

## **BAB IV**

### **HASIL DAN PEMBAHASAN**

Untuk mengetahui kinerja dari sistem navigasi waypoint pada quadcopter ini sesuai dengan harapan, maka perlu dilakukan pengujian terhadap alat tersebut. Untuk memudahkan pengujian alat, maka akan dilakukan pengujian secara terpisah dan secara keseluruhan, tujuan pengujian adalah untuk mengetahui apakah sistem dapat bekerja dengan baik sesuai dengan perencanaan atau tidak

#### **4.1. Pengukuran**

Proses pengukuran dilakukan pada komponen agar mempermudah dalam menganalisa kesalahan serta memperbaiki untuk kesempurnaan. Pengujian sistem meliputi beberapa bagian, yaitu :

1. Pengujian Modul Arduino UNO.
2. Pengujian Modul Kompas HMC5883L.
3. Pengujian Sensor *Gyro* L3G4200D.
4. Pengujian GPS NEO-7m.
5. Pengujian Motor *Brushless*.
6. Pengujian Komunikasi Serial Menggunakan Modul *Wireless* 3DR.
7. Pengujian Keseimbangan Quadcopter.
8. Pengujian GUI.
9. Pengujian Sistem Terintegrasi (Keseluruhan).

##### **4.1.1. Pengujian Modul Arduino UNO**

Pengujian *arduino uno* dilakukan bertujuan untuk mengetahui sistem *board arduino uno* dapat bekerja dengan baik sesuai dengan yang diharapkan. Pengujian dilakukan dengan menyalakan sebuah *LED* (L1) yang tersedia pada *board arduino*

*uno* sendiri yang terhubung langsung dengan pin 13 (*pin digital*). Berikut ini adalah penggalan *listing program* pada Arduino Uno untuk menyalakan LED.

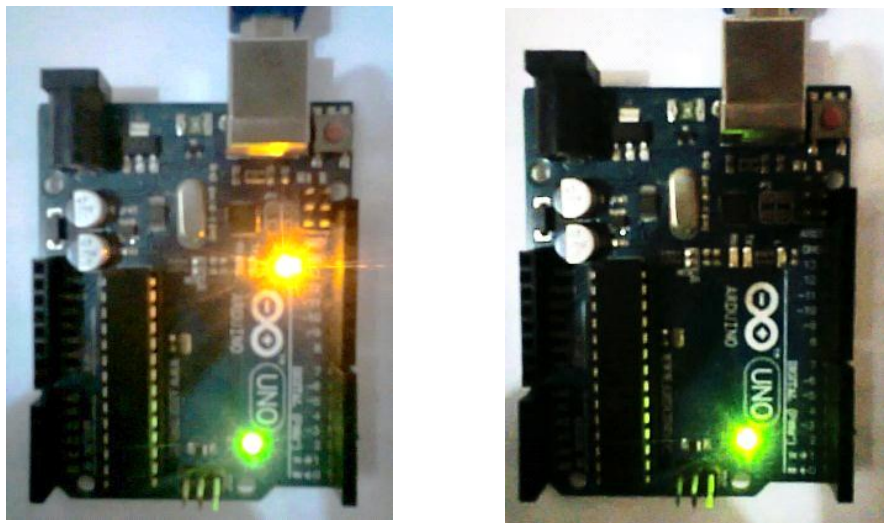
```
int ledPin = 13;

void setup()
{
  pinMode(ledPin, OUTPUT);
}

void loop()
{
  digitalWrite(ledPin, HIGH);
  delay(1000);
  digitalWrite(ledPin, LOW);
  delay(1000);
}
```

**Gambar 4.1** *Listing program menyalakan dan mematikan LED*

Berdasarkan pengujian dan analisa yang telah dilakukan, berikut gambar 4.1 hasil pengujian menyalakan dan mematikan *LED* (L1) yang terdapat pada *board arduino uno* sesuai dengan *listing program* yang digunakan.



**Gambar 4.2** *Mematikan dan menyalakan LED pada board arduino uno*

#### 4.1.2. Pengujian Sensor Kompas HMC5883L

Pengujian sensor kompas HMC5883L bertujuan untuk mengetahui derajat pada tiap arah mata angin. Berikut adalah penggalan *listing program* dan rangkaian sensor kompas HMC5883L untuk mengetahui data derajat pada tiap arah mata angin.

```
#include <Wire.h>
#include "HMC5883L.h"

HMC5883L compass;
int error = 0;

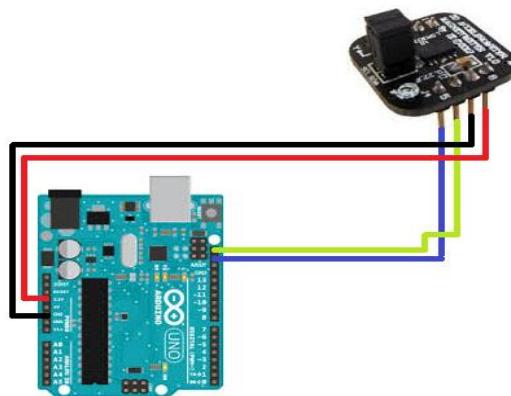
void setup()
{
  Serial.begin(9600);

  Serial.println("Starting the I2C interface.");
  Wire.begin();

  Serial.println("Constructing new HMC5883L");
  compass = HMC5883L();

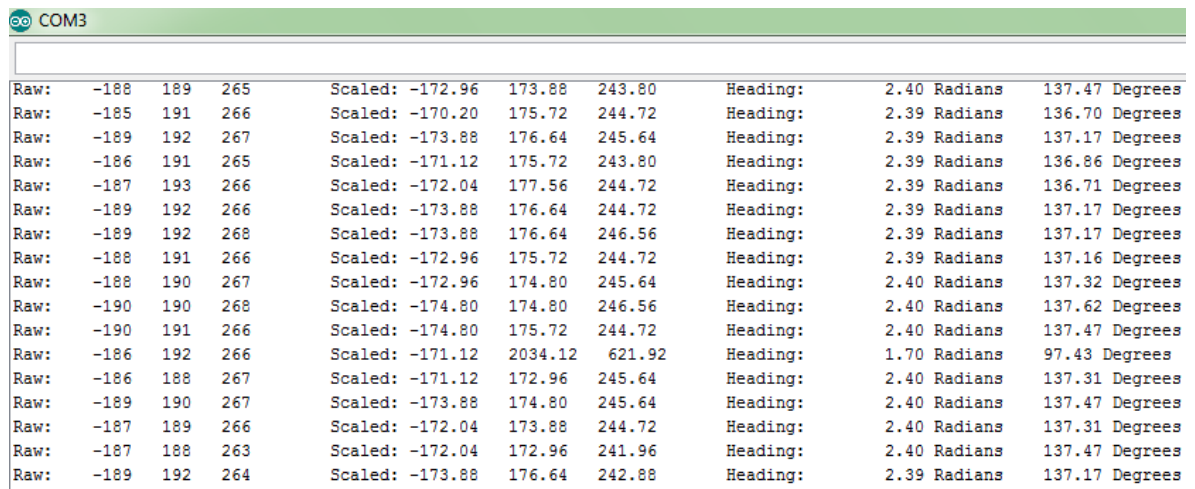
  Serial.println("Setting scale to +/- 1.3 Ga");
  error = compass.SetScale(1.3);
  if(error != 0)
    Serial.println(compass.GetErrorText(error));
}
```

**Gambar 4.3** *Listing program pengujian sensor kompas HMC5883L*



**Gambar 4.4** Skema rangkaian arduino uno dan sensor kompas HMC5883L

Pada gambar di Atas dapat dilihat bahwa sensor kompas dihubungkan dengan Arduino UNO. Tegangan yang digunakan sensor kompas ini adalah 5 volt. Pengambilan data sensor ini menggunakan komunikasi serial I2C dengan port SDA dan SCL yang dihubungkan pada Arduino UNO.



Raw:	-188	189	265	Scaled:	-172.96	173.88	243.80	Heading:	2.40 Radians	137.47 Degrees
Raw:	-185	191	266	Scaled:	-170.20	175.72	244.72	Heading:	2.39 Radians	136.70 Degrees
Raw:	-189	192	267	Scaled:	-173.88	176.64	245.64	Heading:	2.39 Radians	137.17 Degrees
Raw:	-186	191	265	Scaled:	-171.12	175.72	243.80	Heading:	2.39 Radians	136.86 Degrees
Raw:	-187	193	266	Scaled:	-172.04	177.56	244.72	Heading:	2.39 Radians	136.71 Degrees
Raw:	-189	192	266	Scaled:	-173.88	176.64	244.72	Heading:	2.39 Radians	137.17 Degrees
Raw:	-189	192	268	Scaled:	-173.88	176.64	246.56	Heading:	2.39 Radians	137.17 Degrees
Raw:	-188	191	266	Scaled:	-172.96	175.72	244.72	Heading:	2.39 Radians	137.16 Degrees
Raw:	-188	190	267	Scaled:	-172.96	174.80	245.64	Heading:	2.40 Radians	137.32 Degrees
Raw:	-190	190	268	Scaled:	-174.80	174.80	246.56	Heading:	2.40 Radians	137.62 Degrees
Raw:	-190	191	266	Scaled:	-174.80	175.72	244.72	Heading:	2.40 Radians	137.47 Degrees
Raw:	-186	192	266	Scaled:	-171.12	2034.12	621.92	Heading:	1.70 Radians	97.43 Degrees
Raw:	-186	188	267	Scaled:	-171.12	172.96	245.64	Heading:	2.40 Radians	137.31 Degrees
Raw:	-189	190	267	Scaled:	-173.88	174.80	245.64	Heading:	2.40 Radians	137.47 Degrees
Raw:	-187	189	266	Scaled:	-172.04	173.88	244.72	Heading:	2.40 Radians	137.31 Degrees
Raw:	-187	188	263	Scaled:	-172.04	172.96	241.96	Heading:	2.40 Radians	137.47 Degrees
Raw:	-189	192	264	Scaled:	-173.88	176.64	242.88	Heading:	2.39 Radians	137.17 Degrees

**Gambar 4.5** Tampilan data sensor kompas HMC5883L pada serial monitor arduino

Dari gambar serial monitor di Atas ditampilkan nilai 3-axis dari sensor kompas dan heading dari kompas dalam derajat dan radians. Pada saat pengambilan data ini, posisi kompas dalam keadaan diam.

#### 4.1.3. Pengujian Sensor Gyro L3G4200D

Pengujian sensor gyro L3G4200D bertujuan untuk mengetahui perubahan sudut yang terjadi pada sensor dengan 3axis sensor sebagai koreksi posisi pada *quadcopter*. Berikut ini adalah listing program dan rangkaian untuk pengujian sensor gyro.

```

#include <Wire.h>

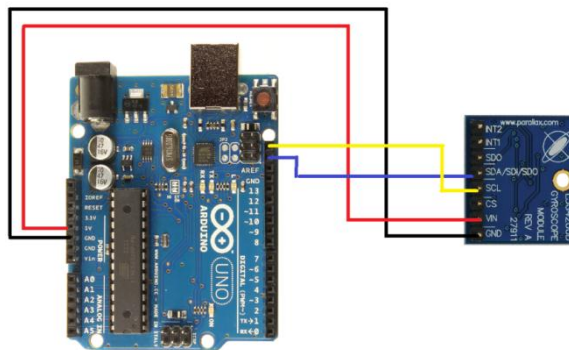
int cal_int;
unsigned long UL_timer;
double gyro_pitch, gyro_roll, gyro_yaw;
double gyro_roll_cal, gyro_pitch_cal, gyro_yaw_cal;
byte highByte, lowByte;

void setup(){
  Wire.begin();
  Serial.begin(57600);

  Wire.beginTransmission(105);
  Wire.write(0x20);
  Wire.write(0x0F);
  Wire.endTransmission();
  Wire.beginTransmission(105);
  Wire.write(0x23);
  Wire.write(0x80);
  Wire.endTransmission();
  delay(250);

```

**Gambar 4.6** Listing program pengujian sensor gyro L3G4200D



**Gambar 4.7** Skema rangkaian arduino uno dan sensor Gyro L3G4200D

Pengujian modul gyro dilakukan dengan merubah posisi *pitch*, *roll* dan *yaw* gyro. Pengujian bertujuan untuk mengetahui perubahan data pada gyro. Berikut ini adalah gambar perubahan data gyro berdasarkan *pitch*, *roll* dan *yaw*.

```

COM8 (Arduino/Genuino Uno)

Pitch:+7 NoU Roll:+2 --- Yaw:+0 ---
Pitch:+14 NoU Roll:+0 --- Yaw:+0 ---
Pitch:+13 NoU Roll:+1 --- Yaw:-1 ---
Pitch:+10 NoU Roll:-0 --- Yaw:+0 ---
Pitch:+13 NoU Roll:-1 --- Yaw:+2 ---
Pitch:+12 NoU Roll:+0 --- Yaw:-1 ---
Pitch:+11 NoU Roll:-1 --- Yaw:-0 ---
Pitch:+10 NoU Roll:-1 --- Yaw:+0 ---
Pitch:+8 NoU Roll:-0 --- Yaw:+1 ---
Pitch:+11 NoU Roll:+1 --- Yaw:+1 ---
Pitch:+10 NoU Roll:+0 --- Yaw:-0 ---
Pitch:+9 NoU Roll:-1 --- Yaw:+0 ---
Pitch:+9 NoU Roll:-1 --- Yaw:+1 ---
Pitch:+9 NoU Roll:-1 --- Yaw:+1 ---
Pitch:+9 NoU Roll:+1 --- Yaw:+1 ---

```

(a)

```

COM8 (Arduino/Genuino Uno)

Pitch:-12 NoD Roll:+0 --- Yaw:-1 ---
Pitch:-22 NoD Roll:+0 --- Yaw:-1 ---
Pitch:-26 NoD Roll:-0 --- Yaw:-1 ---
Pitch:-7 NoD Roll:+1 --- Yaw:+2 NoR
Pitch:-1 --- Roll:+0 --- Yaw:-1 ---
Pitch:-8 NoD Roll:+0 --- Yaw:-1 ---
Pitch:-18 NoD Roll:-1 --- Yaw:-0 ---
Pitch:-20 NoD Roll:+0 --- Yaw:-1 ---
Pitch:-9 NoD Roll:-1 --- Yaw:-0 ---
Pitch:-8 NoD Roll:+1 --- Yaw:+1 ---
Pitch:-15 NoD Roll:+1 --- Yaw:+1 ---
Pitch:-21 NoD Roll:+1 --- Yaw:+1 ---
Pitch:-20 NoD Roll:-0 --- Yaw:-2 ---
Pitch:-11 NoD Roll:-1 --- Yaw:-2 ---
Pitch:-16 NoD Roll:+0 --- Yaw:-1 ---
Pitch:-6 NoD Roll:-1 --- Yaw:-2 ---

```

(b)

**Gambar 4.8** (a) Tampilan data sensor Gyro dimiringkan ke depan (b) Tampilan data sensor Gyro dimiringkan ke belakang

Berdasarkan data pada gambar di Atas, pada saat modul gyro dimiringkan ke depan, maka data *pitch* bernilai positif dan pada saat gyro dimiringkan ke belakang, maka data *pitch* akan berubah negatif. Hal ini menandakan *pitch* pada gyro berfungsi dengan baik.

```

COM8 (Arduino/Genuino Uno)

Pitch:-3 NoD Roll:+4 RwD Yaw:-3 NoL
Pitch:+1 --- Roll:+9 RwD Yaw:-1 ---
Pitch:+2 NoU Roll:+11 RwD Yaw:+1 ---
Pitch:+1 --- Roll:+11 RwD Yaw:-0 ---
Pitch:+0 --- Roll:+7 RwD Yaw:-1 ---
Pitch:-2 --- Roll:+13 RwD Yaw:-1 ---
Pitch:+1 --- Roll:+5 RwD Yaw:+1 ---
Pitch:-0 --- Roll:+8 RwD Yaw:+0 ---
Pitch:+1 --- Roll:+11 RwD Yaw:+2 NoR
Pitch:-0 --- Roll:+9 RwD Yaw:-0 ---
Pitch:-0 --- Roll:+11 RwD Yaw:-1 ---
Pitch:+1 --- Roll:+7 RwD Yaw:+0 ---
Pitch:+0 --- Roll:+16 RwD Yaw:+1 ---
Pitch:+0 --- Roll:+10 RwD Yaw:+2 ---
Pitch:-2 --- Roll:+12 RwD Yaw:-0 ---

```

(a)

```

COM8 (Arduino/Genuino Uno)

Pitch:-1 --- Roll:-12 RwU Yaw:-1 ---
Pitch:+3 NoU Roll:-20 RwU Yaw:+3 NoR
Pitch:-1 --- Roll:-23 RwU Yaw:-1 ---
Pitch:+1 --- Roll:-25 RwU Yaw:+0 ---
Pitch:-0 --- Roll:-16 RwU Yaw:-1 ---
Pitch:-1 --- Roll:-7 RwU Yaw:-1 ---
Pitch:+0 --- Roll:-6 RwU Yaw:+0 ---
Pitch:-0 --- Roll:-4 RwU Yaw:-0 ---
Pitch:+0 --- Roll:-6 RwU Yaw:+0 ---
Pitch:-0 --- Roll:-8 RwU Yaw:+0 ---
Pitch:-0 --- Roll:-8 RwU Yaw:+0 ---
Pitch:-1 --- Roll:-9 RwU Yaw:-1 ---
Pitch:+2 --- Roll:-5 RwU Yaw:+4 NoR
Pitch:+1 --- Roll:-2 RwU Yaw:+1 ---

```

(b)

**Gambar 4.9** (a) Tampilan data sensor Gyro dimiringkan ke kanan (b) Tampilan data sensor Gyro dimiringkan ke kiri

Berdasarkan data pada gambar di Atas, pada saat modul gyro dimiringkan ke kanan, maka data *roll* bernilai positif dan pada saat gyro dimiringkan ke kiri, maka data *roll* akan berubah negatif. Hal ini menandakan *roll* pada gyro berfungsi dengan baik.

COM8 (Arduino/Genuino Uno)				COM8 (Arduino/Genuino Uno)			
Pitch	Roll	Yaw	Roll	Pitch	Roll	Yaw	Roll
Pitch:-1 ---	Roll:-1 ---	Yaw:+8 NoR		Pitch:+1 ---	Roll:+0 ---	Yaw:-18 NoL	
Pitch:-1 ---	Roll:-0 ---	Yaw:+11 NoR		Pitch:+0 ---	Roll:-0 ---	Yaw:-12 NoL	
Pitch:+0 ---	Roll:+0 ---	Yaw:+12 NoR		Pitch:+1 ---	Roll:-0 ---	Yaw:-13 NoL	
Pitch:+0 ---	Roll:+1 ---	Yaw:+13 NoR		Pitch:+0 ---	Roll:-1 ---	Yaw:-7 NoL	
Pitch:-1 ---	Roll:-1 ---	Yaw:+14 NoR		Pitch:+1 ---	Roll:-0 ---	Yaw:-12 NoL	
Pitch:-0 ---	Roll:-1 ---	Yaw:+11 NoR		Pitch:+1 ---	Roll:-1 ---	Yaw:-17 NoL	
Pitch:+0 ---	Roll:+1 ---	Yaw:+11 NoR		Pitch:+0 ---	Roll:-0 ---	Yaw:-15 NoL	
Pitch:+0 ---	Roll:-1 ---	Yaw:+7 NoR		Pitch:+0 ---	Roll:-0 ---	Yaw:-11 NoL	
Pitch:-0 ---	Roll:-0 ---	Yaw:+13 NoR		Pitch:-0 ---	Roll:+1 ---	Yaw:-8 NoL	
Pitch:-1 ---	Roll:+1 ---	Yaw:+10 NoR		Pitch:+1 ---	Roll:-1 ---	Yaw:-9 NoL	
Pitch:-0 ---	Roll:-0 ---	Yaw:+10 NoR		Pitch:+0 ---	Roll:+0 ---	Yaw:-13 NoL	
Pitch:-0 ---	Roll:-0 ---	Yaw:+8 NoR		Pitch:+0 ---	Roll:+0 ---	Yaw:-10 NoL	
Pitch:+1 ---	Roll:+1 ---	Yaw:+8 NoR		Pitch:+1 ---	Roll:-1 ---	Yaw:-18 NoL	
Pitch:-1 ---	Roll:-2 ---	Yaw:+8 NoR		Pitch:+0 ---	Roll:-0 ---	Yaw:-15 NoL	
Pitch:-0 ---	Roll:+1 ---	Yaw:+11 NoR		Pitch:+0 ---	Roll:-1 ---	Yaw:-8 NoL	

**Gambar 4.10 (a)** Tampilan Data Sensor Gyro diputar ke Kanan **(b)** Tampilan Data Sensor Gyro diputar ke Kiri

Berdasarkan data pada gambar di Atas, pada saat modul gyro diputar ke kanan, maka data *yaw* bernilai positif dan pada saat gyro diputar ke kanan, maka data *yaw* akan berubah negatif. Hal ini menandakan *yaw* pada gyro berfungsi dengan baik.

#### 4.1.4. Pengujian GPS NEO-7m

Pengujian sensor GPS Neo-7m ini bertujuan untuk mengetahui nilai data longitude dan latitude pada beberapa titik dan juga untuk mengetahui apakah modul sensor GPS ini berfungsi dengan baik. Berikut ini adalah penggalan listing program dan rangkaian pengujian GPS NEO-7m.

```

#include <TinyGPS.h>
#include <SoftwareSerial.h>

long lat,lon;

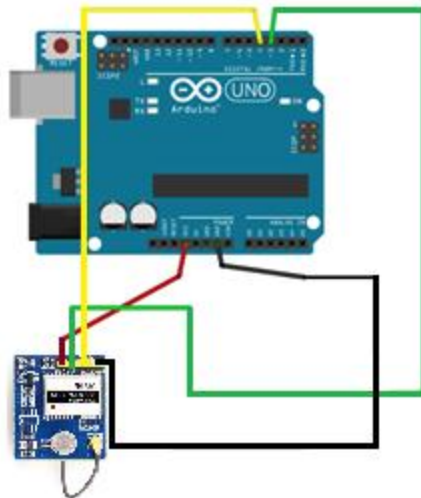
SoftwareSerial gpsSerial(2, 3);
TinyGPS gps;

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

void loop(){
  while(gpsSerial.available()){
    if(gps.encode(gpsSerial.read())){
      gps.get_position(&lat,&lon);
      Serial.print("Position: ");
      Serial.print("lat: ");Serial.print(lat);Serial.print(" ");
      Serial.print("lon: ");Serial.println(lon);
    }
  }
}

```

**Gambar 4.11** Listing program pengujian GPS NEO-7m

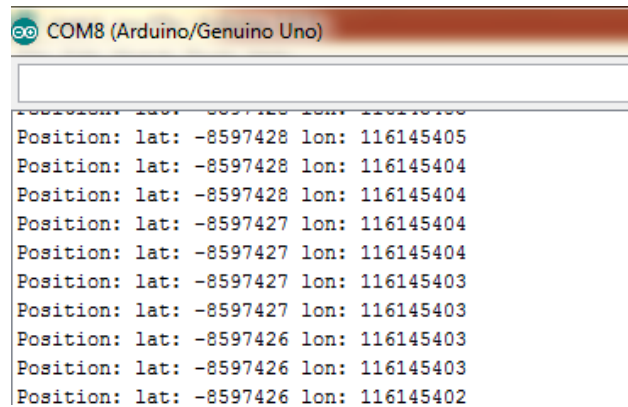


**Gambar 4.12** Skema Rangkaian Arduino uno dan Sensor GPS Neo-7m



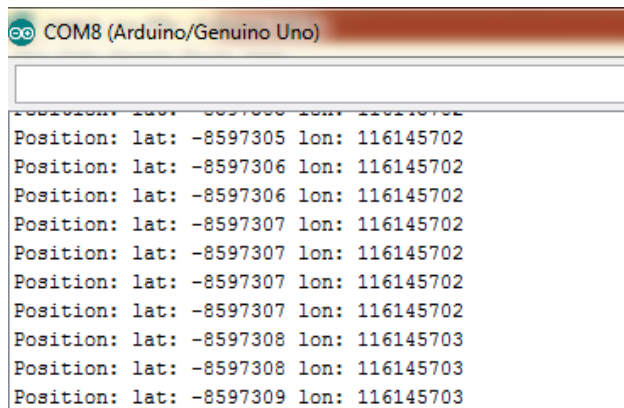
Dari gambar 4.12 kita dapat melihat konfigurasi port sensor GPS yang dihubungkan pada port arduino uno. Kabel hijau merupakan rx,kabel kuning yaitu tx,kabel merah merupakan supply/VCC modul sensor GPS dengan tegangan 3,3v dan kabel hitam adalah ground(negatif).

Pengujian GPS ini adalah dengan berpindah posisi ke 2 arah yang berbeda, untuk melihat perubahan data koordinat posisi dari GPS.



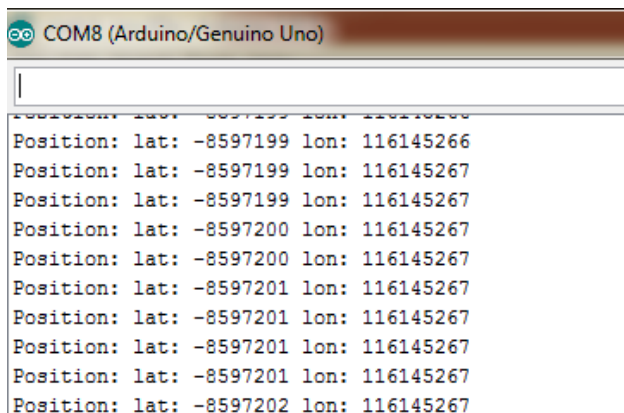
**Gambar 4.13** (a) Tampilan data sensor GPS NEO-7m pada serial monitor arduino awal

Gambar di Atas merupakan nilai terbaca pada GPS pada posisi awal. Nilai koordinat yang terbaca adalah *latitude* dengan nilai  $-8.597426^{\circ}$  dan *longitude* dengan nilai  $116.145402^{\circ}$ .



**Gambar 4.13 (b)** Tampilan data sensor GPS NEO-7m pada serial monitor arduino berpindah ke arah timur

Gambar di Atas merupakan nilai terbaca pada GPS dengan posisi GPS dipindahkan ke arah timur. Nilai koordinat yang terbaca adalah *latitude* dengan nilai  $-8.597309^{\circ}$  dan *longitude* dengan nilai  $116.145703^{\circ}$ .



**Gambar 4.13 (c)** Tampilan data sensor GPS NEO-7m pada serial monitor arduino berpindah ke arah utara

Gambar di Atas merupakan nilai terbaca pada GPS dengan posisi GPS dipindahkan ke arah utara. Nilai koordinat yang terbaca adalah *latitude* dengan nilai  $-8.597202^{\circ}$  dan *longitude* dengan nilai  $116.145267^{\circ}$ .

Dari gambar keseluruhan 4.13 di Atas, dapat kita lihat perubahan data GPS pada *longitude* dan *latitude*. Data yang terlihat pada gambar 4.13b adalah nilai dari *longitude* berubah signifikan, ini menandakan GPS berubah arah *longitudenya* dari

posisi awal. Sedangkan pada gambar 4.13c, nilai yang berubah signifikan adalah nilai *latitude*, ini menandakan GPS berubah arah *latitudenya* dari posisi awal. Kedua percobaan tersebut menandakan bahwa modul GPS dapat berfungsi dengan baik.

#### 4.1.5. Pengujian Motor Brushless

Pengujian motor Brushless bertujuan untuk mengetahui apakah brushless dapat bekerja dengan. Pengujian dilakukan dengan menggunakan *arduino uno* yang telah di program untuk memberikan input ke pada motor Brushless. Berikut ini adalah penggalan *Listing Program* dan rangkaian pengujian motor *Brushless*.

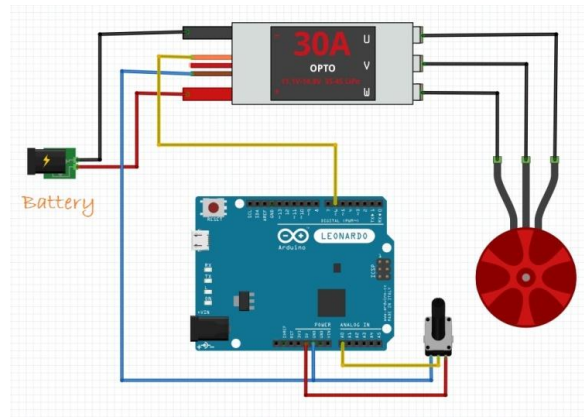
```
#include <Servo.h>

Servo esc;
int throttlePin = 0;

void setup(){
  esc.attach(9);
}

void loop()
{
  int throttle = analogRead(throttlePin);
  throttle = map(throttle, 0, 1023, 0, 179);
  esc.write(throttle);
}
```

**Gambar 4.14** *Listing Program* pengujian motor *Brushless*



**Gambar 4.15** Skema Pengujian Brushless Menggunakan ESC(*driver*)

Motor brushless diberikan PWM yang berubah melalui input arduino, kecepatan pada motor brushless ikut berubah seiring dengan nilai PWM yang diberikan. Hal ini membuktikan bahwa Brushless dan ESC(*driver*) berfungsi dengan baik.

#### **4.1.6. Pengujian Komunikasi Serial Menggunakan Modul *Wireless* 3DR**

Pengujian komunikasi serial pada *arduino uno* dilakukan dengan menghubungkan *port USB* pada *laptop* menggunakan kabel *USB arduino uno*. Pengujian dilakukan bertujuan untuk mengetahui nilai data yang dikirim atau yang diterima sesuai dengan yang diberikan. Perlu diperhatikan pada percobaan ini adalah *BaudRate* pada pengirim dan penerima harus sama, pada pengujian kali ini menggunakan *BaudRate* 57600. Berikut adalah penggalan *Listing Program* pengujian pengiriman data menggunakan komunikasi serial menggunakan modul *wireless* 3DR.

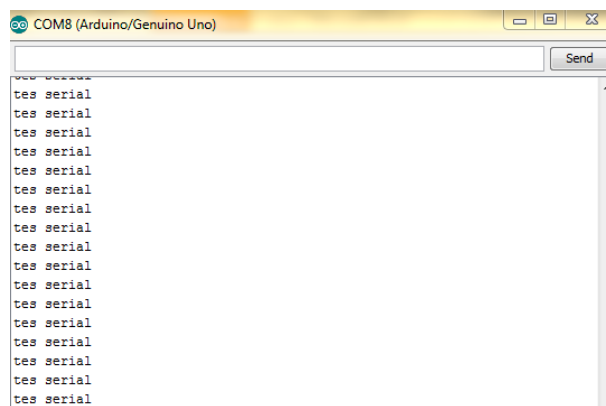
```

void setup(){
  Serial.begin (57600);
}
void loop()
{
  Serial.print (tes serial);
}

```

**Gambar 4.16** *Listing program* pengujian mengirim data pada arduino

Berdasarkan *listing* program diatas, “tes serial” merupakan tipe data *string* yang akan dikirim ke *laptop* melalui komunikasi serial dengan perintah *Serial.println(“tes serial”)*. Gambar 4.17 berikut hasil pengujian yang dapat dilihat pada *menu bar monitor serial framework Arduino Uno*.



**Gambar 4.17** Hasil pengujian mengirim data pada *arduino uno*.

Berikut ini adalah pengujian menerima data pada arduino uno melalui laptop menggunakan *wireless 3DR*. Gambar 4.18 di Bawah ini adalah *listing* program menerima data dari *laptop*.

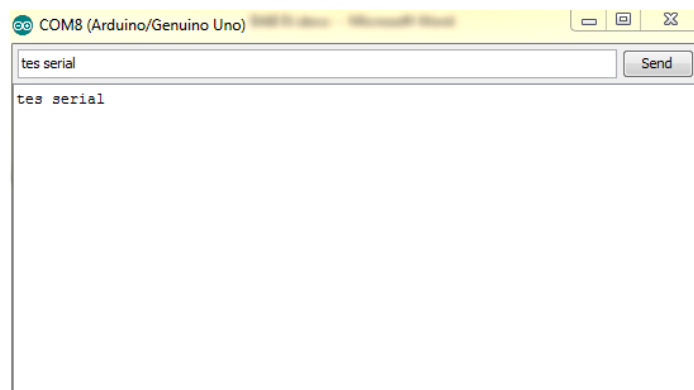
```

char data;
void setup()
{
  Serial.begin(9600);
}
void loop()
{
  if(Serial.available())
  {
    data = Serial.read();
    Serial.print(data);
  }
}

```

**Gambar 4.18** *Listing* program menerima data dari *laptop*

Berdasarkan Gambar 4.18 *listing* program di atas, nilai yang diterima pada *arduino uno* akan disimpan pada variabel “*data*” dengan perintah *Serial.read()*. Karena data yang diterima pada *Arduino* bertipe “*string*” maka *type* data pada *listing* program di atas yaitu “*char*” (karakter). Gambar 4.19 berikut adalah hasil pengujian yang telah dilakukan.



**Gambar 4.19** Hasil pengujian menerima data pada *arduino uno*.

Pengujian selanjutnya yaitu pengujian kemampuan 3DR mentransmisikan data. Pengujian ini dilakukan dengan tanpa halangan. Data yang dikirim berupa karakter “tes serial” dengan variasi jarak untuk mendapatkan jarak yang mnaksimal. Berikut adalah tabel hasil pengujian.

**Tabel 4.1** Pengujian jarak pengiriman data *wireless* 3DR

No	Jarak (m)	Data <i>Transmitter</i>	Data <i>Receiver</i>	Keterangan
1	1	tes serial	tes serial	Terkirim
2	10	tes serial	tes serial	Terkirim
3	20	tes serial	tes serial	Terkirim
4	30	tes serial	tes serial	Terkirim
5	40	tes serial	tes serial	Terkirim
6	50	tes serial	tes serial	Terkirim
7	60	tes serial	tes serial	Terkirim
8	70	tes serial	tes serial	Terkirim
9	80	tes serial	tes serial	Terkirim
10	90	tes serial	tes serial	Terkirim
11	95	tes serial	tes serial	Data kadang tidak terkirim
12	100	tes serial	-	Data tidak terkirim

Dari data pada tabel di Atas, modul *wireless* 3DR dapat mengirimkan data secara sempurna dengan tanpa halangan yaitu pada jarak 90 m. pada jarak 95 m data kadang tidak terkirim dan pada jarak 100 m data tidak terkirim sama sekali.

#### 4.1.7. Pengujian Keseimbangan Quadcopter

Pengujian keseimbangan quadcopter bertujuan untuk melihat apakah quadcopter dapat mempertahankan keseimbangan ketika mendapatkan gangguan dari segala arah.

**Tabel 4.2** Pengujian Keseimbangan Quadcopter

Keadaan Quadcopter	PWM ESC (uS)			
	ESC 1	ESC 2	ESC 3	ESC 4
Posisi Seimbang	1618	1596	1603	1625
Miring ke Depan	1936	1242	1262	2000
Miring ke Belakang	1369	1811	1874	1388
Miring ke Kanan	1876	1742	1345	1479
Miring ke Kiri	1359	1321	1861	1900

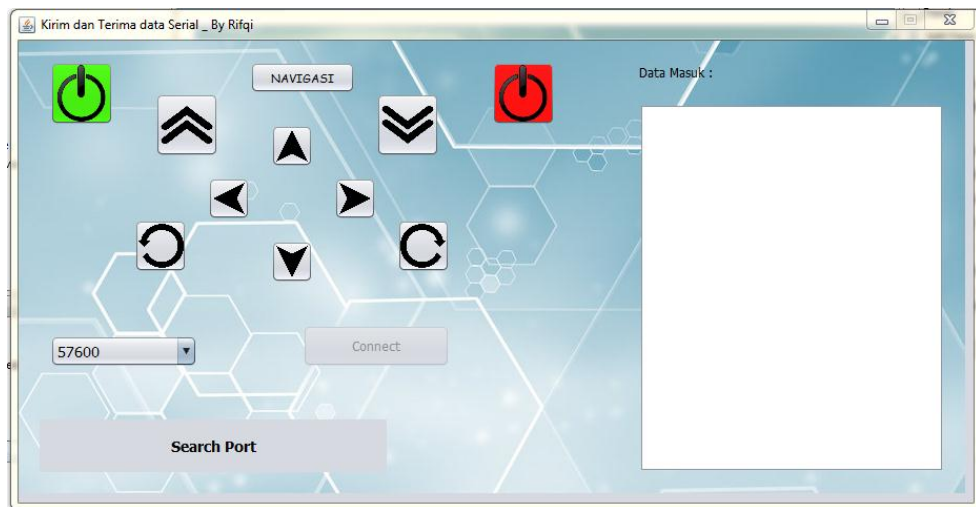
Data pada pengujian di Atas adalah data ketika quadcopter dimiringkan dengan ke 4 arah yang berbeda. Berdasarkan data pada pengujian, terlihat bahwa PWM pada dua buah ESC akan ditambahkan nilai PWM ESC-nya dan dua ESC yang lainnya akan dikurangi PWMnya, sehingga quadcopter akan dapat seimbang. Quadcopter dikatakan seimbang apabila nilai dari PWM dari keempat motor bernilai tidak jauh berbeda seperti pada data tabel di Atas.



#### 4.1.8 Pengujian GUI (*Graphical User Interface*)

Pengujian ini dilakukan untuk menguji tingkat keberhasilan GUI yang telah dibangun. Aplikasi GUI yang digunakan pada perancangan ini adalah NetBeans IDE 8.1. Pengujian dilakukan apakah GUI dengan quadcopter dapat berkomunikasi dengan baik. Langkah-langkah pengujian adalah sebagai berikut :

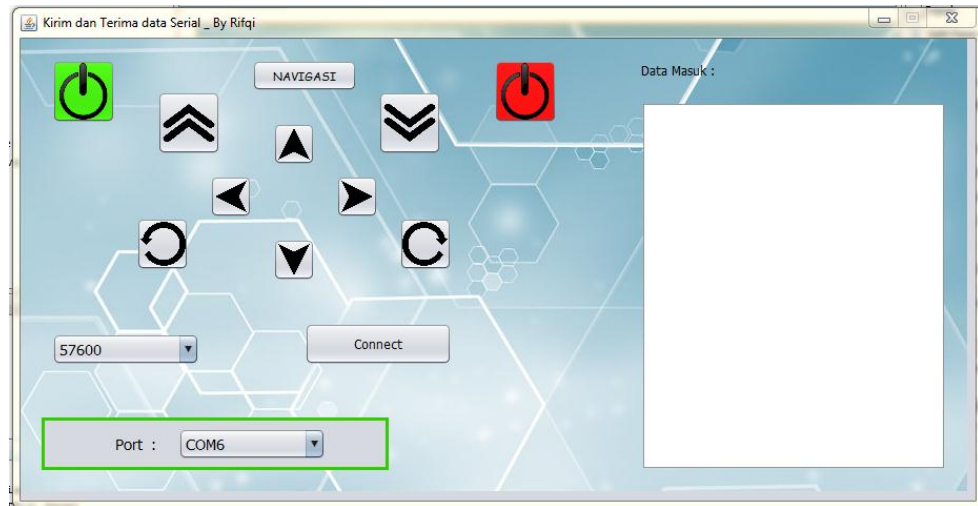
##### 1. *Run Program GUI*



**Gambar 4.20** Tampilan Awal GUI *Quadcopter*

Gambar di Atas adalah tampilan GUI quadcopter ketika pengguna menjalankan GUI *quadcopter* pada program NetBeans. Pengguna diharuskan menghubungkan *modul wireless radio* agar GUI dan *quadcopter* dapat berkomunikasi.

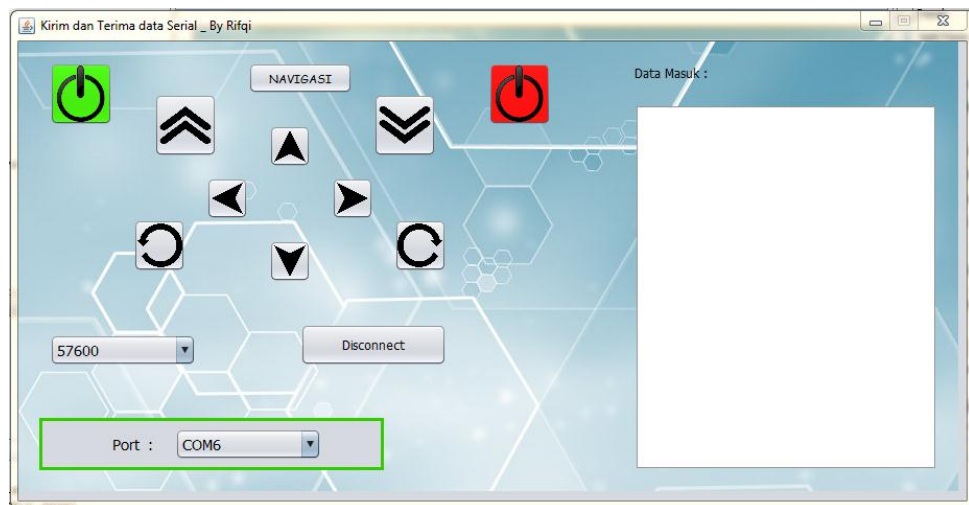
## 2. Memilih *Port Serial*



**Gambar 4.21** Mencari *Port Serial*

Setelah menajalankan GUI, pengguna memilih *port serial* dengan menekan tombol “*Search Port*” sehingga akan menampilkan *port serial* yang tersedia. Setelah tertampil, pengguna memilih port yang digunakan untuk komunikasi serial.

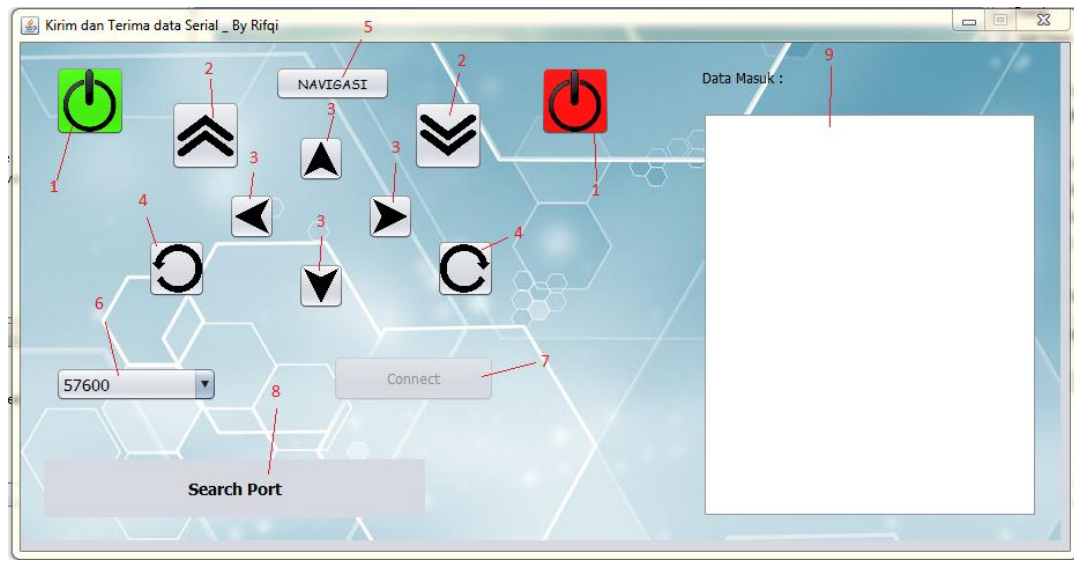
## 3. Menghubungkan GUI dengan Quadcopter



**Gambar 4.22** Menghubungkan GUI dengan *Quadcopter*

Langkah selanjutnya adalah menghubungkan GUI dengan *quadcopter* berdasarkan *port serial* yang telah dihubungkan tadi. Setelah terkoneksi, monitor serial akan menampilkan data pada *quadcopter*.

Adapun fungsi dari tombol pada GUI *quadcopter* adalah :



**Gambar 4.23** Fungsi Tombol GUI *Quadcopter*

1. Tombol *On* dan *Off* berfungsi untuk menyalakan atau mematikan quadcopter.
2. Berfungsi sebagai penaik atau penurun kecepatan putar motor quadcopter.
3. Tombol navigasi manual berfungsi untuk melakukan akselerasi secara manual.
4. Berfungsi untuk memutar arah *quadcopter* ke kanan atau ke kiri.
5. Tombol naavigasi berfungsi sebagai navigasi *waypoint* quadcopter.
6. Berfungsi untuk memilih *baud rate* yang kita gunakan.
7. Merupakan tombol untuk menghubungkan dan mematikan komunikasi serial *quadcopter* dengan GUI.
8. Berfungsi untuk mencari *port serial* yang tersedia.
9. Menampilkan data pada *quadcopter*.

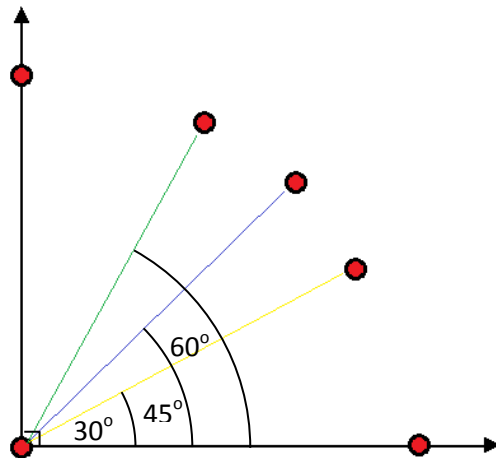
#### 4.1.9 Pengujian Sistem Terintegrasi (Keseluruhan)

Pada pengujian ini seluruh komponen sistem telah dirakit secara utuh. Langkah pertama yang dilakukan setelah proses perakitan selesai adalah membuat kode program (*listing*) pada software IDE (*integrated development environment*) Arduino yang kemudian akan di *upload* kedalam *flash memory* mikrokontroler pada modul. Pada proses *upload*, yang harus dilakukan adalah meng-*compile* program untuk mengetahui bahwa kode program sudah benar (*done compiling*) atau masih terdapat *error*. Selanjutnya memasang USB arduino ke PC untuk melakukan proses *upload* program kedalam mikrokontroler.

Pada pengujian sitem secara terintegrasi (keseluruhan) ini dilakukan pada satu titik koordinat tujuan. Hal ini dilalukan untuk mengetahui apakah quadcopter dapat mencapai titik tersebut ketika dijalankan fungsi *waypoint* pada quadcopter. Quadcopter akan diterbangkan pada ketinggian tertentu, kemudian quadcopter akan bernavigasi menuju titik tujuan ketika menekan tombol navigasi pada GUI. Berikut ini adalah pengujian secara keseluruhan dari *quadcopter*.

##### a. Pengujian dengan Merubah Titik Tujuan

Pengujian dilakukan dengan beberapa tahap, yaitu dengan merubah sudut pada setiap titik tujuan dengan jarak yang sama, yaitu 25 m dan hadap muka quadcopter pada sudut  $0^{\circ}$ . Adapun sudut yang digunakan yaitu  $0^{\circ}$ ,  $30^{\circ}$ ,  $45^{\circ}$ ,  $60^{\circ}$  dan  $90^{\circ}$ . Berikut ini adalah gambar ilustrasi dari pengujian



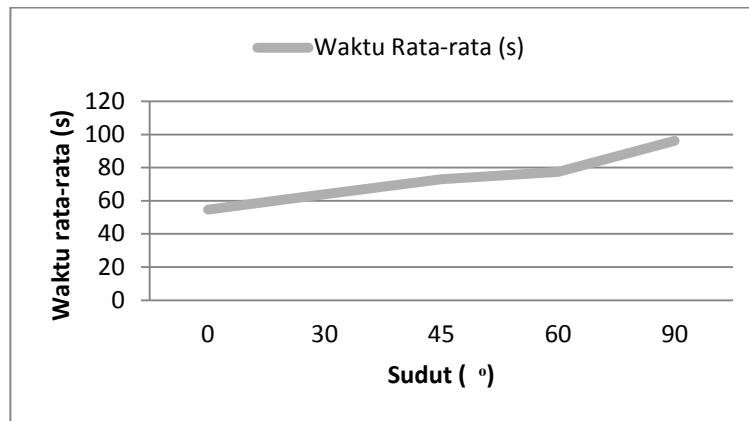
**Gambar 4.24** Ilustrasi Percobaan Titik Tujuan dengan Sudut yang Berbeda

Berikut ini adalah tabel data hasil percobaan waktu tempuh navigasi quadcopter :

**Tabel 4.3** Hasil percobaan waktu tempuh navigasi quadcopter

Sudut ( °)	Waktu (s)					Rata-rata (s)
	I	II	III	IV	V	
0	56.5	43.8	61.2	58.2	53.4	<b>54.6</b>
30	61.3	66.7	57.7	68.4	65.9	<b>64</b>
45	67.6	81.6	72.5	74.2	69.1	<b>73</b>
60	74.5	83.9	75.1	78.2	75.9	<b>77.5</b>
90	98.4	94.4	89.3	103.6	95.1	<b>96.2</b>

Berikut ini adalah grafik yang dapat dibuat berdasarkan data hasil percobaan pada tabel di Atas.

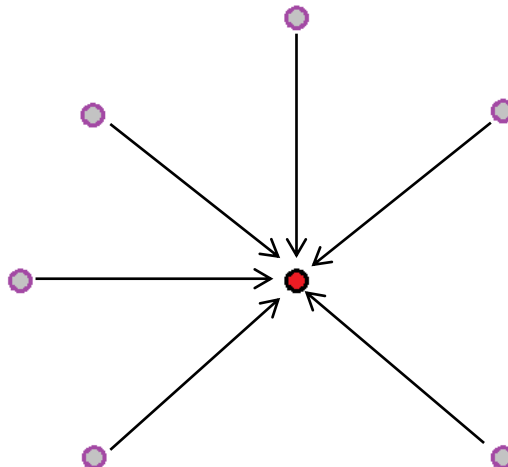


**Gambar 4.25** Grafik Waktu Rata-rata terhadap Nilai Sudut

Dari data pada tabel hasil percobaan di Atas, dapat diketahui pengujian berhasil dilakukan dengan nilai yang memiliki selisih kecil terhadap nilai rata-rata. Terlihat pada grafik semakin besar sudutnya, maka semakin lama pula waktu yang dibutuhkan *quadcopter* untuk mencapai tujuan.

#### b. Pengujian dengan merubah titik awal

Pengujian ini dilakukan pada satu titik tujuan, dengan posisi awal berubah-ubah dengan nilai koordinat titik tujuan yaitu *Latitude* -8.571489 dan *Longitude* 116.102384. Berikut ini adalah gambar ilustrasi pengujian yang dilakukan.



**Gambar 4.26** Ilustrasi Percobaan dengan Titik Awal yang Berbeda

**Tabel 4.4** Hasil Percobaan Waktu Tempuh dan Posisi Akhir *Quadcopter* dengan nilai koordinat titik tujuan *Latitude* -8.571489 dan *Longitude* 116.102384.

Percobaan ke -	Posisi Awal <i>Quadcopter</i> ( ° )		Posisi Akhir <i>Quadcopter</i> ( ° )		Waktu (s)	Jarak (m)
	Longitude	Latitude	Longitude	Latitude		
1	116.102574	-8.571459	116.102360	-8.571501	35.2	3.18
2	116.102516	-8.571317	116.102344	-8.571455	30.5	1.91
3	116.102298	-8.571328	116.102368	-8.571496	28.5	1.68
4	116.102304	-8.571607	116.102376	-8.571487	30.9	1.51
5	116.102264	-8.571615	116.102368	-8.571489	31.6	1.50
6	116.102352	-8.571609	116.102368	-8.571457	33.3	3.10

Berdasarkan data pada tabel 4.4, nilai selisih jarak terkecil yang didapatkan dari posisi akhir *quadcopter* pada saat bernavigasi menuju titik tujuan adalah 1.5 meter dan selisih terjauh adalah 3.18 meter. Ini menunjukkan bahwa *quadcopter* telah mencapai titik tujuan, karena nilai akurasi posisi *quadcopter* terhadap titik tujuan adalah 5 meter, yaitu apa bila *quadcopter* berada pada rentang 5 meter dari titik tujuan, maka *quadcopter* dianggap telah sampai pada titik tujuan.