

Sistem Kendali *Gimbal* 2-Sumbu Sebagai Tempat Kamera Pada *Quadrotor* Menggunakan PID Fuzzy

M. Saiful Haris*¹, Andi Dharmawan², Catur Atmaji³

¹Program Studi S1 Elektronika dan Instrumentasi, FMIPA UGM, Yogyakarta, Indonesia

^{2,3}Departemen Ilmu Komputer dan Elektronika, FMIPA UGM, Yogyakarta, Indonesia

e-mail: *saifulharis911@gmail.com, dharmawan.andi@gmail.com, catur_atmaji@ugm.ac.id

Abstrak

Sistem kendali pada camera gimbal pada penelitian ini berfungsi untuk mengatasi perubahan sudut kemiringan yang terjadi akibat manuver pada quadrotor. Sistem kendali PID dengan metode classical tuning memiliki kelemahan, salah satunya nilai konstanta PID yang tidak dapat secara mandiri menyesuaikan dengan lingkungannya, sehingga diusulkan menggunakan kendali PID fuzzy.

Camera gimbal yang digunakan pada penelitian ini memiliki rancangan mekanik dengan dua buah joint (pitch dan roll) dan motor BLDC sebagai aktuatornya. Perubahan sudut yang terjadi pada sumbu pitch dan roll menjadi umpan balik sistem. Kemudian, logika fuzzy akan menala konstanta PID berdasarkan umpan balik tersebut.

Hasil pengujian sistem pada camera gimbal 2-sumbu menunjukkan kendali PID fuzzy menghasilkan respon yang lebih baik dalam parameter risetime, overshoot, dan settlingtime dibandingkan dengan kendali PID. Rentang nilai masukan error sebesar $[-30^\circ 30^\circ]$ dan delta error sebesar $[-10^\circ 10^\circ]$ pada sumbu pitch dan roll. Rentang nilai keluaran untuk sumbu pitch yaitu, K_p sebesar $[40,2 46,2]$, K_i sebesar $[10,7 20,7]$, dan K_d sebesar $[0,05 0,15]$. Rentang nilai keluaran untuk sumbu roll yaitu, K_p sebesar $[6,4 16,4]$, K_i sebesar $[17,3 27,3]$, dan K_d sebesar $[0,08 0,16]$. Diperoleh kecepatan respon pada sumbu pitch sebesar 0,12 detik dan pada sumbu roll sebesar 1,07 detik.

Kata kunci— BLDC, pitch, roll

Abstract

The function of camera gimbal control system that use in this research is to serves with the angle changes that occur due quadrotor maneuver. The PID control with tuning classical method has weakness, which is the PID variable not independently adjust to the environment, thus proposed using PID fuzzy control.

Gimbal camera used in this study has a mechanical design with two joint (pitch and roll) and the BLDC motor as actuator. The angle changes that occur in the pitch and roll axis will be a feedback system. Then, fuzzy logic will tune the PID variable based on that feedback.

Results of testing the system on 2-axis gimbal camera shows the PID fuzzy control generates better response in parameter risetime, overshoot, and settlingtime compared with PID control. Error input value range of $[-30^\circ 30^\circ]$ and delta error of $[-10^\circ 10^\circ]$ on the pitch and roll axes. The range of the output value for the pitch axis is, K_p at $[40.2 46.2]$, K_i at $[10.7 20.7]$, and K_d of $[0.05 to 0.15]$. The range of the output value for the roll axis is, K_p at $[6.4 16.4]$, K_i at $[17.3 to 27.3]$, and K_d at $[0.08 0.16]$. Speed response speed of pitch axis is 0.12 second and the roll axis is 1.07 seconds.

Keywords— BLDC, pitch, roll

1. PENDAHULUAN

Camera *gimbal* adalah alat penstabil kamera, yang berfungsi untuk menghasilkan gambar atau video tanpa terpengaruh oleh kemiringan atau guncangan yang terjadi ketika pengambilan gambar atau video berlangsung. Memanfaatkan kerja *gimbal* yang dapat bergerak bebas pada porosnya mengakibatkan *camera gimbal* dapat menyesuaikan orientasi pergerakan ketika terjadi perubahan, sehingga kamera yang ditempatkan pada *camera gimbal* akan tetap stabil pada tempatnya.

Pada perkembangannya, *camera gimbal* dapat diaplikasikan pada *quadrotor*. *Quadrotor* merupakan pesawat tanpa awak (*Unmanned Aerial Vehicle*) yang memiliki 4 baling-baling dengan 4 buah rotor sebagai alat penggerakannya. *Camera gimbal* pada *quadrotor* digunakan untuk berbagai macam aplikasi di bidang fotografi udara yaitu, pelacakan target, pemetaan, navigasi otonomus dan lainnya [1].

Pada saat *quadrotor* melakukan manuver, maka keseluruhan dari *body quadrotor* akan berubah kemiringannya termasuk dengan *camera gimbal* yang ditempatkan pada *quadrotor*. Perubahan kemiringan yang terjadi pada *quadrotor* akan menyebabkan *camera gimbal* tidak dapat mempertahankan orientasinya, sehingga akan sangat mempengaruhi proses pengambilan gambar atau video. Oleh karena itu, ketika *camera gimbal* dipasangkan pada *quadrotor* diperlukanlah sebuah sistem kendali untuk *camera gimbal*. Sistem kendali yang diharapkan adalah sistem kendali yang mampu mengatasi perubahan kemiringan yang terjadi dari *quadrotor*.

Salah satu metode kendali yang umum digunakan adalah Kendali PID. Kendali PID memiliki 3 buah *gain* yang bekerja berdasarkan *error* sistem, yaitu Konstanta proporsional (K_p), konstanta integratif (K_i), dan konstanta derivatif (K_d) [2]. Sebuah sistem dapat memiliki respon yang stabil jika dilakukan penalaan dengan menghasilkan konstanta K_p , K_i , dan K_d yang tepat. Salah satu metode umum yang digunakan untuk penalaan K_p , K_i , dan K_d adalah *classical tuning* [3].

Metode penalaan *classical tuning* memiliki kelemahan, salah satunya nilai konstanta K_p , K_i , dan K_d tidak dapat secara mandiri menyesuaikan dengan lingkungannya [4]. Apabila sistem kendali PID dibuat sangat sensitif, maka respon sistem terhadap gangguan menghasilkan *overshoot* besar yang mengakibatkan terjadinya osilasi yang semakin besar. Bila dibuat kurang sensitif, maka akan menghasilkan *overshoot* kecil, tetapi akan memperbesar *risetime*. Oleh karena itu, dibutuhkan suatu kendali yang dapat secara mandiri menala konstanta PID yang menyesuaikan dengan lingkungannya. Salah satu kendali yang dapat menangani masalah tersebut adalah dengan kendali PID *fuzzy*.

Sistem kendali PID *Fuzzy* (*A Fuzzy PID Controller*) menggunakan kontroler PID klasik sebagai pondasi dimana digunakan penalaran *fuzzy* dan variabel ruang input untuk meregulasi *gain* dari PID [5].

2. METODE PENELITIAN

2.1 Analisis Sistem

Gimbal memiliki *joint* untuk menunjang pergerakan pada sebuah sumbu. Pada setiap *joint* terdapat motor sebagai aktuatornya. Konfigurasi *gimbal* dapat bervariasi berdasarkan jumlah sumbu geraknya. Agar dapat mempertahankan orientasi, tiap motor yang terdapat pada *joint* harus dikendalikan antara sudut yang terjadi dengan gerakan motornya, sehingga dibutuhkan kendali untuk mempertahankan orientasi. Ada beberapa parameter yang

mempengaruhi dalam memepertakan orientasi *gimbal* yaitu, gerak motor dan koordinat *body frame* yang ditempatkan kamera atau *end effector*.

Motor yang digunakan sebagai aktuator pada *gimbal* 2-sumbu penelitian ini adalah jenis motor *brushless* DC. Motor *brushless* DC memiliki 3 buah masukan dengan konfigurasi kumparan stator delta. Tiap masukan pada motor *brushless* DC mengatur nilai PWM berbentuk sinusoidal dengan beda fase tiap masukan sebesar 120° yang akan mempengaruhi medan magnet pada stator, sehingga rotor akan bergerak mengikuti perubahan medan magnet yang terjadi pada stator. Jika nilai perubahan PWM yang menyerupai bentuk sinusoidal berlawanan arah jarum jam (*counter clockwise*) maka motor *brushless* DC juga akan bergerak *counter clockwise*. Sebaliknya, jika perubahan sinusoidal PWM searah jarum jam (*clockwise*) maka motor *brushless* DC juga akan bergerak *clockwise*. Perubahan sudut motor dengan nilai perubahan nilai PWM tidak linier, sehingga perlu adanya sebuah sensor sudut sebagai umpan balik untuk dapat bergerak mendekati atau sampai kepada sudut yang diinginkan.

Penempatan kamera pada *gimbal* biasanya ditempatkan pada *end effector* atau *body frame* paling akhir, dalam sistem ini ditempatkan pada *body frame* (2). Karena dalam sistem ini yang ingin dikendalikan adalah orientasi kamera, maka harus ada hubungan antara koordinat kamera dengan sudut *gimbal* yang dihasilkan. maka sensor IMU perlu diletakkan pada *body frame* (2) untuk mengetahui kondisi orientasi kamera. Pada sistem ini jenis *gimbal* yang digunakan adalah *gimbal* 2-sumbu, yang memiliki pergerakan pada rotasi *roll* dan *pitch*. Sensor IMU digunakan untuk membaca nilai dari sudut orientasi *roll* dan *pitch*. Pembacaan sensor IMU dalam sistem ini menggunakan metode *fusion* DMP. Perhitungan dari *accelerometer* dan *gyroscope* akan menghasilkan nilai sudut orientasi *roll* dan *pitch* dengan menggunakan *library* MPU6050.

Rentang nilai yang dapat terjadi pada pergerakan sumbu *pitch* dan *roll* adalah sebesar 30° *clockwise* dan *counter clockwise*, karena *gimbal* akan dipasangkan pada *quadrotor* yang memiliki batas kemiringan sudut sebesar 30° *clockwise* dan *counter clockwise*. Nilai tersebut yang kemudian akan menjadi rentang nilai himpunan masukan *fuzzy* pada sistem kendali *gimbal* 2-sumbu.

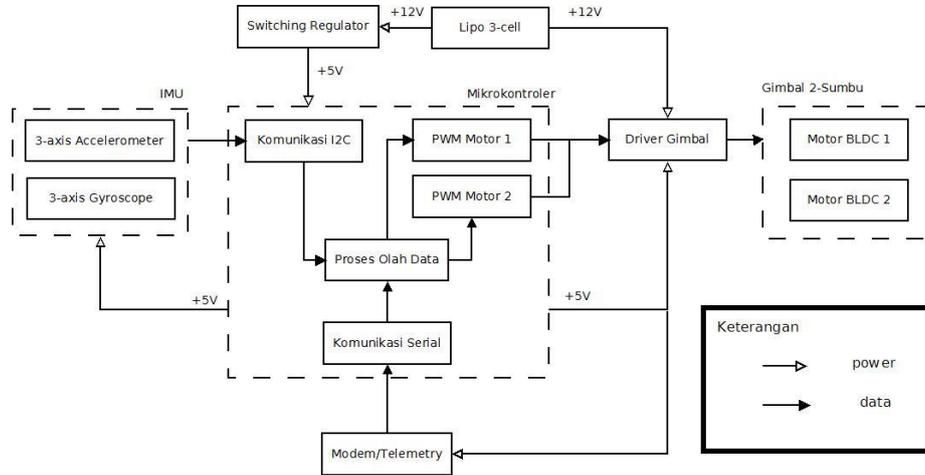
Terdapat beberapa parameter acuan yang dapat dijadikan acuan untuk menentukan tingkat keberhasilan dari penelitian ini. Parameter keberhasilan dari sistem kendali *gimbal* 2-sumbu ini dapat diketahui dari beberapa kondisi yang disebutkan sebagai berikut:

1. Nilai *steady state error* dibawah batas toleransi 5% dari nilai maksimal [2], yaitu sebesar 30° *clockwise* dan *counter clockwise* untuk sudut *roll* dan *pitch* yang dikendalikan, sehingga batas toleransi sudut adalah $1,5^\circ$ *clockwise* dan *counter clockwise* untuk sudut *pitch* dan *roll*.
2. *Risetime* yang dihasilkan untuk sudut *pitch* lebih cepat atau sama dengan 1 detik dan *roll* lebih cepat atau sama dengan 2 detik [6].
3. *Settling time* yang dihasilkan untuk sudut *pitch* lebih cepat dari 2 detik dan *roll* lebih cepat dari 3 detik [6].
4. Maximum *overshoot* untuk sudut *pitch* dan *roll* yang dihasilkan diantara 5° *clockwise* dan *counter clockwise* untuk sudut *pitch* dan 10° *clockwise* dan *counter clockwise* untuk sudut *roll* [7].

2.2 Arsitektur Sistem

Rancangan arsitektur sistem yang digunakan pada penelitian ini secara keseluruhan digambarkan pada Gambar 1. Komponen sensor yang digunakan adalah sensor *inertial measurement unit* (IMU) GY-521. Sensor IMU GY-521 merupakan modul yang berisikan *accelerometer* dan *gyroscope*. Pembacaan Sensor IMU yang digunakan pada sistem ini menggunakan metode sensor *fusion* DMP. Metode sensor *fusion* DMP adalah dengan cara menggabungkan pembacaan dari *accelerometer* dan *gyroscope* pada Sensor IMU untuk

menghasilkan nilai orientasi *roll* dan *pitch* dengan menggunakan *library* MPU6050. Hasil pembacaan dari Sensor IMU kemudian akan diolah mikrokontroler untuk menentukan posisi dari kamera yang ditempatkan pada *gimbal*, sehingga dapat dilakukan kendali pada *gimbal* agar dapat mengatasi perubahan orientasi yang terjadi.



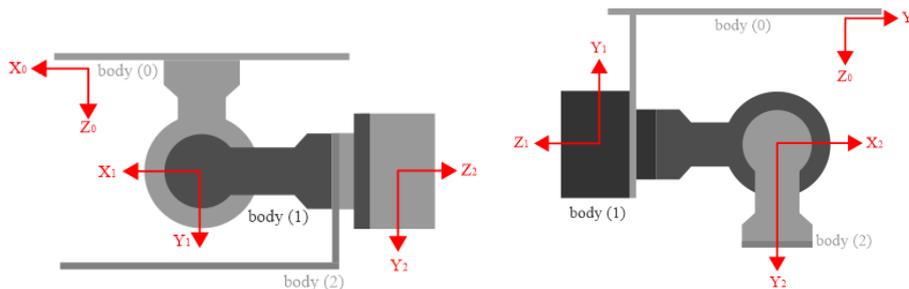
Gambar 1 Arsitektur sistem *gimbal* 2-sumbu

2.3 Pemodelan Gimbal 2-Sumbu

Pemodelan *gimbal* 2-sumbu dibagi menjadi beberapa bagian yaitu, pemodelan matematika dari kinematika *gimbal*, pemodelan motor sebagai aktuator *gimbal*, dan juga kendali PID. Pemodelan *gimbal* yang dibuat pada penelitian ini berfungsi untuk melakukan pendekatan sistem secara simulasi.

2.3.1 Kinematika gimbal 2-sumbu

Gimbal 2-sumbu terdiri dari dua buah *joints* yang dapat berputar pada sudut θ_1 dan θ_2 sepanjang sumbu *pitch* dan sumbu *roll*. *Gimbal* 2-sumbu yang digunakan terdiri dari tiga *body frame* (*body* (0), *body* (1), dan *body* (2)). Pada Gambar 2 ditunjukkan hubungan antar *body frame*. *Body* (0), *body* (1), dan *body* (2) dalam bentuk $X_0Y_0Z_0$, $X_1Y_1Z_1$, dan $X_2Y_2Z_2$.



Gambar 2 Konfigurasi *gimbal* 2-sumbu (kiri) tampak depan, (kanan) tampak samping

Untuk mengetahui pergerakan dari *end effector* atau dalam Gambar 2 ditunjukkan dengan *body frame* (2) dapat dilakukan dengan pendekatan *forward* kinematik menggunakan kaidah Denavit-Hartenberg [8]. *Rotation matrix* antara *body frame* (0) dengan *body frame* (1) ditunjukkan pada persamaan (1).

$${}^0R_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta_1 & -\sin \theta_1 \\ 0 & \sin \theta_1 & \cos \theta_1 \end{bmatrix} \quad (1)$$

Sedangkan *rotation matrix* untuk *body frame* (1) dengan *body frame* (2) ditunjukkan pada persamaan (2).

$${}^1R_2 = \begin{bmatrix} \cos \theta_2 & 0 & \sin \theta_2 \\ 0 & 1 & 0 \\ -\sin \theta_2 & 0 & \cos \theta_2 \end{bmatrix} \quad (2)$$

Sehingga didapatkan total *rotation matrix* untuk *body frame* (0), *body frame* (1), dan *body frame* (2) pada persamaan (3).

$${}^0R_2 = \begin{bmatrix} \cos \theta_2 & 0 & \sin \theta_2 \\ \sin \theta_1 \sin \theta_2 & \cos \theta_1 & -\sin \theta_1 \sin \theta_2 \\ -\cos \theta_1 \sin \theta_2 & \sin \theta_1 & \cos \theta_1 \cos \theta_2 \end{bmatrix} \quad (3)$$

Kemudian untuk menentukan sudut *end effector* atau *body frame* (2) yang diinginkan ditempatkan Sensor IMU. Sensor IMU berfungsi untuk memberikan nilai sudut pada *end effector*. Sehingga dapat diketahui nilai *error* antar sudut yang diinginkan dengan sudut asli dari hasil Sensor IMU. Hubungan tersebut ditunjukkan pada persamaan (4) dan (5).

$$\varepsilon_1 = \alpha_{1 \text{ setpoint}} - \alpha_1 \quad (4)$$

$$\varepsilon_2 = \alpha_{2 \text{ setpoint}} - \alpha_2 \quad (5)$$

Dari persamaan (4) dan (5) dapat dibuat persamaan untuk *rotation matrix* dari *error* yang terjadi pada *body frame* (2) seperti yang ditunjukkan pada persamaan (6),

$${}^2R_e = \begin{bmatrix} \cos \varepsilon_2 & 0 & \sin \varepsilon_2 \\ \sin \varepsilon_1 \sin \varepsilon_2 & \cos \varepsilon_1 & -\sin \varepsilon_1 \sin \varepsilon_2 \\ -\cos \varepsilon_1 \sin \varepsilon_2 & \sin \varepsilon_1 & \cos \varepsilon_1 \cos \varepsilon_2 \end{bmatrix} \quad (6)$$

Dengan menggunakan *inverse kinematik*, didapatkan hubungan antara total *rotation matrix* dengan *error* yang terjadi pada *body frame* (2) seperti yang ditunjukkan pada persamaan (7),

$${}^0R_\varepsilon(\theta, \alpha) = {}^0R_2(\theta) {}^2R_e(\alpha) \quad (7)$$

θ_1 , dan θ_2 adalah sudut motor saat ini. Untuk sudut motor yang baru dapat dilihat pada persamaan (8).

$${}^0R_\varepsilon(\theta_{\text{baru}}) = {}^0R_\varepsilon(\theta, \varepsilon) \quad (8)$$

Dari persamaan (8) dapat dilihat bahwa *body frame* (2) adalah sama dengan *frame error* (ε), Sehingga akan didapatkan persamaan seperti yang ditunjukkan pada persamaan (9),

$${}^0R_\varepsilon(\theta_{\text{baru}}) = {}^0R_2(\theta) {}^2R_e(\alpha) \quad (9)$$

Dari persamaan (9) dapat dibuat hubungan antara *error* terhadap *body frame* 0, permisalan hasil perkalian matriks yang terjadi seperti yang ditunjukkan pada persamaan (10).

$${}^0R_2(\theta) {}^2R_e(\alpha) = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad (10)$$

Pada persamaan (10) r_{ij} merupakan representasi dari komponen didalam matriks ${}^0R_3(\theta)^3R_c(\alpha)$. Dengan menggunakan *inverse* kinematik dapat diketahui bahwa nilai dari sudut motor yang baru, sehingga akan dihasilkan persamaan (11) dan persamaan (12).

$$\theta_{1\text{ baru}} = \tan^{-1}\left(-\frac{r_{23}}{r_{33}}\right) \quad (11)$$

$$\theta_{2\text{ baru}} = \sin^{-1}(r_{13}) \quad (12)$$

$\theta_{1\text{ baru}}$ dan $\theta_{2\text{ baru}}$ adalah sudut motor baru yang diinginkan agar kamera pada *gimbal* dapat mempertahankan orientasi.

2.3.2 Pemodelan Motor

Kita dapat memperoleh model matematik dari motor DC dengan mengaplikasikan hukum *Kirchhoff* kedalam kumparan armature dan hukum II *Newton* kedalam mekanik motor [9]. Sehingga, didapatkan persamaan (13) untuk penerapan Hukum *Kirchhoff* dan persamaan (14) untuk penerapan hukum II *Newton*.

$$L \frac{di}{dt} + Ri = V - K\dot{\theta} \quad (13)$$

$$J\ddot{\theta} + b\dot{\theta} = Ki \quad (14)$$

Kemudian, dari persamaan (13) dan (14) dirubah kedalam bentuk *Laplace* menjadi

$$(Ls + R)I(s) = V(s) - Ks\theta(s) \quad (15)$$

$$s(Js + b)\theta(s) = KI(s) \quad (16)$$

Dari persamaan (15) dan (16) kita dapat mensubtitusikan menjadi persamaan yang ditunjukkan pada persamaan (17).

$$\frac{\theta(s)}{V(s)} = \frac{K}{s[(Ls + R)(Js + B) + K^2]} \quad (17)$$

Dimana, K adalah back emf (Nm/A), L adalah induktansi (H), R adalah resistansi (Ω), J adalah momen inersia dari motor ($\text{Kg m}^2/\text{s}^2$), dan B adalah rasio damping (Nm/rad).

Karena dalam penelitian ini menggunakan 2 buah motor, yaitu motor yang dipasang pada sumbu *roll* dan sumbu *pitch*. maka untuk persamaan (17) berlaku untuk motor pada sumbu *roll* maupun sumbu *pitch*. Namun, nilai J atau inersia berbeda antara motor pada sumbu *roll* dan sumbu *pitch*, karena dipengaruhi beban yang berbeda antara motor pada sumbu *roll* dan sumbu *pitch*. Nilai inersia pada motor sumbu *roll* dan motor sumbu *pitch* ditunjukkan pada persamaan (18) untuk sumbu *roll* dan (19) untuk sumbu *pitch*.

$$I_{xx} = \sum_{j=1}^n \left(I_{G_{xxj}} + m_j(y_j^2 + z_j^2) \right) \quad (18)$$

$$I_{yy} = \sum_{j=1}^n \left(I_{G_{yyj}} + m_j(x_j^2 + z_j^2) \right) \quad (19)$$

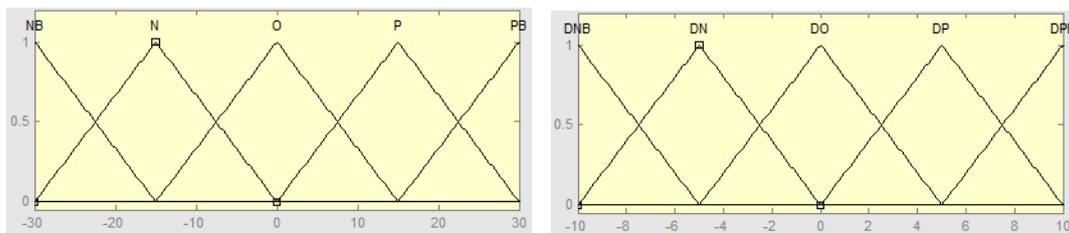
Dimana $I_{G_{xxj}}$ dan $I_{G_{yyj}}$ adalah inersia pada sumbu x atau *pitch* dan y atau *roll*, serta j menunjukkan komponen ke- j yang mempengaruhi inersia; x_j dan y_j adalah jarak *center of gravity* masing masing komponen ke- j ke poros putar motor pada sumbu x dan y; dan m_j adalah massa dari masing-masing komponen ke- j [10].

2.4 Rancangan Sistem Kendali

Salah satu sistem kendali yang dapat digunakan untuk mempertahankan orientasi kamera pada *gimbal* adalah sistem kendali PID (Proportional Integral Derivative). Persamaan PID ditunjukkan pada persamaan (20). Dimana Kp adalah konstanta proporsional, Ki adalah konstanta integral, dan Kd adalah konstanta derivative, dengan keluaran u , dan e merupakan masukan *error* yang berasal nilai masukan referensi r dikurangi dengan nilai masukan sudut y . Serta dimana t adalah waktu saat ini.

$$u(t) = Kp \cdot e(t) + Ki \int_0^t e(t)dt + Kd \frac{de}{dt} \tag{20}$$

Logika *fuzzy* dapat digunakan untuk mengoptimalkan respon dari sistem kendali PID dengan menala konstanta Kp , Ki , dan Kd berdasarkan masukan *error* dan *delta error* [11]. Lima nilai linguistik digunakan dalam rancangan himpunan masukan *fuzzy* yang menggambarkan *error* dan *delta error* ditunjukkan pada Gambar 3. Untuk himpunan *fuzzy error*, nilai-nilai tersebut adalah NB (Negatif Besar), N (Negatif), O (Nol), P (Positif), dan PB (Positif Besar). Untuk himpunan *fuzzy delta error*, nilai-nilai tersebut adalah DNB (Delta Negatif Besar), DN (Delta Negatif), DO (Delta Nol), DP (Delta Positif), dan DPB (Delta Positif Besar).



Gambar 3 Himpunan masukan *fuzzy* (kiri) *error*, (kanan) *delta error*

Tabel 1 merupakan rancangan aturan *fuzzy* yang akan diimplementasikan pada tahap inference. Terdapat sebanyak 25 aturan *fuzzy* (*rule base*) yang ditanamkan pada tahap *inference*. FIS yang digunakan pada kendali ini adalah metode *Mamdani*. Alasan pemilihan FIS dengan metode *Mamdani* yaitu karena keluaran FIS tipe *Mamdani* berupa *fuzzy set* dan bukan sekadar inversi dari fungsi keanggotaan keluaran. Keluaran pada FIS yang diinginkan yaitu berupa nilai pada *fuzzy set* untuk memetakan nilai konstanta PID berdasarkan *error* dan *delta error*.

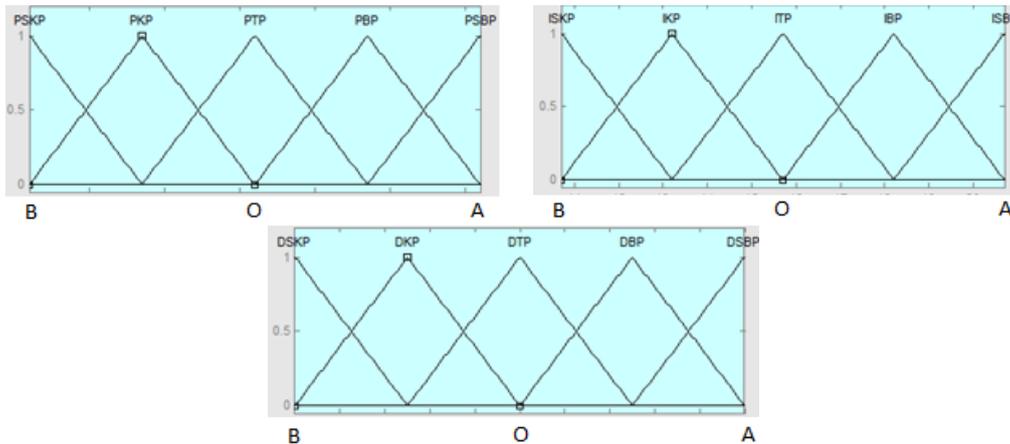
Tabel 1 Aturan *fuzzy* untuk Kp , Ki , dan Kd [12]

Kp dan Ki		Error				
		NB	N	O	P	PB
Delta Error	DNB	SB	B	T	B	SB
	DN	B	T	K	T	B
	DO	T	K	SK	K	T
	DP	B	T	K	T	B
	DPB	SB	B	T	B	SB

Kd		Error				
		NB	N	O	P	PB
Delta Error	DNB	SK	K	T	K	SK
	DN	K	T	B	T	K
	DO	T	B	SB	B	T
	DP	K	T	B	T	K
	DPB	SK	K	T	K	SK

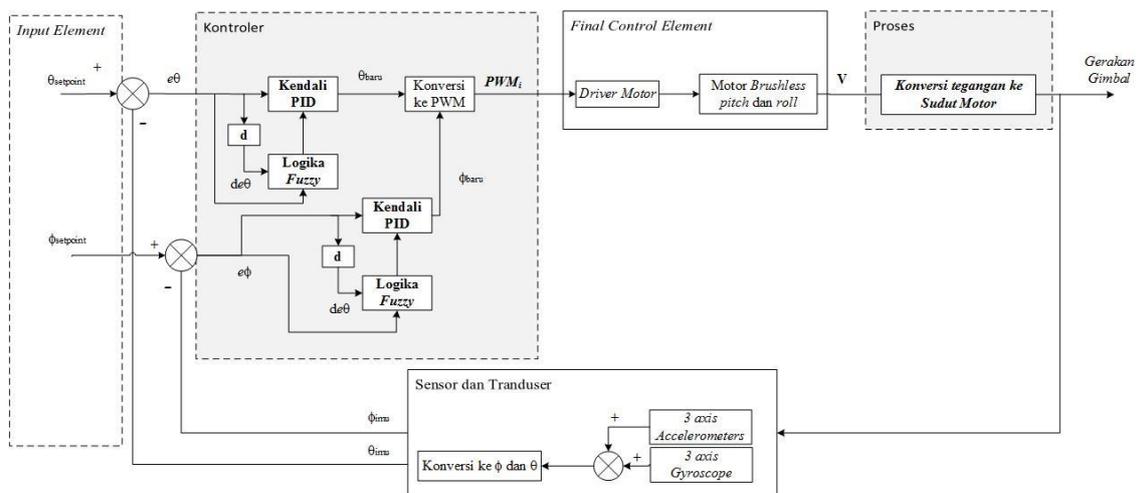
Lima nilai linguistik digunakan dalam rancangan himpunan keluaran *fuzzy* yang menggambarkan nilai Kp , Ki dan Kd . Pada himpunan keluaran *fuzzy* ditunjukkan pada Gambar 4. Nilai-nilai tersebut adalah SK (Sangat Kecil), K (Kecil), T (Tengah), B (Besar), dan SB (Sangat Besar). kemudian untuk inisial huruf awal dari nilai linguistik yaitu P, I dan D merupakan inisial dari konstanta PID, sedangkan huruf akhirnya itu P merupakan inisial untuk sumbu *pitch*. Nilai B dan A pada rentang himpunan akan dicari melalui eksperimen. Sedangkan

nilai 0 pada rentang himpunan merupakan nilai yang didapat melalui penalaan *Ziegler-Nichols*.



Gambar 4 Himpunan masukan *fuzzy* (kiri) K_p , (kanan) K_i , dan (bawah) K_d

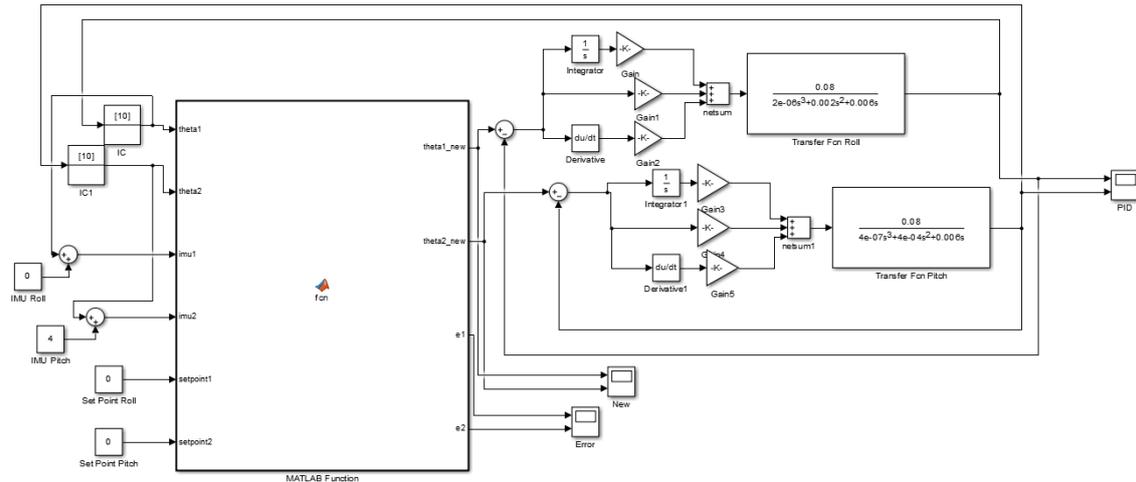
Secara keseluruhan sistem kendali yang akan digunakan pada penelitian ini ditunjukkan pada Gambar 5. Pada Gambar 5, e_θ dan juga e_ϕ menunjukkan *error* yang terjadi pada sumbu *pitch* dan *roll*. Kemudian sistem kendali *fuzzy* memiliki masukan berupa *error* dan juga *delta error* yang diwakili oleh e_θ dan juga $e_\theta(n) - e_\theta(n-1)$ pada sumbu *pitch* dan e_ϕ dan juga $e_\phi(n) - e_\phi(n-1)$ pada sumbu *roll*, sedangkan keluarannya berupa konstanta K_p , K_i , dan K_d yang akan digunakan oleh kendali PID.



Gambar 5 Diagram blok kendali sistem

3. HASIL DAN PEMBAHASAN

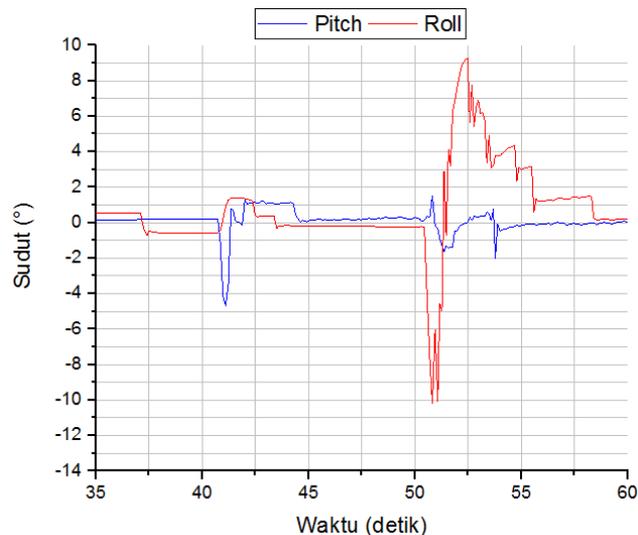
Simulasi pemodelan ditujukan untuk melihat bagaimana respon sistem *gimbal* 2-sumbu bekerja ketika diterapkan kendali PID *Ziegler-Nichols*. Dimulai dengan kinematika, pemodelan motor *brushless* DC pada *gimbal*, hingga kendali PID diimplementasikan pada aplikasi MATLAB menggunakan *SIMULINK toolbox*. Simulasi *gimbal* 2-sumbu yang digunakan pada penelitian ini ditunjukkan pada Gambar 6.



Gambar 6 Simulasi gimbal 2-sumbu

3.1 Hasil Pengujian Kendali PID

Dari simulasi dan validasi hasil pengujian didapatkan nilai dari konstanta K_p , K_i , dan K_d untuk sumbu *pitch* dan *roll*. Untuk sumbu *pitch* didapatkan nilai dari konstanta K_p , K_i , dan K_d masing-masing yaitu 43,2; 15,7; dan 0,1. Sedangkan untuk sumbu *roll* didapatkan nilai dari konstanta K_p , K_i , dan K_d masing-masing yaitu 11,4; 22,3; dan 0,12. Grafik dari pengujian ini dapat dilihat pada Gambar 7.



Gambar 7 Grafik hasil kendali PID pada gimbal 2-sumbu

Pada Gambar 7, untuk grafik sumbu *pitch* pada saat terjadi gangguan sebesar $-4,67^\circ$ respon sistem mengalami *risetime* sebesar 0,12 detik. *Overshoot* yang terjadi yaitu sebesar $1,37^\circ$. *Steady state error* yang terjadi yaitu sebesar $0,42^\circ$ dan *settlingtime* sebesar 0,25 detik.

Sedangkan untuk sumbu *roll* terlihat pada saat terjadi gangguan sebesar $-10,07^\circ$ respon sistem mengalami *risetime* sebesar 1,31 detik. *Overshoot* yang terjadi yaitu sebesar $9,28^\circ$. *Steady state error* yang terjadi yaitu sebesar $0,01^\circ$ dan *settlingtime* sebesar 4,52 detik.

3.2 Hasil Pengujian Kendali PID Fuzzy

Dari percobaan sebelumnya telah didapatkan nilai untuk nilai konstanta PID untuk *pitch* dan *roll*. Nilai konstanta PID tersebut akan menjadi nilai 0 pada rentang nilai himpunan.

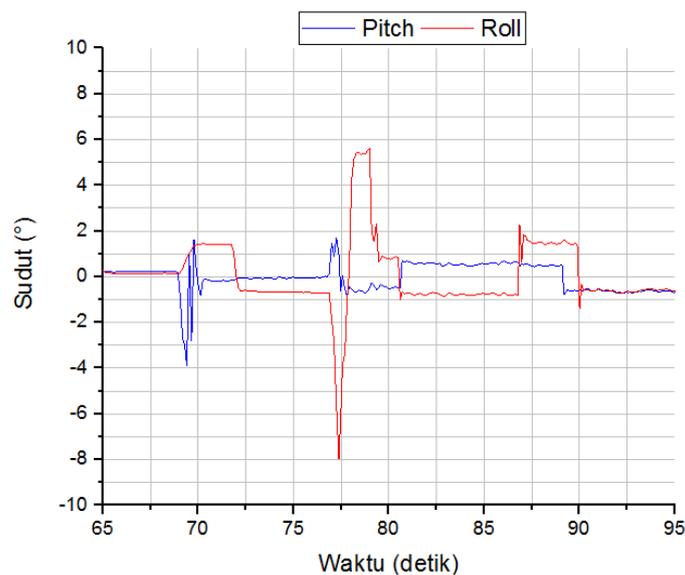
Kemudian berdasarkan nilai O tersebut dilakukan eksperimen untuk Kp dan Ki dengan nilai A yaitu nilai maksimum, sedangkan Kd dengan nilai B yaitu nilai yang minimum. Setelah itu dilakukan eksperimen untuk Kp dan Ki dengan nilai B yaitu nilai yang minimum, sedangkan Kd dengan nilai A yaitu nilai maksimum. Pergeseran nilai Kp, KI, dan Kd dilakukan sedemikian rupa agar interval pergeseran pada nilai maksimum dan minimum dari rentang himpunan bernilai seperti pada Gambar 4. Pengambilan nilai maksimum dan minimum tersebut juga dilihat dari respon yang dihasilkan masih memenuhi spesifikasi sistem yang diinginkan.

Untuk sumbu *pitch*, didapatkan nilai keluaran *fuzzy* berupa nilai Kp dengan rentang nilai 40,2 hingga 46,2, nilai Ki dengan rentang nilai 10,7 hingga 20,7, dan nilai Kd dengan rentang nilai 0,05 hingga 0,15. Untuk sumbu *roll*, didapatkan nilai keluaran *fuzzy* berupa nilai Kp dengan rentang nilai 6,4 hingga 16,4, nilai Ki dengan rentang nilai 17,3 hingga 27,3, dan nilai Kd dengan rentang nilai 0,08 hingga 0,16.

Untuk masukan *fuzzy* pada sumbu *pitch* dan *roll* berupa nilai *error* dengan rentang nilai -30° hingga 30° , dan nilai *delta error* dengan rentang nilai -10° hingga 10° . Nilai rentang masukan *fuzzy* diambil dari pengujian dan batasan dari kinematika *gimbal* yang digunakan. Hasil Grafik dari pengujian ini dapat dilihat pada Gambar 8.

Pada Gambar 8 untuk grafik sumbu *pitch* pada saat terjadi gangguan sebesar $-3,90^\circ$ respon sistem mengalami *risetime* sebesar 0,12 detik. *Overshoot* yang terjadi yaitu sebesar $1,01^\circ$. *Steady state error* yang terjadi yaitu sebesar $0,10^\circ$ dan *settlingtime* sebesar 0,47 detik.

Sedangkan untuk sumbu *roll* terlihat pada saat terjadi gangguan sebesar $-7,97^\circ$ respon sistem mengalami *risetime* sebesar 1,07 detik. *Overshoot* yang terjadi yaitu sebesar $7,80^\circ$. *Steady state error* yang terjadi yaitu sebesar $0,23^\circ$ dan *settlingtime* sebesar 2,10 detik.



Gambar 8 Grafik hasil kendali PID *fuzzy* pada *gimbal* 2-sumbu

3.3 Hasil Pengujian Kendali PID Fuzzy

Setelah telah dilakukan pengujian pada sumbu *pitch* dan *roll* pada *gimbal* dengan kendali PID dan PID *fuzzy*, dibuat tabel untuk memudahkan perbandingan antara kedua metode tersebut yang disajikan pada Tabel 2.

Untuk *risetime* yang baik pada kendali PID dan PID *fuzzy* dihasilkan nilai yang hampir sama pada sumbu *pitch* maupun *roll*. Sedangkan untuk *overshoot* dihasilkan nilai yang lebih baik pada kendali PID *fuzzy* pada sumbu *pitch* maupun *roll*. Kemudian untuk nilai *steady state error* dihasilkan nilai yang lebih baik pada sumbu *pitch* menggunakan kendali PID *fuzzy* sedangkan pada sumbu *roll* menggunakan kendali PID. Untuk nilai *settlingtime* dihasilkan nilai

yang lebih baik pada sumbu *pitch* menggunakan kendali PID sedangkan sumbu *roll* menggunakan kendali PID *fuzzy*.

Tabel 2 Perbandingan hasil kendali PID dan PID *fuzzy* pada *gimbal* 2-sumbu

Error(°)	Pitch		Roll	
	PID	PID <i>fuzzy</i>	PID	PID <i>fuzzy</i>
	-4,67	-3,90	-10,07	-7,97
Risetime(detik)	0,12	0,12	1,31	1,07
Overshoot(°)	1,37	1,01	9,28	7,80
Steadystate(°)	0,42	0,10	0,01	0,23
Settlingtime(detik)	0,25	0,47	4,52	2,10

Untuk sumbu *pitch*, nilai *risetime* dan *overshoot* yang dihasilkan pada kendali PID dan kendali PID *fuzzy* memiliki nilai yang hampir sama. sedangkan *steady state error* dihasilkan nilai yang lebih baik menggunakan kendali PID *fuzzy*. Kemudian untuk nilai *settlingtime* dihasilkan nilai yang lebih baik menggunakan kendali PID.

Untuk sumbu *roll*, nilai *risetime* yang dihasilkan pada kendali PID dan kendali PID *fuzzy* memiliki nilai yang hampir sama. sedangkan *steady state error* dihasilkan nilai yang lebih baik menggunakan kendali PID. Kemudian untuk nilai *overshoot* dan *settlingtime* dihasilkan nilai yang lebih baik menggunakan kendali PID *fuzzy*.

Sudut yang dihasilkan pada sumbu *roll* antara kendali PID dan kendali PID *fuzzy* memiliki perbedaan nilai yang besar yaitu $2,90^\circ$. Hal tersebut terjadi karena mekanik dari *gimbal* yang digunakan mempengaruhi respon sistem. Dapat dilihat pada Gambar 7 dan Gambar 8 saat sumbu *roll* (*body* 1) mendapat gangguan maka sumbu *pitch* (*body* 2) atau *end effector* juga akan mengalami sedikit gangguan sehingga akan mempengaruhi respon dari sumbu *roll* (*body* 1). Pada Gambar 7 dan Gambar 8 dapat dilihat pada saat *roll* mendapat gangguan, kendali PID *fuzzy* dapat menangani gangguan lebih baik pada sumbu *pitch* yang akan mempengaruhi respon sumbu *roll* dibandingkan kendali PID.

4. KESIMPULAN

Berdasarkan hasil pengamatan, pengujian dan analisis pada hasil yang diperoleh dari penelitian dapat disimpulkan, telah berhasil dirancang dan dibuat sistem kendali *gimbal* 2-sumbu yang dapat menjaga orientasi posisi kamera menggunakan metode PID *fuzzy* dengan nilai keluaran *fuzzy* berupa nilai K_p dengan rentang nilai 40,2 hingga 46,2, nilai K_i dengan rentang nilai 10,7 hingga 20,7, dan nilai K_d dengan rentang nilai 0,05 hingga 0,15 pada sumbu *pitch*. sedangkan keluaran *fuzzy* berupa nilai K_p dengan rentang nilai 6,4 hingga 16,4, nilai K_i dengan rentang nilai 17,3 hingga 27,3, dan nilai K_d dengan rentang nilai 0,08 hingga 0,16 pada sumbu *roll*. Untuk masukan *fuzzy* pada sumbu *pitch* dan *roll* berupa nilai *error* dengan rentang nilai -30° hingga 30° , dan nilai *delta error* dengan rentang nilai -10° hingga 10° .

Untuk *risetime*, dihasilkan nilai yang sama antara kendali PID dan kendali PID *fuzzy* baik pada sumbu *pitch* maupun *roll*. Untuk *overshoot*, pada sumbu *pitch* nilai ditunjukkan oleh kendali PID dan PID *fuzzy* bernilai sama, dan pada sumbu *roll* nilai yang lebih baik ditunjukkan oleh kendali PID *fuzzy*. Untuk *steady state error*, pada sumbu *pitch* nilai yang lebih baik ditunjukkan oleh kendali PID *fuzzy*, sedangkan pada sumbu *roll* nilai yang lebih baik ditunjukkan oleh kendali PID. Untuk *settlingtime*, pada sumbu *pitch* nilai yang lebih baik ditunjukkan oleh kendali PID, sedangkan pada sumbu *roll* nilai yang lebih baik ditunjukkan oleh kendali PID *fuzzy*.

5. SARAN

Pada penelitian ini masih terdapat beberapa hal yang perlu disempurnakan. Perlu adanya sistem kendali yang mampu menentukan kecepatan gerak *gimbal* secara adaptif berbanding lurus terhadap perubahan *error* yang terjadi, sehingga *gimbal* akan dapat merespon lebih baik ketika terjadi perubahan orientasi. Selain itu perlu ditambahkan program untuk kalkulasi kecepatan dalam satu kali program berjalan, sehingga dapat memaksimalkan respon *gimbal*.

DAFTAR PUSTAKA

- [1] Rajesh, R.J. and Kavitha, P., 2015, Camera gimbal stabilization using conventional PID controller and evolutionary algorithms, *2015 International Conference on Computer, Communication and Control (IC4)*, September 2015 IEEE., pp. 1–6.
- [2] Ogata, K., 2010, *Modern Control Engineering*, Prentice Hall, New Jersey.
- [3] Passino, K.M. and Yurkovich, S., 1998, *Fuzzy Control*, Addison Wesley Longman, Inc., California.
- [4] Perwira, Z. Y., 2016, Implementasi Metode PID (Proportional Integral Derivative) Fuzzy pada Sistem Penerbangan Quadrotor untuk Penelusuran Lorong Berbelok, *Tesis*, Program Sarjana Elektronika dan Instrumentasi, Univ. Gadjah Mada, Yogyakarta.
- [5] Abbasi, E. and Mahjoob, M., 2013, Controlling of Quadrotor UAV Using a Fuzzy System for Tuning the PID Gains in Hovering Mode, *Tesis*, School of Mechanical Engineering, University of Tehran.
- [6] A. Dharmawan and R. Handoyo, “Sistem Stabilisasi Posisi Kamera pada Quadcopter,” *IJEIS (Indonesian J. Electron. Instrum. Syst.*, vol. 2, no. 2, pp. 109–118, 2012 [Online]. Available: <https://jurnal.ugm.ac.id/ijeis/article/view/2421>. [Accessed: 21-Apr-2017]
- [7] Rajesh, R.J. and Ananda, C.M., 2015, PSO tuned PID controller for controlling camera position in UAV using 2-axis gimbal, *2015 International Conference on Power and Advanced Control Engineering (ICPACE)*, August 2015 IEEE., pp. 128–133,
- [8] Jazar, R.N., 2007, *Theory of Applied Robotics*, Springer US, New York.
- [9] Kleuver, C.A., 2015, *Dynamic System: Modeling, Simulation, and Control*, Wiley.
- [10] Hibbeler, R.C., 2016, *Dynamics*, Prentice Hall, New Jersey.
- [11] Santos, M., López, V. and Morata, F., 2010, Intelligent fuzzy controller of a quadrotor, *Proceedings of 2010 IEEE International Conference on Intelligent Systems and Knowledge Engineering, ISKE 2010*, 141–146.
- [12] Abdo, M., 2014, Modeling control and simulation of two axes gimbal seeker using fuzzy PID controller, *The 22nd Iranian Conference on Electrical Engineering, (Icee)*, 1342–1347,