* sebesar $4,54^\circ$ dan $-7,45^\circ$.
- Kontroler logika *fuzzy* dapat digunakan untuk mengendalikan *quadrotor* menuju posisi yang dikehendaki dengan *error* posisi maksimal sebesar 0,29 meter dan -0,28 meter pada sumbu X dan sumbu Y.
- Dengan menggunakan sistem pengenalan posisi *quadrotor* dan target, *quadrotor* dapat melakukan *tracking* terhadap target yang bergerak di darat dengan *error* posisi maksimal sebesar 0,83 meter dan 1,51 meter pada sumbu X dan sumbu Y.

DAFTAR PUSTAKA

- [1]. Quanser Qball-X4, "User Manual Quanser Qball", Quanser.
- [2]. Milionis, Georgios, "A Framework for Collaborative Quadrotor – Ground Robot Mission". Naval Postgraduate School, Desember 2011.