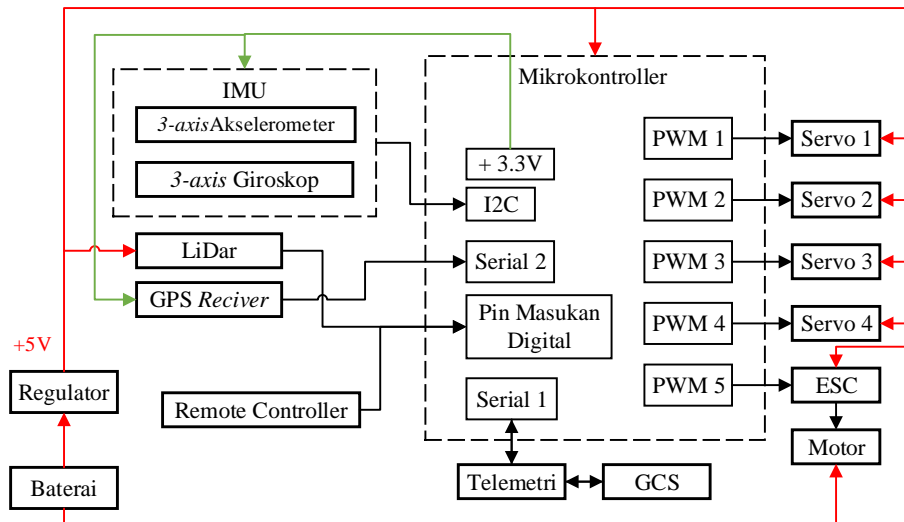


Speed Controller (ESC) yang mendapat masukan PWM dari mikrokontroler. Hubungan antara masukan pemroses dan keluaran arsitektur sistem dijelaskan pada Gambar 4.11



Gambar 4.11 Rancangan sistem elektronik

Berdasarkan rancangan sistem elektronik di atas, maka rancangan untuk skematik pada *board* dapat dibuat. Penelitian ini sensor IMU dan *magnetometer* menggunakan jalur I²C yang dihubungkan melalui SDA dan SCL pada mikrokontroler. Sensor GPS dihubungkan ke pin serial. Servo dan ESC untuk motor *brushless* dihubungkan ke pin PWM. Detail rancangan penggunaan pin terdapat pada Gambar 4.13

Tabel 4.6 Rancangan pin pada mikrokontroler

No	Pin	Keterangan
1.	D29	Pin mode control LiDAR
2.	D30	Pin mode control LiDAR
3.	D20	Pin PWM ESC motor 3
4.	D2	Pin servo aileron kanan
5.	D3	Pin servo elevator
6.	D4	Pin PWM ESC motor
7.	D5	Pin servo aileron kiri
6.	TX1	Pin komunikasi serial dengan telemetry 1
7.	RX1	Pin komunikasi serial dengan telemetry 1
8.	SDA0	Pin komunikasi I ² C HCM5883L dan MPU6050
9.	TX2	Pin komunikasi serial dengan GPS

Tabel 4.7 Rancangan pin pada mikrokontroler

No.	Pin	Keterangan
10.	RX2	Pin komunikasi serial dengan GPS
13.	SCL0	Pin komunikasi I ² C HCM5883L dan MPU6050
14.	D14	Pin <i>interrupt</i> MPU6050
15.	A1	Pin ADC sensor
16.	D36	Pin <i>receiver remote channel aileron</i>
17.	D35	Pin <i>receiver remote channel elevator</i>
18.	D34	Pin <i>receiver remote channel throttle</i>
19.	D33	Pin <i>receiver remote channel rudder</i>
20.	D32	Pin <i>receiver remote channel aux 5</i>
21.	D31	Pin <i>receiver remote channel aux 6</i>



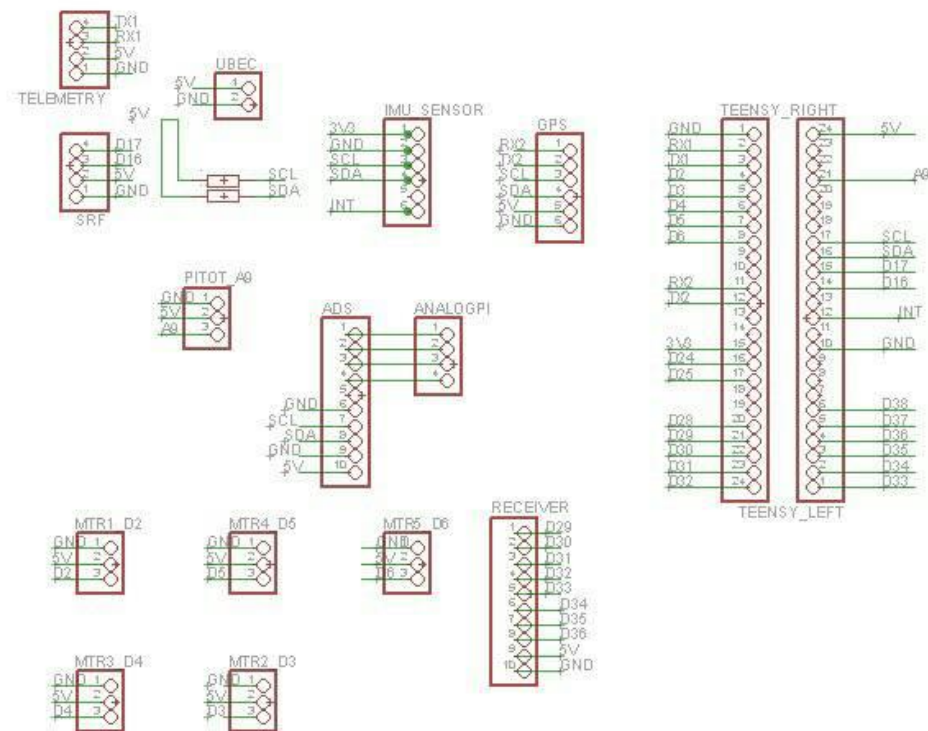
Gambar 4.12 Rancangan mekanik

4.8 Rancangan Program Sistem Penghindar Halangan

Rancangan program sistem penghindar halangan dibuat agar sistem dapat bekerja sesuai dengan analisis sistem yang telah dijabarkan pada subbab 4.3. Rancangan program berfungsi agar dapat mengimplementasikan sistem kendalinya yang berisi deklarasi variabel, fungsi *setup* dan fungsi lainnya. Diagram alir algoritme sistem kendali ditunjukkan pada Gambar 4.14

Program sistem penghindar halangan pada penelitian ini menggunakan bahasa pemrograman C/C++ yang akan berjalan pada modul mikrokontroler. Program utama pada pesawat tanpa awak sayap tetap ini berfungsi untuk mengolah instruksi agar wahana dapat menghindari halangan yang ada didepannya secara otomatis sesuai dengan urutan dan logika yang tertanam. Sub program pada perangkat lunak ini terdiri dari *setup*, sensor *fusion* dan kendali fase

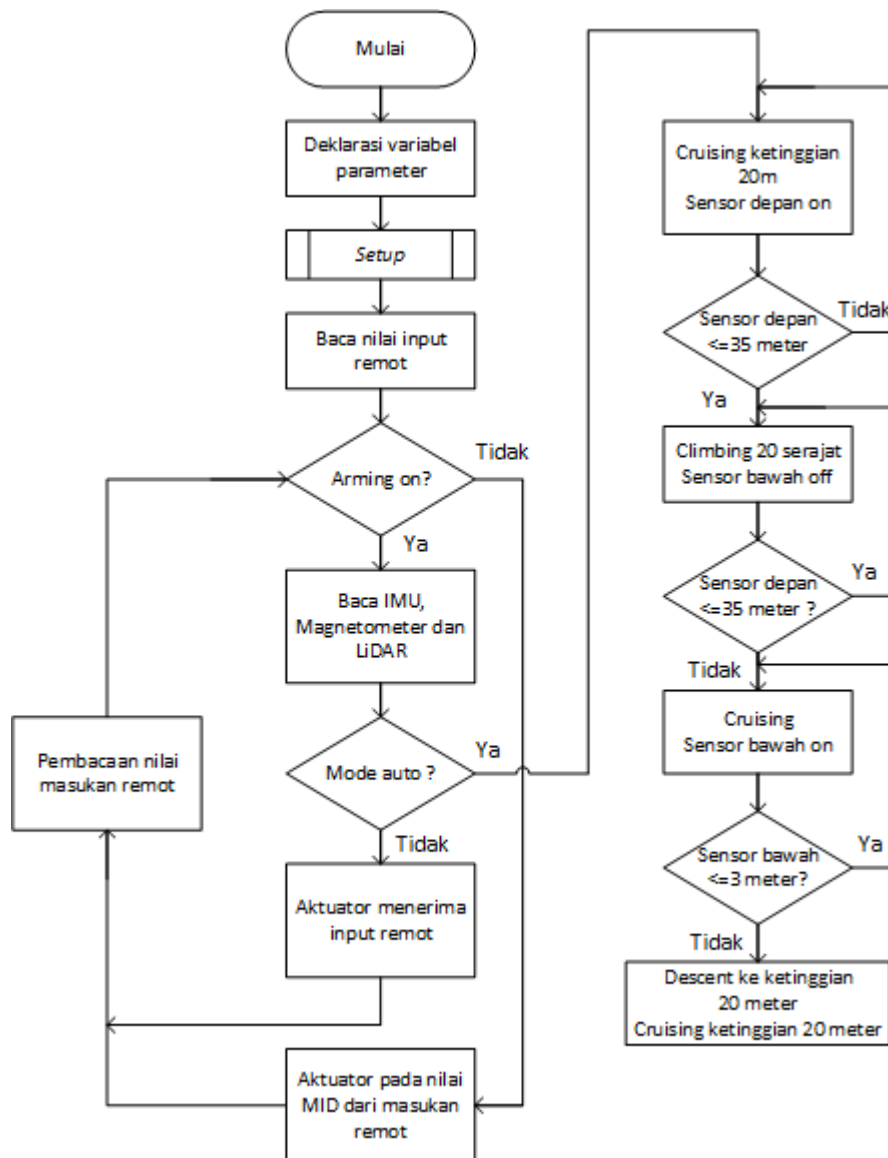
menghindari halangan. Program wahana akan selalu melakukan pengecekan perintah yang berasal dari *ground station*, saat *ground station* telah memberikan *mode auto*, maka wahana akan mulai berjalan sesuai prosedur kendali. Saat prosedur kendali aktif, sistem akan memberikan luaran melalui sensor lalu dibandingkan dengan *state* yang telah ditentukan, setelah itu sistem akan selalu menghitung seluruh kesalahan *state* sehingga didapatkan nilai masukan sistem wahana.



Gambar 4.13 Rancangan skematik elektronik wahana *fixed wing*

4.9 Rencana Pengujian Sistem

Dalam penelitian ini akan dilakukan tahap pengujian yaitu pengujian sensing wahana, pengujian kestabilan sikap *lateral* wahana, pengujian kestabilan sikap *longitudinal* wahana, pengujian kestabilan sikap rotasi wahana saat misi menghindari halangan.



Gambar 4.14 Bagan alir program penghindar halangan

4.9.1 Rancangan pengujian orientasi sikap wahana

Pengujian orientasi sikap wahana menggunakan sensor *fusion* yang dirancang dengan menggunakan metode DMP untuk menghasilkan nilai sudut orientasi wahana. Penggabungan dari nilai sudut *roll*, *pitch*, dan *yaw* digunakan untuk mengetahui orientasi dari wahana. Pengujian sudut kemiringan dilakukan pada bidang datar/*level*. Lalu wahana akan dimiringkan dan nilai kemiringan dihitung menggunakan *waterpass*. Hasil luaran dari perhitungan sensor *fusion* akan dibandingkan dengan luaran nilai yang sebenarnya untuk mendapatkan nilai

ralat *offset* dari hasil *sensing* wahana. Pengujian dilakukan pada masing-masing sumbu secara bergantian dengan menganggap sumbu lain tidak bergerak.

4.9.2 Rancangan pengujian anti-rotasi wahana

Pengujian anti-rotasi pada wahana dilakukan dengan 2 tahap, yaitu tahap uji statis dan dinamis. Pengujian statis dilakukan didarat dengan cara melihat respon defleksi *aileron*, *elevator*, dan *rudder* terhadap kemiringan yang diberikan pada wahana. Pada tahap uji statis dilakukan penalaan nilai **Q** dan **R** untuk anti rotasi menggunakan MATLAB. Nilai optimal *feedback gain* **K** yang didapat dari penalaan pada MATLAB akan dimasukkan kedalam program, lalu tahap pengujian dinamis akan dilakukan ketika wahana terbang sehingga dilihat apakah sistem kendali yang telah dibuat dapat menjaga kestabilan wahana atau tidak.

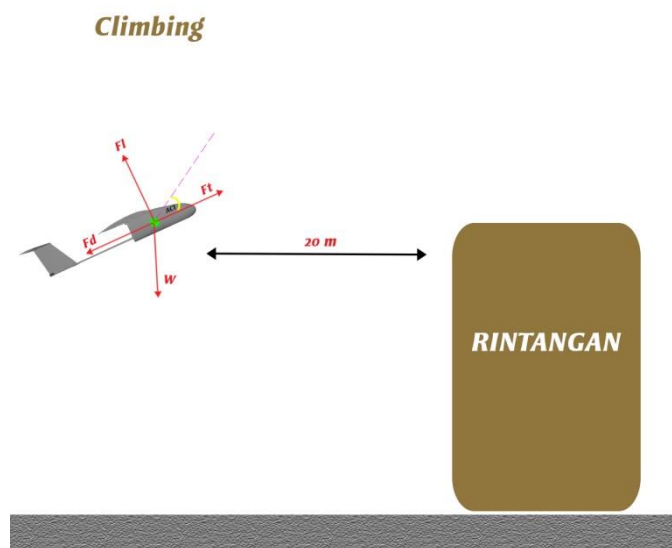
Nilai respon wahana terhadap sudut anti rotasi *roll*, *pitch*, dan *yaw* didapat melalui tahap pengujian dinamis. Nilai respon tersebut merupakan nilai *rise time*, *overshoot*, dan *steady state error*. Hasil yang telah didapat dari tahap uji dinamis akan disimpan dan diamati untuk analisa selanjutnya. Analisa berupa data grafik respon yang kemudian data grafik tersebut akan di validasi dengan hasil simulasi yang telah didapat untuk anti rotasi.

4.9.3 Rancangan pengujian fase menghindari halangan

Pengujian LQR pada fase menghindari halangan dilakukan dengan menerbangkan wahana lalu melihat apakah sistem kendali tersebut mampu mempertahankan ketinggian terbang serta mampu mempertahankan sikap wahana pada saat menghindari halangan.

Pengujian pertama dilakukan dengan menerbangkan wahana yang dikendalikan oleh *radio control* dengan mode *stabilize*. Mode tersebut digunakan agar wahana terbang lurus dan stabil. Selanjutnya, mode wahana diubah menjadi mode *auto*. Mode *auto* akan memberi perintah pada wahana untuk melakukan terbang jelajah lurus (*cruise*) dan data luaran ketinggian akan diamati, lalu data mempertahankan sikap anti rotasi dilihat secara langsung dari sikap terbang wahana.

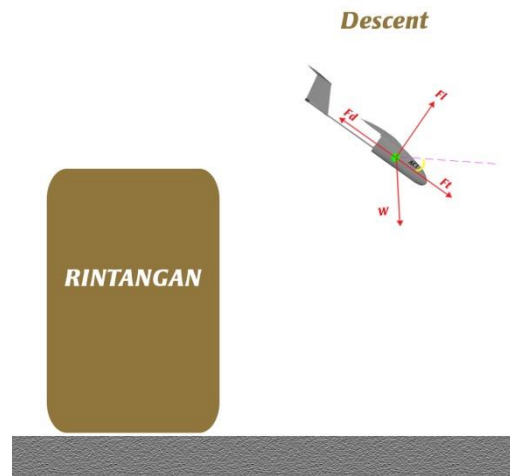
Pengujian kedua yaitu pengujian *climb* atau menanjak. Pengujian ini dilakukan dengan mengatur defleksi sudut *pitch* dengan variasi ketinggian yang berbeda agar dapat melihat respon sistem kendali dalam mempertahankan sudut *pitch* saat *climb*. Pengujian ini juga dilakukan untuk melihat kemampuan sistem kendali dalam mempertahankan sikap wahana saat menanjak. Ilustrasi pengujian kedua ini ditunjukkan pada Gambar 4.15



Gambar 4.15 Tahap *climbing*

Pengujian ketiga yaitu pengujian fase *descent* atau menukik ke bawah. Fase *descent* ini mengatur dan mengamati data defleksi sudut *pitch* serta ketinggian. Fase ini dilakukan agar dapat melihat respon sistem kendali pada saat mempertahankan sikap anti rotasi saat *descent* dan mempertahankan ketinggian. karena pada fase ini wahana akan menukik hingga ketinggian awal sesuai *set point*. Ilustrasi pengujian ini ditunjukkan pada Gambar 4.16.

Kemampuan wahana dalam respon ketinggian dan sikap mempertahankan kestabilan pada fase menghindari halangan akan diamati dan disimpan untuk selanjutnya divalidasi. Urutan dalam pengujian sistem dapat dilihat pada Tabel 4.8



Gambar 4.16 Tahap *descent*

Tabel 4.8 Rencana pengujian sistem

No	Rencana Pengujian	Tujuan Pengujian
1.	Pengujian Kalibrasi sensor	<ul style="list-style-type: none"> - Mendapat nilai ralat hasil luaran sudut orientasi wahana yg didapat dari sensor <i>fusion</i> dan nilai <i>offset</i> dari IMU. - Mencari tingkat presisi sensor ketinggian.
2.	Pengujian kestabilan gerak anti rotasi pada wahana	<ul style="list-style-type: none"> - Membandingkan nilai yang didapat dari hasil simulasi dengan pengujian terbang. - Mendapatkan respon wahana ketika melakukan gerakan rotasi yang terdiri dari <i>risetime</i>, <i>Settling time</i>, <i>overshoot</i> dan <i>steady state error</i> untuk gerakan rotasi <i>roll</i> (ϕ), <i>pitch</i> (θ), dan <i>yaw</i> (ψ) serta kecepatan angular ω_ϕ, ω_θ, ω_ψ.
3.	Pengujian fase menghindari halangan	<ul style="list-style-type: none"> - Mendapatkan respon wahana ketika gerakan <i>climbing</i> menghindari halangan. Selain itu mendapatkan nilai <i>overshoot</i> dan <i>rise time</i> yang dialami sistem, dan mendapatkan indeks performa wahana mengenai kemampuannya dalam menghindari halangan.

4.10 Kalibrasi Sensor Orientasi Wahana

Sebelum digunakan pada sistem, dilakukan kalibrasi terhadap sensor-sensor yang digunakan pada wahana. Orientasi wahana didapatkan menggunakan dua sensor yaitu sensor IMU dan LiDAR. Sensor IMU yang diakses menggunakan *Digital Motion Processing* (DMP) bertujuan untuk mendapatkan nilai sudut *yaw*, *pitch* dan *roll*. Kalibrasi dilakukan dengan membandingkan nilai keluaran hasil pembacaan DMP dengan waterpass sebagai kalibrator fisis dan gawai sebagai kalibrator elektronik. Wahana akan digantung pada salah satu sumbu secara bergantian dan menganggap sudut yang lain tidak bergerak. Selanjutnya wahana akan dimiringkan sesuai dengan tiga acuan sudut yang digunakan, yaitu sudut 0° , 45° , dan 90° .

4.11 Pengujian Gerak Anti-Rotasi Wahana

Pengujian sistem kendali LQR untuk gerak anti rotasi pada wahana dilakukan dengan pengujian dinamis. Pengujian statis dilakukan untuk melakukan penalaan nilai \mathbf{Q} dan \mathbf{R} pada sudut *roll* dan *pitch* dan *yaw*, namun pada penelitian ini untuk mempermudah penalaan maka yang ditala hanya nilai \mathbf{Q} nya saja dan \mathbf{R} diberi nilai 1. Sedangkan pengujian dinamis atau bergerak translasi horisontal dilakukan untuk mengamati sistem kendali LQR dalam menjaga sikap terbang wahana.

Pengujian statis dilakukan dengan melihat respon arah defleksi sayap *aileron* dan *elevator* terhadap arah kemiringan orientasi *fixed wing*. Jika arah defleksi sudah benar, maka pengujian dinamis dapat dilakukan.

Pengujian dinamis dilakukan dengan cara menerbangkan *fixed wing* yang kemudian akan dilakukan pengamatan terhadap sudut rotasi *roll* dan *pitch*. Hal ini dilakukan untuk melihat kestabilan yang dihasilkan dari nilai \mathbf{Q} yang telah didapatkan. Jika kestabilan belum sesuai, maka dapat dilakukan pengujian kembali untuk mencari nilai \mathbf{Q} yang tepat.

4.12 Pengujian Misi Menghindari Halangan

Pengujian sistem kendali LQR untuk kestabilan sikap saat melakukan *climbing* agar dapat menghindari halangan. Pengujian dilakukan secara manual menggunakan remot, wahana akan diterbangkan lurus dan pada jarak 35 meter dari halangan maka sang pilot akan menggerakkan *pitch* agar wahana dapat menghindari halangan tanpa menabrak. Pengujian dilakukan untuk mendapatkan nilai indeks performa wahana sebagai parameter kesuksesan wahana untuk menghindari halangan. Selama penerbangan berlangsung, dilakukan pengamatan apakah masih terjadi *overshoot* dan nilai *rise time* masih besar. Berdasarkan uraian rencana pengujian tersebut, rencana pengujian sistem dirangkum seperti pada Tabel 4.9

Tabel 4.9 Rencana Pengujian Sistem

No	Rencana Pengujian	Target Pengujian
1.	Pengujian gerak anti-rotasi wahana	Mendapatkan parameter kendali berupa <i>risetime</i> , <i>settling time</i> , <i>overshoot</i> , <i>undershoot</i> dan <i>steady state error</i> untuk gerak rotasi <i>roll</i> (ϕ), <i>pitch</i> (θ), kecepatan angular ω_ϕ , ω_θ , dan gerakan translasi sumbu x , y , kecepatan translasi horisontal sumbu x (v_x) dan y (v_y).
2.	Pengujian misi menghindari halangan	Mendapatkan respon wahana ketika gerakan <i>climbing</i> menghindari halangan. mendapatkan nilai <i>overshoot</i> dan <i>rise time</i> yang dialami sistem, dan mendapatkan indeks performa wahana mengenai kemampuannya dalam menghindari halangan.

BAB V

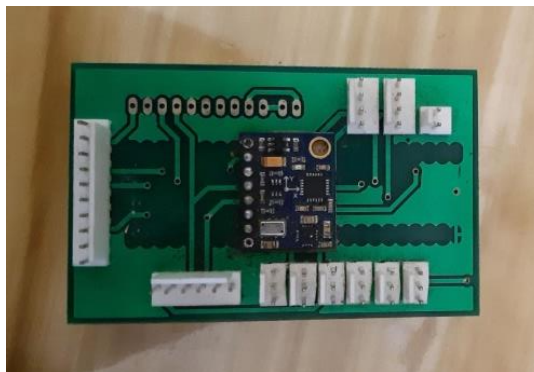
IMPLEMENTASI

Bab ini akan menjelaskan implementasi dari rancangan sistem yang telah dibuat pada bab empat.

5.1 Implementasi Perangkat Keras

5.1.1 Implementasi elektronik

Implementasi elektronik pada wahana terdiri dari Mikrokontroler, sensor IMU, dan sensor LiDAR. Implementasi elektronik ini diawali dengan pembuatan desain perangkat elektronik yaitu *board shield*. Desain *board shield* akan dicetak pada *Printed Circuit Board (PCB)*. Setelah dicetak pada PCB maka komponen pendukung seperti Mikrokontroler, IMU, dan *pin header* akan dipasang pada *board shield*. Board yang telah dicetak ditunjukkan pada Gambar 5.1



Gambar 5.1 *Flight Controller*

5.1.2 Implementasi mekanik

Wahana yang digunakan pada penelitian ini berbentuk pesawat tanpa awak sayap tetap berjenis *skywalker*. Rangka wahana yang terbuat dari *polyfoam* dengan penyambung menggunakan *carbon tube* antara badan dan ekor pesawat. Wahana yang dibuat memiliki panjang badan 133cm dan panjang sayap 140cm. Motor *brushless* terletak pada bagian belakang badan wahana sebagai penggerak utama. Peletakan elektronis disesuaikan pada titik berat wahana agar luaran sensor dapat menggambarkan keadaan yang sebenarnya.



Gambar 5.2 Gambar mekanis wahana

5.2 Penentuan Variabel Kendali

Penentuan parameter kendali merupakan syarat untuk melakukan simulasi sistem kendali. Parameter kendali yang dimaksud adalah dimensi dan inersia dari wahana. Inersia yang digunakan ada tiga yaitu sumbu x , y , dan z dari setiap komponen wahana. Setelah dilakukan perhitungan maka akan didapat hasil inersia pada sumbu x (I_{xx}), inersia pada sumbu y (I_{yy}), dan inersia pada sumbu z (I_{zz}).

5.3 Simulasi Sistem Kendali LQR pada MATLAB

Fungsi LQR pada Matlab mengolah *state space* yaitu matriks **A** dan **B** serta pembobotan LQR yaitu komponen **Q** dan **R** menghasilkan penguatan *full state feedback* **K**. Nilai penguatan **K** hanya dihitung menggunakan Matlab, hal ini disebabkan karena pembuatan *state space* serta perhitungan **K** merupakan perhitungan dengan komputasi yang berat sehingga tidak dapat diterapkan secara langsung pada mikrokontroler.

Tahap simulasi anti rotasi untuk mendapatkan nilai *gain* **K** yang sesuai dengan spesifikasi harus dilakukan penalaan nilai **Q**. Nilai **K** didapatkan dengan menggunakan fungsi `lqr`, proses perhitungan nilai **K** didapat dari penyelesaian persamaan (3.5). Bobot nilai pada matriks **Q** disesuaikan posisinya dengan setiap *state* wahana, sesuai dengan persamaan (4.11) yang ditunjukkan pada Gambar 5.3, lalu nilai *gain* **K** optimal dari hasil simulasi akan dimasukkan kedalam program utama dan dilakukan tahap validasi.

```

11. ....
12. A = [ 0      1      0      0      0 0;
13.      0      0      0 (Iyy-Izz)*r/Ixx 0 0;
14.      0      0      0      1      0 0;
15.      0 (Izz-Ixx)*r/Iyy 0      0      0 0;
16.      0      0      0      0      0 1;
17.      0 (Ixx-Iyy)*q/Izz 0      0      0 0];
18.
19. B = [ 0      0      0;
20.      1/Ixx 0      0;
21.      0      0      0;
22.      0      1/Iyy 0;
23.      0      0      0;
24.      0      0      1/Izz ];
25.
26. C = [1 0 0 0 0 0 ;
27.      0 0 1 0 0 0 ;
28.      0 0 0 0 1 0 ;
29.
30. D = zeros(3,3);
31. Q = [ 58 0 0 0 0 0 ;
32.      0 1.395 0 0 0 0 ;
33.      0 0 35 0 0 0 ;
34.      0 0 0 0.5 0 0 ;
35.      0 0 0 0 85 0 ;
36.      0 0 0 0 0 2.3];
37.
38. %Nilai fullstate feedback gain
39. R = eye(3);
40. K = lqr(A,B,Q,R);

```

Gambar 5.3 kode program simulasi kendali pada MATLAB

5.4 Pengujian Anti Rotasi Wahana

Simulasi gerak rotasi dilakukan untuk mendapatkan respon sistem kendali LQR pada wahana saat bergerak rotasi yang berupa *overshoot*, *rise time*, *settling time*, dan *steady-state error*. Gerak rotasi yang diuji berupa *roll*, *pitch*, dan *yaw*. Hasil simulasi yang berupa *feedback gain* **K** didapatkan dari menala nilai matriks **Q** sehingga mendapatkan respon sistem sesuai dengan spesifikasi yang diinginkan.

Melalui simulasi yang telah dilakukan, maka didapat *nilai feedback gain* **K** yang akan dimasukkan ke program mikrokontroler untuk diuji langsung. Nilai 10 *state* translasi rotasi yang digunakan untuk kendali LQR didapatkan dari pembacaan *sensor accelerometer* dan *gyroscope*. Kode program sistem kendali LQR yang sesuai dengan persamaan (4.13) ditunjukkan pada Gambar 5.4.

```

1. u1 = K_remote*(-throttle_new);
2.
3. u2 = K_roll*((-roll + off_roll) + rollangle + K_gyroRoll*(gx);
4.
5. u3 = K_pitch*(-(pitch + off_pitch) + pitchangle + pitch_reff) +
6. K_gyroPitch*(-gy);
7.
8. u4 = K_yaw*((roll + off_roll) - yawangle);
9.
10. w1 = b * u1/100000;
11.
12. hitroll = u2/u1;
13. hitpitch = u3/u1;
14. hityaw = u4/u1;
15.
16. hitpitch = constrain(hitpitch,-30/TO_DEG,30/TO_DEG);
17. hitroll = constrain(hitroll,-30/TO_DEG,30/TO_DEG);
18. hityaw = constrain(hityaw,-30/TO_DEG,30/TO_DEG);

```

Gambar 5.4 Program kendali anti-rotasi pada mikrokontroler

Nilai **K** yang telah didapat melalui simulasi kemudian akan dimasukkan kedalam program mikrokontroler untuk diuji coba pada wahana. Nilai 6 *state* yang digunakan pada simulasi didapat dari luaran sensor *accelerometer* dan *gyroscope* pada wahana. Setelah itu nilai luaran sensor akan dimasukkan kedalam *state* yang berupa matriks untuk diolah dalam persamaan *fullstate feedback*. Hasil dari proses tersebut berupa vektor masukan sistem wahana yang akan digunakan untuk mengoreksi nilai *state* sistem yang tidak bernilai nol. Nilai vektor **u** selanjutnya akan diubah menjadi nilai PWM dan defleksi sudut servo.

5.5 Pengujian Fase Menghindari Halangan

Luaran sensor LiDAR digunakan untuk mendapatkan nilai ketinggian dan jarak depan yang selanjutnya akan menjadi nilai referensi kedalam *state pitch*. *State* diproses menggunakan persamaan *fullstate feedback* untuk menghasilkan nilai masukan proses berupa sudut *pitch*. Nilai sudut tersebut akan digunakan untuk mengoreksi nilai *state* ketinggian dan jarak depan wahana. Program sistem kendali LQR akan diimplementasikan untuk fase menghindari halangan pada mikrokontroler.

Program fase menghindari halangan dibagi menjadi 3 tahap yaitu, tahap *cruise*, *climb*, dan tahap *descent*. Tahap tersebut bergantung pada hasil luaran sensor ketinggian dan jarak depan. Program untuk fase menghindari halangan

ditunjukkan pada Gambar 5.5 dimana susunan program sudah sesuai dengan diagram alir pada Gambar 4.14.

```

1. void avoidance()
2. {
3.     if(front_distance<35) // ketika terdapat halangan didepan
4.     pesawat kurang dari 35
5.     {
6.         mission.pitch = 20; //CLIMBING
7.         flag_climb = 1;
8.     }
9.     else if(lidar_distance>40 && flag_climb == 1) //ketika
10. jarak posisi perut pesawat jauh dengan halangan & pesawat
11. sedang climbing
12.     {
13.         mission.pitch = 0; //CRUISING
14.         flag_climb = 0;
15.     }
16.     else if(lidar_distance>40 && (attitude.pitch<5 &&
17. attitude.pitch >-5)) //ketika di bawah sudah ada halangan &
18. pesawat sedang cruising
19.     {
20.         mission.pitch = -20; //DESCENDING
21.         flag_descent = 1;
22.     }
23.     else if(attitude.altitude <40)
24.     {
25.         mission.pitch = 0; //CRUISING
26.         flag_descent = 0;
27.     }
28. }

```

Gambar 5.5 Program kendali fase menghindari halangan pada mikrokontroler

5.6 Simulasi Sudut Serang Sayap Wahana Pada Javafoil

Penelitian ini juga melakukan simulasi sudut serang sayap wahana untuk mendapatkan koefisien gaya angkat (C_l) dari sayap wahana. Simulasi dilakukan dengan memasukkan beberapa parameter yaitu, ketebalan dan bentuk sayap. Gambar 5.6 merupakan bentuk dari *airfoil* yang digunakan pada wahana, lalu bentuk *airfoil* tersebut akan disimulasikan ketika mendapatkan sudut serang tertentu. Hasil dari simulasi berupa koefisien gaya angkat pada setiap sudut serang yang dibentuk dari sayap.

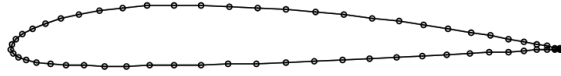


UNIVERSITAS
GADJAH MADA

**KENDALI PENGHINDAR HALANGAN PADA PESAWAT TANPA AWAK SAYAP TETAP
MENGUNAKAN METODE LQR**

Wahyu Agung Wicaksono, Dr. Andi Dharmawan, S.Si., M.Cs. ; Dr. Agfianto Eko Putra, M.Si

Universitas Gadjah Mada, 2019 | Diunduh dari <http://etd.repository.ugm.ac.id/>



Gambar 5.6 Bentuk *airfoil* sayap wahana

BAB VI

HASIL DAN PEMBAHASAN

6.1 Hasil Perhitungan Inersia

Tabel 6.1 menunjukkan hasil perhitungan inersia wahana dan didapatkan total inersia pada sumbu x sebesar $13,40 \times 10^{-2} \text{ kgm}^2$, sumbu y sebesar $5,75 \times 10^{-3} \text{ kgm}^2$, dan sumbu z sebesar $1,85 \times 10^{-2} \text{ kgm}^2$ dengan berat total wahana adalah 1,57 kg. Total inersia yang didapatkan selanjutnya digunakan sebagai parameter kendali pada simulasi MATLAB. Berdasarkan Gambar 5.3 pada BAB V, variabel-variabel I_{xx} , I_{yy} , dan I_{zz} masuk kedalam matriks **A** untuk pemodelan sistem yang disusun sesuai persamaan gerak anti rotasi.

Tabel 6.1 Hasil perhitungan inersia wahana

Komponen	Massa (Kg)	$I_{xx} \text{ (kgm}^2\text{)}$	$I_{yy} \text{ (kgm}^2\text{)}$	$I_{zz} \text{ (kgm}^2\text{)}$
Badan	1,21	$1,32 \times 10^{-2}$	$4,33 \times 10^{-4}$	$1,3 \times 10^{-2}$
Sayap	$2,81 \times 10^{-2}$	$1,48 \times 10^{-4}$	$5,27 \times 10^{-3}$	$5,41 \times 10^{-3}$
<i>Horizontal stabilizer</i>	$1,7 \times 10^{-3}$	$2,76 \times 10^{-5}$	$3,47 \times 10^{-5}$	$6,23 \times 10^{-5}$
Vertical stabilizer	$3,2 \times 10^{-3}$	$2,39 \times 10^{-5}$	$7,41 \times 10^{-5}$	$1,65 \times 10^{-5}$
Propeller	$1,7 \times 10^{-3}$	$6,15 \times 10^{-5}$	$6,15 \times 10^{-5}$	$1,42 \times 10^{-5}$
Total	1,57	$13,4 \times 10^{-2}$	$5,75 \times 10^{-3}$	$1,85 \times 10^{-2}$

6.2 Hasil Kalibrasi Pengukuran Sudut

Pengukuran ini dibagi menjadi dua tahap, yaitu tahap mencari nilai *offset* lalu tahap kedua mencari nilai ralat dari luaran sensor.

Pengujian dilakukan dengan cara meletakkan mikrokontroler pada bidang datar yang telah diukur dengan *waterpass*. Nilai *offset* dari sensor *accelerometer* dan *gyroscope* pada sumbu x, y , dan z didapat melalui program kalibrasi. Nilai *offset* tersebut akan digunakan untuk menormalkan nilai hasil sensor. Bagian bidang datar luaran sensor IMU seharusnya mendekati angka nol. Hasil pengujian sensor ditunjukkan pada Tabel 6.2.

Tabel 6.2 Hasil nilai *offset*

Sensor	Sumbu	Nilai <i>Offset</i>
<i>Accelerometer</i>	X	1273
	Y	-1541
	Z	1375
<i>Gyroscope</i>	X	67
	Y	-4
	Z	21

Nilai hasil pengujian *offset* akan dimasukkan ke dalam prosedur *setup* pada program utama sebagai inisialisasi sensor. Selisih nilai yang sesungguhnya ditunjukkan pada Tabel 6.3

Proses untuk mendapatkan nilai ralat luaran dari sensor dilakukan dengan mencatat nilai luaran sensor dari 0° hingga 30° dengan kenaikan 5° dan interval 5 detik di tiap kenaikan sudutnya. Pengujian diberi batasan karena wahana diuji pada bidang datar, dimana sudut *roll* dan *pitch* pada posisi 0° dan pengujian rotasi hanya diberi gangguan tidak lebih dari 30°. Hasil nilai pengujian sensor *fusion* dengan nilai rata-rata standar deviasi yang dihasilkan dari 50 *sample data* ditunjukkan pada Tabel 6.3. Sebagaimana ditunjukkan pada Tabel 6.3 menunjukkan bahwa pengukuran menggunakan metode sensor *fusion* sudah memenuhi spesifikasi dengan nilai *error* pada setiap data tidak lebih dari 1°.

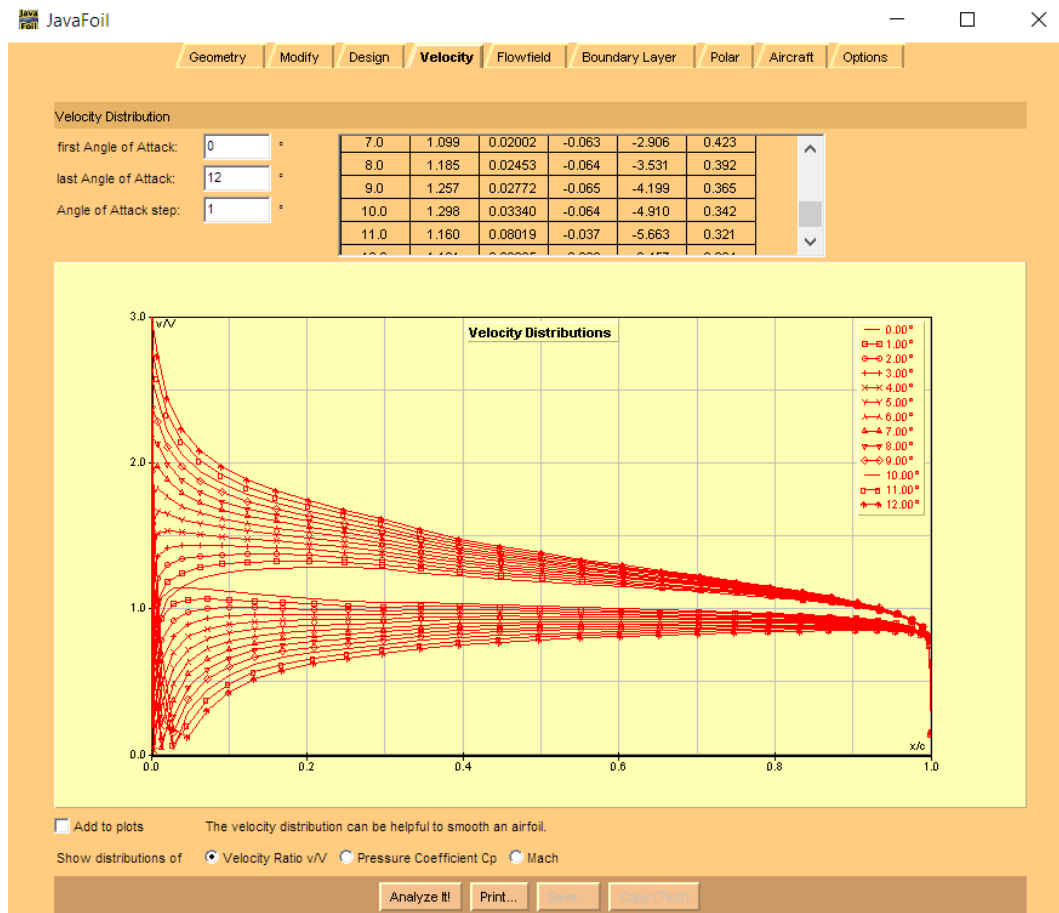
Tabel 6.3 Nilai rata-rata dan standar deviasi sensor

Kalibrator	Luaran Sensor <i>Roll</i>	Luaran Sensor <i>Pitch</i>
0°	(0,05 ± 0,023)°	(0,06 ± 0,005)°
5°	(5,05 ± 0,022)°	(5,04 ± 0,300)°
10°	(9,92 ± 0,009)°	(9,97 ± 0,126)°
15°	(15,06 ± 0,028)°	(15,24 ± 0,017)°
20°	(19,64 ± 0,201)°	(20,07 ± 0,009)°
25°	(25,14 ± 0,060)°	(24,79 ± 0,161)°
30°	(30,37 ± 0,021)°	(30,53 ± 0,373)°

6.3 Hasil Simulasi *Airfoil* Sayap Wahana

Simulasi *airfoil* sayap wahana dilakukan menggunakan perangkat lunak java foil. Simulasi dilakukan dengan memasukkan bentuk dan ketebalan sayap

wahana. Hasil simulasi ditunjukkan pada Gambar 6.1. Hasil dari simulasi ini memperoleh beberapa nilai koefisien gaya angkat (C_l) sesuai dengan nilai sudut serang yang dihasilkan. Nilai sudut serang kritis merupakan keadaan dimana nilai koefisien gaya angkat tersebut menghasilkan gaya angkat ($lift$) yang mendekati gaya berat wahana.



Gambar 6.1 Hasil simulasi *airfoil* sayap wahana

6.4 Hasil Pengujian Anti-Rotasi Wahana

Dalam pengujian anti-rotasi, sistem kendali akan mengendalikan sikap terbang wahana dengan menggerakkan aktuator pada *aileron*, *elevator*, dan *rudder*.

Simulasi anti-rotasi dilakukan dengan memberikan variasi pembobotan nilai Q berupa Q_ϕ , Q_θ , Q_ψ sebagai Q sudut dan Q_p , Q_q , Q_r sebagai Q kecepatan sudut. Variasi dari nilai Q tersebut akan menghasilkan nilai *fullstate feedback*

gain **K** yaitu $K_{1,1}$ dan $K_{1,2}$ untuk kendali anti-roll, $K_{2,3}$ dan $K_{2,4}$ untuk kendali anti-pitch, $K_{3,5}$ dan $K_{3,6}$ untuk kendali anti-yaw. Pembobotan nilai **Q** diawali dengan memberikan nilai 1 kemudian ditambah atau dikurangi hingga diperoleh respon sudut anti-rotasi yang sesuai dengan spesifikasi sistem, sedangkan untuk nilai **R** dianggap 1. Data berupa respon sudut anti-rotasi dan respon aktuator ditampilkan sehingga respon sistem terhadap perubahan nilai **Q** dapat diamati.

Variasi pembobotan nilai **Q** dapat diamati pada Tabel 6.4 Pembobotan nilai **Q** yang menghasilkan respon sistem sesuai dengan spesifikasi sistem adalah pembobotan terbaik pada penelitian ini. Nilai **Q** sudut terbaik adalah 58, 35, dan 85 untuk Q_ϕ , Q_θ , Q_ψ dan **Q** untuk kecepatan sudut sebesar 1,395, 0,5, dan 2,3 untuk Q_p , Q_q , Q_r . Nilai **Q** tersebut menghasilkan *rise time* yang sesuai dengan spesifikasi serta nilai **Q** kecepatan sudut yang cukup untuk meredam *overshoot* dan tidak terlalu mengurangi respon sistem sehingga memenuhi spesifikasi. Hasil simulasi dengan **Q** terbaik ditunjukkan pada Tabel 6.4

Tabel 6.4 Pembobotan matriks **Q** dengan nilai **K** yang dihasilkan

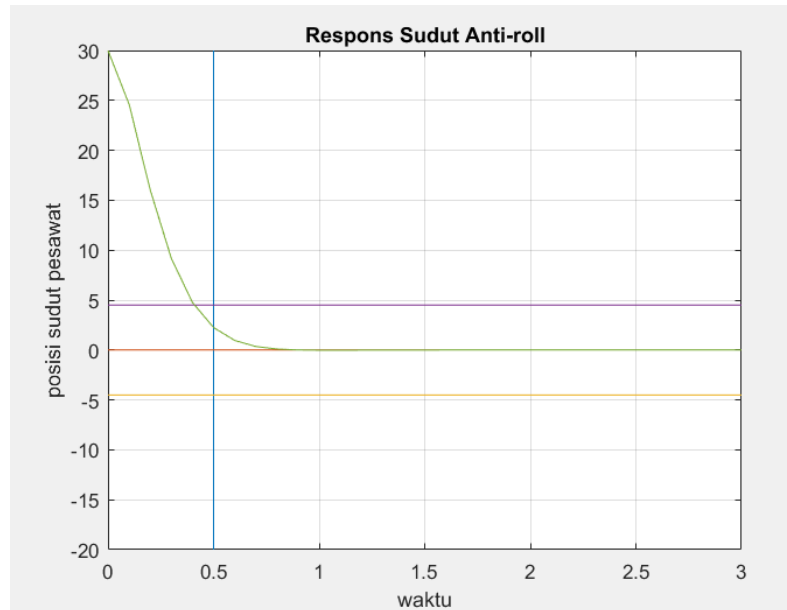
Q	K
$\begin{bmatrix} 58 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1,395 & 0 & 0 & 0 & 0 \\ 0 & 0 & 35 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0,5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 85 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2,3 \end{bmatrix}$	$\begin{bmatrix} 7,615 & 1,856 & 0 & 0 & 0 & 0 \\ 0 & 0 & 5,916 & 1,086 & 0 & 0 \\ 0 & 0 & 0 & 0 & 9,219 & 2,39 \end{bmatrix}$

Nilai **K** setelah penalaan nilai **Q** diambil yang terbaik dan selanjutnya dimasukkan ke dalam program utama untuk dilakukan validasi. Validasi dilakukan dengan cara menerbangkan wahana secara langsung pada ketinggian tertentu lalu *ground control station* akan memberikan perintah untuk masuk ke dalam mode *stabilize*. Setelah wahana terbang stabil secara kasat mata lalu akan diberikan gangguan berupa masukan sudut secara singkat dari *ground control station* untuk melihat respon dari wahana ketika mengembalikan posisi stabil nya. Hasil dari validasi dilihat dari data sudut yang dikirim ke *ground control station*. Setelah itu data yang didapat akan diolah menjadi kedalam bentuk grafik. Proses

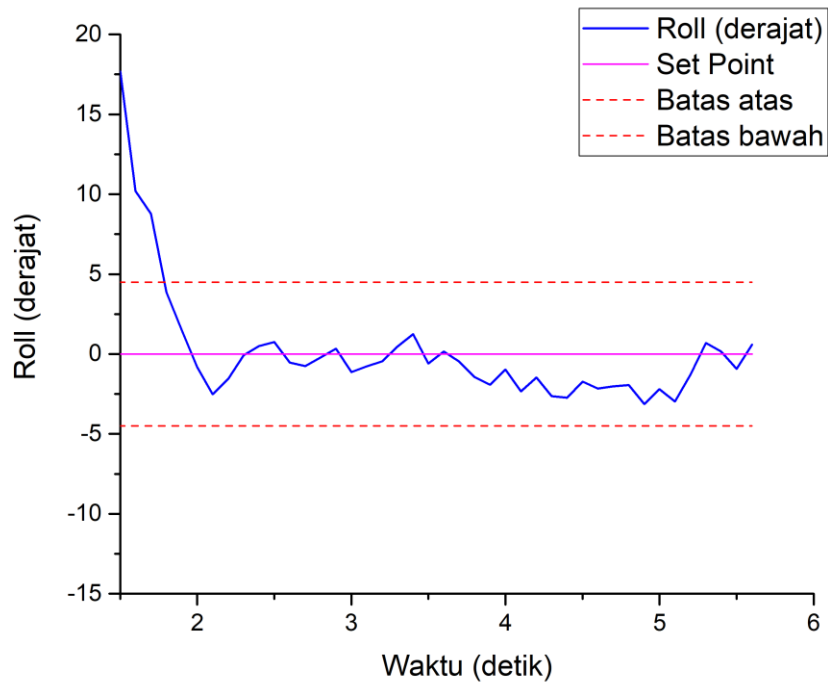
validasi dilakukan pada masing-masing sudut *roll*, *pitch* dan *yaw*. Grafik respon hasil simulasi dan validasi anti *roll* ditunjukkan pada Gambar 6.2.

Simulasi anti *roll* yang ditunjukkan pada Gambar 6.2 (a), sistem menghasilkan *rise time* sebesar 0.5 detik dan tidak menghasilkan *multi overshoot*. Hasil tersebut didapatkan dengan menala nilai \mathbf{Q} terbaik, yaitu Q_ϕ sebesar 58 dengan Q_p sebesar 1,395 sehingga menghasilkan nilai *gain* K_{11} sebesar 7,615 dan K_{12} sebesar 1,856, jika nilai \mathbf{Q} yang diberikan semakin besar, maka respon sistem akan semakin cepat dan nilai \mathbf{Q} yang besar memungkinkan terjadinya *multi overshoot*, sehingga diperlukan peredaman menggunakan Q_p agar tidak timbul *multi overshoot*. Nilai 58 untuk Q_ϕ dan 1,395 untuk Q_p dianggap paling optimal untuk sistem yang dibuat.

Saat tahap validasi, wahana akan diterbangkan hingga ketinggian tertentu dalam mode *stabilize*. Wahana akan diberikan gangguan *roll* dengan nilai sudut tertentu untuk menguji kendali anti rotasinya. Berdasarkan Gambar 6.2 (b), wahana diberikan gangguan dengan nilai puncak sebesar 30° . Gangguan yang diberikan menyebabkan ketidakstabilan pada wahana sehingga kendali *fullstate feedback* sangat berperan untuk mengatasi gangguan tersebut. Nilai *gain* \mathbf{K} yang dihasilkan pada simulasi LQR digunakan untuk mengatur gerak *aileron* sehingga dihasilkan torsi yang berlawanan dengan arah gangguan yang diberikan. Tujuan dari gerakan *aileron* tersebut adalah agar wahana kembali ke posisi *roll* referensinya, yaitu 0° , setelah melakukan validasi anti *roll*, data yang didapatkan kemudian diolah dan hasilnya menunjukkan bahwa sistem kendali telah mampu mempertahankan gerak *roll* sesuai dengan nilai referensinya dengan nilai *rise time* sebesar 0,5 detik dan *settling time* sebesar 1 detik. Namun nilai sudut *roll* tidak selalu berada pada referensinya.



(a)



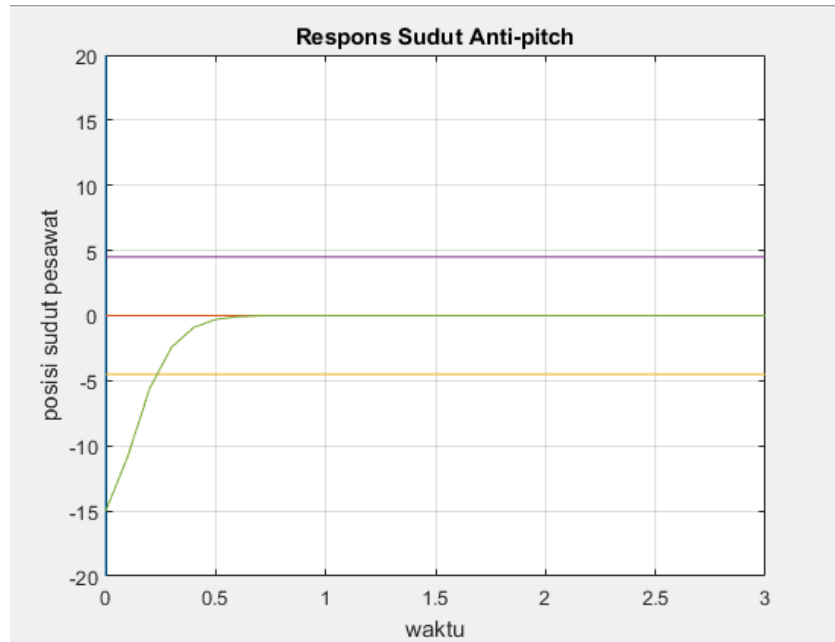
(b)

Gambar 6.2 Grafik respon anti *roll* (a) simulasi (b) validasi

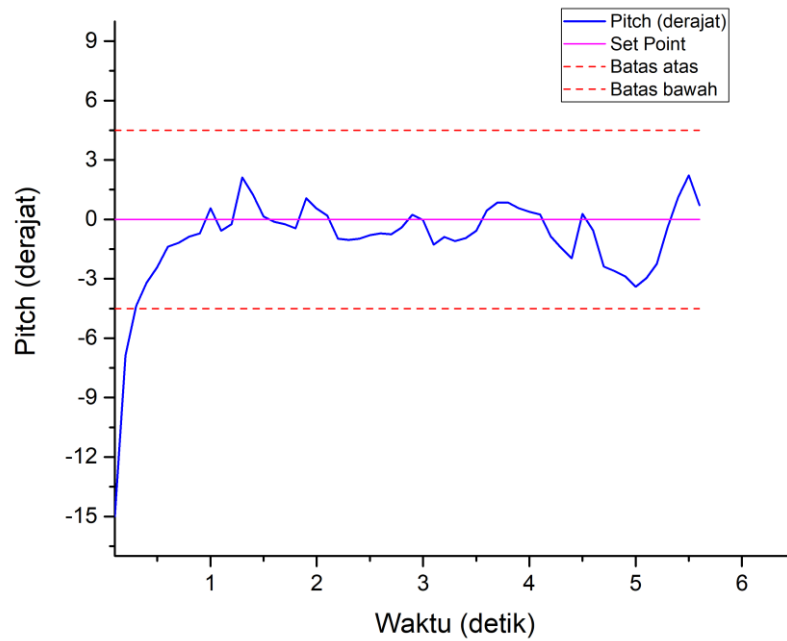
Ketidakstabilan tersebut dapat disebabkan oleh beberapa faktor diantaranya gangguan dari luar sistem dan pada simulasi LQR tidak disertakan simulasi untuk mengatasi gangguan dari luar sistem sehingga pada tahap validasi wahana tidak dapat mengatasi gangguan dari luar sistem. Efek dari gangguan luar tersebut didapatkan kecenderungan *steady state error* sebesar $1,21^\circ$. Meskipun terdapat *steady state error* pada anti *roll*, nilai yang dihasilkan masih berada dalam batas toleransi, sesuai dengan spesifikasi pada subbab 4.3.

Gambar 6.3 menunjukkan hasil simulasi dan validasi anti *pitch*. Grafik (a) terlihat ketika simulasi anti *pitch* sistem menghasilkan *risetime* sebesar 0,4 detik dan tidak terdapat *multi overshoot*. Hasil tersebut diperoleh dengan penalaan nilai terbaik yaitu, Q_θ sebesar 35 dengan Q_q sebesar 0,5 sehingga menghasilkan nilai *gain* K_{23} sebesar 5,916 dan K_{24} sebesar 1,086. Jika nilai Q yang diberikan semakin besar maka respon sistem akan semakin cepat, namun nilai Q yang besar akan memungkinkan terjadinya *multi overshoot*, sehingga diperlukan peredaman menggunakan Q_q agar tidak timbul *multi overshoot*. Nilai 35 untuk Q_θ dan 0,5 untuk Q_q dianggap paling optimal untuk sistem yang dibuat.

Setelah melakukan validasi anti *pitch*, data diolah dan hasilnya menunjukkan bahwa sistem kendali yang dibuat telah mampu mempertahankan gerak *pitch* sesuai nilai referensinya dengan nilai *risetime* sebesar 0,4 detik dan *settling time* sebesar 1 detik. Namun sudut *pitch* tidak selalu berada pada nilai referensinya, ketidakstabilan tersebut bisa disebabkan oleh beberapa faktor diantaranya faktor luar, dan pada simulasi LQR tidak disertakan simulasi untuk mengatasi gangguan luar sistem. Efek dari gangguan luar tersebut didapatkan kecenderungan *steady state error* sebesar $0,89^\circ$. Meskipun terdapat kecenderungan *steady state error* pada kendali *pitch*, nilai yang dihasilkan masih berada pada batas toleransi yang sesuai pada subbab 4.3. seperti yang terlihat pada grafik (b).



(a)



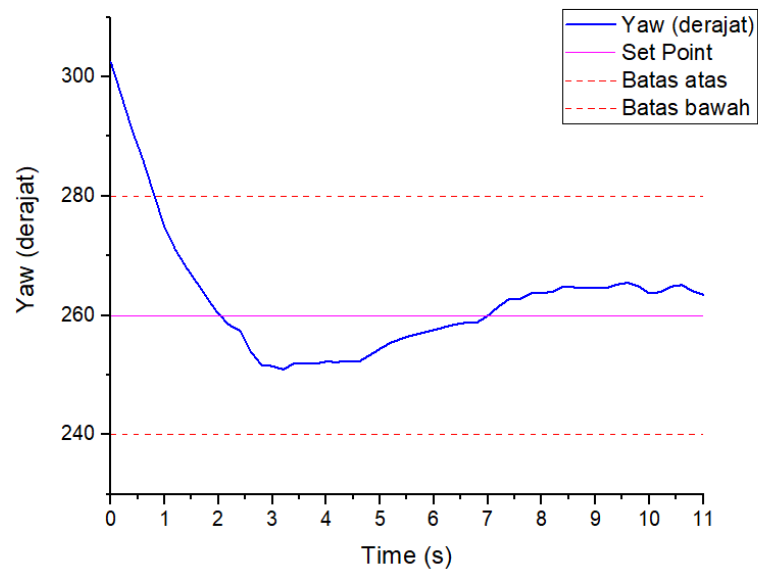
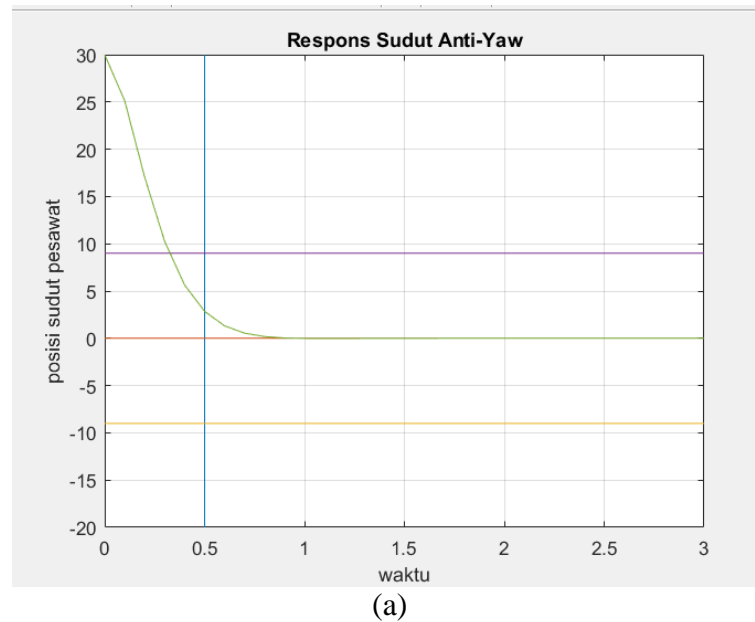
(b)

Gambar 6.3 Grafik respon anti *pitch* (a) simulasi (b) validasi

Simulasi dan validasi anti *yaw* ditunjukkan pada Gambar 6.4. Berdasarkan Gambar 6.4 (a) menunjukkan ketika simulasi anti *yaw*, sistem menghasilkan *rise time* sebesar 0,5 detik dan tidak terdapat *multi overshoot*. Hasil tersebut dapat dicapai dengan menala menala nilai \mathbf{Q} terbaik, yaitu Q_ψ sebesar 85 dengan Q_r sebesar 2,3 sehingga menghasilkan nilai *gain* K_{35} sebesar 9,219 dan K_{36} sebesar 2,39. Jika nilai \mathbf{Q} yang diberikan semakin besar maka respon sistem akan semakin cepat, namun nilai \mathbf{Q} yang besar akan memungkinkan terjadinya *multi overshoot*, sehingga diperlukan peredaman menggunakan Q_r agar tidak timbul *multi overshoot*. Nilai 85 untuk Q_ψ dan 2,3 untuk Q_r dianggap paling optimal untuk sistem yang dibuat.

Wahana akan diterbangkan dengan ketinggian tertentu dengan mode *stabilize*. Wahana akan diberi gangguan *yaw* dengan nilai sudut tertentu dari *ground control station* untuk menguji kendali anti rotasinya. Gambar 6.4 (b) memperlihatkan wahana diberi gangguan gerak *yaw* dengan nilai puncak $302,44^\circ$. Nilai *gain* \mathbf{K} yang dihasilkan pada simulasi LQR digunakan untuk mengatur gerak *rudder* sehingga menghasilkan nilai torsi yang berlawanan dengan arah gangguan. Gerakan *rudder* tersebut bertujuan agar wahana dapat kembali ke posisi *yaw* referensi yaitu 260° .

Penentuan nilai referensi *yaw* diambil ketika wahana mulai masuk dalam mode *stabilize*. Pembacaan nilai sudut *yaw* memiliki rentang antara 0° - 360° , di mana nilai 0° menunjukkan arah utara dan bertambah nilainya sesuai arah jarum jam. Setelah itu dilakukan validasi anti *yaw*, data yang didapat diolah dan hasilnya menunjukkan bahwa sistem kendali telah mampu mempertahankan gerak *yaw* sesuai dengan nilai referensinya dengan *rise time* sebesar 1,6 detik. Namun, nilai sudut *yaw* tidak selalu berada pada nilai referensinya. Ketidakstabilan ini disebabkan beberapa faktor diantaranya ketika melakukan simulasi LQR tidak disertakan simulasi untuk mengatasi gangguan luar sistem, sehingga saat melakukan validasi wahana tidak mampu mengatasi gangguan dari luar. Meskipun terdapat ketidakstabilan, namun nilai yang dihasilkan masih berada pada batas toleransi yang sesuai dengan spesifikasi pada subbab 4.3.



Gambar 6.4 Grafik respon anti anti *yaw* (s) simulasi (b) validasi

6.5 Pengujian Menghindari Halangan

Setelah kendali anti rotasi dapat menjaga wahana terbang stabil, maka kendali kestabilan pada fase menghindari halangan dapat diuji. Pengujian dilakukan dengan menerbangkan wahana pada ketinggian 20 meter. Pengujian

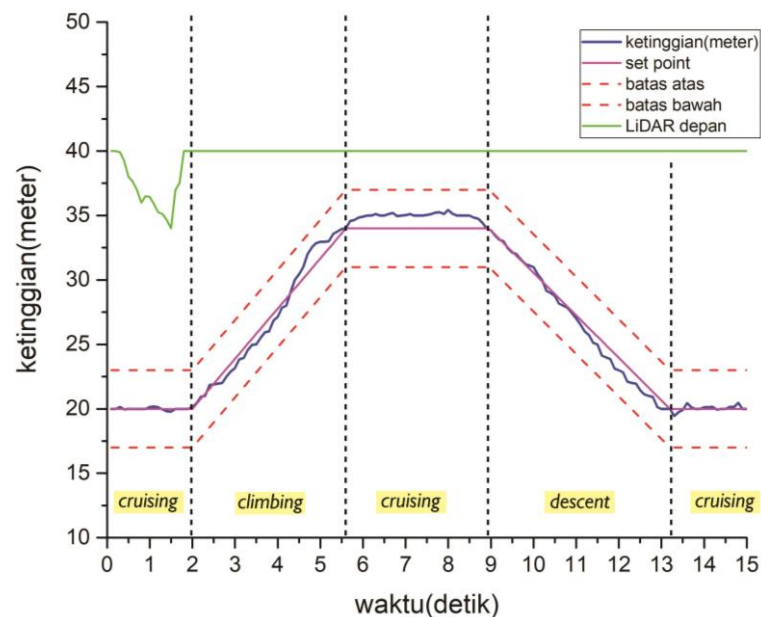
fase menghindari halangan dibagi menjadi fase *cruise*, *climb*, dan *descent*. tahap *cruise* akan dimulai ketika wahana diberi perintah dari *ground station*.

Wahana diminta untuk menjaga ketinggian terbang jelajah pada 20 meter. Saat tahap terbang jelajah nilai PWM diatur pada 1500us. Pemberian nilai PWM ini diharapkan agar wahana memiliki kecepatan terbang yang konstan. Tahap *cruise*, wahana akan terbang hingga sensor depan membaca jarak kurang dari 35 meter maka diasumsikan ada hambatan di depan wahana dan wahana akan memasuki tahap *climb*. Tahap *climb*, sudut *pitch* diatur sebesar 20° dan nilai PWM akan naik menjadi 1900us agar wahana dapat menanjak naik menghindari hambatan. Ketika jarak depan sudah besar dari 30 meter maka diasumsikan hambatan sudah tidak ada dan wahana akan memasuki tahap *cruise* kembali dan sensor jarak bawah akan mendeteksi apakah masih ada hambatan atau tidak, ketika sensor jarak bawah membaca nilai lebih besar dari 10 meter, maka diasumsikan sudah tidak ada hambatan di bawah wahana dan wahana akan memasuki tahap *descent* dengan mengatur referensi sudut *pitch* 20° ke bawah hingga mencapai nilai ketinggian referensi yaitu 20 meter. Setelah mencapai ketinggian referensi di 20 meter maka wahana akan kembali melakukan fase terbang jelajah seperti di awal sesuai dengan rancangan pengujian pada sub bab (4.9.3).

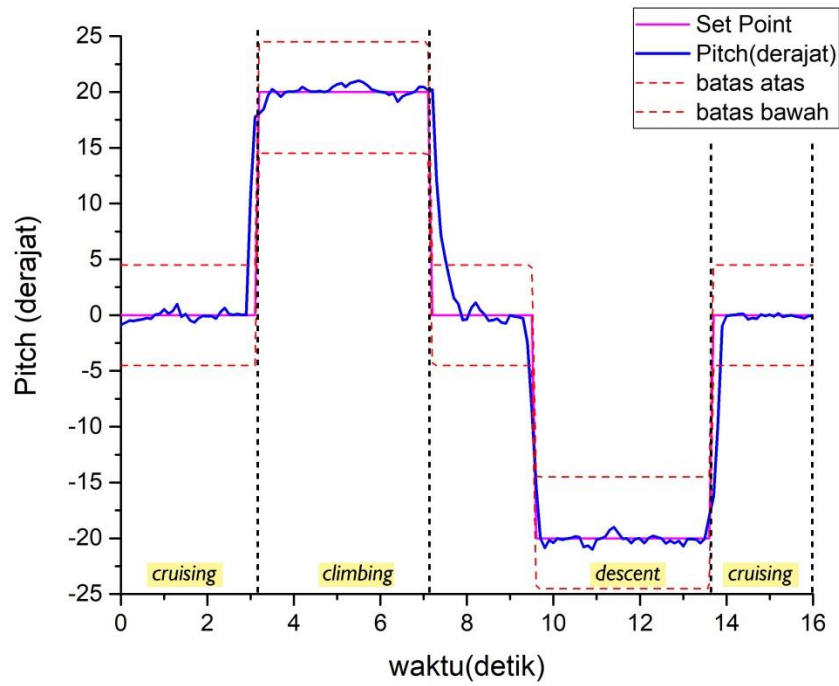
Gambar 6.5 menunjukkan grafik ketinggian wahana yang sesungguhnya dan jarak depan wahana terhadap halangan, grafik ketinggian didapat menggunakan sensor barometer dan jarak depan wahana terhadap halangan menggunakan sensor LiDAR. Gambar 6.5, ketika nilai bacaan sensor LiDAR berada di bawah 35 meter, maka wahana akan memasuki fase *climbing* sesuai dengan skenario penghindar halangan pada sub bab (4.4). Terlihat pada Gambar 6.6, pada detik ke-3 hingga 7,1 wahana mulai memasuki fase *climb* dengan sudut referensi 20° , lalu wahana akan memasuki fase *cruise* pada detik ke-7,1 hingga 9,5, pada fase *cruise* ini wahana berada di atas halangan, dan ketika sensor bawah mendeteksi tidak adanya halangan, maka wahana akan masuk fase *descent* pada detik ke-9,6 hingga 13,7 dengan sudut referensi 20° ke bawah dan detik ke-13,7 wahana sudah mencapai ketinggian referensi 20 meter dan wahana

memasuki fase *cruise* kembali di ketinggian 20 meter. Selama melakukan misi menghindari halangan, perubahan nilai ketinggian dan jarak wahana mengalami *overshoot* yang disebabkan oleh faktor luar, seperti kondisi cuaca yang tidak stabil, kondisi halangan yang dapat berubah ketika diterpa angin dan sebagainya. *Overshoot* yang dihasilkan dikarenakan pada simulasi tidak dimasukkan simulasi untuk mengatasi adanya faktor luar tersebut. Namun *overshoot* yang dihasilkan masih berada di dalam batas toleransi.

Selama fase menghindari halangan, sistem kendali LQR akan mengatur sikap anti rotasi untuk gerak *roll*, *pitch*, dan *yaw* wahana terhadap gangguan yang terjadi. Waktu yang tercatat selama fase menghindari halangan dimulai dari fase *climb* hingga *descent* menuju ketinggian referensi adalah 10,5 detik. Gambar 6.6 merupakan grafik respon sudut *pitch* ketika wahana melakukan misi menghindari halangan. Terlihat pada Gambar 6.6 bahwa wahana mampu mempertahankan nilai *pitch* sesuai dengan referensinya dengan batas toleransi $\pm 5^\circ$ dan nilai *gain* yang diberikan sudah mampu meredam *multiovershoot* selama fase menghindari halangan. Dari hasil tersebut, wahana dapat dikatakan telah mampu menjaga kestabilan wahana dalam melakukan fase menghindari halangan menggunakan metode LQR dalam kendali *fullstate feedback*.



Gambar 6.5 Grafik ketinggian riil wahana dan jarak depan wahana selama misi



Gambar 6.6 Respon sudut *pitch* saat fase menghindari halangan

BAB VII

PENUTUP

7.1 Kesimpulan

Berdasarkan pengujian, pengamatan, dan analisis hasil yang telah diperoleh, maka didapat kesimpulan sebagai berikut. Wahana telah mampu melakukan fase menghindari halangan dengan stabil menggunakan metode *Linear Quadratic Regulator* (LQR) dan mampu mempertahankan respon sudut *pitch* pada nilai referensinya. Hasil tersebut diperoleh dengan nilai *fullstate feedback gain* **K** sebesar 7,615 dan 1,856 untuk anti *roll*, 5,916 dan 1,086 untuk anti *pitch*, serta 9,219 dan 2,39 untuk anti *yaw*. Nilai *feedback gain* **K** tersebut akan menghasilkan nilai *Rise Time* 0,5 detik, *Steady State Error* 1,21° untuk anti *roll*, *Rise Time* 0,4 detik, *Steady State Error* 0,89° untuk anti *pitch*, dan *Rise Time* 1,6 detik, *Steady State Error* 2,47° untuk anti *yaw*.

7.2 Saran

Masih terdapat beberapa hal yang perlu disempurnakan dari penelitian skripsi ini. Saran-saran yang dapat digunakan untuk menyempurnakan penelitian skripsi selanjutnya sebagai berikut:

1. Perlu adanya penambahan parameter kendali untuk mengantisipasi adanya gangguan dari luar seperti angin, sehingga hasil simulasi yang diperoleh semakin dekat dan menyerupai keadaan lapangan.
2. Perlu adanya penambahan data *air speed* untuk mengkompensasi data kecepatan wahana ketika kondisi angin tinggi.
3. Dibutuhkan kendali untuk menghindari wahana dengan menggunakan *pitch* dan *roll* agar dapat lebih baik lagi ketika menghindari halangan.

DAFTAR PUSTAKA

- Ahsan, M., Shafique, K., Mansoor, A. Bin dan Mushtaq, M., 2013, Performance comparison of two altitude-control algorithms for a fixed-wing UAV, *2013 3rd IEEE International Conference on Computer, Control and Communication, IC4 2013*, [Online] 0–4, tersedia di DOI:10.1109/IC4.2013.6653744.
- Fossen, T.I., 2011, Mathematical Models for Control of Aircraft and Satellites, *Department of Engineering Cybernetics*, (January), 1–8,
- Gageik, N., Benz, P. dan Montenegro, S., 2015, Obstacle detection and collision avoidance for a UAV with complementary low-cost sensors, *IEEE Access*, [Online] 3599–609, tersedia di DOI:10.1109/ACCESS.2015.2432455.
- Kohno, S. dan Uchiyama, K., 2014, Design of robust controller of fixed-wing UAV for transition flight, *2014 International Conference on Unmanned Aircraft Systems, ICUAS 2014 - Conference Proceedings*, [Online] 1111–1116, tersedia di DOI:10.1109/ICUAS.2014.6842364.
- Kwag, Y.K., Choi, M.S., Jung, C.H. dan Hwang, K.Y., 2007, Collision avoidance radar for UAV, *CIE International Conference of Radar Proceedings*, [Online] 0–3, tersedia di DOI:10.1109/ICR.2006.343231.
- Lavretsky, E. dan Wise, K., 2013, *Robust and Adaptive Control*, [Online]. tersedia di DOI:10.1007/978-1-4471-4396-3.
- Liu, W., Yu, C., Wang, X., Zhang, Y. dan Yu, Y., 2017, The Altitude Hold Algorithm of UAV Based on Millimeter Wave Radar Sensors, *Proceedings - 9th International Conference on Intelligent Human-Machine Systems and Cybernetics, IHMSC 2017*, [Online] 1436–439, tersedia di DOI:10.1109/IHMSC.2017.106.
- Prahasta, R. dan Dharmawan, A., 2016, Implementasi Metode LQR (Linear Quadratic Regulator) pada Pengendali Terbang Quadrotor untuk Kestabilan Sikap Pesawat, *Tesis*, Universitas Gadjah Mada, Yogyakarta, Indonesia.
- Warsi, F.A., Hazry, D., Ahmed, S.F., Joyo, M.K., Tanveer, M.H., Kamarudin, H. dan Razlan, Z.M., 2014, *Yaw, Pitch and Roll controller design for fixed-wing*