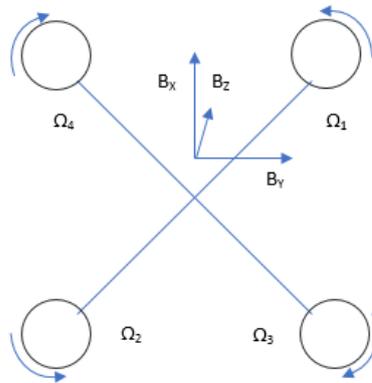


BAB II

TINJAUAN PUSTAKA

2.1 Konsep Dasar Quadcopter

Quadcopter merupakan kendaraan tanpa awak memiliki empat baling-baling yang terhubung melalui lengan yang saling bersilangan membentuk huruf “X”. Baling-baling ini memiliki *pitch* dan diameter tertentu sehingga saat digerakkan oleh motor elektrik menghasilkan gaya angkat yang lebih besar dari gaya hambatnya.



Gambar 2.1 Konsep Dasar *Quadcopter* dalam posisi *hover*

Arah depan pada *quadcopter* di Gambar (2.1) mewakili sumbu X pada badan *quadcopter*. Perputaran pada keempat baling-baling tidak semuanya sama yaitu pada lengan diagonal baling-baling 1 dan 2 akan berputar berlawanan arah jarum jam. Sedangkan pada baling-baling 3 dan 4 akan berputar searah jarum jam. Keuntungan dari arah perputaran baling-baling ini adalah untuk meniadakan efek momen inersia dan *quadcopter* dapat melayang stabil diudara. Dari kombinasi pergerakan keempat motor tersebut, *quadcopter* memiliki empat pergerakan sebagai berikut:

a) *Thrust*

Thrust adalah gaya yang menimbulkan *quadcopter* bergerak searah sumbu Z vertikal dari bumi. Untuk menimbulkan gaya *thrust*, kecepatan keempat motor harus sama dan secara bersamaan motor tersebut ditambah

kecepatannya hal ini akan menimbulkan gaya angkat lebih besar dan *quadcopter* akan bergerak naik keatas begitu juga sebaliknya.

b) *Roll*

Roll adalah torsi yang mengakibatkan *quadcopter* berputar disepanjang sumbu X dari bodinya. Hal ini terjadi ketika dua motor berputar lebih cepat dari dua motor lainnya. Misalnya jika *quadcopter* akan melakukan *roll* kearah kanan maka yang terjadi adalah dua motor sebelah kiri akan berputar lebih cepat dari dua motor sebelah kanan begitu juga sebaliknya.

c) *Pitch*

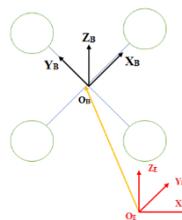
Pitch adalah torsi yang mengakibatkan *quadcopter* berputar disepanjang sumbu Y dari bodinya. Hal ini terjadi ketika dua motor berputar lebih cepat dari dua motor lainnya. Misalnya jika *quadcopter* akan melakukan *pitch* kedepan maka yang terjadi adalah dua motor bagian belakang akan berputar lebih cepat dari dua motor bagian depan begitu juga sebaliknya.

d) *Yaw*

Yaw adalah torsi yang mengakibatkan *quadcopter* berputar disepanjang sumbu Z dari bodinya. Pergerakan ini dipengaruhi oleh perubahan kecepatan dari keempat motor. Jika kecepatan motor depan dan belakang diperlambat sedangkan kecepatan motor kanan dan kiri dipercepat maka *quadcopter* akan bergerak menyimpang kekiri begitu pula sebaliknya.

2.1.1 Kinematika

Kinematika adalah cabang dari mekanika klasik yang membahas gerak benda dan sistem benda tanpa mempersoalkan gaya penyebab Gerakan [2]. *Quadcopter* juga dipelajari tentang kinematika, sehingga untuk mempermudah analisa *quadcopter* memiliki keluaran 6 *degree of freedom* (DOF). 6 DOF *rigid-body* di diskripsikan dalam dua buah *frame*. *Frame* yang digunakan sebagai referensi yaitu *earth frame* (*E-frame*) dan *body frame* (*B-frame*).



Gambar 2.2 *Quadcopter Frame*

Posisi linier *quadcopter* (Γ^E) ditentukan dari koordinat vektor 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 (2.1) dan (2.2)

$$r^E = [X \quad Y \quad Z]^T, \quad (2.1)$$

$$\Theta^E = [\phi \quad \theta \quad \psi]^T. \quad (2.2)$$

Salah satu yang diperlukan untuk mentransformasikan nilai dari kedua *frame* tersebut adalah dengan matriks rotasi. Matrik rotasi ini terdiri dari 3 buah matrik rotasi yang masing – masing berotasi terhadap sumbu X bumi (\square_{\square}), sumbu Y bumi (\square_{\square}), dan sumbu Z bumi (\square_{\square}).

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*.

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*.

c) Rotasi Sumbu Z

Rotasi sepanjang sumbu z dilambangkan dengan $R(\psi, z)$ yang merupakan rotasi *quadcopter* sepanjang sumbu z menghasilkan sudut yang diberi nama sudut *yaw*. Ketiga matriks rotasi dari tiap-tiap sumbu putar didapatkan persamaan matriks rotasi sebagai berikut:

$$R_{\Theta} = R(\psi)R(\theta)R(\phi). \quad (2.3)$$

Kecepatan pada *quadcopter* diekspresikan terhadap *body frame*. Kecepatan *quadcopter* terdiri dari kecepatan linier dan kecepatan angular yang disajikan dalam persamaan (2.4) dan (2.5)

$$V^B = [u \quad v \quad w]^T, \quad (2.4)$$

$$\omega^B = [p \quad q \quad r]^T. \quad (2.5)$$

Persamaan (2.6) merupakan kombinasi nilai linier dan angular untuk memberikan representasi pada *space* terhadap *E-frame*

$$\xi = [r^E \ \Theta^E]^T = [X \ Y \ Z \ \phi \ \theta \ \psi]^T. \quad (2.6)$$

Kemudian generalisasi dari vektor kecepatan linier *quadcopter* dan kecepatan angular *quadcopter* pada *B-frame* terdapat pada persamaan (2.7)

$$V = [V^B \ \omega^B]^T = [u \ v \ w \ p \ q \ r]^T. \quad (2.7)$$

Hubungan antara kecepatan linier dan *B-frame* dan salah satu faktor pada *E-frame* V^E dapat dilihat pada persamaan (2.8)

$$V^E = \Gamma^E = R_{\Theta} V^B. \quad (2.8)$$

Dimana R_{Θ} adalah matriks rotasi dari *B-frame* ke *E-frame*. Hubungan kecepatan angular pada *E-frame* (Θ^E) ke *B-frame* ω^B atau sebaliknya dapat dilihat pada persamaan (2.9) dan (2.10)

$$\omega^B = T_{\Theta}^{-1} \Theta^E, \quad (2.9)$$

$$\Theta^B = T_{\Theta} \omega^E. \quad (2.10)$$

Ketika pada saat *quadcopter* terbang akan menghasilkan matriks rotasi yaitu transformasi *B-Frame* ke *E-Frame* menggunakan matriks transfer T_{θ} pada persamaan (2.11)

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = T_{\theta}^{-1} \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix}. \quad (2.11)$$

Persamaan (2.12) matriks transformasi dari *B-Frame* menuju *E-Frame*

$$T_{\theta}^{-1} = \begin{bmatrix} 1 & 0 & -s_{\theta} \\ 0 & c_{\phi} & c_{\theta} s_{\phi} \\ 0 & -s_{\phi} & c_{\phi} c_{\theta} \end{bmatrix}, \quad (2.12)$$

$$T_{\theta}^{-1} = \begin{bmatrix} 1 & -s_{\phi} t_{\theta} & s_{\theta} t_{\theta} \\ 0 & c_{\phi} & -s_{\phi} \\ 0 & \frac{s_{\phi}}{c_{\theta}} & \frac{c_{\phi}}{c_{\theta}} \end{bmatrix}. \quad (2.13)$$

Persamaan yang sudah didapat akan dibentuk suatu hubungan antara kecepatan terhadap *E-frame* dan *B-frame* dapat dilihat pada persamaan (2.14)

$$\dot{\xi} = J_{\Theta} v. \quad (2.14)$$

Dimana $\dot{\xi}$ adalah vektor kecepatan yang mengacu pada *E-frame*, v adalah vektor kecepatan mengacu pada *B-frame* dan J_{Θ} adalah matriks *Jacobian*.

2.1.2 Dinamika

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 [3]. Isaac Newton menciptakan hukum-hukum fisika yang menjadi panduan dalam fisika dinamika. Secara umum, dinamika sangat berkaitan erat dengan hukum kedua Newton tentang gerak.

Persamaan (2.6)–(2.11) itu didapatkan persamaan umum *quadcopter* yang diturunkan dari metode *Euler-Newton*

$$\ddot{X} = \frac{U_1}{m} (\sin\psi \sin\phi + \cos\psi \sin\theta \cos\phi), \quad (2.15)$$

$$\ddot{Y} = \frac{U_1}{m} (-\cos\psi \sin\phi + \sin\psi \sin\theta \cos\phi), \quad (2.16)$$

$$\ddot{Z} = -g + \frac{U_1}{m} (\cos\theta \cos\phi), \quad (2.17)$$

$$\dot{p} = \frac{I_{YY} - I_{ZZ}}{I_{XX}} qr - \frac{J_{TP}}{I_{XX}} q\Omega + \frac{U_2}{I_{XX}}, \quad (2.18)$$

$$\dot{q} = \frac{I_{ZZ} - I_{XX}}{I_{YY}} pr - \frac{J_{TP}}{I_{YY}} p\Omega + \frac{U_3}{I_{YY}}, \quad (2.19)$$

$$\dot{r} = \frac{I_{XX} - I_{YY}}{I_{ZZ}} pr - \frac{U_4}{I_{ZZ}}. \quad (2.20)$$

Sedangkan input dari persamaan umum *quadcopter* (2.15)–(2.20) ditentukan dari persamaan (2.21)–(2.25) yang merupakan kecepatan baling-baling apabila terjadi gangguan dari luar

$$U_1 = b (\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2), \quad (2.21)$$

$$U_2 = bl (-\Omega_2^2 + \Omega_4^2), \quad (2.22)$$

$$U_3 = bl (-\Omega_1^2 + \Omega_4^2), \quad (2.23)$$

$$U_4 = d (-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2), \quad (2.24)$$

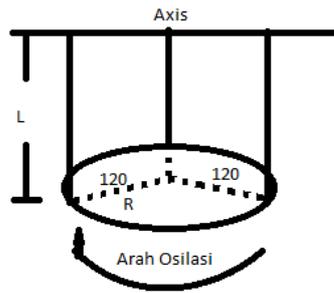
$$\Omega = -\Omega_1 + \Omega_2 - \Omega_3 + \Omega_4. \quad (2.25)$$

2.2 Properti Fisik *Quadcopter*

Properti fisik *quadcopter* merupakan nilai parameter yang dibutuhkan untuk perhitungan model matematis dari *quadcopter* seperti momen inersia. Perhitungan momen inersia diperlukan untuk mengetahui nilai I_{xx} , I_{yy} dan I_{zz} dari *plant quadcopter*. Ada dua metode untuk menghitung momen inersia dari *quadcopter* yang telah dibangun yaitu metode eksperimen dan metode analitis.

a. Metode Eksperimen

Metode eksperimen merupakan metode pengukuran karakteristik *plant* dari *quadcopter* menggunakan pengukuran secara fisik seperti pengukuran momen inersia dengan metode trifial pendulum. Metode trifial pendulum dipilih karena dalam implementasinya sangat mudah. Pengukuran dilakukan dalam sumbu X, Y dan Z.



Gambar 2.3 Teorema trifial pendulum

Setelah melakukan percobaan pada Gambar (2.3) sebanyak 10 kali osilasi maka dilakukan perhitungan pada I_{xx} , I_{yy} dan I_{zz} pada masing-masing sumbu *quadcopter*.

$$I_{xx,yy,zz} = \frac{M \cdot g \cdot R^2 T_{x,y,z}^2}{4 \cdot \pi^2 \cdot L} \quad (2.26)$$

M = Berat total *quadcopter* termasuk piringan pada trifial pendulum dalam kg

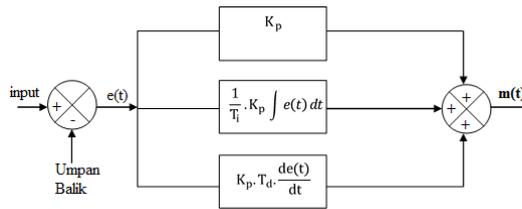
G = Gravitasi bumi 9.8 m/s

R = Jari-jari pada piringan trifial pendulum dalam meter

T = Periode satu osilasi dalam detik

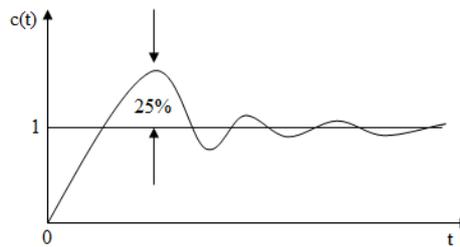
L = Panjang benang pengait piringan dalam meter

$I_{xx,yy,zz}$ = Massa momen inersia objek dalam sumbu x, y dan z



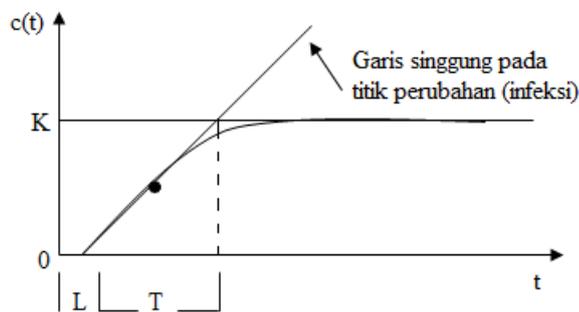
Gambar 2.5 Blok diagram kontroler PID

Ziegler-Nichols pertama kali memperkenalkan metodenya pada tahun 1942. Metode ini memiliki dua cara, metode osilasi dan kurva reaksi. Kedua metode ditujukan untuk menghasilkan respon sistem dengan lonjakan maksimum sebesar 25%. Gambar (2.6) memperlihatkan kurva dengan lonjakan 25%.



Gambar 2.6 Kurva respon rangga ratuan yang memperlihatkan 25% lonjakan maksimum.

Sistem kendali PID yang dibangun akan digunakan untuk mengatur *Pitch*, *Roll* dan *Yaw* dari *quadcopter*. Setelah mengetahui fungsi alih dari *plant quadcopter* maka dilakukan simulasi dengan menggunakan *software* MATLAB dengan tidak menambahkan kontroler untuk mengetahui *step* respon dari *plant* tersebut. Setelah itu penalaan dengan menggunakan metode Ziegler-Nichols dilakukan. Karena sistem ini merupakan rangkaian *close-loop* yang akan mendapatkan masukan berupa *error* dari sensor (*feedback*).



Gambar 2.7 Kurva respon berbentuk S

Tabel 2.1 Penalaan parameter PID dengan metode osilasi

Type Kontroller	Kp	T_i	Td
P	T/L	-	0
PI	0,9 T/L	L/0,3	0
PID	1,2 T/L	2L	0,5L

2.4 Model Linearisasi

Kinematika dan dinamika telah dijelaskan pada bab sebelumnya. Namun konsep yang paling penting dapat diringkas kedalam persamaan (2.30), (2.31) dan (2.32). Dinamika *quadcopter* harus disederhanakan untuk memberikan model yang mudah dan dapat diimplementasikan dalam algoritma kontrol [6]. Persamaan (2.18), (2.19) dan (2.20) dapat disusun ulang berdasarkan tiga pertimbangan sebagai berikut:

- Karena gerakan *quadcopter* dapat diasumsikan dekat dengan kondisi melayang, perubahan sudut kecil terjadi (terutama untuk *roll* dan *pitch*). Oleh karena itu istilah-istilah ini dapat disederhanakan.
- Mendefinisikan hubungan antara kecepatan sudut dalam *E-frame* dan yang ada dalam *B-frame*. Karena dalam kondisi melayang dekat dengan matriks identitas, persamaan percepatan telah dirujuk langsung ke percepatan sudut Euler.
- Algoritma kontrol keseluruhan digunakan untuk memberikan sinyal yang tepat kepada baling-baling. Karena *quadcopter* mempunyai empat baling-baling, tidak lebih dari empat variabel dapat dikontrol dalam *loop*. Menurut pilihan ini, persamaan yang menggambarkan posisi X dan Y telah dihapus.

$$\ddot{\phi} = \frac{U_2}{I_{xx}}, \quad (2.30)$$

$$\ddot{\theta} = \frac{U_3}{I_{yy}}, \quad (2.31)$$

$$\ddot{\psi} = \frac{U_4}{I_{zz}}. \quad (2.32)$$

2.5 Sistem Komputer Pendamping

Komputer pendamping merupakan *hardware* tambahan yang berfungsi untuk membuat keputusan cerdas. *Flight controller* saat ini mungkin dapat membuat keputusan namun terbatas hanya meliputi *autonomous* yang standar. Sebelum komputer pendamping, pengembang UAV tidak mempunyai pilihan lebih selain terpaku pada kode pengendali penerbangan yang terbatas [7]. Sekarang dengan hadirnya komputer pendamping, pengembang *drone* dapat membangun aplikasi *drone* canggih dengan intelijen *onboard*, konektivitas tanpa batas ke *cloud* melalui 4G / LTE, mengintegrasikan muatan *custom*, membangun & menghubungkan antarmuka *Web / Mobile custom*. Jalur komunikasi yang digunakan melalui jaringan kabel dengan modul *driver* FTDI yang dapat dilihat pada gambar (2.8) [8]. *Driver* ini digunakan untuk mengonversi saluran komunikasi serial ke *Universal Asynchronous Receiver-Transmitter (UART)* pada pengendali penerbangan.



Gambar 2.8 Sambungan Komunikasi Odroid ke Pixhawk

Dalam pengoperasiannya sistem operasi yang terpasang pada komputer pendamping yaitu linux ubuntu dan telah terinstal *mavproxy* dan *library dronekit*. *Mavproxy* atau MAVLink adalah protokol pengiriman pesan yang sangat ringan untuk berkomunikasi dengan *drone* dan antara komponen *drone* ke *onboard computer* sedangkan *dronekit* merupakan *platform open source* yang menangani algoritma penerbangan berbasis bahasa python. Berikut merupakan perintah dasar pada pemrograman python pada *library dronekit*:

```
“# mavproxy.py --master=/dev/ttyUSB0 --baudrate 1500000 --aircraft MyCopter”
```

Dimana *ttyUSB0* merupakan *port* modul *driver* FTDI, nilai 1500000 merupakan *baudrate* yang digunakan untuk komunikasi antara Odroid dan Pixhawk.