BAB IV

PENGUJIAN DAN ANALISIS

Pada bab ini akan diuraikan mengenai proses pengujian dari sistem yang dirancang. Dimana pengujian yang dilakukan meliputi pengukuran terhadap komponen masukan, proses dan output keluaran yang terdapat dalam sistem yang dirancang kemudian dilanjutkan dengan menganalisis hasil pengukuran tersebut. Hal ini bertujuan untuk melihat performa sistem yang dirancang

4.1 Pengujian dan Analisis Bagian Masukan

Pengujian dan analisis bagian masukan input terdiri dari pengujian sensor *barometer*, sensor *accelerometer* dan sensor *gyroscope*

4.1.1 Pengujian Sensor *Barometer*

Sensor ini bekerja berdasarkan tekanan dan temperatur lingkungan, semakin tinggi posisi sensor maka semakin kecil tekanan udara. Sensor ini sangatlah sensitif terhadap perubahan ketinggian, sehingga sensor ini dapat menetukan ketinggian suatu objek. Sensor barometer ini terletak pada pcb flight controller. Cara pengujian nya adalah dengan cara mereset barometer lalu ukur ketinggian quadcopter dari tempat take off

Tabel 4.1 Hasil pengukuran ketinggian sensor barometer

Pengukuran secara manual (cm)	**Ketinggian terukur sensor (cm)**
10	13
20	21
50	55
100	105
120	125

Pada saat *quadcopter* diangkat sebesar 10cm , bacaan dari barometer menunjukan bahwa ketinggian *quadcopter* sebesar 13cm, setelah itu *quadcopter* diangkat dengan ketinggian 20cm dengan pengukuran manual, bacaan barometer menunjukan ketinggian 21cm

Dari hasil pengujian tabel 4.1 menggunakan sensor barometer, dapat diketahui bahwa pengukuran yang dilakukan sensor mendekati ketinggian sesungguhnya.

4.1.2 Pengujian Sensor Accelerometer

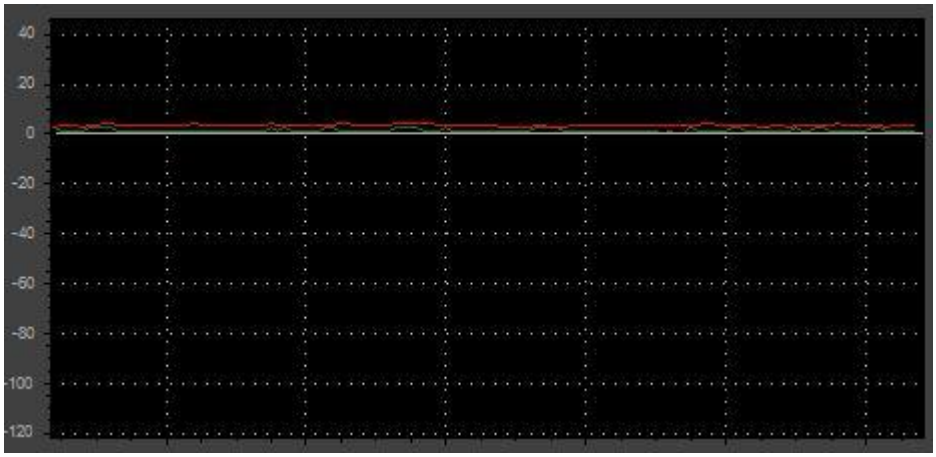
Pengujian *accelerometer* bertujuan untuk mengetahui tingkat keakuratan dari *accelerometer* dalam membaca perubahan sudut dari quadcopter. *Accelerometer* sudah terdapat dalam pcb flight controller kemudian data dari *accelerometer* akan dibuat input dan keluarannya adalah torsi dari motor brushless. Sumbu *roll* merupakan sumbu X dari *accelerometer,* sumbu *pitch* merupakan sumbu Y dari *accelerometer*

4.2 Tabel hasil pengujian accelerometer

No.	Arah Mekanik	Roll (x)	Pitch (y)
1	Miring kedepan	-10	150
2	Miring kebelakang	3	-150
3	Permukaan datar	0	0
4	Miring ke kanan	250	2
5	Miring ke kiri	-250	10

Pada saat *quadcopter* dimiringkan ke depan bacaan dari *accelerometer* menunjukan bahwa nilai sumbu dari *pitch* menjadi 150, untuk kemiringan kearah belakang bacaan dari *accelerometer* menunjukan angka -150, selanjutnya pengujian pada sumbu *roll* yang dilakukan dengan cara memiringkan permukaan dari *quadcopter*. Pengujian sumbu *roll* mendapatkan hasil yang sesuai dengan permukaan dari *quadcopter*

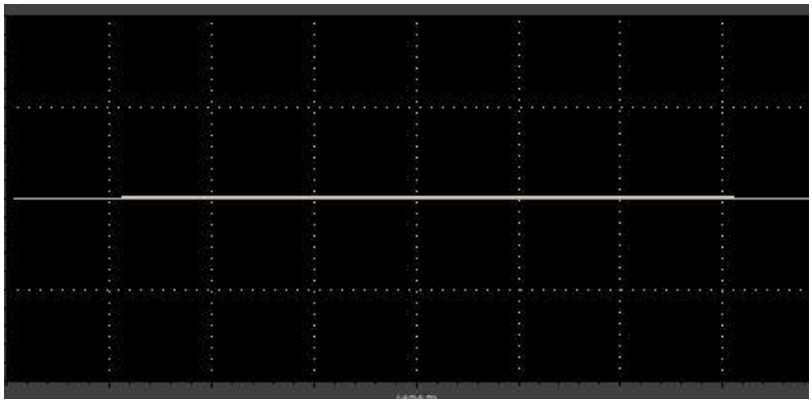
Dari tabel di atas dapat diketahui bahwa nilai setiap sumbu accelerometer sudah sesuai dengan arah mekanik robot



Gambar 4.1 Grafik accelerometer ketika diam

4.1.3 Pengujian Sensor Gyroscope

Pengujian *gyroscope* bertujuan untuk mengetahui tingkat keakuratan dari *gyroscope* dalam membaca perubahan kecepatan sudut yang dinamis dari quadcopter. Sensor *gyroscope* yang digunakan akan mempunyai nilai keluaran jika sedang berotasi searah jarum jam pada sumbu Y, maka tegangan keluarannya akan mengecil. Sedangkan jika berotasi berlawanan arah jarum jam, maka tegangan keluarannya akan membesar. Jika sensor *gyroscope* tidak berotasi (keadaan diam) maka keluaran tegangan *gyroscope* akan bernilai sama dengan nilai offsetnya.



Gambar 4.2 sensor gyroscope pada saat diam

Tabel 4.3 hasil pengujian gyroscope saat diam

No	**Data *Gyroscope* Dalam Keadaan Diam**		
	Roll	**Pitch**	**Yaw**
1	0	0	0
2	0	0	0
3	0	0	0
4	0	0	0
5	0	0	0

Tabel 4.4 hasil pengujian gyroscope saat berputar searah jarum jam

No.	**Data *Gyroscope* Dalam Keadaan berputar searah jarum jam**		
	Roll	**Pitch**	**Yaw**
1	22	0	-2
2	26	-5	1
3	27	-3	1
4	11	1	0
5	41	-15	-16

Data pengujian sensor gyroscope didapatkan melalui logs data yang dihasilkan oleh GUI dari program *flight controller* yang digunakan.

4.2 Pengujian dan Analisis Bagian Proses

Tujuan dari pengujian ini untuk mengimplementasikan masukan, proses dan keluaran. Kendali PID digunakan untuk kestabilan permukaan (*level)* dan kestabilan ketinggian menggunakan kendali fuzzy logic. Analisis yang dilakukan pada kendaliler PID auto level. Cara yang dilakukan untuk mendapatkan parameter kendali pid dengan cara *tuning trial and error*

4.2.1 Pengujian dan Analisis kendali sudut Pitch(Y) dan Roll(X)

Pengujian kendali pada sudut pitch dan roll dilakukan dengan cara melihat respon dari sistem dengan masukan dari telemetry, tuning dilakukan dengan cara trial and error, dan juga melihat referensi tuning pid. Tuning setiap sudut dilakukan karena bentuk dan pembagian berat dari quadcopter tidaklah simetris sehingga parameter pid nya berbeda

Tabel 4.5 Pengujian kendali proporsional sumbu roll

Kp	Ki	Kd	Respon sudut
6	0	0	Respon sangat lambat, osilasi tinggi
7	0	0	Overshoot cukup besar, osilasi cukup tinggi
9.5	**0**	**0**	**Respon lebih cepat, osilasi mengecil**

Tabel 4.6 Pengujian kendali proporsional dan integral sumbu roll

Kp	Ki	Kd	Respon sudut
9.5	0,005	0	Tidak bisa mempertahankan posisi
9,5	**0,01**	**0**	**Attitude cukup tertahan**
9,5	0,15	0	Osilasi yang tinggi

Tabel 4.7 pengujian kendali proporsional, integral dan differensial sumbu roll

Kp	Ki	Kd	Respon sudut
9.5	0,01	25	Respon Osilasi mengecil
9.5	**0,01**	**50**	**Respon cepat, osilasi sangat kecil**
9.5	0,01	60	Respon cepat akan tetapi terjadi osilasi

Tabel 4.8 pengujian kendali proporsional sumbu pitch

Kp	Ki	Kd	Respon sudut
5	0	0	Respon sangat lambat
8	0	0	Respon masih lambat
12,8	**0**	**0**	**Respon lebih cepat, osilasi mengecil**

Tabel 4.9 pengujian kendali proporsional dan integral sumbu pitch

Kp	Ki	Kd	Respon sudut
12.8	**0,013**	**0**	**Overshoot mengecil dan dapat mempertahankan posisi**
12.8	0,020	0	Overshoot mengecil, osilasi cukup tinggi
12.8	0,025	0	Osilasi semakin tinggi

Tabel 4.10 pengujian kendali proporsional, integral dan differensial sumbu pitch

Kp	Ki	Kd	Respon sudut
12.8	0,013	20	Respon cepat , Osilasi mengecil
12.8	**0,013**	**60**	**Respon melambat, osilasi sangat kecil**
12.8	0,013	80	Respon lambat dan terjadi osilasi

Penggunaan parameter PID roll dan pitch disesuaikan dengan penggunaan quadcopter, pemilihan setiap parameter bertujuan agar quadcopter dapat melayang dengan stabil

4.2.2 Pengujian dan Analisis Kendali A*uto Level PID*

Pengujian kendali pada auto level PID dengan cara melihat respon dari sistem, tuning dilakukan dengan cara trial and error, dan juga melihat referensi tuning pid

Tabel 4.11 pengujian kendali proporsional

Kp	Ki	Kd	Respon sudut
5	0	0	Respon sangat lambat, osilasi tinggi
8	0	0	Respon cukup cepat, osilasi semakin mengecil
10	0	0	Respon lebih cepat, osilasi mengecil

Tabel 4.12 Pengujian kendali proporsional dan integral

Kp	Ki	Kd	Respon sudut
10	0,05	0	Tidak bisa mempertahankan attitude
10	**0,1**	**0**	**Attitude cukup tertahan**
10	0,15	0	Terjadi osilasi yang cukup besar

Tabel 4.13 Pengujian kendali proporsional, integral dan differensial

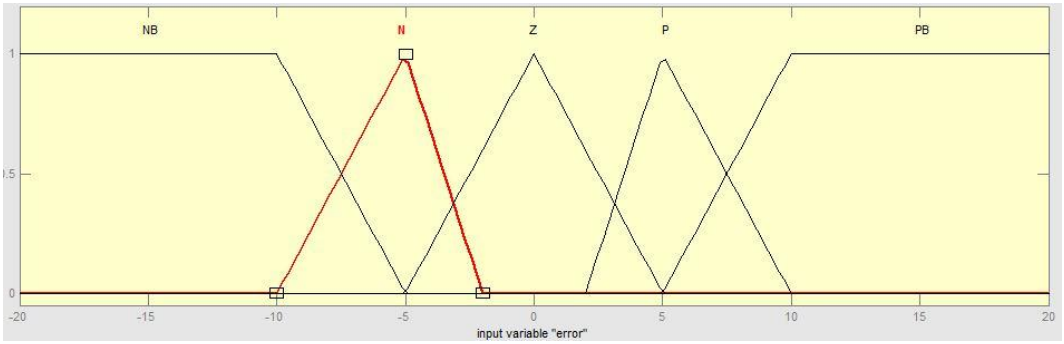
Kp	Ki	Kd	Respon sudut
10	0,1	60	Respon cepat, osilasi tinggi
10	0,1	75	Respon cukup cepat, osilasi mengecil
10	**0,1**	**100**	**Respon cepat dan stabil, dengan osilasi yang sangat kecil**

Tabel diatas merupakan hasil tuning PID pada kendalier quadcopter yang menyebabkan quadcopter mampu menjaga dan mempertahankan posisi

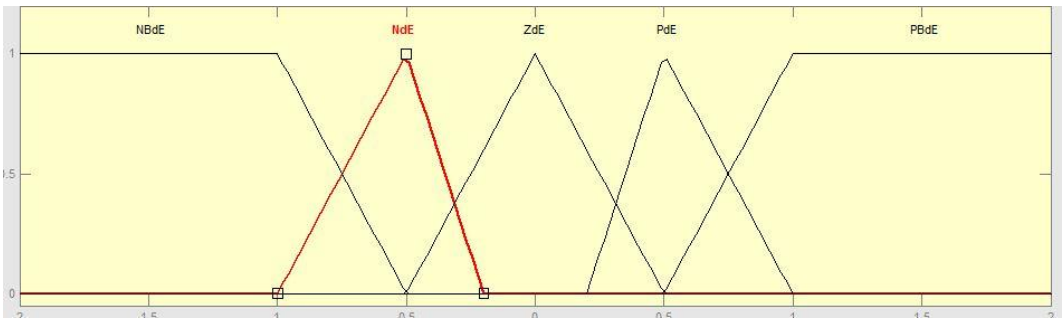
4.2.3 pengujian dan analisis kendali *fuzzy*

Tujuan dari pengujian ini adalah untuk mengimplementasikan masukan, proses dan keluaran dengan *fuzzy rule* kedalam *flight controller*. Pengujian dilakukan melalui simulasi fuzzy Matlab. Pengamatan hasil dimaksudkan untuk mendapatkan nilai torsi yang diharapkan. Berikut ini adalah fungsi keanggotaan *error* dan *d_error.*

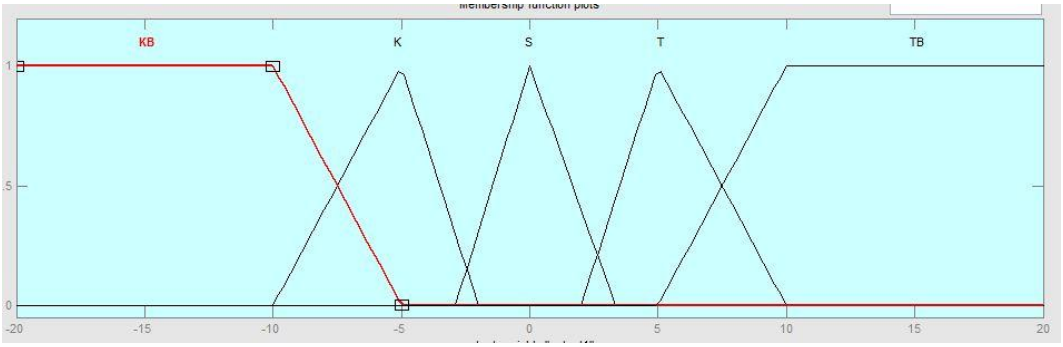
Perancangan kendali fuzzy dibuat berdasarkan nilai dan parameter yang telah dijelaskan pada bab sebelumnya



Gambar 4.3 Fungsi keanggotaan error



Gambar 4.4 Fungsi keanggotaan d_error



Gambar 4.5 Fungsi keanggotaan output

Pada fungsi keanggotaan error dan derror nilai masukannya berupa jarak dalam centimeter, sedangkan dalam fungsi keanggotaan output nilai keluaran nya adalah nilai torsi yang akan di proses di *flight controller*.

Berikut ini adalah listing program yang diterapkan pada mikrokontroller dengan menggunakan arduino dan *Embeded Fuzzy Logic Library* (eFLL).

```
FuzzySet* Z = new FuzzySet(-0.5,0,0,0.5);
FuzzySet* N = new FuzzySet(-0.10,-0.5,-0.5,-0.2);
FuzzySet* NB = new FuzzySet(-2,-2,-1,-0.5);
FuzzySet* P = new FuzzySet(0.2,0.5,0.5,1);
FuzzySet* PB = new FuzzySet(0.5,1,2,2);

FuzzySet* ZdE = new FuzzySet(-0.5,0,0,0.5);
FuzzySet* NdE = new FuzzySet(-1,-0.5,-0.5,-0.2);
FuzzySet* NBdE = new FuzzySet(-2,-2,-1,-0.5);
FuzzySet* PdE = new FuzzySet(0.2,0.5,0.5,1);
FuzzySet* PBdE = new FuzzySet(0.5,1,2,2);

void setup(){
  Serial.begin(9600);

  // FuzzyInput
  FuzzyInput* error = new FuzzyInput(1);
  error->addFuzzySet(NB);
  error->addFuzzySet(N);
  error->addFuzzySet(Z);
  error->addFuzzySet(P);
  error->addFuzzySet(PB);
  fuzzy->addFuzzyInput(error);

  // FuzzyInput
  FuzzyInput* derror = new FuzzyInput(2);
  derror->addFuzzySet(ZdE);
  derror->addFuzzySet(NdE);
  derror->addFuzzySet(NBdE);
```

Gambar 4.6 Tampilan listing Program input Fungsi Keanggotaan

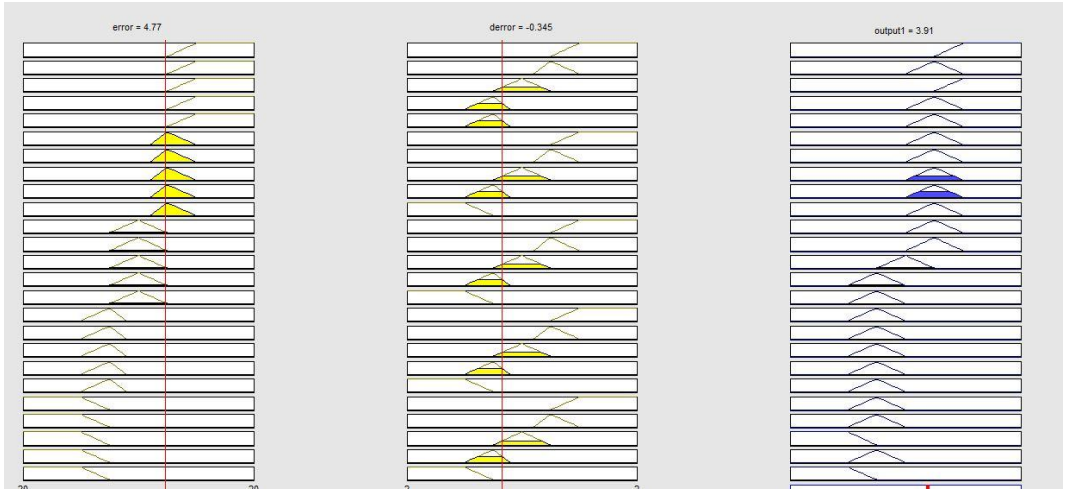
```
// Building FuzzyRule
FuzzyRuleAntecedent* iferrorPB= new FuzzyRuleAntecedent();
iferrorPB->joinSingle(PB);
FuzzyRuleAntecedent* iferrorPBAnddErrorPBdE= new FuzzyRuleAntecedent();
iferrorPBAnddErrorPBdE->joinWithAND(PB, PBdE);
FuzzyRuleConsequent* thentorsiTB = new FuzzyRuleConsequent();
thentorsiTB->addOutput(TB);
FuzzyRule* fuzzyRule1 = new FuzzyRule(1, iferrorPBAnddErrorPBdE,thentorsiTB);
fuzzy->addFuzzyRule(fuzzyRule1);

FuzzyRuleAntecedent* iferrorP= new FuzzyRuleAntecedent();
iferrorP->joinSingle(P);
FuzzyRuleAntecedent* iferrorPAnddErrorZdE= new FuzzyRuleAntecedent();
iferrorPAnddErrorZdE->joinWithAND(P, ZdE);
FuzzyRuleConsequent* thentorsiT = new FuzzyRuleConsequent();
thentorsiT->addOutput(T);
FuzzyRule* fuzzyRule2 = new FuzzyRule(2, iferrorPAnddErrorZdE,thentorsiT);
```

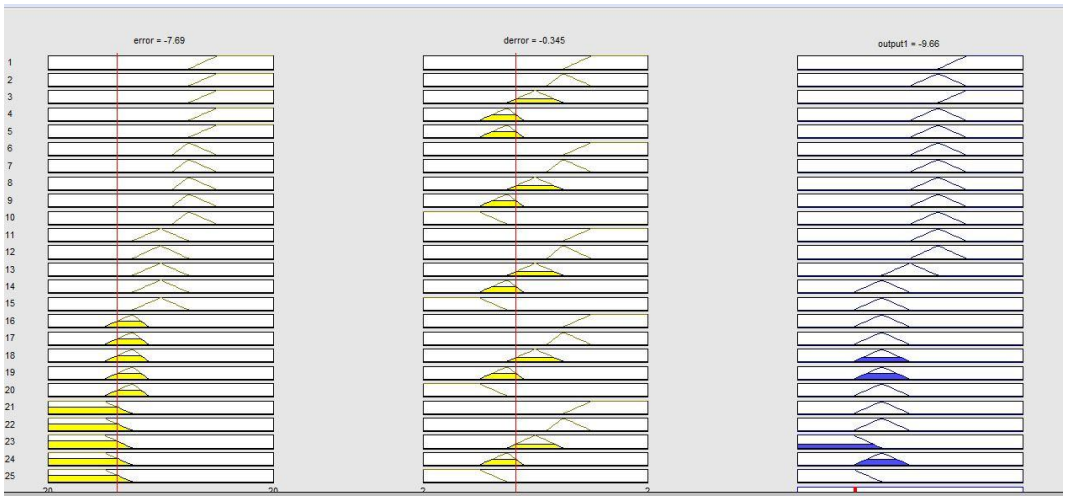
Gambar 4.7 Tampilan listing Program Fuzzy Rule

```
fuzzy->setInput(1, error);
fuzzy->setInput(2, d_error);
fuzzy->fuzzify();
float output1 = fuzzy->defuzzify(1);
```

Gambar 4.8 Tampilan listing Program defuzzyfikasi



Gambar 4.9 Hasil simulasi MATLAB error= 4.77 dan d_error=-0.345



Gambar 4.10 Hasil simulasi MATLAB error= -7 dan d_error=-0.345

hasil dari sistem fuzzy akan dibulatkan terlebih dahulu, setelah itu akan ditambahkan nilai torsi pada setiap motor. Pada saat nilai error ketinggian sebesar 4.77 cm dan d_error nya -0.345 maka output torsi nya adalah 3.91, nilai output tersebut akan ditambahkan pada nilai torsi pada setiap motor

4.3 Pengujian dan Analisis Bagian Keluaran

Pengujian dilakukan untuk mengetahui respon keluaran terhadap input transmitter, apabila diberikan masukan berupa kendali propeller, output motor berupa nilai duty cycle

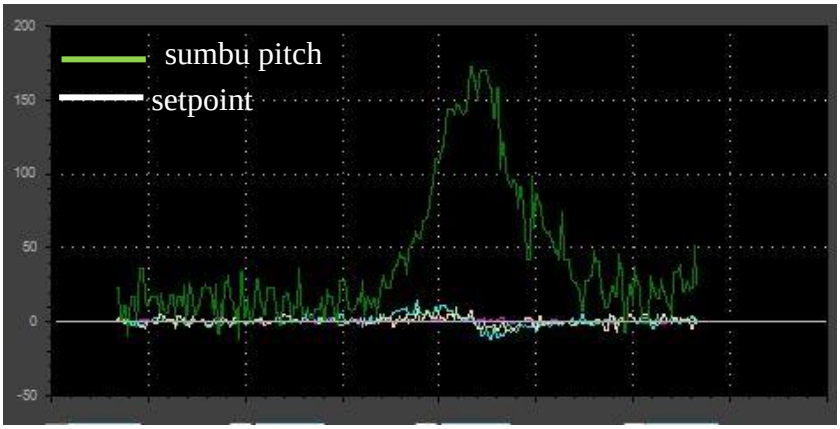
Tabel 4.14 Respon motor terhadap inputan transmittter

Input transmitter	Nilai torsi motor *brushless (rpm)*			
	Motor 1	Motor 2	Motor 3	Motor 4
Pitch depan	1000	1000	1500	1500
Pitch belakang	1500	1500	1000	1000
Roll kanan	1000	1500	1000	1500
Roll kiri	1500	1000	1500	1000
Yaw CW	1500	1000	1000	1500
Yaw CCW	1000	1500	1500	1000

Pengujian keluaram motor dilakukan untuk mengetahui respon dari motor terhadap perubahan nilai dari transmitter. Pada saat transmitter memberikan input pitch yang besar dari nilai netral *stick* , maka motor akan bermanuver kedepan , begitu juga pada saat diberikan nilai input roll yang besar, keluaran dari motor sesuai dengan yang diharapkan

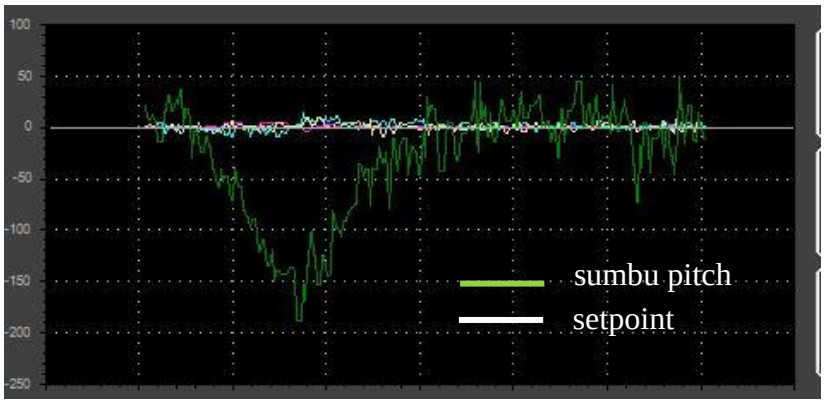
4.4 Hasil Implementasi Pada Sistem

Implementasi kendali auto level dilaksanakan dengan cara memberi gangguan terhadap quadcopter dengan cara memiringkan sumbu x dan sumbu y. Hasil dari respon sudut *roll* dan *pitch* dengan adanya gangguan ditunjukan pada gambar dibawah ini :



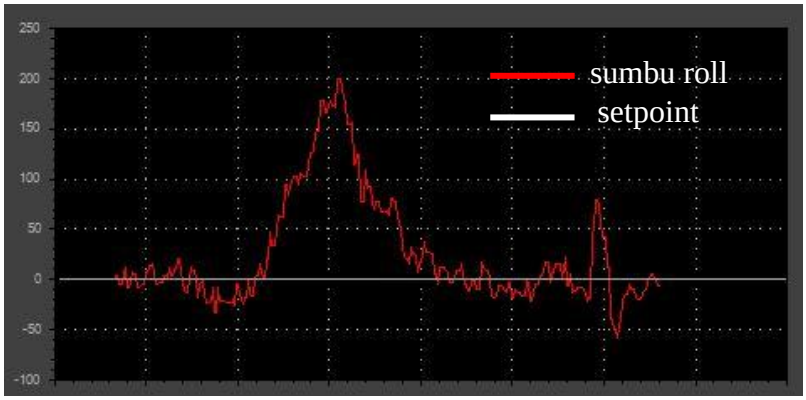
Gambar 4.11 Respon sudut *pitch* dimiringkan sebesar 10 derajat kedepan

Dari gambar di atas menunjukan bahwa *quadcopter* dapat mempertahankan posisi dengan bentuk gangguan dimiringkan sebesar 10 derajat, akan tetapi pada grafik diatas terdapat nilai osilasi yang cukup sedikit yang menyebabkan *quadcopter* drift kearah depan.



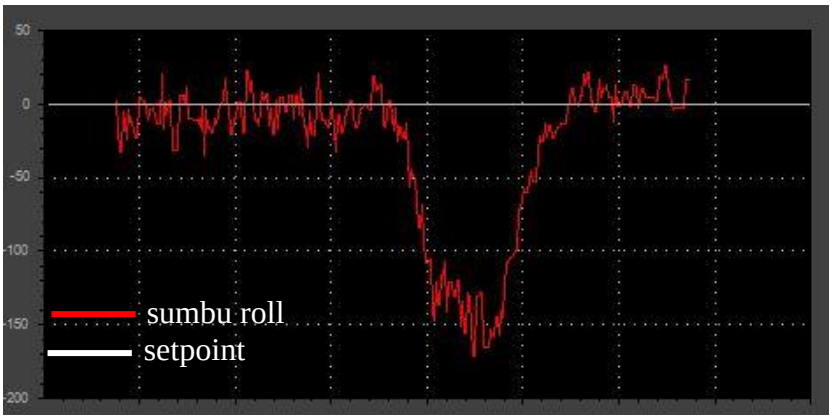
Gambar 4.12 Respon sudut *pitch* dimiringkan sebesar 10 derajat kebelakang

Pada pengujian sudut pitch dengan memiringkan *quadcopter* ke belakang, dapat disimpulkan bahwa sudut *pitch* dari *quadcopter* dapat merespon dengan cepat gangguan yang diberikan akan tetapi nilai dari *steady state error* yang tinggi yang menyebabkan pergerakan *quadcopter* sedikit maju.



Gambar 4.13 Respon sudut *roll* dimiringkan sebesar 10 derajat kekanan

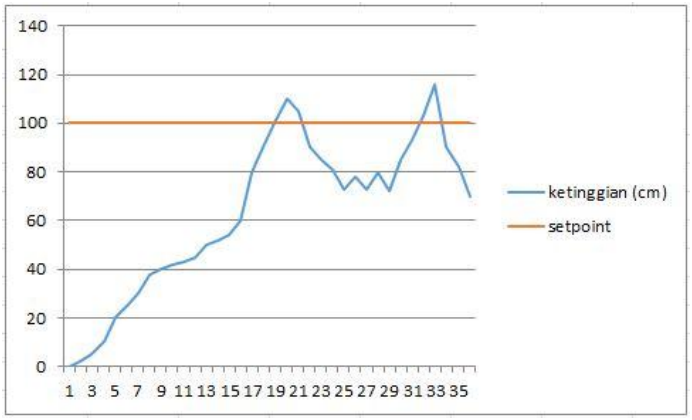
Pada pengujian sudut *roll* kearah positif dapat disimpulkan bahwa sudut *roll* mempunyai nilai steady state error yang kecil sehingga nilai pergerakan kestabilan sumbu *roll* tidak menyebabkan *drifting* pada *quadcopter*.



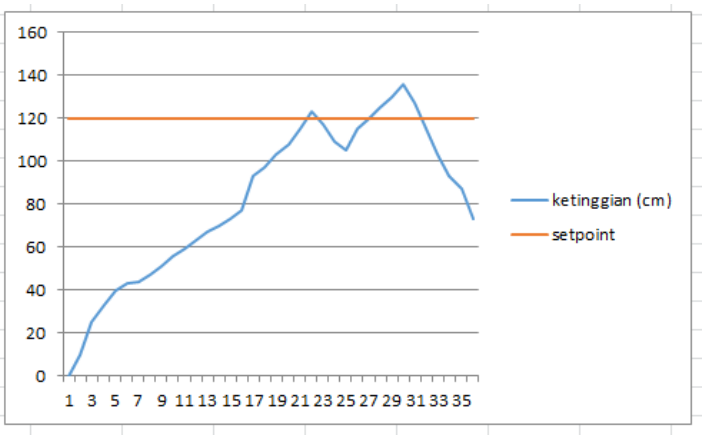
Gambar 4.14 Respon sudut roll dimiringkan sebesar 10 derajat kekiri

Pengujian *auto level* tuning pid secara keseluruhan dapat disimpulkan bahwa sistem kendali *auto level* sudah berjalan sesuai dengan yang diharapkan, pembacaan nilai *steady state error* yang ditunjukan oleh grafik tidak terlihat jelas oleh mata, akan tetapi nilai *steady state error* tersebut cukup mempengaruhi dalam *hovering quadcopter*.

Kendali fuzzy digunakan untuk menahan ketinggian, dibawah ini merupakan hasil respon sistem kendali fuzzy altitude hold.



Gambar 4.15 Grafik respon sistem fuzzy pada ketinggian 1m



Gambar 4.16 Respon sistem fuzzy pada ketinggian 1.2m

Pada saat quadcopter pertama kali take off dikendalikan oleh telemetry sehingga kecepatan naik ke setpoint dapat ditentukan dengan cepat, akan tetapi sistem kendali belum bekerja dengan baik sehingga terjadi osilasi untuk mencapai setpoint, error yang terjadi cukup lah besar dan sangat terlihat oleh karena itu dibutuhkan pemahaman yang lebih dalam mengenai program kendali fuzzy dan program Multiwii untuk dapat menyempurnakan sistem kendali yang dirancang