

BAB II

TINJAUAN PUSTAKA

2.1. Penelitian Terkait

Dalam penelitian Tugas Akhir ini dilakukan studi literatur yang merupakan pencarian teori serta referensi yang relevan dengan kasis dan permasalahan yang akan diselesaikan, teori dan referensi didapat dari jurnal, *paper*, buku dan sumber lainnya.

Pada penelitian yang berjudul “Pengendalian Pergerakan *Hover* Quadcopter Menggunakan Metode PID Jaringan Syaraf Tiruan” dimana pengendali JST berperan sebagai pengendali untuk memilih nilai K_p , K_i dan K_d melalui proses pembelajaran JST sehingga mampu untuk mengendalikan pergerakan *hover*. Pada kecepatan sudut menggunakan pengendali PID dan PD untuk posisi sudut, didapati hasil respon keluaran posisi sudut *roll* saat *set point* 0,1 yaitu eror 3,6%. Pada sudut *pitch set point* 0,1 dengan eror 4% dan sudut *yaw set point* 0,1 memiliki eror 5%. Dan diberi gangguan saat simpangan posisi sudut *roll* menuju ke 0,0065 rad kecepatan sudutnya ikut bertambah dan saat posisi sudut meninggalkan posisi 0,0065 rad maka kecepatannya ikut turun. Ini karena dikendalikan dengan nilai PID hasil linierisasi saat nilai p dan q diantara 0 sampai 0,2 rad yang menyebabkan respon kecepatan sudut *roll* kembali ke *set point*. Begitu juga dengan respon saat sudut *pitch* dan *yaw*. Sehingga hasil desain kendali PD untuk kendali posisi sudut dan nilai K_p , K_i dan K_d hasil linierisasi untuk kendali kecepatan sudut cocok untuk *plant* Quadcopter [3].

Analisa Kestabilan Terbang Quadcopter dengan Pengendali PD terhadap Gangguan [1]. Menggunakan pengendali PD dimana didapati hasil saat diberi gangguan pada respon ketinggian untuk mengoreksi gangguan ketika diturunkan 1 m di detik ke-20 dilakukan dengan baik dan dapat kembali ke *set point* setelah 2-3 detik. Sehingga koreksi pada sudut *roll*, *pitch* dan *yaw* saat diberi gangguan dapat dikatakan baik dan sesuai yang diinginkan. Terbukti dengan waktu *settling time* untuk ketiga respon kurang dari 3 detik.

Perancangan dan Implementasi Sistem Pengaturan Optimal LQR untuk Menjaga Kestabilan *Hover* pada Quadcopter [5]. Pada penelitan tersebut didapatkan hasil simulasi dapat terbang *hover* pada ketinggian 2 m, dan dapat mengatasi gangguan dengan *rise time* selama 0.1332 detik. Respon hasil simulasi terbang *hover* dengan *set point* ketinggian 100 cm masih berisolasi antara 50 cm sampai 200 cm, dan respon kestabilan sudut lebih lambat yaitu 0.23 detik.

Perancangan dan Implementasi Kontrol PID *Gain Scheduling* untuk Gerakan Lateral *Way-to-Way Point* pada UAV Quadcopter [7]. Pada pergerakan *hover* adalah saat yang paling riskan pada Quadcopter. Sehingga pengaturan *hover* dari Quadcopter menjadi prioritas pada riset ini, karena apabila terjadi kesalahan *hover* akan menyebabkan perubahan gerak terhadap sumbu X, Y maupun Z. Peneliti menggunakan kendali PID dan metode *gain scheduling* dengan parameter penguatan yang diperoleh secara eksperimental untuk pengaturan gerak lateral menyusuri lintasan. Apabila menggunakan PID biasa, maka diperoleh penurunan ketinggian yang drastis hingga 50% dari total ketinggian Quadcopter, namun apabila menggunakan metode PID *gain scheduling* pada simulasi hanya terjadi penurunan ketinggian sebesar 0.35% dan saat implementasi mengalami sekitar 10% dari total ketinggian yang dicapai.

Perancangan dan Implementasi Kontrol Fuzzy-PID pada Pengendalian *Auto Take-Off* Quadcopter UAV [8]. Fase *take-off* pada Quadcopter merupakan fase kritis yang harus dijaga kestabilannya. Pengaruh turbulensi angin yang dihasilkan dari perputaran motor merupakan salah satu gangguan yang menyebabkan sulitnya proses *take-off*. Terlebih dengan ukuran *plant* yang cukup berat. Untuk mencapai ketinggian sekitar 0 hingga 1 meter banyak terjadi gangguan, dapat dicapai Quadcopter dalam waktu 4.85 detik. Kemudian terjadi *offset* sebesar 0.96% pada kondisi *steady state* yang dicapai dalam waktu 5 detik.

2.2. Quadcopter

Quadcopter adalah jenis *Unmanned Aerial Vehicle* (UAV) yang termasuk kategori UAV mikro dan banyak digunakan oleh industri, instansi pendidikan, maupun masyarakat umum. Robot Quadcopter ini memiliki ciri yang *unique* yang mudah untuk dikenali yaitu sesuai namanya memiliki empat buah baling –baling motor pada keempat sisinya yang digunakan sebagai penggerakannya.



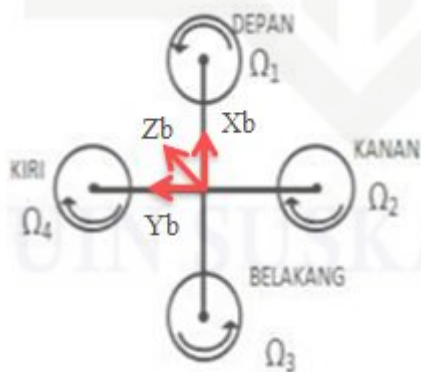
Gambar 2.1. Bentuk fisik quadrotor[1]

Gambar di atas adalah salah satu bentuk robot Quadcopter dengan konfigurasi *plus* (+). Empat buah baling – baling yang terpasang memudahkan Quadcopter untuk bermanuver sehingga dapat bergerak ke segala arah. Quadcopter dilengkapi dengan beberapa sensor, diantaranya sensor *Global Position System* (GPS) yang digunakan untuk navigasi, sensor *Inertial Measurement Unit* (IMU) yang berfungsi untuk menghitung percepatan serta orientasi arah pergerakan, sensor *ultra sonic* yang mendeteksi ketinggian Quadcopter, dan sensor – sensor lainnya yang mendukung fungsi dan kinerja Quadcopter [1].

Quadcopter memiliki beberapa kelebihan yang membuatnya cocok untuk melakukan pekerjaan tertentu. Salah satunya karena bentuknya yang kecil membuat Quadcopter dapat bergerak secara bebas walaupun ditempat yang sulit untuk dijangkau. Quadcopter tidak memerlukan landasan pacu untuk terbang, karena Quadcopter dapat terbang vertikal. Selain itu Quadcopter juga dapat bergerak kedelapan arah mata angin [3].

Quadcopter juga memiliki beberapa kekurangan, diantaranya Quadcopter hanya dapat bergerak dalam jangka waktu yang singkat. Hal ini dikarenakan *power supply* dari Quadcopter adalah baterai yang memiliki kapasitas terbatas. Jangka waktu tersebut secara tidak langsung akan mempengaruhi kapasitas bawaan, kecepatan terbang, dan jarak tempuh dari Quadcopter [3].

2.2.1. Konsep Dasar Quadcopter[3]



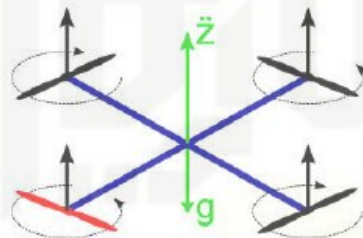
Gambar 2.2 Quadcopter dalam kondisi *hover* atau gerak melayang

Kondisi diatas menunjukkan motor depan dan belakang berputar berlawanan arah jarum jam, motor kanan dan kiri berputar searah jarum jam. Sedangkan Z_B , X_B , dan Y_B adalah *body frame* (*B-frame*) dari Quadcopter. Kombinasi dari perubahan kecepatan

keempat motor akan menghasilkan beberapa pergerakan. Berikut adalah beberapa pergerakan yang ada dalam Quadcopter:

a. Gaya *Thrust* ($U_1[N]$)

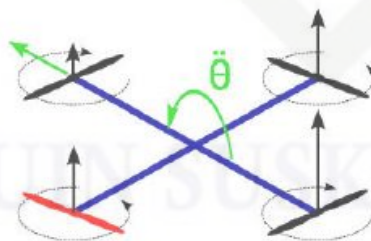
Gaya *thrust* adalah gaya yang mengakibatkan Quadcopter bergerak naik keatas atau turun kebawah searah sumbu Z *earth frame*. Gaya ini terjadi jika kecepatan putarab keempat motor sama dan secara bersamaan ditambah kecepatannya dengan nilai yang sama, maka Quadcopter akan menghasilkan gaya angkat (*thrust*). Begitu juga sebaliknya, jika kecepatan dari keempat motor diperlambat secara bersamaan, Quadcopter akan bergerak kebawah [3].



Gambar 2.3. Gerakan Vertikal Quadcopter[9]

b. Torsi *Roll* ($U_2[Nm]$)

Torsi *roll* adalah torsi yang mengakibatkan Quadcopter berputar disepanjang sumbu x *body frame* (X_B). Torsi ini dipengaruhi jika kecepatan motor kiri dipercepat sedangkan motor kanan diperlambat maka Quadcopter akan bergerak mengguling ke kanan. Begitu pula jika sebaliknya, jika motor kanan yang dipercepat dan motor kiri yang diperlambat maka Quadcopter akan bergerak mengguling ke kiri [3].

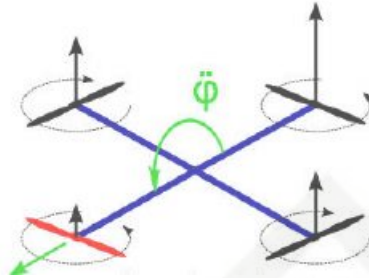


Gambar 2.4. Gerakan *Roll* Quadcopter[9]

c. Torsi *Pitch* ($U_3[Nm]$)

Torsi *Pitch* adalah torsi yang mengakibatkan Quadcopter berputar disepanjang sumbu y pada *body frame* Quadcopter (Y_B). Gerak ini dipengaruhi jika kecepatan motor depan diperlambat sedangkan kecepatan motor belakang dipercepat, maka Quadcopter akan bergerak mengguguk kedepan. Begitu pula jika kecepatan motor depan dipercepat

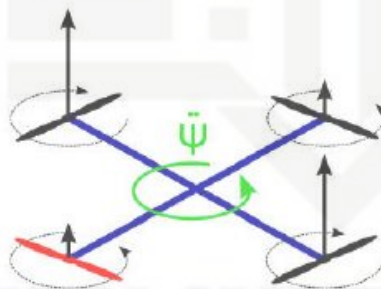
dan kecepatan motor belakang diperlambat maka Quadcopter akan bergerak mengganggu kebelakang [3].



Gambar 2.5. Gerakan Pitch Quadcopter[9]

d. Torsi Yaw (U_4 [Nm])

Torsi yaw adalah torsi yang mengakibatkan Quadcopter berputar disepanjang sumbu Z pada *body frame* Quadcopter (Z_B). Gerak ini dipengaruhi oleh perubahan kecepatan dari keempat motor Quadcopter. Jika kecepatan motor depan dan belakang diperlambat sedangkan kecepatan motor kanan dan kiri dipercepat maka Quadcopter akan bergerak menyimpang ke kiri. Begitu pula sebaliknya, jika kecepatan motor depan dan belakang dipercepat sedangkan motor yang lain diperlambat, maka Quadcopter akan bergerak menyimpang ke kanan. Perubahan kecepatan yang terjadi pada keempat motor memiliki nilai yang sama [3].

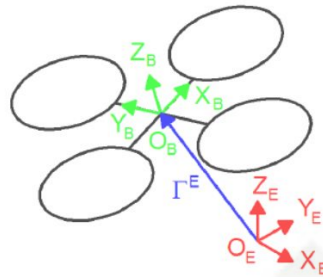


Gabar 2.6. Gerakan Yaw Quadcopter[9]

2.2.2. Kinematika[3][9]

Kinematika adalah cabang dari mekanika klasik yang membahas gerak benda dan sistem benda tanpa mempersoalkan gaya penyebab gerakan. Quadcopter juga dipelajari tentang kinematika, sehingga untuk mempermudah analisa Quadcopter memiliki keluaran 6 *degree of freedom* (DOF). 6 DOF *rigid-body* di deskripsikan dalam dua buah *frame*.

Frame yang digunakan sebagai referensi yaitu *earth frame* (*E-frame*) dan *body frame* (*B-frame*) [3].



Gambar 2.7. *Frame* Quadcopter[3]

Gambar diatas menunjukkan terdapat 2 buah *frame*. *Frame* dengan warna merah adalah *earth frame* (*E-frame*) dengan masing–masing sumbu *frame* mengarah pada utara (X_E), barat (Y_E), arah keatas (Z_E), dan origin (O_E). Sedangkan *frame* dengan warna hijau adalah *body frame* (*B-frame*) dengan masing–masing sumbu *frame* mengarah pada depan (X_B), kiri (Y_B), keatas (Z_B), dan origin (O_B).

Posisi linier quadrotor (Γ_E) ditentukan dari koordinat vector antara origin *B-frame* serta origin dari *E-frame* dengan memperhatikan *E-frame*. Posisi angular Quadcopter (Θ_E) ditentukan dari orientasi *B frame* terhadap *E-frame*. Persamaan posisi linier dan posisi angular masing – masing ditulis pada persamaan dibawah [9].

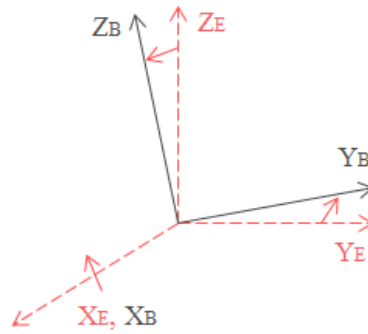
$$\Gamma^E = [X \ Y \ Z]^T \tag{2.1}$$

$$\Theta^E = [\phi \ \theta \ \psi]^T \tag{2.2}$$

Salah satu yang diperlukan untuk mentransformasi suatu nilai dari *body frame* (*B-frame*) ke *earth frame* (*E-frame*) adalah matrik rotasi. Matrik rotasi ini terdiri dari 3 buah matrik rotasi yang masing–masing berotasi terhadap sumbu x bumi (X_E), sumbu y bumi (Y_E), dan sumbu z bumi (Z_E).

a. Rotasi Sumbu X

Rotasi sepanjang sumbu x dilambangkan dengan $R(\phi,x)$ yang merupakan rotasi Quadcopter sepanjang sumbu x menghasilkan sudut yang diberi nama sudut *roll*. Rotasi tersebut direpresentasikan dengan Gambar 2.8 [9].



Gambar 2.8. Rotasi sumbu X

$$\begin{cases} X_E = X_B \\ Y_E = Y_B \cos \phi - Z_B \sin \phi \\ Z_E = Y_B \sin \phi + Z_B \cos \phi \end{cases} \quad (2.3)$$

Persamaan 2.3 dapat dibuat dalam bentuk matriks menjadi Persamaan 2.4 [9].

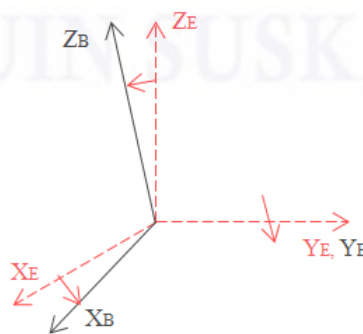
$$\begin{bmatrix} X_E \\ Y_E \\ Z_E \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{bmatrix} \begin{bmatrix} X_B \\ Y_B \\ Z_B \end{bmatrix} \quad (2.4)$$

Adapun matrik rotasi pada sumbu x ditunjukkan pada persamaan 2.5 [9].

$$R(\phi, X) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{bmatrix} \quad (2.5)$$

b. Rotasi Sumbu Y

Rotasi sepanjang sumbu y dilambangkan dengan $R(\theta, y)$ yang merupakan rotasi Quadcopter sepanjang sumbu y menghasilkan sudut yang diberi nama sudut *pitch*. Rotasi tersebut direpresentasikan dengan Gambar 2.9 [9].



Gambar 2.9 Rotasi sumbu Y

$$\begin{cases} X_E = X_B \cos\theta + Z_B \sin\theta \\ Y_E = Y_B \\ Z_E = -X_B \sin\theta + Z_B \cos\theta \end{cases} \quad (2.6)$$

Persamaan 2.6 dapat dibuat dalam bentuk matriks menjadi Persamaan 2.7. [9]

$$\begin{bmatrix} X_E \\ Y_E \\ Z_E \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} X_B \\ Y_B \\ Z_B \end{bmatrix} \quad (2.7)$$

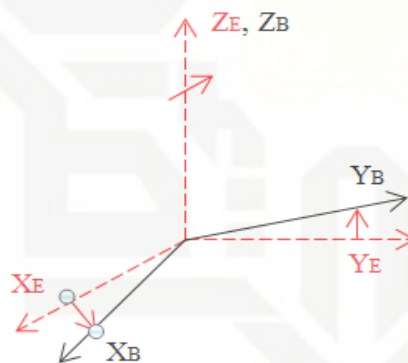
Adapun matriks rotasi pada sumbu y ditunjukkan pada persamaan 2.8. [9]

$$R(\theta, y) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \quad (2.8)$$

c. Rotasi Sumbu Z

Rotasi sepanjang sumbu z dilambangkan dengan $R(\psi, z)$ yang merupakan rotasi Quadcopter sepanjang sumbu z yang menghasilkan sudut yang diberi nama sudut *yaw*.

Rotasi tersebut direpresentasikan dengan Gambar 2.10 dan persamaan 2.9 [9].



Gambar 2. 10 Rotasi sumbu Z

$$\begin{cases} X_E = X_B \cos\psi - Y_B \sin\psi \\ Y_E = X_B \sin\psi + Y_B \cos\psi \\ Z_E = Z_B \end{cases} \quad (2.9)$$

Persamaan 2.9 dapat dibuat dalam bentuk matriks menjadi persamaan 2.10 [9].

$$\begin{bmatrix} X_E \\ Y_E \\ Z_E \end{bmatrix} = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_B \\ Y_B \\ Z_B \end{bmatrix} \quad (2.10)$$

Adapun matriks rotasi pada sumbu z ditunjukkan pada persamaan 2.11 [9].

$$R(\varphi, z) = \begin{bmatrix} \cos\varphi & -\sin\varphi & 0 \\ \sin\varphi & \cos\varphi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.11)$$

Persamaan rotasi dari tiap – tiap sumbu putar didapat satu persamaan matrik rotasi dimana dengan keterangan :

$$\begin{aligned} c &= \cos \\ s &= \sin \\ t &= \tan[9]. \end{aligned}$$

$$R_{\Theta} = R(\psi) R(\theta) R(\phi) \quad (2.12)$$

$$R_{\Theta} = \begin{bmatrix} c\psi c\theta & -s\psi c\theta + c\psi s\theta s\phi & s\psi s\theta + c\psi s\theta s\phi \\ s\psi c\theta & c\psi c\theta + s\psi s\theta c\phi & -c\psi s\theta + s\psi s\theta c\phi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix} \quad (2.13)$$

Sedangkan untuk kecepatan dalam Quadcopter diekspresikan terhadap *body frame* (*B-frame*). Kecepatan Quadcopter terdiri dari kecepatan linier V^B dan kecepatan angular ω^B . Komposisi vektornya disajikan dalam persamaan 2.14 dan 2.15 [9].

$$V^B = [u \ v \ w]^T \quad (2.14)$$

$$\omega^B = [p \ q \ r]^T \quad (2.15)$$

Diperlukan kombinasi nilai linear dan angular untuk memberikan representasi yang lengkap dalam *space*. ξ merupakan komposisi dari vektor posisi linier Γ^E (m) dan vektor posisi sudut Θ^E (rad) terhadap *earth frame* (*E-frame*) seperti terlihat pada persamaan 2.16 [9].

$$\xi = [\Gamma^E \ \Theta^E]^T = [X \ Y \ Z \ \phi \ \theta \ \psi]^T \quad (2.16)$$

V merupakan generalisasi dari vektor kecepatan linier Quadcopter V^B ($m \ s^{-1}$) dan kecepatan angular Quadcopter ω^B ($rad \ s^{-1}$) pada *body frame* (*B-frame*) seperti dibawah ini [9]:

$$V = [V^B \ \omega^B]^T = [u \ v \ w \ p \ q \ r]^T \quad (2.17)$$

Hubungan antara kecepatan linear pada *body frame* (*B-frame*) dan salah satu faktor pada *earth frame* (*E-frame*) V^E (atau $\dot{\Gamma}^E$) [m/s] dapat dilihat pada Persamaan 2.18 [9].

$$V^E = \dot{\Gamma}^E = R_{\Theta} V^B \quad (2.18)$$

Dimana R_{Θ} adalah matrik rotasi dari *body frame* (*B-frame*) ke *earth frame* (*E-frame*). Seperti pada kecepatan linear, hal tersebut juga berlaku untuk menghubungkan kecepatan angular pada *E-frame* (atau kecepatan Euler) $\dot{\Theta}^E$ [rad/s] ke *B-frame* ω^B atau sebaliknya. Hubungan itu dapat dilihat pada persamaan 2.19 [9].

$$\omega^B = T_{\Theta}^{-1} \dot{\Theta}^E \quad (2.19)$$

$$\dot{\Theta}^E = T_{\Theta} \omega^B \quad (2.20)$$

dimana T_{Θ} adalah matriks transformasi. Matriks transformasi T_{Θ} dapat ditetapkan dengan menggunakan laju Euler $\dot{\Theta}^E$ dalam *B-frame*, seperti pada Persamaan berikut[9].

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + R(\phi)^{-1} \begin{bmatrix} \dot{\theta} \\ \dot{\theta} \\ 0 \end{bmatrix} + R(\phi)^{-1} R(\theta)^{-1} \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} \quad (2.21)$$

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = T_{\Theta}^{-1} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (2.22)$$

Persamaan 2.13 dan 2.19 maka diperoleh matriks transformasi dari *body frame* (*B-frame*) menuju *earth frame* (*E-frame*) [9].

$$T_{\Theta}^{-1} = \begin{bmatrix} 1 & 0 & -s\theta \\ 0 & c\phi & c\theta s\phi \\ 0 & -s\phi & c\phi c\theta \end{bmatrix} \quad (2.23)$$

$$T_{\Theta} = \begin{bmatrix} 1 & s\phi t\theta & c\phi t\theta \\ 0 & c\phi & -s\phi \\ 0 & s\phi / c\theta & c\phi / c\theta \end{bmatrix} \quad (2.24)$$

Persamaan yang sudah didapat, maka dibentuk suatu hubungan antara kecepatan terhadap *earth frame* (*E-frame*) dan *body frame* (*B-frame*) [9]. Dimana ξ adalah vektor kecepatan yang mengacu pada *earth frame* $\xi[+]$ (*E-frame*), v adalah vektor kecepatan mengacu *body frame* (*B-frame*) dan J_{Θ} adalah matrik *jacobian*. Matrik *jacobian* terdiri dari 4 *sub*-matrik sebagaimana Persamaan 2.26 [9].

$$\dot{\xi} = J_{\Theta} v \quad (2.25)$$

$$J_{\Theta} = \begin{bmatrix} R_{\Theta} & 0_{3 \times 3} \\ 0_{3 \times 3} & T_{\Theta} \end{bmatrix} \quad (2.26)$$

2.2.3. Dinamika[3][9]

Dinamika adalah cabang dari ilmu fisika (terutama mekanika klasik) yang mempelajari gaya, torsi dan efeknya pada gerak. Dinamika merupakan kebalikan dari kinematika, yang mempelajari gerak suatu objek tanpa memperhatikan penyebabnya. Isaac Newton menciptakan hukum-hukum fisika yang menjadi panduan dalam fisika dinamika. Secara umum, dinamika sangat berkaitan erat dengan hukum kedua Newton tentang gerak [3].

Berdasarkan aksioma pertama Euler dari Hukum Kedua Newton, turunan dari komponen linear dari gerakan suatu benda dapat dilihat pada Persamaan 2.27 [9].

$$\begin{cases}
 m \ddot{\Gamma}^E = F^E \\
 m R_{\Theta} \dot{V}^B = R_{\Theta} F^B \\
 m (R_{\Theta} \dot{V}^B + R_{\Theta} V^B) = R_{\Theta} F^B \\
 m R_{\Theta} (\dot{V}^B + \omega^B \times V^B) = R_{\Theta} F^B \\
 m (\dot{V}^B + \omega^B \times V^B) = F^B
 \end{cases} \quad (2.27)$$

Dimana m (kg) adalah massa quadrotor, $\ddot{\Gamma}^E$ ($m s^{-2}$) adalah vector percepatan linear quadrotor yang mengacu pada E-frame, F^E (N) adalah vektor gaya quadrotor terhadap E-frame, \dot{V}^B ($m s^{-2}$) adalah percepatan linear Quadcopter terhadap B-frame, dan \dot{R}_{Θ} adalah turunan pertama matriks rotasi. Dan simbol \times merupakan perkalian produk suatu vektor. Berdasarkan aksioma kedua Euler dari Hukum Kedua Newton, dengan cara yang sama, turunan dari gerakan angular komponen dari suatu benda dapat dilihat pada Persamaan 2.28 [9].

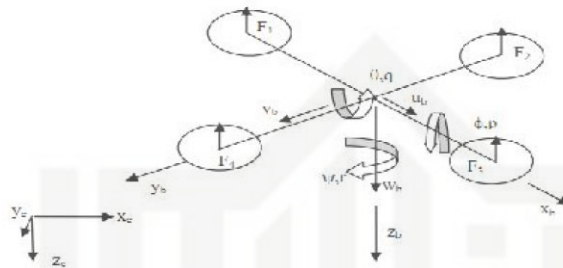
$$\begin{cases}
 I \ddot{\Theta}^E = \tau^E \\
 I T_{\Theta} \dot{\omega}^B = T_{\Theta} \tau^B \\
 I \dot{\omega}^B + \omega^B \times (I \omega^B) = T_{\Theta} \tau^B
 \end{cases} \quad (2.28)$$

Persamaan 2.28, I ($N m s^2$) adalah matriks inersia Quadcopter (pada B-frame), $\ddot{\Theta}^E$ ($rad s^{-2}$) adalah vektor percepatan sudut Quadcopter terhadap E-frame, $\dot{\omega}^B$ ($rad s^{-2}$) adalah vektor percepatan sudut Quadcopter terhadap B-frame, dan τ^E (N m) adalah torsi Quadcopter terhadap E-frame.

Persamaan 2.27 dan 2.28, dapat dibentuk suatu persamaan baru dengan 6 derajat kebebasan (DOF). Persamaan 2.29 menunjukkan formulasi matriks dari dinamika sistem [9].

$$\begin{bmatrix} mI_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & I \end{bmatrix} \begin{bmatrix} \dot{V}^B \\ \dot{\omega}^B \end{bmatrix} + \begin{bmatrix} \omega^B \times (mV^B) \\ \omega^B \times (I\omega^B) \end{bmatrix} = \begin{bmatrix} F^B \\ \tau^B \end{bmatrix} \quad (2.29)$$

Dimana notasi $I_{3 \times 3}$ berarti matriks identitas 3×3 . Sedangkan $0_{3 \times 3}$ adalah matriks nol 3×3 . Quadcopter memiliki empat buah motor yang menghasilkan gaya dorong seperti terlihat pada gambar 2.11.



Gambar 2.11 E-frame dan B-frame Quadcopter beserta empat gaya disetiap propeller[6]

Vektor gaya yang terjadi pada Quadcopter dapat didefinisikan melalui persamaan[9]:

$$\Lambda = [F^B \ \tau^B]^T = [F_x \ F_y \ F_z \ \tau_x \ \tau_y \ \tau_z]^T \quad (2.30)$$

Persamaan 2.30 dapat juga ditulis dalam bentuk formulasi matriks seperti pada persamaan berikut[9]:

$$M_B \dot{v} + C_B(v)v = \Lambda \quad (2.31)$$

di mana v adalah vektor percepatan Quadcopter terhadap B-frame. M_B adalah matriks inersia sistem dan C_B adalah matriks sentripetal Coriolis. Persamaan 2.32 menunjukkan matriks M_B [9]

$$M_B = \begin{bmatrix} mI_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & I \end{bmatrix} = \begin{bmatrix} m & 0 & 0 & 0 & 0 & 0 \\ 0 & m & 0 & 0 & 0 & 0 \\ 0 & 0 & m & 0 & 0 & 0 \\ 0 & 0 & 0 & I_{xx} & 0 & 0 \\ 0 & 0 & 0 & 0 & I_{yy} & 0 \\ 0 & 0 & 0 & 0 & 0 & I_{zz} \end{bmatrix} \quad (2.32)$$

Dapat dilihat bahwa matriks M_B adalah matriks diagonal, sedangkan matriks C_B ditunjukkan persamaan dibawah ini[9].

$$C_B = \begin{bmatrix} 0_{3 \times 3} & -mS(V_B) \\ 0_{3 \times 3} & -S(I \omega_B) \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & mw & -mv \\ 0 & 0 & 0 & -mw & 0 & mu \\ 0 & 0 & 0 & mv & -mu & 0 \\ 0 & 0 & 0 & 0 & I_{zz}r & -I_{yy}q \\ 0 & 0 & 0 & -I_{zz}r & 0 & I_{xx}p \\ 0 & 0 & 0 & I_{yy}q & -I_{xx}p & 0 \end{bmatrix} \quad (2.33)$$

Kontribusi yang kedua mengenai efek giroskopis yang dihasilkan oleh perputaran propeler. Selama dua propeler yang berhadapan berputar searah jarum jam dan dua propeler lainnya berputar berlawanan arah jarum jam, terjadi ketidak-seimbangan saat jumlah aljabar dari kecepatan motor tidak sama dengan nol.

Kontribusi yang ketiga adalah perhitungan gaya dan torsi yang dihasilkan oleh pergerakan input. Pergerakan matriks E_B dikali dengan Ω^2 untuk memperoleh vektor perpindahan $U_B(\Omega)$. Efek aerodinamis (*factor thrust b* (N s²) dan *drag d* (N m s²)) berpengaruh pada gaya dan torsi yang dihasilkan. Persamaan input kecepatan *propeller* (baling-baling) dapat dilihat dalam Persamaan 2.32 [9].

$$\begin{cases} U_1 = b (\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \\ U_2 = bl (-\Omega_2^2 + \Omega_4^2) \\ U_3 = bl (-\Omega_1^2 + \Omega_3^2) \\ U_4 = d (-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2) \\ \Omega = -\Omega_1 + \Omega_2 - \Omega_3 + \Omega_4 \end{cases} \quad (2.34)$$

dimana l (m) adalah jarak antara pusat Quadcopter dengan pusat *propeller*. U_1 , U_2 , U_3 , dan U_4 adalah komponen vektor gerak. Hubungan komponen vektor tersebut dengan *propeller* dapat diperoleh dari perhitungan aerodinamis.

Persamaan-persamaan sebelumnya bukan dalam bentuk matriks, melainkan dalam bentuk sistem persamaan dibawah ini [9].

$$\begin{cases} \dot{p} = \frac{I_{YY}-I_{ZZ}}{I_{XX}} q r - \frac{J_{TP}}{I_{XX}} q \Omega + \frac{U_2}{I_{XX}} \\ \dot{q} = \frac{I_{ZZ}-I_{XX}}{I_{YY}} p r - \frac{J_{TP}}{I_{YY}} p \Omega + \frac{U_3}{I_{YY}} \\ \dot{r} = \frac{I_{XX}-I_{YY}}{I_{ZZ}} p q - \frac{U_4}{I_{ZZ}} \end{cases} \quad (2.35)$$

Bagian rotasi dari Quadcopter berasal dari konsep persamaan gerak rotasi dan dengan menggunakan metode Newton-Euler yang berasal dari kerangka tubuh Quadcopter dengan rumus umum seperti pada persamaan 2.28.

Agar lebih sederhana, momen gyroscopic tidak digunakan karena inersia yang dihasilkan oleh Quadcopter jauh lebih besar dari yang inersia yang dihasilkan oleh rotor. Jadi, persamaan gerak rotasi yang digunakan sebagai berikut: [10]

Persamaan gerak rotasi diperoleh dengan persamaan gerak Euler sebagai berikut

[11]:

$$\sum \tau = I\ddot{\theta} \quad (2.36)$$

$$U_R - C_B(v)\dot{\theta} = I\ddot{\theta} \quad (2.37)$$

dimana I adalah matriks inersia sistem, U_R adalah vektor gaya rotasi dan $C_B(v)$ adalah matriks Coriolis-centripetal [8].

$$\ddot{\theta} = \begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} \quad (2.38)$$

$$I = \begin{bmatrix} I_{XX} & 0 & 0 \\ 0 & I_{YY} & 0 \\ 0 & 0 & I_{ZZ} \end{bmatrix} \quad (2.39)$$

$$U_R = \begin{bmatrix} U_2 \\ U_3 \\ U_4 \end{bmatrix} \quad (2.40)$$

$$C_B(v) = \begin{bmatrix} 0 & I_{ZZ}\dot{\psi} & -I_{YY}\dot{\theta} \\ -I_{ZZ}\dot{\psi} & 0 & I_{XX}\dot{\phi} \\ I_{YY}\dot{\theta} & I_{XX}\dot{\phi} & 0 \end{bmatrix} \quad (2.41)$$

sehingga Persamaan 2.35 dapat dituliskan menjadi: [8]

$$\begin{bmatrix} U_2 \\ U_3 \\ U_4 \end{bmatrix} - \begin{bmatrix} 0 & I_{ZZ}\dot{\psi} & -I_{YY}\dot{\theta} \\ -I_{ZZ}\dot{\psi} & 0 & I_{XX}\dot{\phi} \\ I_{YY}\dot{\theta} & I_{XX}\dot{\phi} & 0 \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} I_{XX} & 0 & 0 \\ 0 & I_{YY} & 0 \\ 0 & 0 & I_{ZZ} \end{bmatrix} \begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} \quad (2.42)$$

$$\begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} I_{XX} & 0 & 0 \\ 0 & I_{YY} & 0 \\ 0 & 0 & I_{ZZ} \end{bmatrix}^{-1} \left(\begin{bmatrix} U_2 \\ U_3 \\ U_4 \end{bmatrix} - \begin{bmatrix} 0 & I_{ZZ}\dot{\psi} & -I_{YY}\dot{\theta} \\ -I_{ZZ}\dot{\psi} & 0 & I_{XX}\dot{\phi} \\ I_{YY}\dot{\theta} & I_{XX}\dot{\phi} & 0 \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \right) \quad (2.43)$$

Persamaan dinamika rotasi untuk masing-masing gerak roll, pitch dan yaw sebagaimana dituliskan pada Persamaan 2.44 [8][10].

$$\begin{cases} \ddot{\phi} = \frac{U_2 l}{I_{XX}} - \frac{\dot{\theta} \psi}{I_{XX}} (I_{ZZ} - I_{YY}) \\ \ddot{\theta} = \frac{U_3 l}{I_{YY}} - \frac{\dot{\phi} \psi}{I_{YY}} (I_{XX} - I_{ZZ}) \\ \ddot{\psi} = \frac{U_4}{I_{ZZ}} - \frac{\dot{\phi} \dot{\theta}}{I_{YY}} (I_{YY} - I_{XX}) \end{cases} \quad (2.44)$$

2.3 Model Matematika Quadcopter

Dari analisis kinematika dan dinamika yang telah dibahas, diperoleh persamaan model matematika dari sistem gerak rotasi Quadcopter secara keseluruhan ditunjukkan pada persamaan 2.45 [8][10].

$$\begin{cases} \ddot{\phi} = \frac{U_2 l}{I_{XX}} - \frac{q r}{I_{XX}} (I_{ZZ} - I_{YY}) \\ \ddot{\theta} = \frac{U_3 l}{I_{YY}} - \frac{p r}{I_{YY}} (I_{XX} - I_{ZZ}) \\ \ddot{\psi} = \frac{U_4}{I_{ZZ}} - \frac{p q}{I_{ZZ}} (I_{YY} - I_{XX}) \end{cases} \quad (2.45)$$

Simbol p q r diubah menjadi x_8 x_{10} x_{12} karena mengikuti penamaan persamaan differensial pada simulink MATLAB. Dalam bentuk persamaan state-space, Persamaan 2.45 dapat dinyatakan sebagai [8]:

$$\begin{cases} \dot{x}_7 = x_8 \\ \dot{x}_8 = \frac{U_2 l}{I_{XX}} - \frac{(I_{ZZ} - I_{YY})}{I_{XX}} x_{10} x_{12} \\ \dot{x}_9 = x_{10} \\ \dot{x}_{10} = \frac{U_3 l}{I_{YY}} - \frac{(I_{XX} - I_{ZZ})}{I_{YY}} x_8 x_{12} \\ \dot{x}_{11} = x_{12} \\ \dot{x}_{12} = \frac{U_4}{I_{ZZ}} - \frac{(I_{YY} - I_{XX})}{I_{ZZ}} x_8 x_{10} \end{cases} \quad (2.46)$$

atau

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

$$\begin{bmatrix} \dot{x}_7 \\ \dot{x}_8 \\ \dot{x}_9 \\ \dot{x}_{10} \\ \dot{x}_{11} \\ \dot{x}_{12} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_7 \\ x_8 \\ x_9 \\ x_{10} \\ x_{11} \\ x_{12} \end{bmatrix} + \begin{bmatrix} 0 \\ -\frac{(I_{ZZ}-I_{YY})}{I_{XX}} x_{10}x_{12} \\ 0 \\ -\frac{(I_{XX}-I_{ZZ})}{I_{YY}} x_8x_{12} \\ 0 \\ -\frac{(I_{YY}-I_{XX})}{I_{ZZ}} x_8x_{10} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{l}{I_{XX}} U_2 \\ 0 \\ \frac{l}{I_{YY}} U_3 \\ 0 \\ \frac{1}{I_{ZZ}} U_4 \end{bmatrix} \quad (2.47)$$

dimana

$$\begin{aligned} x_7 &= \phi, & x_8 &= \dot{\phi}, \\ x_9 &= \theta, & x_{10} &= \dot{\theta}, \\ x_{11} &= \psi, & x_{12} &= \dot{\psi} \\ \dot{\phi} &= p, & \ddot{\phi} &= \dot{p} \\ \dot{\theta} &= q, & \ddot{\theta} &= \dot{q} \\ \dot{\psi} &= r, & \ddot{\psi} &= \dot{r} \end{aligned}$$

Keterangan:

U_1 = gaya angkat/*thrust*

U_2 = gaya untuk melakukan *roll*

U_3 = gaya untuk melakukan *pitch*

U_4 = gaya untuk melakukan *yaw*

Φ = posisi sudut *roll*

Θ = posisi sudut *pitch*

ψ = posisi sudut *yaw*

p = kecepatan sudut *roll*

q = kecepatan sudut *pitch*

r = kecepatan sudut *yaw*

\dot{p} = percepatan sudut *roll*

\dot{q} = percepatan sudut *pitch*

\dot{r} = percepatan sudut *yaw*

m = massa total *quadrotor*

I_{xx} = momen inersia *frame* pada sumbu X

I_{yy} = momen inersia *frame* pada sumbu Y

I_{zz} = momen inersia *frame* pada sumbu Z

J_{TP} = momen inersia total pada sumbu *propeller*

2.3.1 Konstanta Inersia (I_{xx} , I_{yy} , dan I_{zz})[9]

Momen inersia adalah ukuran kelembaman suatu benda untuk berotasi terhadap porosnya. Momen inersia berperan dalam dinamika rotasi seperti massa dalam dinamika dasar, dan menentukan hubungan antara momentum sudut dan kecepatan sudut, momen gaya dan percepatan sudut, dan beberapa besaran lain. Sesuai dengan dasar teori yang telah di jelaskan sebelumnya, maka dapat dihitung nilai parameter I_{xx} , I_{yy} , dan I_{zz} dengan mengasumsi dari bentuk Quadcopter. Persamaan yang digunakan untuk menghitung I_{xx} , I_{yy} , dan I_{zz} akan di tunjukkan pada persamaan di bawah ini [3].

$$I_{xx} = \frac{mr^2}{4} + \frac{mh^2}{6} + 2mr^2 + \frac{MR^2}{4} + \frac{MH^2}{12} \quad (2.48)$$

$$I_{yy} = \frac{mr^2}{4} + \frac{mh^2}{6} + 2mr^2 + \frac{MR^2}{4} + \frac{MH^2}{12} \quad (2.49)$$

$$I_{zz} = \frac{MR^2}{2} + 4mr^2 \quad (2.50)$$

Untuk medapatkan nilai I_{xx} , I_{yy} , dan I_{zz} , maka harus diketahui dahulu massa, tinggi, dan jari-jari pada Quadcopter.

Tinggi, massa dan jari-jari pada Quadcopter adalah tinggi (H) = 0.137 m, massa (M) = 0.66 Kg, jari-jari (R) = 0.06 m. Sedangkan untuk tinggi, massa dan jari-jari pada motor Quadcopter adalah tinggi (h) = 0.034 m, massa (m) = 0.075 Kg, jari-jari (r) = 0.014

Dengan memasukkan nilai massa, tinggi, dan jari-jari sesuai nilai diatas ke dalam persamaan I_{xx} , I_{yy} , dan I_{zz} , maka bisa di dapatkan nilai parameter I_{xx} adalah $1,680157 \times 10^{-3} \text{ Kgm}^2$, I_{yy} adalah $1,680157 \times 10^{-3} \text{ Kgm}^2$, dan I_{zz} adalah $1,20466 \times 10^{-3} \text{ Kgm}^2$

2.3.2 Konstanta Inersia Motor

Untuk mengetahui nilai momen inersia pada motor Quadcopter, maka digunakan rumus pada persamaan 2.51 [9].

$$I_r = \frac{4(m \times r^2)}{2} \quad (2.51)$$

Dengan memasukkan nilai panjang, massa, dan jari-jari motor sesuai gambar diatas ke dalam Persamaan 2.46, maka di dapatkan nilai dari momen inersia motor adalah $0,025872 \text{ Kgm}^2$

Dari model matematik yang diperoleh dengan pemodelan fisis, maka konstanta yang harus diperoleh untuk menentukan parameter *plant* Quadcopter akan ditampilkan pada Tabel 2.1.

Tabel 2.1 Identifikasi Konstanta[3]

| KONSTANTA | NILAI | SATUAN |
|---|---------------------------|-------------------|
| Massa Quadcopter | 1,2 | Kg |
| Momen inersia rotasi sumbu X (I_{xx}) | $1,680157 \times 10^{-3}$ | Kgm^2 |
| Momen inersia rotasi sumbu Y (I_{yy}) | $1,680157 \times 10^{-3}$ | Kgm^2 |
| Momen inersia rotasi sumbu Z (I_{zz}) | $1,20466 \times 10^{-3}$ | Kgm^2 |
| Momen inersia motor <i>propeller</i> (J_{TP}) | 0,025872 | Kgm^2 |
| Konstanta Thrust | $2,2478 \times 10^{-6}$ | N.sec^2 |
| Konstanta Drag | $2,5 \times 10^{-7}$ | Nm.sec^2 |
| Panjang dari motor ke pusat Quadcopter | 0.06 | m |

2.4. Pengendali Quadcopter

Banyak pengendali yang dapat digunakan pada Quadcopter. Mulai dari kendali cerdas, kendali optimal, kendali adaptif dan kendali konvensional. Dibawah ini akan dijelaskan mengenai pengendali yang akan digunakan dalam tugas akhir ini.

2.4.1. Logika Fuzzy

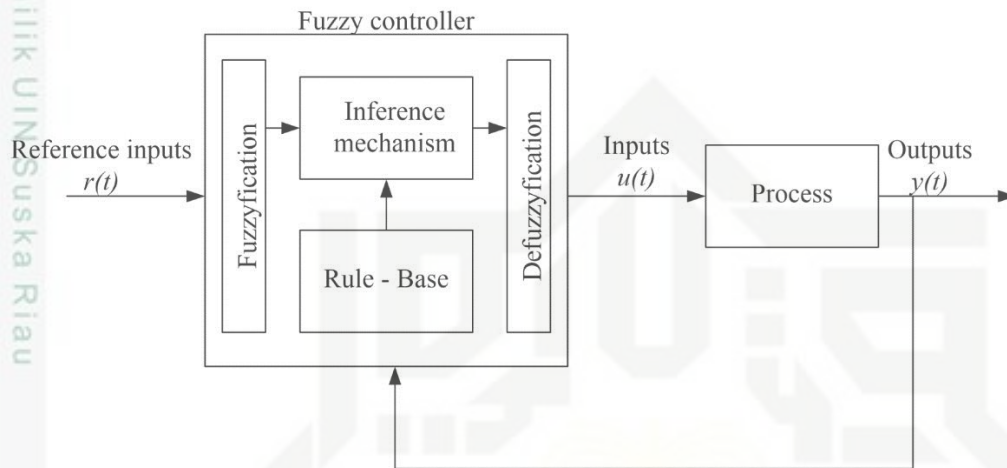
Sebelum logika Fuzzy diperkenalkan, orang telah mengetahui konsep logika klasik yang membagi sifat parameter menjadi dua hal yang berlawanan secara tegas, seperti benar atau salah, 0 atau 1. Konsep ini ternyata memiliki kekurangan dalam penerapan di kehidupan manusia, karena manusia lebih mengenal sesuatu dengan tidak eksak atau samar. Seperti penilaian terhadap penentuan nilai suatu apakah tidak enak, cukup enak, enak dan sangat enak.

Tujuan utama dari logika Fuzzy adalah memetakan sebuah ruang input kedalam sebuah ruang output, dan mekanisme utama untuk melakukan ini adalah kumpulan dari pernyataan *if-then* yang disebut dengan *rules* (aturan). Semua *rule* dievaluasi secara paralel, dan orde dari *rule* tersebut tidaklah penting. *Rule* ini sangat berguna karena mengacu pada kata sifat yang menggambarkan variabel tersebut[11].

Logika Fuzzy pertama kali diperkenalkan oleh Prof. Lotfi A. Zadeh pada tahun 1965. Dasar logika Fuzzy adalah teori himpunan Fuzzy, peranan derajat keanggotaan sebagai penentu keberadaan elemen dalam suatu himpunan sangatlah penting[11].

2.4.1.1. Aturan Dasar Logika Fuzzy

Dalam logika Fuzzy struktur umum kendali logika Fuzzy terdiri dari : *fuzzification*, penalaran (sistem inferensi Fuzzy), dan *defuzzification*.



Gambar 2.13 Struktur Umum Kendali Logika Fuzzy[5]

2.4.1.2. Fuzzyfikasi

Fuzzifikasi adalah proses mengubah nilai *crisp*/tegas variabel input menjadi nilai linguistik yang berupa derajat keanggotaan (*degree of membership*) dari setiap himpunan Fuzzy (Fuzzy set) variabel input. Untuk mengubah *crisp* input menjadi Fuzzy input, terlebih dahulu harus menentukan fungsi keanggotaan (*membership function*) untuk *crisp* input, kemudian proses fuzzifikasi akan mengubah *crisp* input dan membandingkan dengan fungsi keanggotaan (*membership function*) yang telah ada untuk menghasilkan harga Fuzzy input [11].

2.4.1.3. Knowledge Base

Knowledge base merupakan komponen yang terpenting dalam merancang logika Fuzzy. Secara umum, *knowledge base* hanya terdiri dari aturan-aturan dasar (*rule base*) yang berfungsi untuk memenuhi tujuan perancangan logika Fuzzy. Pada pengendali logika Fuzzy, *knowledge base* berkembang menjadi dua bagian yaitu: *data base* dan Fuzzy *rulesbase*.

1. *Data Base*

Pada bagian *data base*, ada empat prinsip dalam mendesain parameter dari pengendali Fuzzy yaitu : diskritisasi dan normalisasi dari *universe of discourses*, partisi input dan output Fuzzy, dan fungsi keanggotaan dari himpunan Fuzzy.

a) Diskritisasi *universe of discourse*

Universe of discourses dapat bersifat kontinyu atau diskrit. Jika *universe* bersifat kontinyu, maka diperlukan proses diskritisasi untuk menjadikan *universe* tersebut bersifat diskrit. Proses diskritisasi juga sering disebut dengan kuantisasi. Kuantisasi merupakan proses mengubah *universe* menjadi segmen – segmen penomoran. Jumlah *level* kuantisasi merupakan hal penting yang dapat mempengaruhi performansi logika Fuzzy, sehingga diperlukan jumlah kuantisasi yang cukup besar untuk memberikan performansi yang memadai. Perlu pertimbangan dalam penentuan jumlah *level* kuantisasi dalam hal kualitas performansi Fuzzy dan kapasitas memori pada komputer.

Untuk proses diskritisasi, dibutuhkan skala pemetaan yang berfungsi untuk mengubah variabel yang diukur kedalam nilai diskrit. Pemetaan dapat berupa *uniform (linear)*, *nonuniform (nonlinear)*, atau keduanya.

b) Normalisasi *universe of discourse*

Proses ini merupakan proses diskritisasi kedalam rentang normalisasi. *Universe* normalisasi terdiri dari segmen – segmen yang mempunyai nilai terbatas.

c) Partisi ruang input dan output Fuzzy

Sebuah variabel linguistik dalam *antecedent* dari sebuah *rule base* akan membentuk suatu ruang input Fuzzy, sementara itu *consequent* dari sebuah *rule base* membentuk ruang output Fuzzy. Secara umum, variabel linguistik dikaitkan dengan himpunan kata (*term set*). Partisi Fuzzy berfungsi untuk menentukan berapa banyak kata yang seharusnya berada pada sebuah himpunan kata. Terdapat tujuh kata linguistik yang sering digunakan dalam Fuzzy *inference* :

NB: *negative big*

NM: *negative medium*

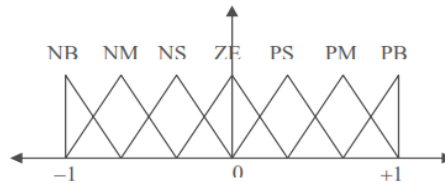
NS: *negative small*

ZE: *zero*

PS: *positive small*

PM: *positive medium*

PB: positive big



Gambar 2.14 Contoh partisi Fuzzy dengan kata linguistik

d) Fungsi keanggotaan dari himpunan Fuzzy

Terdapat beberapa tipe fungsi keanggotaan seperti bentuk segitiga, trapesium dan bell. Sebagai contoh, istilah NM didefinisikan seperti :

$$\mu_{NM}(-6) = 0.3$$

$$\mu_{NM}(-5) = 0.7$$

$$\mu_{NM}(-4) = 1.0$$

$$\mu_{NM}(-3) = 0.7$$

$$\mu_{NM}(-2) = 0.3$$

2. Rule base

Rule base merupakan sebuah cara untuk menyatakan suatu pernyataan kondisi berdasarkan pengetahuan manusia yang direpresentasikan dalam istilah linguistik.

Pernyataan kondisi ini dinyatakan dalam bentuk :

IF premise (antecedent), THEN conclusion (consequent)

Bentuk ekspresi diatas lebih dikenal sebagai bentuk *IF-THEN rule base*. Fuzzy rule memiliki dua bagian yang berbeda : evaluasi *antecedent* dari sebuah *rule* (bagian IF dari *rule*) dan *implication* atau menerapkan hasil evaluasi ke *consequent rule* tersebut (bagian THEN dari *rule*).

Dalam menentukan *rule base* yang digunakan, maka diperlukan metode untuk menentukan *rule base* yang mampu memenuhi kriteria perancangan logika Fuzzy. Terdapat dua cara pendekatan untuk menurunkan *rule base* Fuzzy. Pertama, metode heuristik, yang mana *rule base* dibentuk dengan menganalisa perilaku dari proses yang dikendalikan. Penurunan ini mengandalkan pengetahuan kualitatif dari perilaku proses. Pendekatan kedua yaitu metode penentuan, yang mana metode ini dapat menentukan secara sistematis struktur linguistik dari *rule base*. Berikut ini merupakan empat mode untuk menurunkan *rule base* Fuzzy. Empat mode ini memang tidak begitu eksklusif, tetapi

mode ini diperlukan untuk digabungkan sehingga didapatkan sebuah sistem yang mampu bekerja efektif.

- Pengalaman para pakar dan pengetahuan *control engineer*: pengoperasian manual dan metode kuisisioner
- Berdasarkan aksi operator kendali: pengamatan terhadap manusia sebagai pengendali dalam hal pengoperasian input dan output data.
- Berdasarkan model Fuzzy dari sebuah proses: gambaran linguistik karakter dinamis dari suatu proses.
- Berdasarkan pembelajaran: kemampuan untuk memodifikasi *rulebase* seperti pengendali *self-organizing*.

2.3.1.4 Sistem Inferensi Fuzzy

Dalam membangun sebuah sistem Fuzzy dikenal beberapa metode penalaran yaitu metode Tsukamoto, metode Mamdani (*max – min*) dan metode Takagi Sugeno.

1. Metode Tsukamoto

Metode Tsukamoto merupakan perluasan dari penalaran monoton, pada metode Tsukamoto, setiap konsekuen pada aturan yang berbentuk *If-Then* harus direpresentasikan dengan suatu himpunan Fuzzy dengan fungsi keanggotaan yang monoton. Sebagai hasilnya, output hasil penarikan kesimpulan (*inference*) dari tiap-tiap aturan diberikan secara tegas (*crisp*) berdasarkan α -predikat (*fire strength*). Hasil akhir diperoleh dengan menggunakan rata-rata berbobot (*weight average*) [12].

2. Metode Mamdani (*Max – Min*)

Metode Mamdani sering juga dikenal dengan nama Metode *Max-Min*. Metode ini diperkenalkan oleh Ebrahim Mamdani pada tahun 1975 [12]. Untuk mendapatkan output, diperlukan 4 tahapan :

1. Pembentukan himpunan Fuzzy.

Pada proses fuzzifikasi langkah pertama adalah menentukan *variable* Fuzzy dan himpunan Fuzzy. Kemudian tentukan derajat kesepadanan (*degree of match*) antara data masukan Fuzzy dengan himpunan Fuzzy yang didefinisikan untuk setiap variabel masukan sistem dari setiap aturan Fuzzy. Pada metode mamdani,

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

baik variabel input maupun variabel output dibagi menjadi satu atau lebih himpunan Fuzzy [12].

2. Aplikasi fungsi implikasi

Pada metode mamdani, fungsi implikasi yang digunakan adalah *Min*. Lakukan implikasi Fuzzy berdasar pada kuat penyulutan dan himpunan Fuzzy terdefinisi untuk setiap variabel keluaran di dalam bagian konsekuensi dari setiap aturan. Hasil implikasi Fuzzy dari setiap aturan ini kemudian digabungkan untuk menghasilkan keluaran infrensi Fuzzy [12].

3. Komposisi Aturan

Tidak seperti penalaran monoton, apabila sistem terdiri dari beberapa aturan, maka inferensi diperoleh dari kumpulan dan korelasi antar aturan. Ada 3 metode yang digunakan dalam melakukan inferensi sistem Fuzzy, yaitu: *max*, *additive* dan probabilistik OR [12].

4. Penegasan (*deFuzzy*)

Input dari proses defuzzifikasi adalah suatu himpunan Fuzzy yang diperoleh dari komposisi aturan-aturan Fuzzy, sedangkan output yang dihasilkan merupakan suatu bilangan pada domain himpunan Fuzzy tersebut. Sehingga jika diberikan suatu himpunan Fuzzy dalam *range* tertentu, maka harus dapat diambil suatu nilai *crisp* tertentu sebagai *output* [12].

3. Metode Sugeno

Penalaran dengan metode Sugeno hampir sama dengan penalaran Mamdani, hanya saja *output* (konsekuen) sistem tidak berupa himpunan Fuzzy, melainkan berupa konstanta atau persamaan linear. Metode ini diperkenalkan oleh Takagi Sugeno Kang pada tahun 1985, sehingga metode ini sering juga dinamakan dengan metode TSK [12].

2.3.1.5 DeFuzzyfikasi (Penegasan)[12]

DeFuzzyfikasi merupakan langkah terakhir dalam proses sistem Fuzzy. Hasil dari deFuzzyfikasi ini merupakan langkah yang diambil oleh sistem kendali Fuzzy. Pada proses ini hasil dari inferensi Fuzzy diubah kebentuk tegas (*crisp*). Ada beberapa metode deFuzzyfikasi antara lain [12] :

1. Metode *Centroid*

Metode *centroid* dilakukan dengan mengambil titik pusat daerah Fuzzy. Secara umum dirumuskan dengan :

$$z = \frac{\int z\mu(z)dz}{\int \mu(z)dz} \text{ atau } z = \frac{\sum_{j=1}^n z_j\mu(z_j)}{\sum_{j=1}^n \mu(z_j)} \quad (2.52)$$

2. Metode *Bisector*

Metode *bisector* dilakukan dengan mengambil nilai dari domain Fuzzy yang memiliki nilai keanggotaan setengah dari nilai keanggotaan Fuzzy. Secara dirumuskan dengan :

$$z_p \text{ _ sedemikian _ hingga _ } \int_{R_1}^p \mu(z)dz = \int_p^{R_n} \mu(z)dz \quad (2.53)$$

3. Metode *Mean of Maximum (MOM)*

Metode MOM dilakukan dengan mengambil nilai rata-rata domain yang memiliki nilai keanggotaan maksimum.

4. Metode *Largest of Minimum (LOM)*

Metode LOM dilakukan dengan mengambil nilai terbesar dari domain yang memiliki nilai keanggotaan maksimum.

5. Metode *Smallest of Maximum (SOM)*

Metode LOM dilakukan dengan mengambil nilai terkecil dari domain yang memiliki nilai keanggotaan maksimum.

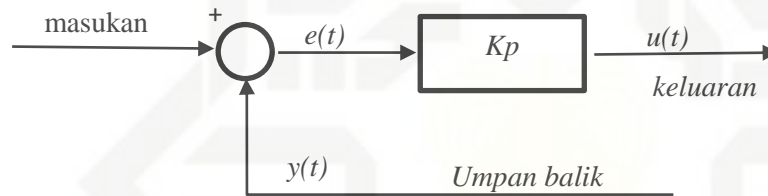
2.4.2. *Proportional, Integral, Derivative (PID)*[13]

Setiap kekurangan dan kelebihan dari masing-masing kendali Proporsional, Integral, dan Diferensial dapat saling menutupi dengan menggabungkan ketiganya menjadi kendali proporsional *plus* integral *plus* diferensial (kendali PID). Elemen-elemen

pengendali P, I, dan D masing-masing secara keseluruhan bertujuan untuk mempercepat reaksi sebuah sistem, menghilangkan *offset* dan menghasilkan perubahan awal besar.

2.4.2.1. Proportional

Kendali proposional memiliki keluaran yang sebanding atau proposional dengan besarnya sinyal kesalahan (selisih antara besaran yang diinginkan dengan harga aktualnya). Secara lebih sederhana dapat dikatakan, bahwa keluaran kendali proporsional merupakan perkalian antara konstanta proporsional dengan masukannya. Perubahan pada sinyal masukan akan segera menyebabkan sistem secara langsung mengubah keluarannya sebesar konstanta pengalinya. Blok diagram kendali proporsional ini ditunjukkan pada Gambar 2.15. Persamaan matematis kendali proporsional ditunjukkan pada Persamaan 2.59.



Gambar 2.15. Diagram Blok Kendali *Proportional*[13]

$$u(t) = K_p e(t) \tag{2.54}$$

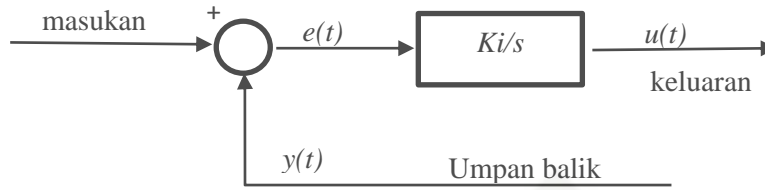
Dimana K_p adalah konstanta proporsional. Ciri-ciri kendali proporsional harus diperhatikan ketika kendali tersebut diterapkan pada suatu sistem. Secara eksperimen, pengguna kendali proporsional harus memperhatikan ketentuan-ketentuan berikut ini:

1. Jika nilai K_p kecil, kendali proporsional hanya mampu melakukan koreksi kesalahan yang kecil, sehingga akan menghasilkan respon sistem yang lambat.
2. Jika nilai K_p dinaikkan, respon sistem menunjukkan semakin cepat mencapai keadaan mantapnya (*steady state*).
3. Jika jika nilai K_p diperbesar sehingga mencapai harga yang berlebihan, akan mengakibatkan sistem bekerja tidak stabil, atau respon sistem akan berosilasi.

2.4.2.2. Integral

Kendali integral berfungsi menghasilkan respon sistem yang memiliki kesalahan keadaan tunak (*error steady state*) nol. Jika sebuah *plant* tidak memiliki unsur *integrator*, kendali proporsional tidak akan mampu menjamin keluaran sistem dengan kesalahan

keadaan tunaknya nol. Dengan kendali integral, respon sistem dapat diperbaiki, yaitu mempunyai kesalahan keadaan mantapnya nol. Gambar 2.16, menampilkan diagram blok kendali integral dengan masukan $E(s)$ dan keluaran $U(s)$.



Gambar 2.16. Diagram Blok Kendali *Integral*[13].

Pengendali integral memiliki karakteristik seperti halnya sebuah integral. Keluaran pengendali sangat dipengaruhi oleh perubahan yang sebanding dengan nilai sinyal kesalahan. Jika sinyal kesalahan tidak mengalami perubahan, keluaran akan menjaga keadaan seperti sebelum terjadinya perubahan masukan. Persamaan matematis kendali integral ditunjukkan pada Persamaan 2.60.

$$u(t) = \frac{1}{T_i} \int_0^t e(t) dt \quad (2.55)$$

Konsekuensi dengan memperbesar *gain* integral atau memperkecil konstanta waktu integral dapat membuat sistem lebih cepat “mengejar” nilai keadaan tunak, namun sistem cenderung akan berosilasi. Dalam keadaan seperti ini, diperlukan sebuah metode yang dapat memperkecil amplitudo osilasi dan nilai *overshoot* dari respon. Ketika digunakan, pengendali integral mempunyai beberapa karakteristik, antara lain :

1. Keluaran pengendali membutuhkan selang waktu tertentu, sehingga pengendali integral cenderung memperlambat respon.
2. Ketika sinyal kesalahan berharga nol, keluaran pengendali akan bertahan pada nilai sebelumnya.
3. Jika sinyal kesalahant idak berharga nol, keluaran akan menunjukkan kenaikan atau penurunan yang dipengaruhi oleh besarnya sinyal kesalahan dan nilai K_i .
4. Konstanta integral K_i yang berharga besar akan mempercepat hilangnya *offset*.

Tetapi semakin besar nilai konstanta K_i akan mengakibatkan peningkatan osilasi dari sinyal keluaran pengendali.

2.4.2.3. Pengendali Derivative

Secara intuitif, respon sistem yang berosilasi disebabkan pada beberapa hal. Proses dinamik dari sebuah *plant* menyebabkan respon sebuah *plant* tidak langsung berubah

dengan adanya perubahan sinyal kendali, namun membutuhkan waktu proses. Waktu ini akan membuat sistem kendali mengalami keterlambatan untuk mengoreksi kesalahan. Untuk itu dibutuhkan pengendali yang dapat memprediksi kesalahan dari sebuah sistem seperti pada Gambar 2.17. Nilai kesalahan prediksi dapat diturunkan dari persamaan Taylor sebagai berikut.

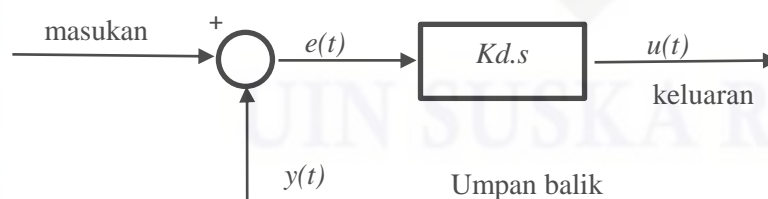
$$e(t + Td) \approx e(t) + Td \frac{de(t)}{dt} \tag{2.56}$$

Dengan mengatur gradien dari garis prediksi, maka akan diperoleh kesalahan prediksi yang lebih akurat, dan persamaan menjadi:

$$u(t) = K_p (e(t) + Td \frac{de(t)}{dt}) \tag{2.57}$$

Sebagai validasi, seperti yang telah dipaparkan bahwa osilasi dari respon sistem terjadi karena sinyal kontrol berlebihan yang diberikan ke *plant*. Maka diperlukan mekanisme untuk mengurangi nilai sinyal kontrol yang berlebihan tersebut. Kontroler derivatif yang ditambahkan pada kontroler proporsional dapat mengurangi aksi kontrol yang berlebihan.

Dapat dilihat bahwa keadaan awal saat sistem memiliki kesalahan positif maka kontroler proporsional akan berusaha mengejar nilai *setpoint*. Dengan adanya respon tersebut, maka perubahan kesalahan proporsional dengan negatif kesalahan. Hal ini akan membuat kontroler derivatif akan berusaha meredam aksi kontroler proporsional yang berlebihan. Berkurangnya nilai sinyal kontrol akan berdampak pada kecepatan respon sistem yang semakin berkurang. Gambar 2.17 menampilkan diagram blok kontrol derivative.



Gambar 2.17. Diagram Blok Kendali *Derivatif*[13]

Ketika digunakan, kontroler derivatif mempunyai beberapa karakteristik, antara lain:

1. Kontroler ini tidak dapat menghasilkan keluaran bila tidak ada perubahan pada masukannya (berupa sinyal kesalahan).

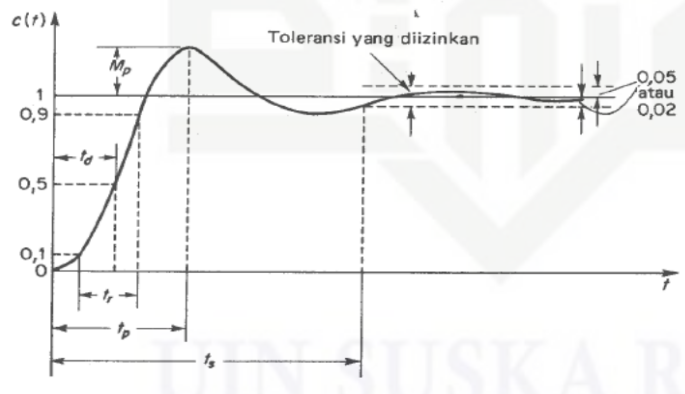
2. Jika sinyal kesalahan berubah terhadap waktu, maka keluaran yang dihasilkan kontroler tergantung pada nilai Kd dan laju perubahan sinyal kesalahan.
3. Kontroler derivatif mempunyai suatu karakter untuk mendahului, sehingga kontroler ini dapat menghasilkan koreksi yang signifikan sebelum pembangkit kesalahan menjadi sangat besar. Jadi kontroler derivatif dapat mengantisipasi pembangkit kesalahan, memberikan aksi yang bersifat korektif, dan cenderung meningkatkan stabilitas sistem.

2.5. Identifikasi Sistem

Untuk mengidentifikasi sistem *rotary inverted pendulum* yang digunakan mengacu pada respon transien sistem dalam kondisi *close loop*.

2.5.1 Respon Transien Sistem Orde Kedua

Metode ini digunakan untuk mengidentifikasi sistem berdasarkan pengamatan grafis terhadap masukan *step*. Dimana sinyal uji diberikan kepada sistem untuk mengetahui respon sistem *open loop*. Dari respon sistem tersebut, diketahui bahwa terdapat karakteristik-karakteristik penting dari sistem. Sistem yang digunakan pada rotasi Quadcopter merupakan sistem orde kedua.[8][14][15]



Gambar 2.18 Respon Sistem Orde Kedua[15].

Respon transien sistem orde kedua:

1. Waktu tunda atau *delay time* (t_d) : Waktu yang dibutuhkan respon mulai $t=0$ sampai respon mencapai 50% dari nilainya pada keadaan tunak (*steady state*). Waktu tunda menyatakan besarnya faktor keterlambatan respon akibat proses *sampling*.

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

- a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
- b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

2. Waktu naik atau *rise time* (t_r) : Harga waktu yang menyatakan bahwa respon sistem telah naik dari 5% ke 95% atau 10% ke 90% dari nilai respon pada keadaan tunak (*steady state*).
3. Waktu puncak (t_p) : Waktu yang diperlukan respon untuk mencapai puncak pertama *overshoot* (puncak tertinggi diawal).
4. Maksimum *overshoot* (M_p) : Nilai yang menyatakan perbandingan antara nilai maksimum respon (*overshoot*) yang melampaui nilai *steady state* dibanding dengan nilai *steady state*. Besarnya persen *overshoot* dinyatakan sebagai berikut:

$$\% M_p = \frac{c(t_p) - c(\infty)}{c(\infty)} \times 100\% \quad (2.58)$$

Overshoot hanya terjadi jika sistem kurang teredam sehingga harga ($\zeta < 1$).

5. Waktu tunak atau *settling time* (t_s) : Waktu yang dibutuhkan kurva respon sistem untuk menetap pada daerah stabil. Rumus untuk mendapatkan nilai *settling time* (t_s)