

# Perancangan dan Implementasi *Autonomous Landing* Menggunakan *Behavior-Based* dan *Fuzzy Controller* pada *Quadcopter*

Fadjri Andika Permadi, Rusdhianto Effendi AK, Ali Fatoni  
Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember (ITS)  
Jl. Arief Rahman Hakim, Surabaya 60111  
*E-mail: fatoni@ee.its.ac.id*

**Abstrak** -- Perkembangan teknologi sistem kendali pesawat sayap berputar (*copter*) semakin pesat salah satunya pada pesawat baling-baling empat (*quadcopter*). *Landing* merupakan bagian tersulit dalam penerbangan *quadcopter*. Ukuran *quadcopter* yang kecil mengakibatkan susahny pengendalian kestabilan dan kecepatan turun.

Cara mengatasi permasalahan ini adalah dengan *autonomous landing* yang menggunakan algoritma kendali *behavior-based* (berbasis perilaku). Tugas akhir ini merancang dan mengimplementasikan algoritma kendali *behavior-based* (berbasis perilaku) pada proses *autonomous landing quadcopter* dan kontroler PD (*Proporsional, Diferensial*) pada untuk kestabilan sudut *pitch* dan *roll*, sedangkan untuk jarak *landing* menggunakan kontroler logika *fuzzy*.

Pada Tugas Akhir ini, didapatkan nilai parameter kontroler PD *roll* dan kontroler PD *pitch* dari hasil *tuning* terstruktur pada simulasi  $K_p=500$  dan  $K_d=30$ . Sedangkan kendali *landing* menggunakan kontroler logika *fuzzy* dengan parameter  $K_e=4$   $K_{de}=175$  dan  $K_u=1$  pada simulasi dapat melakukan proses *landing* selama 8 detik dari ketinggian 3 meter. Respon hasil implementasi pada *quadcopter* belum sesuai dengan hasil simulasi. Proses *landing* pada implementasi lebih cepat dengan waktu 3.5 detik dari ketinggian 2 meter, selain itu koreksi sudut *roll* dan sudut *pitch* masih terhadap error  $\pm 3^\circ$ .

**Kata Kunci** : *quadcopter, autonomous landing, behavior base dan fuzzy controller.*

## I. PENDAHULUAN

Pada beberapa tahun terakhir, dunia penerbangan telah mengalami perkembangan yang sangat cepat. Banyak sekali penemuan-penemuan serta perkembangan yang terjadi di dalamnya. Salah satunya adalah dibuatnya kendaraan udara tak berawak. Kendaraan udara jenis inilah yang biasa disebut sebagai UAV (*Unmanned Aerial Vehicle*). Penggunaan UAV saat ini sangat dibutuhkan baik untuk keperluan militer maupun sipil misalnya untuk pencarian dan penyelamatan korban bencana alam serta penginderaan jarak jauh seperti monitoring hutan, monitoring lalu lintas dan keperluan monitoring daerah perbatasan.

UAV dapat digunakan untuk pekerjaan yang berbahaya sekalipun seperti memata-matai musuh pada saat perang atau

menjangkau daerah yang cukup berbahaya untuk misi penyelamatan. Karena ukurannya yang mini dan tidak mengeluarkan suara bising layaknya kendaraan udara lain seperti *quadcopter*, UAV dapat terbang menyatu dengan langit dan sulit untuk diketahui oleh manusia. Walaupun terbang dalam ketinggian yang rendah kemampuan untuk menyerupai binatang seperti serangga atau burung membuatnya sulit untuk diketahui. Oleh karena itu penggunaan UAV lebih banyak digunakan oleh departemen pertahanan dalam melakukan pengawasan wilayah.

Namun karena secara penuh dikontrol oleh pilot yang berada jauh dari *quadcopter*, seringkali menjadikannya tidak stabil sehingga rawan mengalami kecelakaan terbang. Secara garis besar ada tiga macam kategori tantangan yang dihadapi dalam pengembangan UAV antara lain: efisiensi aerodinamika, peningkatan pembebanan dan yang terpenting adalah masalah kontrol dan stabilitas. Secara garis besar fase penerbangan gerakan longitudinal pada *quadcopter* dibagi dalam 3 fase utama, yaitu *take off* (tinggal landas), *hovering* (melayang) , dan *landing* (pendaratan). Dari ketiga fase terbang tersebut, fase *landing* merupakan fase yang paling kritis di mana resiko terjadi kecelakaan adalah paling besar.

Penggunaan konsep *autopilot* pada proses *landing* akan menjadikannya kendaraan terbang otomatis yang bisa melakukan pendaratan dengan aman. Untuk proses *landing quadcopter* akan mengimplementasikan algoritma kendali *behavior-based*(berbasis-perilaku), yang diterapkan pada proses *landing* terdiri dari dua level yaitu *hovering* dan *decrease of the distance*. Masalah stabilisasi *hovering* pada *quadcopter* dapat diatasi dengan penggunaan kontroler *fuzzy*. Penggunaan algoritma *fuzzy* dilakukan untuk mengolah sinyal kesalahan yang mana digunakan untuk mengatur sudut *roll* dan *pitch* pada proses *landing*.

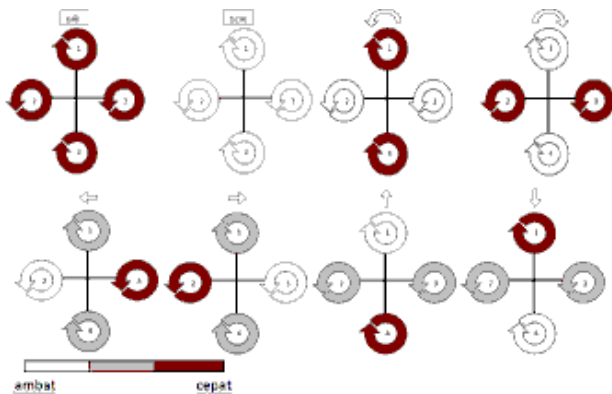
## II. DASAR TEORI

### A. *Quadcopter*[7]

*Quadcopter* memiliki 6 *degree of freedom (DoF)* dengan 12 state, 6 dari keluaran 12 state ini menentukan *attitude* dari *quadcopter*. *Quadcopter* memiliki 4 buah rotor sebagai penggerak baling-baling yang digunakan untuk menghasilkan gaya angkat. Baling-baling yang sering disebut dengan propeller yang dipasang bersama 4 buah brushless motor. Tipe dari *quadcopter* sendiri terdiri dari 2 jenis, tipe X dan tipe + tergantung pada gerak laju pada *quadcopter*. *Quadcopter*

memiliki *frame* dimana sebagai tulang tumpu pada koordinat x dan y.

Untuk bergerak naik, diperlukan kecepatan yang sama dan cukup besar pada keempat rotornya. Terlihat pada Gambar 2.1, pengaruh kecepatan rotor terhadap gerakan *quadcopter*.



Gambar 2. 1 Pergerakan dari *Quadcopter*

Pada Gambar 2.1 bagian depan *quadcopter* adalah rotor dengan nomor 1. Jika kecepatan motor 2 dan 3 dinaikan/ diturunkan maka akan terjadi gerakan rotasi yang dinamakan *roll* dan mengakibatkan gerak translasi pada sumbu y. Sebaliknya jika kecepatan motor 1 dan 4 dinaikan/ diturunkan maka akan terjadi gerakan rotasi yang dinamakan *pitch* dan mengakibatkan gerak translasi pada sumbu x. Pada Gambar 2.1 pergerakan *quadcopter* rotasi terhadap sumbu z dinamakan *yaw*, selain itu jika kecepatan keempat motor sama akan terjadi pergerakan translasi terhadap sumbu z.

Persamaan dinamika dari *quadcopter* pada gerak translasi dan rotasi dapat dituliskan pada Persamaan 2.1[1].

$$\ddot{x} = (\cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi) \frac{U_1}{m} \tag{2.1}$$

$$\ddot{y} = (\cos \phi \sin \theta \cos \psi - \sin \phi \sin \psi) \frac{U_1}{m} \tag{2.2}$$

$$\ddot{z} = -g + (\cos \phi \cos \theta) \frac{U_1}{m} \tag{2.3}$$

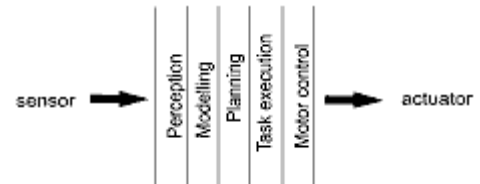
$$\ddot{\phi} = \frac{I_{yy} - I_{zz}}{I_{xx}} qr + \frac{Jr}{I_{xx}} q\Omega + \frac{U_2}{I_{xx}} \tag{2.4}$$

$$\ddot{\theta} = \frac{I_{zz} - I_{xx}}{I_{yy}} pr - \frac{Jr}{I_{yy}} p\Omega + \frac{U_3}{I_{yy}} \tag{2.5}$$

$$\ddot{\psi} = \frac{I_{xx} - I_{yy}}{I_{zz}} pq + \frac{U_4}{I_{zz}} \tag{2.6}$$

**B. Behavior Based Robotic[2]**

Pada sistem kendali robot, pendekatan yang biasa digunakan adalah dengan menguraikan setiap masalah kedalam rangkaian unit fungsional sebagaimana ditunjukkan pada Gambar 2.2.



Gambar 2.2 Teknik Penguraian Tradisional untuk Sistem Kendali *Mobile Robot* ke dalam unit-unit Fungsional [2]

Berbeda dengan pendekatan di atas, *behavior based robotic* mendesain sistem kendali robot menggunakan pendekatan *task achieving behaviors* (perilaku dalam menunaikan tugas) sebagaimana ditunjukkan oleh Gambar 2.3.



Gambar 2.3 Dekomposisi Sistem Kendali *Mobile Robot* dengan *Task Achieving Behaviors* [2]

Metode dekomposisi ini memiliki arsitektur *mobile robot* yang sangat berbeda dengan dekomposisi yang berdasarkan unit fungsional (Gambar 2.3). Berbeda secara *hardware*, dan sejumlah kelebihan lain seperti *robustness*, *buildability* dan *testability*.

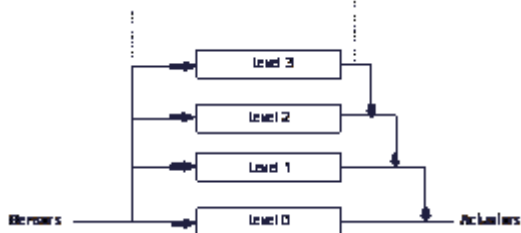
**Arsitektur Subsumption**

Arsitektur *subsumption* adalah struktur BBR yang diusulkan oleh Rodney Brooks [2]. Dalam membangun robotnya, Rodney Brooks menguraikan permasalahan sistem kendali robot sesuai dengan manifestasi luar yang diinginkan oleh sistem kendali robot, tidak berdasarkan pada operasi internal dari sistem kendali robot sebagaimana yang telah dilakukan oleh beberapa peneliti sebelumnya. Oleh karena itu, Brooks mendefinisikan sejumlah level kompetensi pada *mobile robot* mandiri. Level kompetensi adalah spesifikasi informal dari sekelompok perilaku yang diinginkan robot bekerja pada semua lingkungan yang akan dihadapi. Level kompeten yang lebih tinggi menunjukkan kelompok perilaku yang lebih khusus/spesifik.

Tiap level kompetensi memasukkan sub kelompok dari level kompetensi sebelumnya. Karena level kompetensi mendefinisikan kelompok perilaku yang valid, dapat dianggap bahwa level yang lebih tinggi memberikan tambahan batasan pada kelompok perilaku tersebut. Rodney Brooks memulai dengan membangun sistem kendali robot yang melaksanakan level kompetensi nol. Perbaikan kesalahan dilakukan dengan teliti. Brooks tidak pernah mengubah sistem ini dan menyebutnya sistem kendali level ke nol. Selanjutnya, dibangun lapisan kendali yang lain yang disebut sistem kendali level kesatu. Level ini dapat menguji data dari level nol dan juga memberikan data ke dalam *internal interface* level nol, serta menekan data normal yang mengalir.

Lapisan ini, dengan tambahan dari lapisan nol melaksanakan level kompetensi pertama. Lapisan ke nol melanjutkan untuk bekerja tanpa mengetahui lapisan di atasnya yang terkadang mengganggu aliran data. Proses yang

sama diulangi untuk mendapatkan level kompetensi yang lebih tinggi sebagaimana ditunjukkan oleh Gambar 2.4.



Gambar 2.4 Arsitektur *Subsumption*[2]

### III. PERANCANGAN SISTEM

#### A. Spesifikasi Sistem

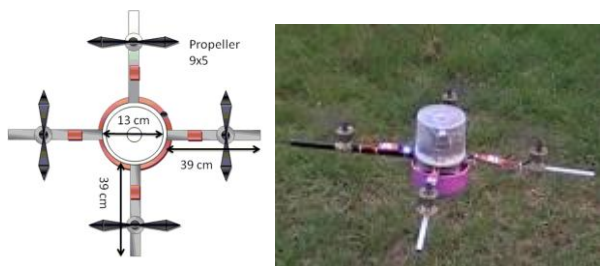
Kebutuhan sistem dalam penelitian ini adalah sebuah *quadcopter* RC sebagai *plant* yang akan dikontrol, *gyroscope* dan *accelerometer* sensor yang digunakan untuk pembacaan sudut *roll*, *pitch*, dan *yaw*, serta untuk mengukur ketinggian *quadcopter* akan digunakan sensor PING)). Seperangkat rangkaian mikrokontroler yang berfungsi sebagai kontroler digunakan untuk mengatur kestabilan terbang dan kecepatan masing-masing motor untuk proses landing. Untuk pembacaan data yang dikirimkan dari *quadcopter* melalui komunikasi wireless akan diterima oleh ground station.

#### B. Perancangan dan Implementasi Perangkat Keras

Perancangan perangkat keras pada tugas akhir ini terdiri dari dua bagian yaitu rancang bangun mekanik *quadcopter* dan desain sistem elektronik.

##### 1. Desain Mekanik

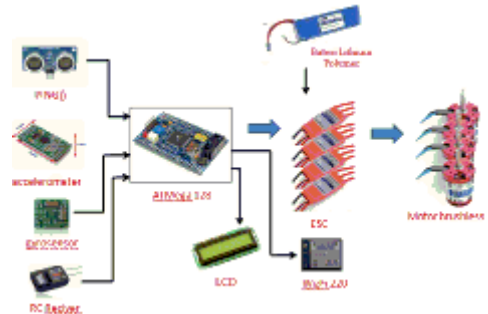
Pada pembuatan Tugas Akhir ini kebutuhan *quadcopter* model merupakan salah satu komponen utama. *Quadcopter* model merupakan komponen yang menjadi bagian utama sebagai *plant* yang akan dikendalikan dalam sistem autopilot yang akan dirancang. Adapun spesifikasi *quadcopter* model yang digunakan adalah jenis plush (+) dengan frame berdasar dari aluminium dan penyangga tengah frame terbuat dari *acrylic fiber*. Bahan dasar tersebut yang menjadikan *quadcopter* model ini lebih ringan dalam terbang dari pada pesawat model dengan bahan yang lain. Pada *quadcopter* model juga terdapat pengaman rangkaian dan penutup batrai juga berfungsi sebagai kaki *quadcopter* yang terbuat dari bahan plastik yang berbentuk *cylinder*. Spesifikasi dari *quadcopter* model terlihat pada Gambar 3.1.



Gambar 3.1 Rancangan Desain Mekanik *Quadcopter* dan Hasil Implementasi Rancangan Mekanik *Quadcopter*

##### 2. Desain Sistem Elektronik

Dengan mengacu kepada diagram blok sistem elektronik Gambar 3.2, terdiri dari bagian masukan, bagian kendali, bagian keluaran dan bagian catu daya (baterai). Pada bagian masukan berupa satu buah sensor ultrasonik (PING)), satu buah sensor *accelerometer* dan *gyrosensor* berfungsi untuk membaca sudut dan percepatan sudut kemiringan *quadcopter*. Pada bagian kendali menggunakan mikrokontroler Atmega128. Pada bagian keluaran berupa penampil LCD 2\*16, wifizi 220 untuk pengiriman data, empat buah ESC /driver motor sebagai penggerak empat aktuator (*brushless motor*).



Gambar 3.2 Diagram Blok Sistem Elektronik

#### C. Desain Autonomous Landing dengan Behavior-Based

Aksi-aksi yang diperlukan *quadcopter* dalam proses *autonomous landing* adalah:

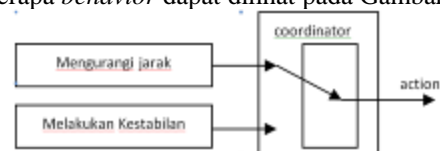
1. Melakukan proses kestabilan *quadcopter*.
2. Mengurangi delta kecepatan ke empat motor untuk menurunkan jarak.

Gambar 3.3 merupakan rancangan behavior based pada proses landing *quadcopter*.



Gambar 3.3 Rancangan Behavior-based pada *Quadcopter*

Untuk menghubungkan dua *behavior* (level-0 dan level-1), ditambahkan satu bagian koordinator *arbitrase*. Fungsinya adalah menentukan siapa yang akan memegang kendali aktuator serta menentukan aksi dari *quadcopter*. *Arbitrase* bertugas mengatur kapan perilaku gerakan *houversing* (stabilisasi) mengambil alih kendali dari perilaku mengurangi kecepatan keempat motor (jarak) yang ada di level sebelumnya. Koordinator *arbitrase* ini menggunakan *competitive coordinator*, pengambilan keputusan pada masing-masing *behavior* dapat diatur melalui level kompetensinya, serta antar sesama *behavior* tidak dapat saling mempengaruhi. Hasil rancangan koordinator kompetitif terhadap beberapa *behavior* dapat dilihat pada Gambar 3.4.

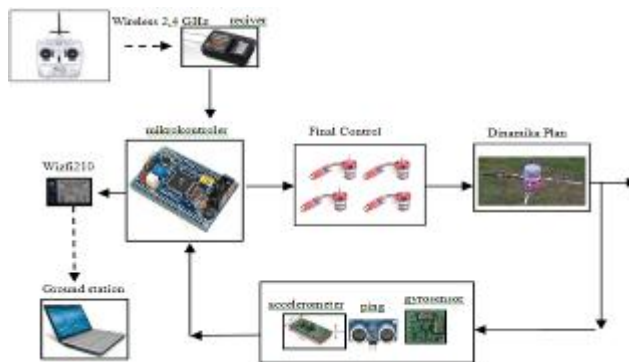


Gambar 3.4 Rancangan Arbitrasecompetitive Coordinator

Pada awalnya pada saat saklar autolanding aktif, maka proses mengurangi kecepatan/ jarak berlangsung jika terjadi gangguan yang mengakibatkan posisi kestabilan dari *quadcopter* terganggu maka *coordinator arbitrase* menjalankan aksi stabilisasi posisi *quadcopter* dengan hover pada ketinggian tersebut, setelah posisi stabil kembali *coordinator* akan kembali memilih aksi penurunan kecepatan/jarak.

**D. Perancangan Kontroler pada Quadcopter**

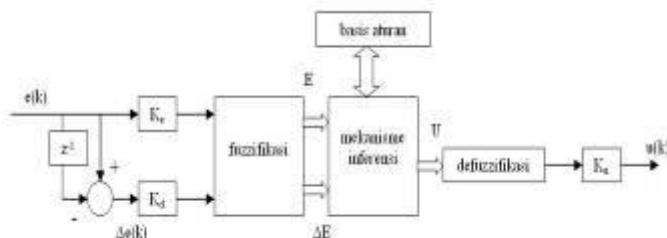
Pada penelitian ini menggunakan kontroler fuzzy PD untuk mengendalikan ketinggian (sumbu z), sedangkan sudut pitch dan roll dikendalikan menggunakan kontroler PD (propotional difrensial). Setiap variable memiliki nilai variable terukur (PV). *Present Value* (PV) adalah nilai pembacaan sensor saat itu atau variabel terukur yang di umpan balik oleh sensor (sinyal *feedback*). Deviasi/simpangan antar variabel terukur (PV) dengan nilai acuan (SP) disebut error (galat). *Setpoint* (SP) adalah suatu prameter nilai acuan atau nilai yang diinginkan. Diagram blok sistem kendali *quadcopter* ditunjukkan pada Gambar 3.5.



Gambar 3. 5 Diagram Blok Sistem Kontrol *Quadcopter*

**1. Kontroler Fuzzy ketinggian quadcopter**

Perancangan dan pembuatan kontroler logika fuzzy pada prinsipnya menggunakan teorema fuzzy logic secara keseluruhan. Secara rinci, perancangan perangkat lunak terdiri dari proses kuantisasi, fuzzifikasi, inference, dan defuzzifikasi. Gambar 3.6 adalah blok dari perancangan control fuzzy.



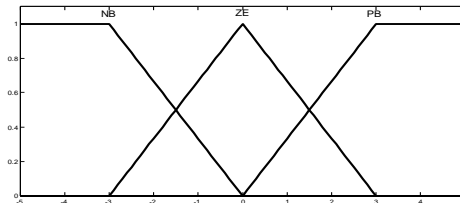
Gambar 3. 6 Blok Kontrol Fuzzy Ketinggian *Quadcopter*

Masukkan diproses oleh mikrokontroler untuk mendapatkan aksi kontrol untuk mengendalikan plan sesuai dengan set point yang diinginkan. Setpoint yang diharapkan akan dibandingkan dengan ketinggian sebenarnya pada plant. Sehingga diperoleh selisih yang berupa error, sinyal error ini nantinya akan oleh kontroler fuzzy PD untuk menentukan aksi kontrol.

Tahap perancangan kontroler logika fuzzy adalah sebagai berikut:

- a) Menentukan derajat fungsi keanggotaan

Pada perancangan ini digunakan 3 variabel linguistik, yaitu NB (*negative big*), ZE(*zero*) dan PB (*positif big*). *Setpoint* yang dirancang adalah sesuai kemampuan sensor mengukur jarak, yaitu 3 meter, maka error yang akan terjadi diperkirakan antara (-3 , 0 , 3). Dan besaran d-error yang dirancang sama seperti error yaitu antara (-3 , 0 , 3). Himpunan fuzzy yang digunakan terlihat seperti pada Gambar 3.7.



Gambar 3. 7 Fuzifikasi Error dan Derror

Nilai masukan error dan derror akan dikonversi ke variable linguistic sebanyak keanggotaan fuzzy.

- b) Menentukan fungsi keanggotaan untuk variable keluaran.

Variable keluaran dalam tugas akhir ini adalah lebar pulsa masukan ESC motor yang mana range pulse 40 untuk kecepatan minimum dan 100 untuk kecepatan maximum. Untuk fungsi keanggotaan fuzzy keluaran nilainya seperti fungsi keanggotaan eror dan deror yaitu (-3,0,3).

- c) Menentukan Rule Base

Setelah dari fuzifikasi, masuk ke rule base yang mana sinyal kontrol akan di tentukan oleh rule base dengan bantuan inference. Membuat aturan dasar fuzzy yang digunakan untuk menentukan sinyal kontrol berupa lebar pulsa motor. Aturan kontroler fuzzy ditunjukkan pada Tabel 3.4. Cara pembacaan dari rule base adalah IF <fungsi keanggotaan 1> AND error <fungsi keanggotaan 2> THEN lebar pulsa injeksi <fungsi keanggotaan 3> .

**Tabel 3.1** Rule base Kontrol Logika Fuzzy Ketinggian *Quadcopter*

Error \ Derror	NB	ZE	PB
NB	NB	NB	ZE
ZE	NB	ZE	PB
PB	ZE	PB	PB

Inference yang digunakan adalah inference metode mamdani yang mana memiliki Persamaan 3.33

$$\mu_y(k) = \max [\min \{\mu_u(k), \mu_R(E(i), \Delta E(j))\}] \quad (3.33)$$

- d) Untuk mendapatkan nilai aksi kontrol (u) perlu dilakukan proses defuzzifikasi, dalam hal ini dipilih defuzzifikasi dengan metode COA (*center of area*).

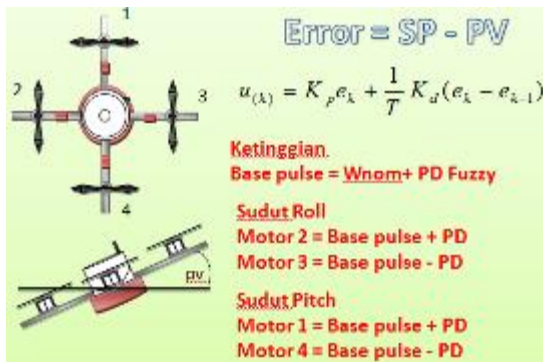
Proses defuzzifikasi ini berdasarkan dari sinyal masukan berupa kecepatan referensi dan error untuk menghasilkan sinyal kontrol lebar pulsa kecepatan motor dalam nilai angka.

- e) Tuning eksperimen mencari parameter gain error (K<sub>e</sub>), gain deltaerror (K<sub>d</sub>) dan gain kontroler (K<sub>u</sub>).

2. *Kontroler PD kestabilan quadcopter.*

Pembuatan program kontroler PD ini dilakukan berdasarkan persamaan kontroler PD digital. Pertama-tama akan dibuat dua buah variabel berupa *error*, *last\_error*. Gunanya adalah untuk menyimpan data *error* dan *last\_error* yang akan digunakan pada perhitungan aksi kontroler PD.

Setiap satu kali *looping* program, *error* akan diperbaharui dengan data yang diambil dari sensor, dan sebelumnya akan disimpan di *last\_error*. Keluaran dari perhitungan program kontroler PD ini adalah nilai pulsa ECS. Nilai pulsa ini dapat bernilai positif ataupun negatif. Untuk kestabilan sudut roll, keluaran dari kontrol PD akan masuk ke motor 2 dan motor 3 yang mana antara motor 2 dan motor 3 sifatnya bertolak belakang. Begitu juga untuk kestabilan sudut pitch, yang bertolak belakang antara motor 1 dengan motor 4. Ilustrasi aplikasi kontroler PD dapat dijelaskan pada diagram Gambar 3.8.



Gambar 3. 7 Ilustrasi Kontroler PID pada *Quadcopter*

IV. PENGUJIAN DAN ANALISA

A. *Analisis Kontroler pada Quadcopter*

1. Hasil simulasi

Pada pengendalian kestabilan pada *quadcopter*, kontroler yang digunakan adalah kontroler PD untuk mengendalikan sudut pitch dan roll pada set point 0. Analisis yang dilakukan pada kontroler PD pitch dan roll ialah dipusatkan pada proses penentuan parameter kontroler PD yang tepat. Pada umumnya, ada dua cara yang dilakukan untuk mendapatkan parameter tersebut, yang pertama adalah dengan cara *tuning* analitik dan *tuning* eksperimen. Pada *quadcopter* sendiri, *tuning* analitik tidak bisa dilakukan karena kondisi plant yang terdiri dari banyak masukan dan banyak keluaran (MIMO) jadi proses *tuning* parameter difokuskan ke *tuning* eksperimen. Cara *tuning* eksperimen parameter pada *quadcopter* diadopsi dari [5].

Tabel 4.1, Tabel 4.2 merupakan hasil *tuning* nilai-nilai parameter kontroler PD pada *quadcopter* dengan acuan pengujian ini adalah *quadcopter* diberi *setpoint* 0 dengan initial condition sudut 1 rad lalu diamati respon masing-masing parameter kontroler PD yang menyebabkan mampu menjaga dan mempertahankan sudut ke posisi 0.

Tabel 4. 1 Pengujian kontroler proporsional

Kp	Kd	Respon Sudut Simulasi
10	0	Respon tidak stabil dan osilasi sangat tinggi
250	0	Respon terlihat cepat, awal osilasi tinggi, semakin lama osilasi mengecil
<b>500</b>	<b>0</b>	<b>Respon terlihat lebih cepat, osilasi awal osilasi tinggi dan lama-lama mengecil</b>

Tabel 4. 2 Pengujian kontroler proporsional dan diferensial

Kp	Kd	Respon Sudut Simulasi
500	5	Respon cepat, masih ada osilasi tinggi
<b>500</b>	<b>30</b>	<b>Respon cepat, stabil, tanpa osilasi</b>
500	50	Respon agak melambat tanpa osilasi

*Simulasi Kontrol Ketinggian (sumbu Z)*

Tabel 4.3, Tabel 4.4 merupakan hasil *tuning* nilai-nilai parameter kontroler fuzzy pada ketinggian *quadcopter* dengan acuan pengujian ini adalah *quadcopter* diberi *setpoint* 0 dengan initial condition sudut 3 meter lalu diamati respon masing-masing parameter kontroler fuzzy yang menyebabkan mampu menuju posisi 0 meter.

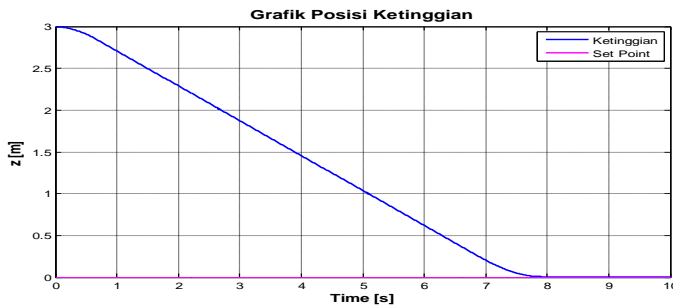
Tabel 4. 3 Tuning parameter Ku kontroler Fuzzy

Ke	Kde	Ku	Respon Ketinggian Simulasi
1	1	1	Respon lambat, tidak stabil dan osilasi sangat tinggi
3	1	1	Respon agak cepat, tidak stabil dan osilasi sangat tinggi
<b>4</b>	<b>1</b>	<b>1</b>	<b>Respon agak cepat, tidak stabil dan osilasi sangat tinggi</b>

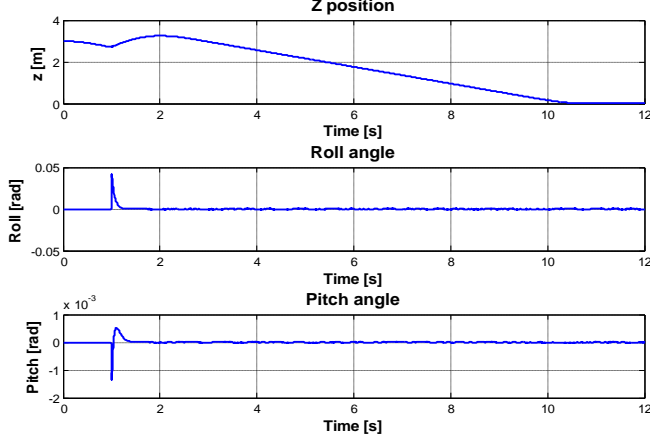
Tabel 4. 4 Tuning parameter Kde kontroler Fuzzy

Ke	Kde	Ku	Respon Ketinggian Simulasi
4	100	1	Respon agak cepat, masih ada osilasi
4	150	1	Respon agak cepat, masih ada osilasi tapi agak berkurang
<b>4</b>	<b>175</b>	<b>1</b>	<b>Respon agak cepat, stabil, tanpa osilasi</b>

Dari *tuning* eksperimen gain diferensial eror (Kde) semakin besar gain Kde maka semakin lambat respon, pilih gain Kd dimana menghasilkan lama respon dari *initial condition* ke *set point* yang sesuai dengan kebutuhan, tanpa *overshoot* dan stabil. Pada hasil *tuning* parameter yang cocok adalah nilai Ku=1, Ke=4 dan Kde=175. Berikut respon *landing* menggunakan fuzzy kontroler dengan parameter Ku=1, Ke=4 dan Kde=175 *initial condition* 3 meter ditampilkan pada Gambar 4.1.



Gambar 4. 1 Respon Ketinggian dengan Kontroler Fuzzy,  $K_u=10$ ,  $K_d=50$  dengan *Initial Condition* =3 meter

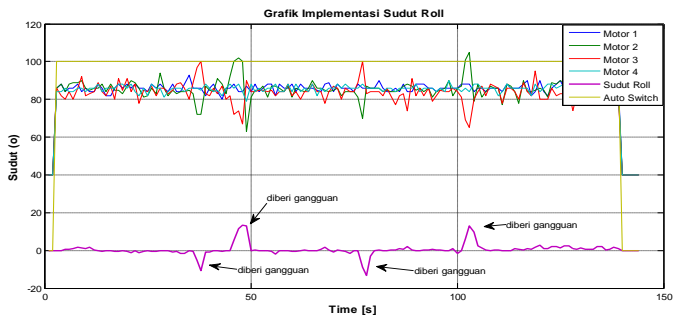


Gambar 4. 2 Respon Ketinggian dengan Kontroler Fuzzy,  $K_u=10$ ,  $K_d=50$  pada saat diberi Gangguan Sudut Roll

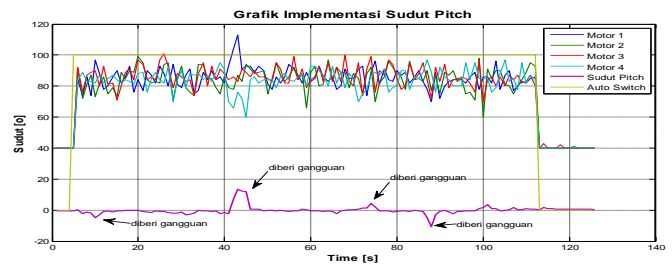
Dapat dilihat pada respon ketinggian Gambar 4.1, dengan parameter fuzzy yang telah di tuning waktu lama mendarat tanpa gangguan selama 8 detik, sedangkan pada saat diberi gangguan pada sudut roll sebesar 0.04 rad yang ditampilkan pada Gambar 4.2 maka behaviornya akan aktif, coordinator akan memilih mengaktifkan kontroler roll dan pitch akan bekerja mengendalikan ke set point=0. Sedangkan kontroler ketinggian tidak bekerja seperti yang ditampilkan pada Gambar 4.2.

2. Hasil Implementasi kontroler

Implemen kontroler roll dan pitch dilaksanakan menggunakan dudukan frame yang bebas salah satu sumbu rotasinya. Implementasi parameter kontroler PD pada kendali sudut roll dan pitch sama,  $K_p=2.5$  dan  $K_d=12$ . Hasil dari respon sudut roll dan pitch dengan adanya gangguan ditunjukkan pada Gambar 4.3 dan 4.4.

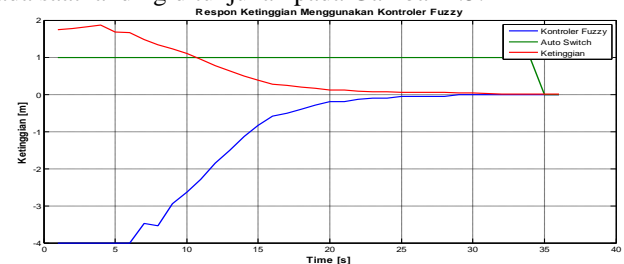


Gambar 4. 3 Respon Implementasi Sudut Roll



Gambar 4. 4 Respon Implementasi Sudut Pitch

Pada plan ketinggian menggunakan kontroler fuzzy dimana parameter kontroler fuzzy yang dipasang sesuai dengan tuning eksperimen adalah  $K_e=1$ ,  $K_d=1$  dan  $K_u=0.05$ . Hasil implementasi kontroler fuzzy terhadap plan ketinggian pada saat landing ditunjukkan pada Gambar 4.5.



Gambar 4. 5 Grafik Respon Ketinggian dengan Kontroler Fuzzy

V. KESIMPULAN/RINGKASAN

Dari hasil penelitian yang telah dilakukan pada pengerjaan tugas akhir ini, maka dapat diperoleh beberapa simpulan diantaranya:

1. Untuk masalah *landing*, dengan penambahan kontroler PD Fuzzy pada ketinggian *quadcopter* dapat melakukan dengan lama respon menuju set point 0 meter. Dari simulasi dapat kita lihat dengan parameter  $K_e=4$   $K_d=175$  dan  $K_u=1$  lama respon landing dari initial kondisi 3 menuju 0 selama 8 detik.
2. Kontrol *behavior-based* pada *autonomous landing* pada *quadcopter* sudah dapat mengatasi problem *landing* yang sering dihadapi. Jika mendapat gangguan pada sudut roll atau pitch melebihi 0.5 radian kontrol *behavior-based* tidak dapat mengatasinya.

DAFTAR PUSTAKA

- [1] Bresciani, Tommaso. 2008. *Modelling, Identification and Control of a Quadrotor Helicopter*. Department of Automatic Control, Lund University.
- [2] Brooks, R. (1986). "A robust layered control system for a mobile robot", IEEE Journal of Robotics and Automation Vol. 2, No. 1, hal.14–23.
- [3] Fahmizal, "Implementasi Sistem Navigasi Behavior Base dan Kontroler PID pada Manuver Robot Maze", Tugas Akhir, Institut Teknologi Sepuluh Nopember, Surabaya, 2011.
- [4] Pirjanian, P. (1999). "Behavior coordination mechanisms—State-of-the-art", Technical Report IRIS (Institute of Robotics and Intelligent Systems), University of Southern California, hal 99- 375.
- [5] William, C. (2006). "Feedback and Temperature Control" diakses pada 21 Desember 2010, Tuning a PID Temperature Controller: <http://newton.ex.ac.uk/teaching/CDHW/feedback/set-up-PID.html>.