

Perancangan *Autonomous Landing* pada *Quadcopter* Menggunakan *Behavior-Based Intelligent Fuzzy Control*

Chalidia Nurin Hamdani, Rusdhianto Effendie A.K., dan Eka Iskandar
Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember (ITS)
Jl. Arief Rahman Hakim, Surabaya 60111
e-mail: ditto@ee.its.ac.id

Abstrak— *Quadcopter* adalah salah satu *platform unmanned aerial vehicle* (UAV) yang saat ini banyak diriset karena kemampuannya melakukan take-off dan landing secara vertikal. Karena menggunakan 4 motor *brushless* sebagai penggerak utama, *quadcopter* memiliki kompleksitas yang cukup tinggi baik dalam pemodelan maupun pengendalian. Landing merupakan salah satu mekanisme pada *quadcopter* yang membutuhkan kecepatan yang akurat dan aman dengan tetap mempertahankan keseimbangan. Pada penelitian ini, penulis menggunakan *Behavior-Based Intelligent Fuzzy Control* (BBIFC) sebagai dasar kontrol untuk penerapan *autonomous landing* pada *quadcopter*. BBIFC adalah salah satu skema *high-level control* di mana desain kontrol terdiri dari beberapa layer. Ada 2 layer yang digunakan pada penelitian ini yaitu layer untuk pengendalian sudut pitch, roll, yaw dan layer untuk pengendalian ketinggian. Setiap layer memiliki mekanisme kontrol yang berbeda yang didesain menggunakan *Intelligent Fuzzy Controller* dan kontroler PID. Dengan metode ini dihasilkan algoritma untuk mekanisme *safe autonomous landing* dengan mengikuti sinyal eksponensial di mana *quadcopter* mencapai titik 0 (nol) meter dalam waktu 15 detik dan Kontroler PID dapat mengendalikan keseimbangan *quadcopter* dalam waktu 7.97 detik untuk roll dan pitch serta 1.25 detik untuk yaw sejak gangguan sudut diberikan.

Kata Kunci— *Unmanned Aerial Vehicle, quadcopter, autonomous landing, behavior-based intelligent fuzzy control.*

I. PENDAHULUAN

Pada dekade terakhir, dunia penerbangan mengalami perkembangan pesat dengan hadirnya kendaraan udara tak berawak atau yang sering disebut *Unmanned Aerial Vehicle* (UAV). Penggunaan UAV ini bisa dikategorikan cukup luas terutama untuk keperluan militer, monitoring hutan, monitoring perbatasan dan yang lainnya[1].

Salah satu jenis UAV yang banyak diteliti saat ini adalah *quadcopter*, UAV jenis helicopter yang menggunakan empat motor untuk menggerakkan propeler/baling-balingnya. Kelebihan *quadcopter* adalah kemampuannya dalam melakukan take-off dan landing secara vertikal atau biasa dikenal dengan VTOL (*Vertical Take-Off and Landing*). Landing merupakan istilah yang sering digunakan untuk mekanisme saat pesawat mengurangi ketinggian hingga ketinggian 0 meter [9].

Pada umumnya, *quadcopter* dikendalikan melalui radio transmitter dari jarak jauh oleh manusia sehingga sering sekali terjadi kesalahan pengendalian terutama dalam mekanisme landing. Hal ini terjadi karena mekanisme landing merupakan salah satu mekanisme paling kritis dalam pengendalian

quadcopter yang membutuhkan keakuratan kecepatan dengan tetap mempertahankan keseimbangan.

Untuk itu perlu adanya mekanisme *autonomous landing* pada *quadcopter* sehingga faktor kesalahan manusia selama mekanisme landing dapat diatasi. Mekanisme *autonomous landing* ini dibuat dengan menggunakan algoritma berbasis *Behavior-Based Intelligent Fuzzy Control* (BBIFC) di mana strategi pengendalian dibagi menjadi beberapa layer. BBIFC ini digunakan untuk menyederhanakan algoritma pengendalian sehingga didapat sebuah algoritma yang mampu mengendalikan *quadcopter* dalam melakukan mekanisme *autonomous landing* secara aman [3].

BBIFC merupakan modifikasi dari *Behavior-Based Control* dengan menambahkan *Intelligent Fuzzy Controller* ke dalamnya. *Behavior-Based Control* sendiri merupakan skema *high-level control* yang sering digunakan dalam pengendalian robot. Algoritma *Behavior-Based Control* didasarkan dari perilaku makhluk hidup yang berperilaku sesuai keadaan lingkungan [5].

Intelligent Fuzzy Controller ini merupakan kontroler fuzzy seperti PID yang didesain dan diimplementasikan untuk mengendalikan *quadcopter*, baik untuk gerak rotasi maupun gerak translasi. Gerak rotasi terdiri dari gerak roll, pitch dan yaw. Gerak translasi terdiri dari gerak di sumbu X, Y dan Z. Input dari *Intelligent Fuzzy Controller* ini adalah error dan turunan error untuk setiap variabel. Outputnya adalah gaya yang dibutuhkan untuk gerak rotasi dan translasi. Tantangan dalam penelitian ini adalah untuk menghasilkan pengendalian yang akurat sehingga mekanisme landing bisa dilakukan dengan aman.

Konsep dasar dan model matematika *quadcopter* akan dijelaskan pada Bagian 2. Perancangan kontroler akan dibahas di Bagian 3. Dari model matematika dan kontroler yang telah dirancang, didapatkan hasil yang dibahas di Bagian 4. Kesimpulan dan saran untuk pengembangan di masa yang akan datang disampaikan di Bagian 5.

II. DASAR TEORI

A. Gerakan Dasar *Quadcopter* [2]

Quadcopter adalah salah satu jenis *platform* UAV dengan 4 motor *brushless* yang digunakan untuk menggerakkan baling-baling atau propeller. Pada penelitian ini, keempat motor *brushless* tersebut disinkronisasi dengan konfigurasi frame plus (+). Motor *brushless* bagian depan dan belakang

berputar searah jarum jam, sedangkan bagian kanan dan kiri berputar berlawanan arah jarum jam. Berikut gerakan dasar pada *quadcopter*.

- *Throttle* atau *height*

Gerak ini dapat dilakukan dengan menaikkan atau menurunkan kecepatan semua *propeller* dalam jumlah yang sama.

- *Roll*

Gerak ini dapat dilakukan dengan menambah atau mengurangi kecepatan salah satu *propeller* yang kiri atau yang kanan.

- *Pitch*

Gerak ini dapat dilakukan dengan menambah atau mengurangi kecepatan salah satu *propeller*, yang depan atau yang belakang.

- *Yaw*

Gerak ini dapat dilakukan dengan menambah atau mengurangi *propeller* depan belakang dan kanan-kiri secara bersama.

B. Model Matematika *Quadcopter*

Berdasarkan analisis kinematika dan dinamika yang dilakukan oleh Tommasso, didapatkan persamaan yang merupakan model matematika dari *quadcopter* secara umum seperti pada Persamaan 1.

$$\begin{cases} \ddot{X} = \frac{U_1}{m} (\cos \psi \sin \theta \cos \phi + \sin \psi \sin \phi) \\ \ddot{Y} = \frac{U_1}{m} (\sin \psi \sin \theta \cos \phi - \cos \psi \sin \phi) \\ \ddot{Z} = -g + \frac{U_1}{m} (\cos \theta \cos \phi) \\ \dot{p} = \frac{I_{yy} - I_{zz}}{I_{xx}} qr - \frac{I_r}{I_{xx}} q\Omega + \frac{U_2}{I_{xx}} \\ \dot{q} = \frac{I_{zz} - I_{xx}}{I_{yy}} pr + \frac{I_r}{I_{yy}} p\Omega + \frac{U_3}{I_{yy}} \\ \dot{r} = \frac{I_{xx} - I_{yy}}{I_{zz}} pq + \frac{U_4}{I_{zz}} \end{cases} \quad (1)$$

Dengan persamaan kecepatan sebagaimana Persamaan 2.

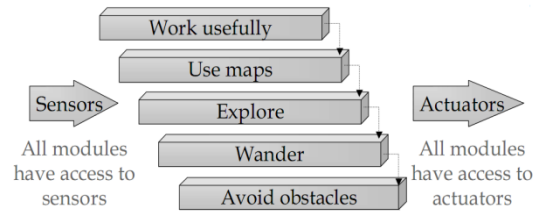
$$\begin{cases} U_1 = b (\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \\ U_2 = l b (-\Omega_2^2 + \Omega_4^2) \\ U_3 = l b (-\Omega_1^2 + \Omega_3^2) \\ U_4 = d (-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2) \\ \Omega = -\Omega_1 + \Omega_2 - \Omega_3 + \Omega_4 \end{cases} \quad (2)$$

C. Behavior-Based Control

Behavior-Based Control adalah salah satu skema *high-level control* yang sering dipakai untuk kendali robot. Algoritma *behavior-based* diturunkan dari sifat-sifat alami makhluk hidup yaitu bertingkah laku sesuai dengan keadaan lingkungannya. Pada gambar 3 ditunjukkan prinsip kerja algoritma *behavior-based* di mana respon robot merupakan hasil stimuli dari lingkungan robot.

Ada 2 macam arsitektur *Behavior Based Control*, yaitu sekuensial dan paralel. Namun, arsitektur yang banyak

digunakan saat ini adalah arsitektur paralel dimana semua mekanisme yang direncanakan berjalan secara bersama dalam satu waktu. Dalam arsitektur paralel seperti pada Gambar 2 semua modul memiliki akses ke sensor dan aktuator.

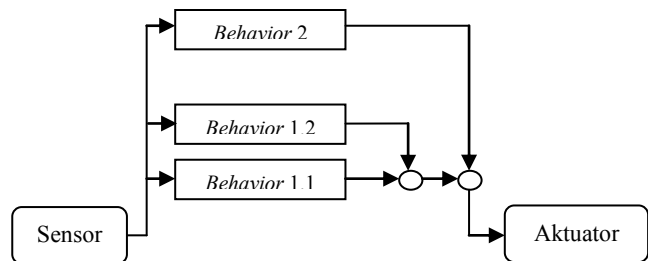


Gambar. 1. Arsitektur paralel dalam *Behavior-Based Control* [4]

Metode *Subsumption* [6]

Subsumption merupakan sebuah metode yang seleksi mekanisme yang berbasis pada *distributed selection*. Dalam metode *subsumption* seperti yang ditunjukkan oleh Gambar 2.11, *behavior* (1,2,3,...N) didefinisikan sebagai tingkah laku robot yang alami. Misalnya seperti jalan maju hingga menemukan halangan, belok kanan, belok kiri, mencari objek, dan seterusnya. Setiap *behavior* bergantung pada stimuli yang diterimanya. Stimuli dalam robotik padarobotik pada dasarnya adalah data-data berbagai macam sensor yang difungsikan. *Behavior* yang memiliki prioritas yang lebih tinggi dapat membatalkan aksi dari *behavior* yang lebih rendah.

Arsitektur yang akan digunakan adalah *colony-style subsumption* di mana *behavior* dipecah menjadi *behavior* yang lebih kecil seperti pada Gambar 2.[7]



Gambar. 2. Arsitektur *colony-style subsumption*

D. Fuzzy Logic Controller (FLC)[8]

Fuzzy logic merupakan salah satu komponen pembentuk *soft computing*. Logika ini pertama kali diperkenalkan oleh Prof. Lotfi A. Zadeh pada tahun 1965. Dasar logika ini adalah teori himpunan *fuzzy*. Pada teori himpunan *fuzzy*, peranan derajat keanggotaan sebagai penentu keberadaan elemen dalam suatu himpunan sangat penting. Nilai keanggotaan atau derajat keanggotaan atau *membership function* menjadi ciri utama dari penalaran dengan *fuzzy logic* ini.

Secara umum FLC mempunyai empat bagian pokok yang dapat dilihat pada Gambar 2.18. Keempat bagian tersebut mempunyai fungsi sebagai berikut.

- *Fuzzifier*

Berfungsi mentransformasikan sinyal masukan yang bersifat *crisp* (bukan *fuzzy*) ke himpunan *fuzzy* dengan menggunakan operator *fuzzifier*

- *Fuzzy Rule base*

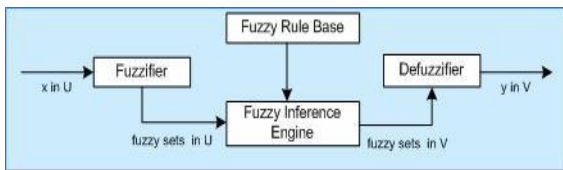
Berisi data aturan dasar yang mendefinisikan himpunan *fuzzy* atas daerah-daerah masukan dan keluaran dalam perangkat aturan kontrol

- *Fuzzy Inference Engine*

Merupakan inti dari FLC yang mempunyai kemampuan dalam mengambil keputusan. Aksi kontrol ini disimpulkan dengan menggunakan implikasi *fuzzy* dan mekanisme inferensi *fuzzy*.

- *Defuzzifier*

Berfungsi mentransformasikan output yang bersifat *fuzzy* menjadi angka *crisp* sesuai dengan objek kontrol.



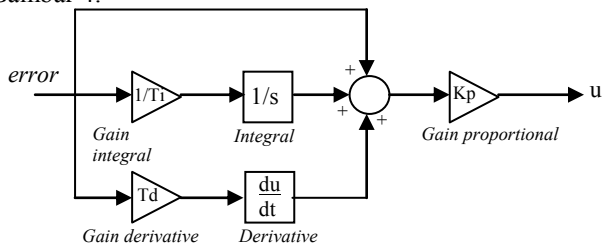
Gambar. 3. Struktur dasar FLC

E. Kontroler PID (Proportional Integral Derivative)[10]

Kontroler PID merupakan metode kendali yang terdiri dari tiga konsep matematika yaitu *proportional*, *integral* dan *derivative*. Kontroler ini menggunakan *negative feedback control*. Input kontroler PID adalah sinyal *error* $e(t)$ dan outputnya adalah sinyal kontrol $u(t)$. Hubungan antara input dan output kontroler PID dapat dituliskan sebagaimana Persamaan 3.

$$u(t) = K_p \left(e(t) + T_d \frac{de(t)}{dt} + \frac{1}{T_i} \int e(t)dt \right) \quad (3)$$

Dari hubungan pada Persamaan 2.11 dapat digambarkan ke dalam diagram blok kontroler PID yang dapat dilihat pada Gambar 4.

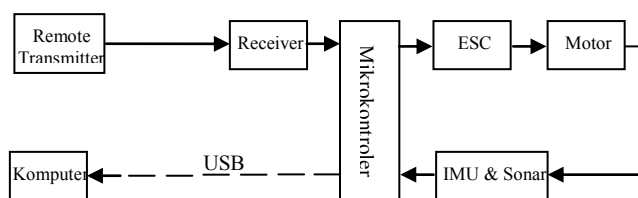


Gambar. 4. Diagram blok kontroler PID

III. PERANCANGAN SISTEM

A. Rancangan Sistem Quadcopter

Sistem *quadcopter* yang dirancang untuk bisa melakukan *autonomous landing* dapat dilihat pada Gambar 5.



Gambar. 5. Rancangan Sistem *Quadcopter*

Pada sistem tersebut, komputer digunakan sebagai *ground station* untuk melakukan pemrograman pada mikrokontroler melalui modul XBee. Mikrokontroler selaku otak dari semua pergerakan *quadcopter* ini akan diprogram dengan metode yang digunakan yakni *Behavior-Based Intelligent Fuzzy Control*. Output dari mikrokontroler ini berupa sinyal pulsa yang dirubah menjadi tegangan oleh ESC dan kemudian tegangan ini yang masuk ke motor. Pada *quadcopter* ini juga terdapat sinyal *feedback* dari beberapa sensor yakni *accelerometer*, *gyro* dan sensor ultrasonik PING. *Remote transmitter* digunakan sebagai alat *switch* untuk mengubah mode penerbangan dari manual ke otomatis dengan mengirimkan sinyal pada *receiver* yang telah terhubung ke mikrokontroler.

B. Rancangan Perangkat Keras Quadcopter

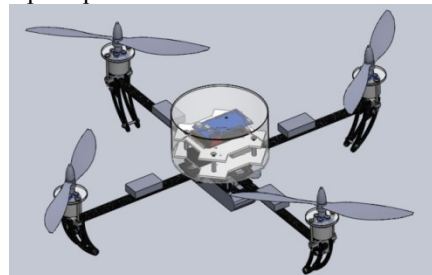
Beberapa perangkat keras yang dibutuhkan dalam perancangan sistem *quadcopter* antara lain: motor *brushless*, *frame quadcopter*, mikrokontroler ArduPilotMega 1, *Electronic Speed Controller*, modul Xbee, *Inertial Measurement Unit*, sensor sonar, *gyroscope*, dan *accelerometer*. Perangkat-perangkat ini dirangkai menjadi sistem *quadcopter* seperti pada Gambar 6.



Gambar. 6. Sistem *quadcopter*

C. Identifikasi Parameter Fisik Quadcopter

Model matematika sistem *quadcopter* telah disebutkan di bagian 2. Pengukuran dan perhitungan parameter fisik dilakukan dengan menggunakan *software* SolidWorks 2012. Dengan bantuan *software* ini dapat dibuat model 3 dimensi (3D) setiap komponen dengan bantuan jangka sorong sebagai alat ukur besaran panjang. Setiap komponen diatur massa jenisnya agar beratnya sesuai dengan massa di dunia nyata. Semua komponen dimodelkan dalam bentuk 3D kemudian dirangkai menjadi satu sehingga terbentuk 3D *quadcopter* seperti pada Gambar 7.



Gambar. 7. Model 3D *quadcopter*

Tabel 1.
Data Parameter Fisik

No.	Parameter Fisik	Nilai
1.	Massa <i>quadcopter</i> (m)	1145 gram
2.	Momen inersia	
	a. Momen inersia rotasi terhadap sumbu X (I_{xx})	$17,64123795 \times 10^{-3}$ kg.m ²
	b. Momen inersia rotasi terhadap sumbu Y (I_{yy})	$17,51065052 \times 10^{-3}$ kg.m ²
	c. Momen inersia rotasi terhadap sumbu Z (I_{zz})	$32,84422323 \times 10^{-3}$ kg.m ²
	d. Momen inersia motor-propeler (I_r)	$7,92 \times 10^{-5}$ kg.m ²
3.	Konstanta <i>thrust</i> (b)	$2,2478 \times 10^{-6}$ N.sec ²
4.	Konstanta <i>drag</i> (d)	2.5167×10^{-7} N.sec ²

Dengan memasukkan parameter fisik pada persamaan 1, maka didapatkan model matematika untuk *quadcopter* khusus untuk penelitian ini seperti pada persamaan 4.

$$\left\{ \begin{aligned} \ddot{X} &= \frac{U_1}{1,145} (\cos \Psi \sin \theta \cos \phi + \sin \Psi \sin \phi) \\ \ddot{Y} &= \frac{U_1}{1,145} (\sin \Psi \sin \theta \cos \phi - \cos \Psi \sin \phi) \\ \ddot{Z} &= -g + \frac{U_1}{1,145} (\cos \theta \cos \phi) \end{aligned} \right. \quad (4)$$

$$\begin{aligned} \dot{p} &= -0,8691891 \quad qr - 0,0448948 \quad q\Omega + 56,685364 \quad U_2 \\ \dot{q} &= 0,8682136 \quad pr + 0,0452296 \quad q\Omega + 57,108101 \quad U_3 \\ \dot{r} &= 0,00397596 \quad pq + 30,446754 \quad U_4 \end{aligned}$$

D. Rancangan Kontroler

Kontroler yang akan diterapkan pada sistem *quadcopter* adalah *Behavior-Based Intelligent Fuzzy Control*. Tujuan perancangan kontroler ini agar *quadcopter* bisa melakukan *autonomous landing*.

Behavior-Based Control

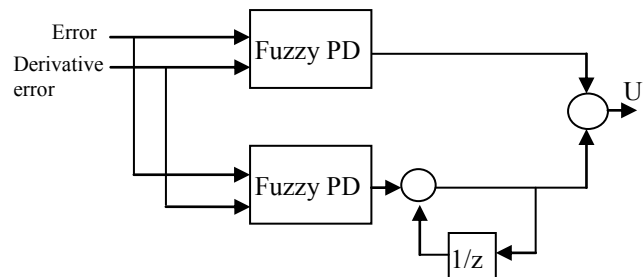
Dalam merancang kontroler dengan *behavior-based control* ditentukan rencana kontrol yang akan digunakan. Untuk tugas akhir ini, akan dirancang *Behavior Based Control* untuk *autonomous landing* sehingga *quadcopter* dapat melakukan *autonomous landing* dengan aman. Secara garis besar rencana kontrol yang akan diterapkan antara lain :

1. Menjaga kestabilan (*hovering*)
 - a. Mendeteksi kestabilan
 - b. Mengendalikan sudut *Roll, Pitch dan Roll*
2. Mengurangi ketinggian (*Landing*)
 - a. Mendeteksi ketinggian
 - b. Mengendalikan ketinggian

Kontroler Ketinggian

Untuk mengatur ketinggian *quadcopter*, dirancang kontroler Z dengan menggunakan *Intelligent Fuzzy Controller* untuk diterapkan dalam strategi pengendalian *Behavior_based Control*. *Intelligent Fuzzy Controller* yang dimaksud adalah FLC tipe PID (*Proportional, Integral & Derivative*). Karakteristik FLC yang digunakan adalah sebagai berikut.

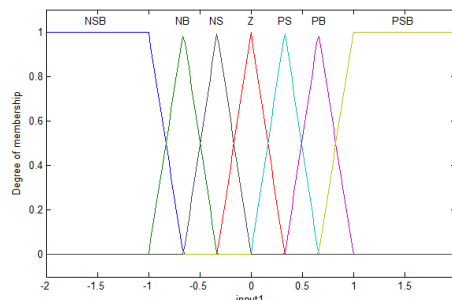
- Inferensi tipe mamdani
- Menggunakan *membership function* segitiga dan trapesium
- Menggunakan metode defuzzifikasi centroid
- Menggunakan operator AND minimum
- Menggunakan implikasi fungsi minimum
- *Tuning gain* untuk input dan output dilakukan dengan *trial dan error*



Gambar. 8. Diagram blok *Intelligent Fuzzy Controller*

Kontroler ini dirancang memiliki 4 input dan 1 output seperti pada Gambar 8. Dua input berupa *error* dan *derivative error* untuk menghasilkan sinyal kontrol PD (*Proportional Derivative*) dan dua input berupa *error* dan *derivative error* yang akan menghasilkan sinyal kontrol yang kemudian diintegrasikan untuk menghasilkan sinyal kontrol *Integral*. *Error* yang dimaksud adalah *error* ketinggian atau selisih *set point* dengan ketinggian sebenarnya yang terbaca oleh sensor *ultrasonic*.

Pada kontroler ini, himpunan *fuzzy* untuk input dan output menggunakan semesta pembicaraan [-1 1] dengan 7 *membership function* yang dinyatakan dalam linguistik NSB (*Negative Super Big*), NB (*Negative Big*), NS (*Negative Smal*), Z(*Zero*), PS (*Positive Smal*), PB (*Positive Big*), dan PSB (*Positive Super Big*) seperti pada Gambar 9



Gambar. 9. *Membership function*

Rule base pada kontroler ini menggunakan *rule base* Mack Vicar-Whelan seperti pada Gambar 3.18. *Rule base* ini memungkinkan untuk menciptakan respon dengan waktu relatif cepat dan tanpa *overshoot*. Karena jumlah *membership function* setiap input ada 7, maka menggunakan *rule base* 7 x 7 MackVicar-Whelan.

Tuning parameter dilakukan secara manual *Tuning* parameter kontroler dilakukan secara *trial & error* sampai didapatkan respon sistem yang terbaik dari kontroler ini. Hasil *tuning* dapat dilihat di Tabel 2.

Tabel 2.
Parameter *Intelligent Fuzzy Controller*

Kel	Kd1	Ke2	Kd2	K	Ki
5	300	2	15	4	0.001

Kontroller Roll, Pitch dan Yaw

Pengendalian sudut *roll*, *pitch* dan *yaw* dilakukan dengan menggunakan kontroler PID (*Proportional*, *Integral* dan *Derivative*). Tujuan pengendalian sudut-sudut ini adalah agar *quadcopter* tetap berada dalam kondisi seimbang selama proses *landing*. Sebelum melakukan perancangan kontroler PID, perlu dilakukan linearisasi pada model matematika untuk *roll*, *pitch* dan *yaw* sehingga didapatkan Persamaan 5, 6 dan 7.

$$\ddot{\phi} = \frac{U_2}{I_{xx}} \tag{5}$$

$$\ddot{\theta} = \frac{U_2}{I_{yy}} \tag{6}$$

$$\ddot{\Psi} = \frac{U_4}{I_{zz}} \tag{7}$$

Tuning parameter kontroler PID dari sistem ini dilakukan dengan *trial & error*. Parameter kontroler yang didapatkan setelah proses *tuning* dilakukan dapat dilihat di Tabel 3.

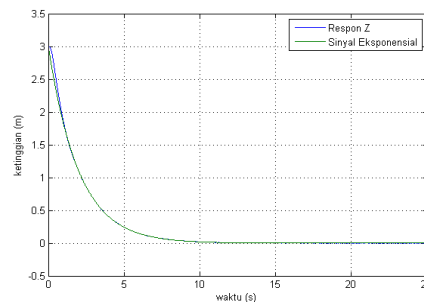
Tabel 3.
Tabel parameter kontroler PID

Parameter Kontroler PID – Roll		
K _p	K _i	K _d
8	1.9	1
Parameter Kontroler PID – Pitch		
K _p	K _i	K _d
8	1.9	1
Parameter Kontroler – Yaw		
K _p	K _i	K _d
8	1.9	2.05

IV. SIMULASI DAN ANALISIS

Simulasi ini dibuat menggunakan *software* MATLAB 2010 dengan menggunakan fasilitas SIMULINK. Simulasi yang dibuat meliputi simulasi translasi Z (ketinggian), sudut *roll*, sudut *pitch* dan sudut *yaw* terhadap model matematik dari quadrotor, sehingga dapat dilakukan analisa sebelum melakukan implementasi metode secara langsung.

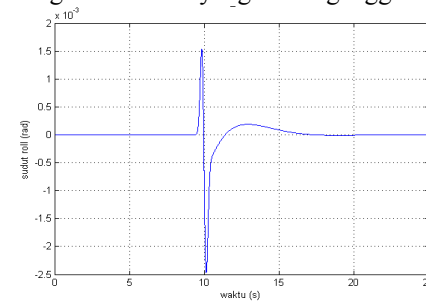
Pada simulasi ini, *Quadcopter* diberi kondisi awal untuk ketinggian (sumbu z) sebesar 3 meter dan untuk sudut *roll*, *pitch*, *yaw* sebesar 0 rad. U1 nominal diberikan sebagai gaya awal untuk melawan gaya gravitasi sebesar 9.81 m/s². U1 nominal ini diberikan agar *quadcopter* dapat menjaga ketinggian pada kondisi awal akibat adanya gravitasi bumi. Setelah penambahan U1, respon kondisi awal akan tetap di ketinggian 3 m. *Quadcopter* akan melakukan mekanisme landing dengan mengurangi ketinggian hingga mencapai titik nol (0) meter. Penurunan ketinggian tersebut dalam simulasi ini dilakukan dengan mengurangi gaya angkat U1 mengikuti sinyal eksponensial terbalik seperti yang ditunjukkan pada Gambar 11.



Gambar. 10. Respon ketinggian *quadcopter*

Dari Gambar 10, dapat dianalisis bahwa respon dapat mengikuti sinyal eksponensial, dan sampai di titik nol dalam waktu 15 detik. Pada respon terjadi riple ketika hampir mencapai titik nol (0) sebesar 3 mm. Dengan mengikuti sinyal eksponensial, kecepatan menurun *quadcopter* berkurang seiring dengan semakin dekatnya dengan titik 0 (nol). Hal ini untuk mengurangi momentum yang terjadi ketika *quadcopter* menyentuh tanah.

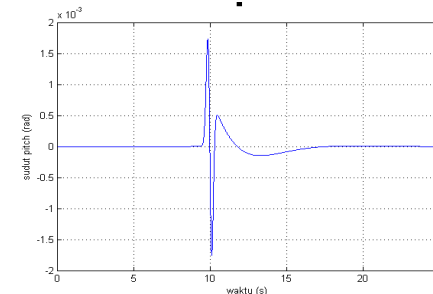
Untuk menguji kontroler PID yang telah dirancang untuk *roll*, *pitch* dan *yaw*, diberikan gangguan berupa sinyal gaussian dengan time peak 10 detik, lebar pulsa 0.2 dan tinggi puncak 0.2. Gambar 11 menunjukkan respon sudut *roll* dengan kontroler yang diberi gangguan.



Gambar. 11. Respon sudut *roll* dengan kontroler PID

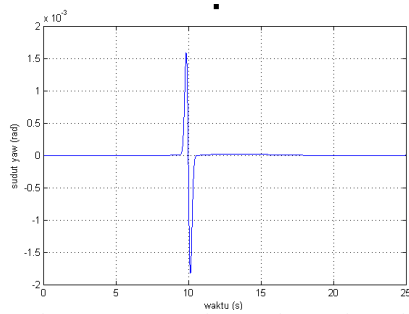
Dari respon pada Gambar 11 dapat dilihat bahwa kontroler PID dapat meredam gangguan dan mengembalikan nilai sudut *roll* ke 0 (nol) radian dalam waktu 7.97 detik sejak gangguan diberikan.

Dengan kontroler PID dan menggunakan parameter pada Tabel 3.6 maka diperoleh respon sudut *pitch* seperti pada Gambar 12.



Gambar. 12. Respon sudut *pitch* dengan kontroler PID

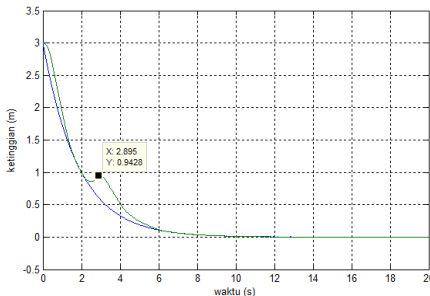
Dari respon tersebut kontroler PID mampu mengendalikan gangguan yang diberikan dan mengembalikan sudut ke nilai 0 (nol) radian dalam waktu 7.97 detik sejak gangguan diberikan. Respon sudut *yaw* dengan menggunakan kontroler PID seperti pada Gambar 13.



Gambar. 13. Respon sudut yaw dengan kontroler PID

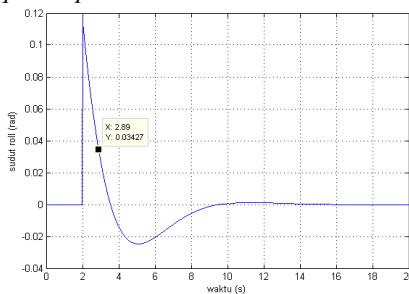
Dari respon pada Gambar 13 dapat dilihat bahwa kontroler PID dapat meredam gangguan dan mengembalikan nilai sudut *yaw* ke 0 (nol) radian dalam waktu 1.25 detik sejak gangguan diberikan.

Untuk menguji *Behavior-Based Control* yang dirancang, diberikan gangguan berupa sinyal step dengan *final value* 0.12. Respon translasi z setelah diberi gangguan pada salah satu sudut seperti pada Gambar 14



Gambar. 14. Respon translasi z setelah sudut *roll* diganggu sebesar 0.12 rad

Dari respon pada Gambar 14 dapat dilihat bahwa setelah detik ke-2 di mana gangguan diberikan, respon ketinggian cenderung meninggalkan sinyal referensi. Hal ini terjadi karena ketika *quadcopter* dianggap tidak stabil, maka sinyal kontrol yang digunakan adalah sinyal kontrol tepat sebelum *quadcopter* tidak stabil.



Gambar. 15 Respon sudut *roll* dengan gangguan 0.12 radian

Pada Gambar 15, ditunjukkan respon sudut *roll* ketika diberi gangguan sebesar 0.12 rad. Dari Gambar 15 dapat dilihat bahwa sudut *roll* kembali ke rentang stabil pada waktu 2.89 detik. Jika ditinjau kembali pada Gambar 14, maka terlihat setelah waktu 2.89 detik kontroler ketinggian kembali digunakan. Hal ini terlihat dengan respon kembali menajar nilai *set point*

V. KESIMPULAN

Dari penelitian Tugas Akhir yang telah dilaksanakan, dapat disimpulkan bahwa:

1. *Intelligent Fuzzy Controller* dapat mengendalikan translasi di sumbu z dengan mengikuti sinyal eksponensial terbalik dan mencapai titik nol dalam waktu 15 detik
2. *Behavior-Based Control* dapat bekerja dengan baik sehingga pengendalian translasi di sumbu Z hanya dilakukan ketika *quadcopter* dalam keadaan setimbang.
3. Kontroler PID dapat mengendalikan sudut *roll*, *pitch* dan *yaw* sehingga kembali ke nilai 0 (nol) radian dalam waktu 7.97 detik untuk *roll* dan *pitch* serta 1.25 detik untuk *yaw*.

DAFTAR PUSTAKA

- [1] Domingues, Jorge M.B. *Quadrotor Prototype*. Universidade Tecnica de Lisboa. Dissertacio. 2009
- [2] Bresciani, Tommaso. *Modelling, Identification and Control of a Quadrotor Helicopter*. Departement of Automatic Control Lund University. 2008.
- [3] Pitowarno, Endra. *Robotika: desain, kontrol, dan kecerdasan buatan*. Andi Offset. Yogyakarta, 2006.
- [4] Fitzpatrick, Paul. *Behavior-based Control in Mobile Robot*. University of Limerick lecture. Ireland.
- [5] Stenzel, R. *A behaviour-based control architecture*. IEEE international conference on systems, man, and cybernatics. 3235-3240 vol 5. 2000.
- [6] Brooks, A.R. *A Robust Layered Control System for A Mobile Robot*. IEEE jurnal of robotics and automation. 14-23 vol 2.1986
- [7] Christopher J.H. & Alan F.T.W. *A Methodology for Provably Stable Behavior-Based Intelligent Control*. ScienceDirect jurnal of Robotics and Autonomous System. 52-73 vol 54. 2006
- [8] Sri Kusumadewi & Hari Purnomo. *Aplikasi Logika Fuzzy : untuk Pendukung Keputusan*. Graha Ilmu. Yogyakarta. 2010
- [9] Santos, M. and Morata, F. *Intelligent fuzzy controller of a quadrotor*. IEEE International conference on intelligent system and knowledge engineering. 141-146. 2010
- [10] Gamayanti, Nurlita. *Diktat Mata Kuliah Dasar Sistem Pengaturan*. Teknik Sistem Pengaturan. Jurusan Teknik Elektro FTI-ITS. Surabaya. 2010