

## BAB III

### METODOLOGI PENELITIAN

#### 3.1 Waktu dan Tempat Penelitian

Penelitian tugas akhir sistem umpan balik kendali *close loop* antenna penjejak 2 axis akan dilaksanakan :

Waktu Pelaksanaan : 17 Oktober 2018 s.d Selesai

Tempat Pelaksanaan : Laboratorium Teknik Elektro Universitas  
Muhammadiyah Yogyakarta.

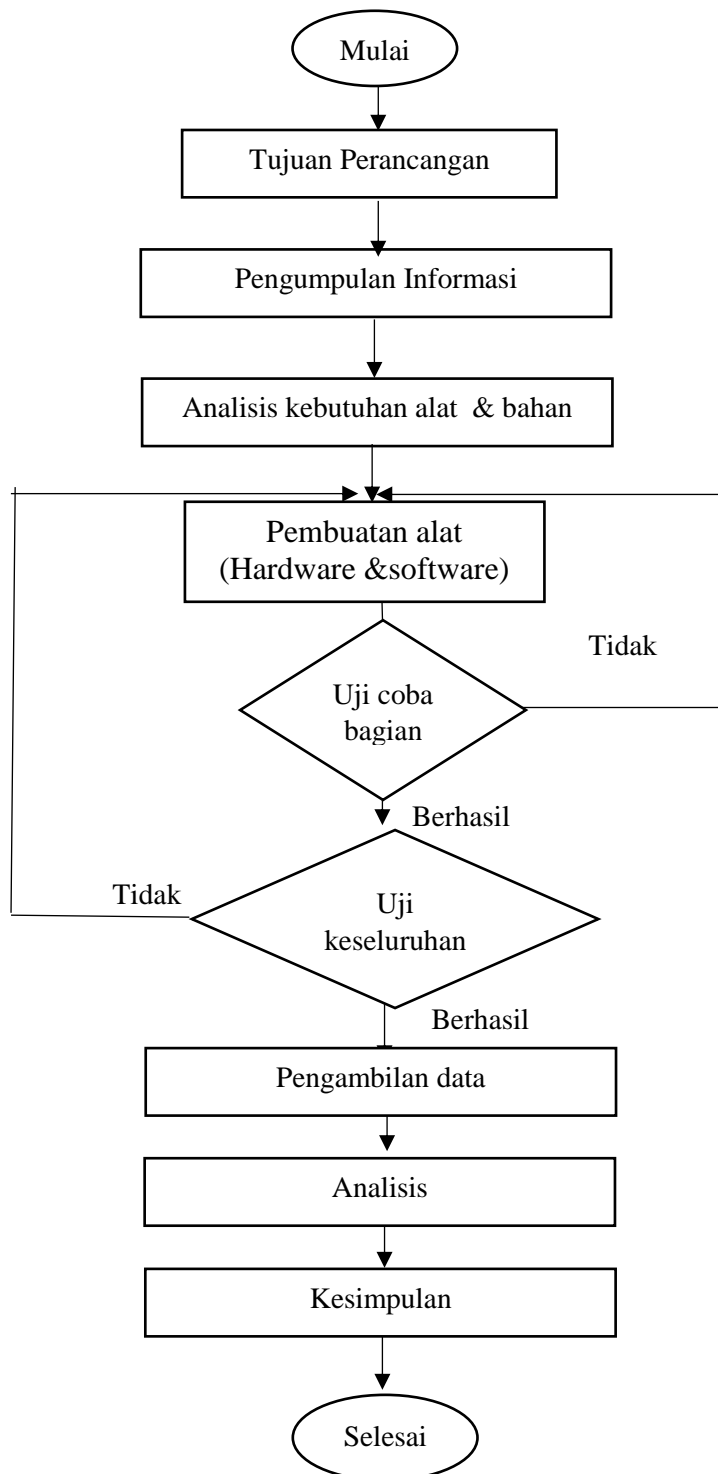
#### 3.2 Alat dan Bahan

Pelaksanaan pembuatan tugas akhir ini dibutuhkan alat dan bahan yang berupa

Tabel 3. 1 Alat dan Bahan

Alat	Bahan
Laptop asus	Sensor Autonics EP50S8-1024-2F-N-24
<i>Logic Analyzer</i>	Kapasitor
AFC	Resistor
<i>Osilloscope</i>	Arduino mega 2560
<i>Power supply</i>	Sensor HMC5883L
Mesin Laminating	RS-485
Multimeter	Db25f PCB
Solder	Pin sisir
Tang crimping	Kabel jumper
Obeng	USB serial
Tang Potong	LED
<i>Cutter</i>	PCB
Gunting	74HC4050
CNC	Kabel <i>male</i> dan <i>female</i>
-	Transistor
-	Db25m
-	Black Housing
-	LCD 16x2

### 3.3 Diagram Alir Penelitian



Gambar 3. 1 Diagram Alir Penelitian

### 3.3.1 Penjelasan Diagram Alir Penelitian

#### 1. Mulai

Tahap ini merupakan tahap awal yang dilakukan penulis untuk memulai tahap penulisan tugas akhir.

#### 2. Tujuan Perancangan

Pelaksanaan pembuatan tugas akhir ini harus memiliki tujuan dasar mengenai pembuatan sistem umpan balik kendali *close loop* pada antena penjejak dua axis.

#### 3. Pengumpulan Informasi

Tahap pengumpulan informasi merupakan tahap dimana informasi didapatkan dengan membaca buku, jurnal maupun artikel di internet. Tahap ini bertujuan untuk mengumpulkan informasi dan data yang berhubungan dengan judul penelitian tugas akhir.

#### 4. Analisis Kebutuhan Alat dan Bahan

Setelah mengumpulkan informasi melalui jurnal, buku ataupun internet, data - data yang didapatkan akan dianalisis sesuai kebutuhan. Pada tahap ini penulis dapat menentukan alat serta bahan yang akan digunakan untuk merancang alat. Selain itu jenis perangkat lunak yang dibutuhkan untuk menunjang penelitian ini juga ditentukan pada tahap ini. Analisis kebutuhan alat dan bahan akan disesuaikan dengan kondisi alat yang akan dibuat. Tahap ini merupakan tahap penentu sebelum proses pembuatan alat.

#### 5. Pembuatan Alat

Tahap pembuatan alat merupakan tahap setelah dilakukannya perancangan alat. Tahap perancangan terbagi atas dua bagian yaitu tahap perancangan perangkat lunak dan tahap perancangan perangkat keras. Tahap perancangan perangkat lunak meliputi program yang digunakan untuk mengakses sensor, simulasi, desain dan lain-lain. Tahap perancangan perangkat keras meliputi sistem pengkabelan, perancangan antar komponen dan lain-lain. Setelah tahap perancangan telah selesai, pembuatan alat dapat segera dilakukan.

#### 6. Uji Coba Bagian

Tahap uji coba bagian merupakan tahap pengujian yang dilakukan satu demi satu. Pada tahap pembuatan alat, terdapat perancangan perangkat lunak dan perangkat

keras. Perangkat lunak dan perangkat keras diuji coba secara masing-masing. Hal ini bertujuan agar setiap bagian dapat bekerja dengan baik sehingga sistem yang akan dibuat dapat sesuai dengan tujuan.

#### 7. Uji Keseluruhan

Tahap uji keseluruhan merupakan tahap pengujian suatu sistem. Tahap ini melakukan uji coba terhadap bagian perangkat lunak dan perangkat keras yang telah digabungkan menjadi satu. Hasil dari tahap uji keseluruhan adalah sensor dapat memberikan nilai umpan balik berupa azimuth dan elevasi sesuai dengan yang diinginkan. Nilai azimuth dan elevasi akan dibandingkan dengan nilai input. Diharapkan nilai *error correction* menjadi lebih kecil atau mendekati 0.

#### 8. Pengambilan Data

Tahap pengambilan data dilakukan setelah tahap uji keseluruhan sukses. Ketika tahap uji keseluruhan menyatakan sistem dikatakan baik, maka tahap pengambilan data akan dimulai. Tahap ini data yang akan diambil merupakan data nilai *feedback* dari perangkat yang telah dirancang. Banyaknya jumlah data yang akan diambil akan disesuaikan dengan kebutuhan tugas akhir.

#### 9. Analisis Data

Analisis data dapat dilakukan ketika tahap pengambilan data telah selesai. Pada tahap ini semua data akan dikumpulkan dan dianalisis. Data yang didapatkan merupakan data yang berasal dari hasil pengujian alat yang telah dibuat. Hasil pengujian merupakan keseluruhan sistem yang telah dibuat.

#### 10. Kesimpulan

Kesimpulan merupakan tahap pembuatan intisari analisis data yang diperoleh dari penelitian.

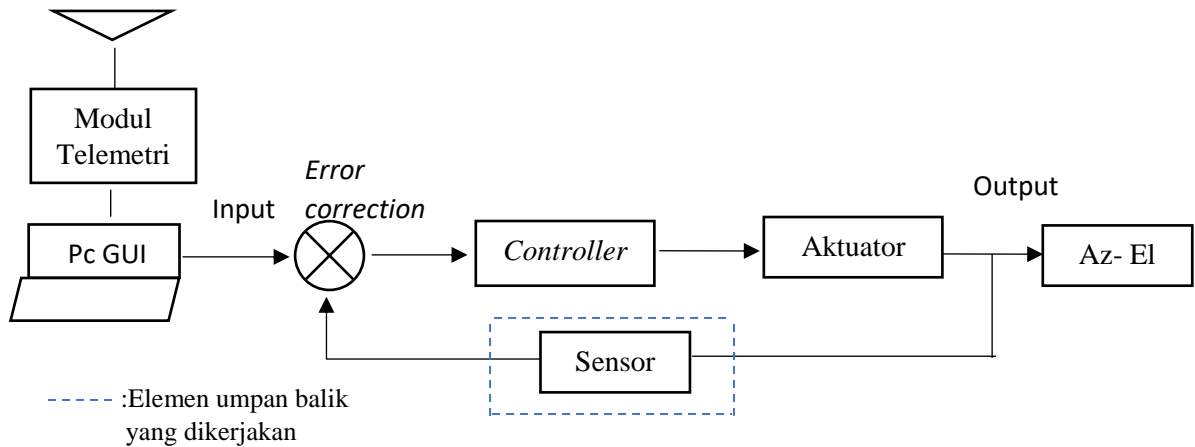
#### 11. Selesai

Pada tahap ini semua proses penelitian tugas akhir telah selesai dan siap digunakan.

### 3.4 Perancangan Alat Sistem Umpan Balik

#### 3.4.1 Deskripsi Sistem Perancangan

Sistem umpan balik kendali *close loop* antenna penjejak dua axis akan dirancang dalam bentuk blok diagram di bawah ini

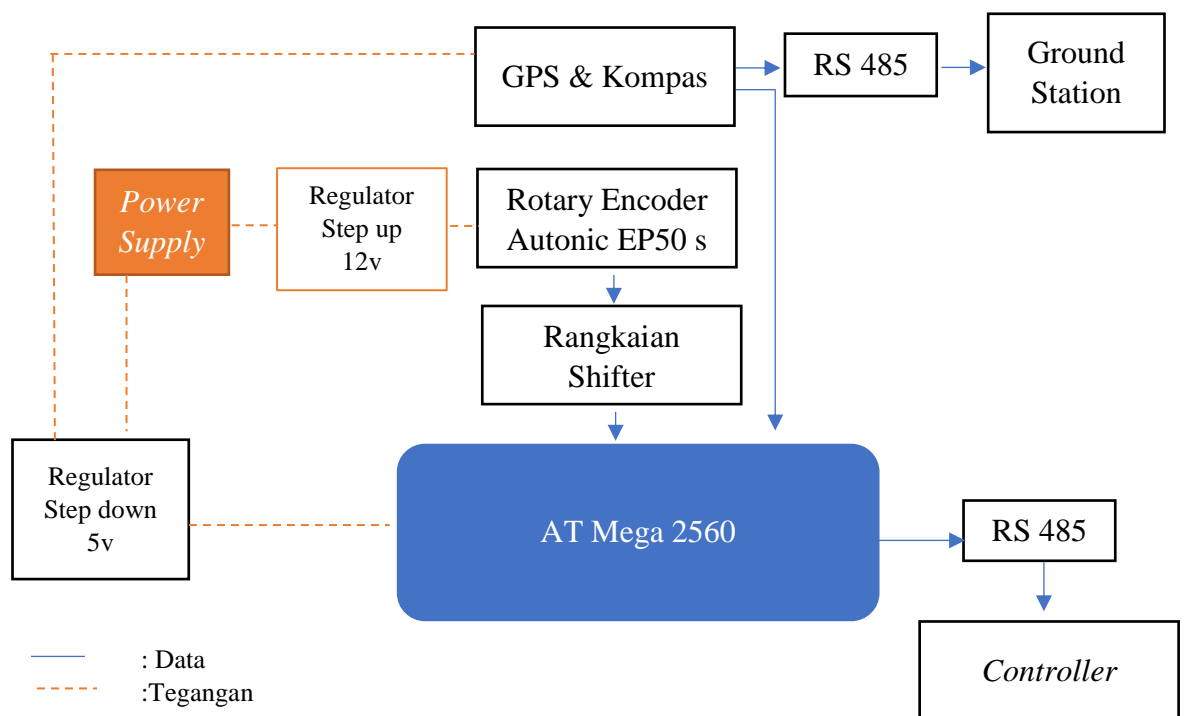


Gambar 3. 2 Diagram Blok Sistem Umpan balik Kendali Close Loop

Blok diagram di 3.2 merupakan sistem umpan balik kendali *close loop* antenna penjejak dua axis. Penggunaan sistem kendali *close loop* berguna agar nilai *output* yang dihasilkan sesuai dengan yang diharapkan. Sistem perancangan di atas berawal dari pengiriman data yang berasal dari muatan menggunakan radio telemetri. Data yang diterima pada ground station akan diproses oleh aplikasi GUI (Graphic User Interface). Pada mulanya data koordinat posisi yang diterima oleh *receiver* dari muatan masih berbentuk data *longitude*, *latitude*, dan *altitude*. Kemudian dilakukan akuisisi data posisi ketinggian, sikap vertikal dan horizontal pada antenna *ground station*. Data pada perhitungan GUI akan dijadikan sebagai nilai masukkan pada sistem kendali antenna penjejak. Kemudian, dilakukan kalkulasi nilai *set point* untuk mendapatkan keluaran nilai azimuth dan elevasi. Data keluaran yang berupa nilai azimuth dan elevasi akan masuk ke mikrokontroler untuk menggerakkan motor penggerak antenna. Pada proses ini terdapat elemen umpan balik pada sistem.

Elemen umpan balik pada diagram blok di atas adalah nilai sensor. Sensor yang digunakan berupa sensor sudut untuk menentukan posisi elevasi dan azimuth antenna penjejak. Nilai umpan balik akan mempengaruhi aksi pengontrolan. Nilai umpan balik dan *set poin* akan dibandingkan secara terus-menerus menggunakan komparator untuk mendapatkan hasil *output* yang diinginkan sehingga motor akan bergerak ke arah posisi aktual muatan. Sensor GPS pada antenna berfungsi untuk memberitahu posisi *ground control station*.

### 3.4.2 Rancangan Perangkat Keras



Gambar 3. 3 Diagram Blok Perancangan Hardware

Perancangan perangkat keras merupakan salah satu dari tahap perancangan sistem kendali umpan balik *close loop*. Gambar 3.3 merupakan diagram blok tentang perancangan perangkat keras sistem umpan balik kendali *close loop* yang akan dibuat. Pembuatan perangkat keras pada sistem diatas menggunakan dua buah sensor yaitu sensor *rotary encoder* dan sensor kompas. Sensor *rotary encoder* akan digunakan sebagai sensor penentu sudut elevasi sedangkan sensor kompas merupakan sensor penentu sudut azimuth.

Pada sensor kompas terdapat sensor lain yang berupa sensor GPS. Sensor GPS ini digunakan sebagai penentu titik *Ground Control station* berada. Data sensor GPS tidak diolah dengan mikrokontroler melainkan langsung dikirimkan secara serial ke *ground control station* menggunakan RS485. Data pada GPS akan diumpankan pada sebuah aplikasi untuk mendapatkan nilai latitude, longitude dan altitude.

Pada sensor *rotary encoder* sebelum data diolah oleh mikrokontroler terdapat rangkaian *shifter*. Rangkaian *shifter* berfungsi sebagai penerjemah antara tegangan yang lebih tinggi ke tegangan rendah. Besarnya tegangan *input* dan *output* yang dibutuhkan oleh sensor rotary dapat menyebabkan kerusakan pada mikrokontroler. Hal ini disebabkan oleh tegangan rekomendasi input mikrokontroler hanya 5V sehingga dibutuhkan rangkaian pengkondisi sinyal yang dapat menyelaraskan kedua komponen tersebut.

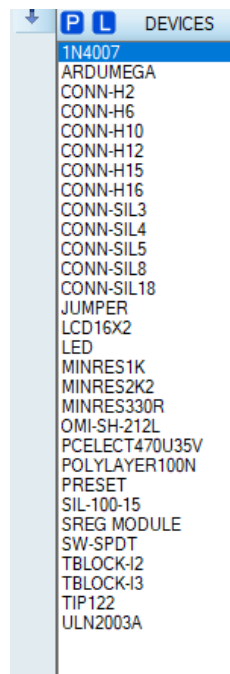
Selain itu terdapat regulator yang dipasang sebelum mikrokontroler. Hal itu bertujuan untuk menurunkan tegangan input dari 7.2 V menjadi 5V. Tegangan 5v merupakan tegangan operasi arduino mega sedangkan 7.2V merupakan tegangan dari baterai. Data yang telah berhasil dikonversi dalam derajat akan dikirimkan kepada kontroler. Data akan dikirimkan secara serial menggunakan komponen RS485.

Dalam pembuatan rancangan perangkat keras dibutuhkan beberapa *software* pendukung. *Software* pendukung ini dapat digunakan untuk membuat simulasi alat, skematik alat, desain, dan PCB. Pembuatan simulasi dilakukan dengan menggunakan *software* proteus 8 profesional. *Software* ini dapat membantu dan mempermudah dalam melakukan perancangan *hardware*. Namun, *software* ini memiliki kelemahan pada *library* komponen sehingga tidak semua komponen elektronika berada pada *software proteus*. Adapun langkah perancangan perangkat keras sistem kendali *close loop* adalah:



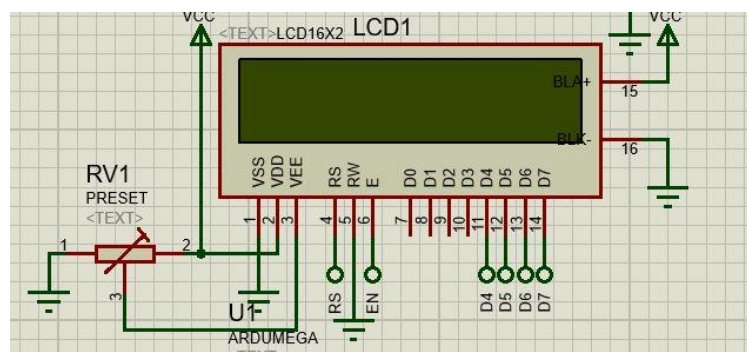


komponen elektronika seperti LCD 16x2, arduino mega 2560, pin konektor dan lain-lain. Penyusunan pembuatan skematik pada alat memiliki maksud untuk mempermudah saat melakukan perangkaian. Selain itu pembuatan skematik juga mempermudah dalam pembuatan PCB.



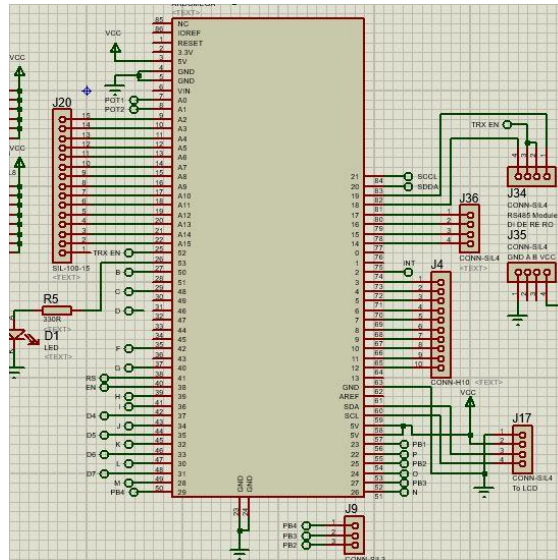
Gambar 3. 5 Daftar Komponen yang Digunakan pada Skematik

Gambar 3.5 merupakan daftar komponen yang diambil pada *component mode* melalui *Pick Mode*. *Pick Mode* akan menampilkan banyak jenis dan variasi dari komponen. Pemilihan komponen dapat disesuaikan dengan kebutuhan alat. Untuk mengetahui spesifikasi komponen dapat diketahui melalui *data sheet*.



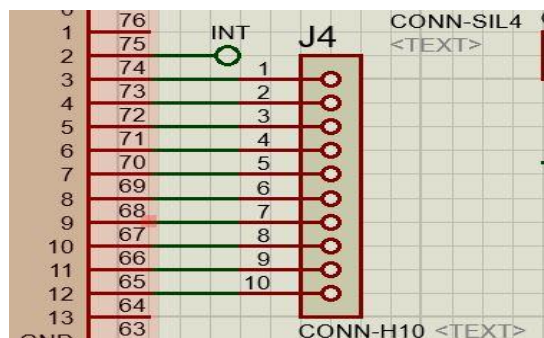
Gambar 3. 6 LCD 16x2

Gambar komponen 3.6 adalah gambar LCD (*Liquid Crystal Display*) 16 x 2 yang dipasang pada alat sistem umpan balik. LCD digunakan untuk menampilkan nilai azimuth dan elevasi yang telah diproses oleh mikrokontroler. Penggunaan resistor variabel pada layar berfungsi untuk mengatur kecerahan cahaya pada LCD.



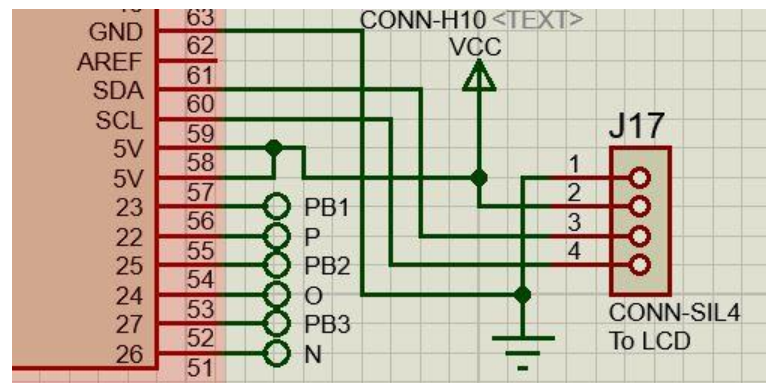
Gambar 3. 7 Arduino Mega 2560

Arduino mega merupakan salah satu jenis mikrokontroler IC AT Mega 2560. Arduino ini memiliki keunggulan yang terdapat pada pin input/output yang lebih banyak. Arduino mega memiliki 54 *input/output*. Pada penelitian ini, penggunaan arduino mega disebabkan karena banyaknya pin *input/output* yang dibutuhkan oleh komponen. Salah satu contohnya adalah sensor autonics EP50S8-1024-2F-N-24 yang digunakan sebagai sensor sudut yang membutuhkan 10 pin *input/output*.



Gambar 3. 8 Konektor Pin

Pada gambar 3.8 merupakan gambar konektor yang menggunakan pin 3 hingga pin 12. Pin konektor tersebut merupakan pin yang akan digunakan oleh sensor Autonics EP50S8-1024-2F-N-24 sebagai sensor sudut. Pin tersebut akan digunakan sebagai inputan yang akan disambungkan dari *output* CD4050.

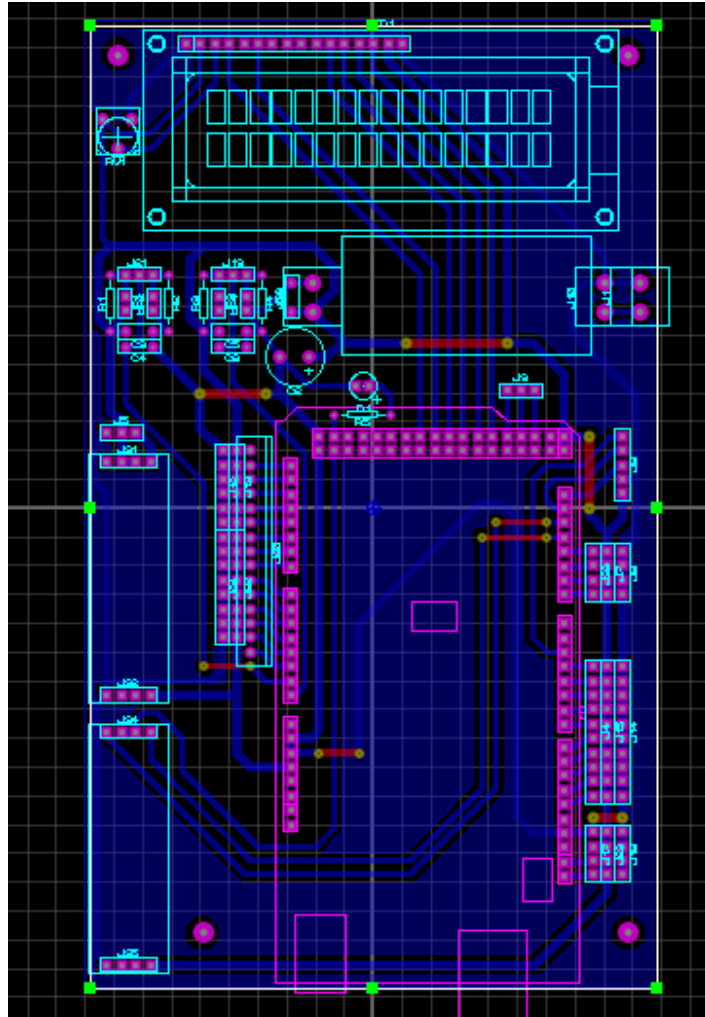


Gambar 3. 9 Pin SDA dan SCL

Pin SDA dan SCL pada gambar 3.9 akan digunakan oleh sensor HMC5883L sebagai sensor kompas digital. Sensor ini menggunakan protokol komunikasi data berupa I2C. Protokol komunikasi I2C merupakan komunikasi serial dua arah yang memiliki saluran khusus sebagai pengirim dan penerima. SCL (*Serial Clock Line*) bertugas untuk menghantarkan sinyal clock sedangkan SDA (*Serial Data*) berguna untuk mentransmisikan data. Piranti dari sistem I2C bus menggunakan *master* dan *slave*. *Master* merupakan piranti yang mengirim dan *slave* merupakan piranti yang dialamati.

Sistem pada I2C bus dimulai ketika *master* mengirimkan sinyal *start* dan diakhiri dengan sinyal *stop*. Sinyal *start* terjadi apabila terjadi perubahan tegangan pada SDA dari 1 menjadi 0 pada saat SCL bernilai satu. Sedangkan sinyal *stop* terjadi apabila nilai tegangan SDA berubah dari 0 menjadi 1. Selain itu terdapat sinyal lain yang bernama sinyal ACK (*acknowledge*) sinyal ini merupakan sinyal yang dikirimkan *slave* sebagai tanda keberhasilan dalam pengiriman data. Sinyal ini merubah nilai SDA dari satu menjadi 0. Dikirmkannya sinyal 0 oleh *slave* menandakan pengiriman data 8 bit telah selesai.

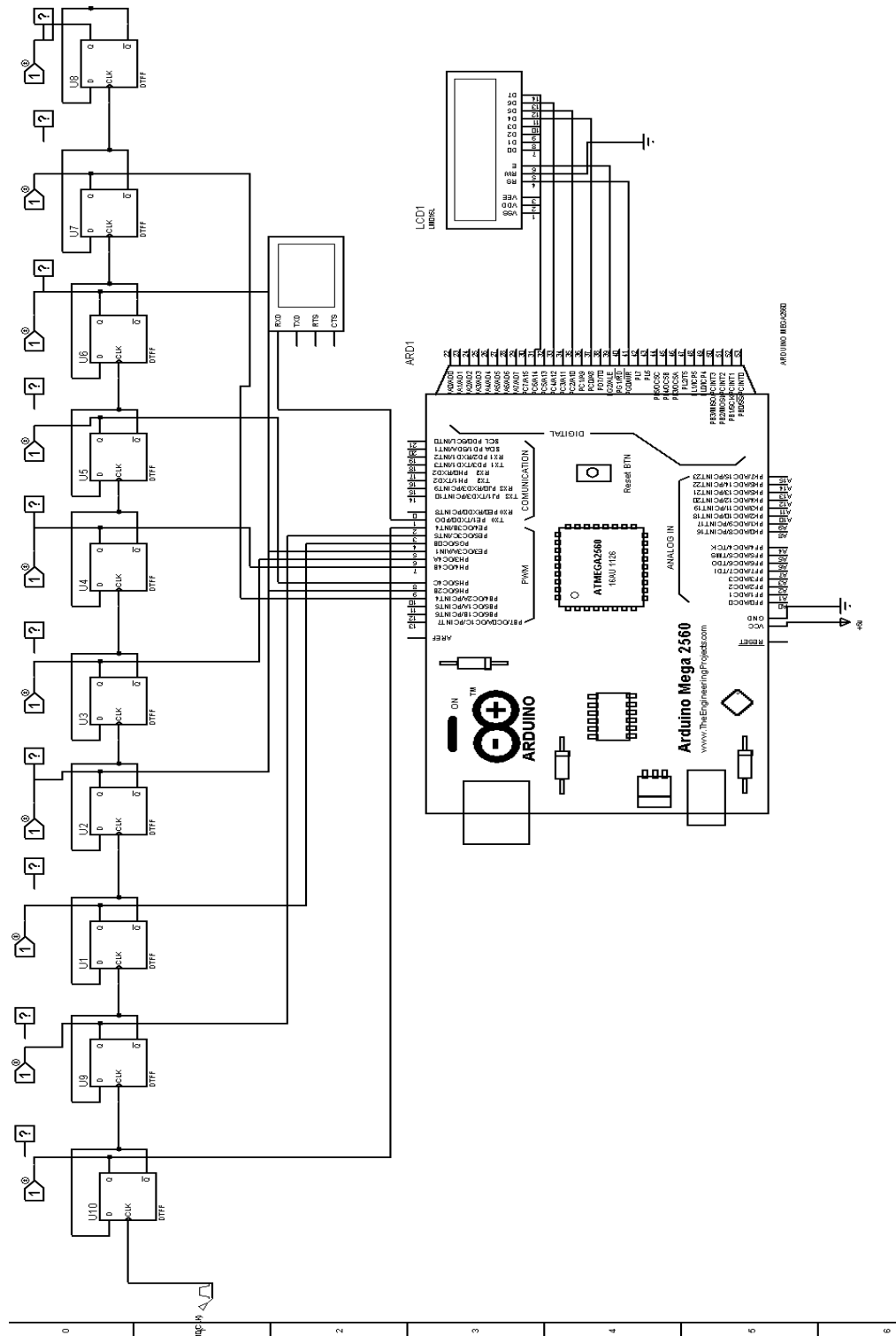
## 2. Pembuatan layout PCB



Gambar 3. 10 PCB Sistem Umpan Balik

Tahap ini merupakan tahap pembuatan PCB. Pembuatan PCB dapat dilakukan ketika pembuatan skematik telah selesai dengan benar. Dalam pembuatan PCB, hal yang paling penting adalah desain peletakan komponen. Jalur pada komponen tidak boleh saling bersilangan dan menyentuh dengan jalur lainnya. Diharapkan desain PCB lebih rapat dan meminimalisir penggunaan jumper. Pada PCB diatas menggunakan *layer selector Top Copper* dan *Bottom Copper*.

3. Pembuatan Program dan Simulasi Sensor Autonic EP50S8-1024-2F-N-24



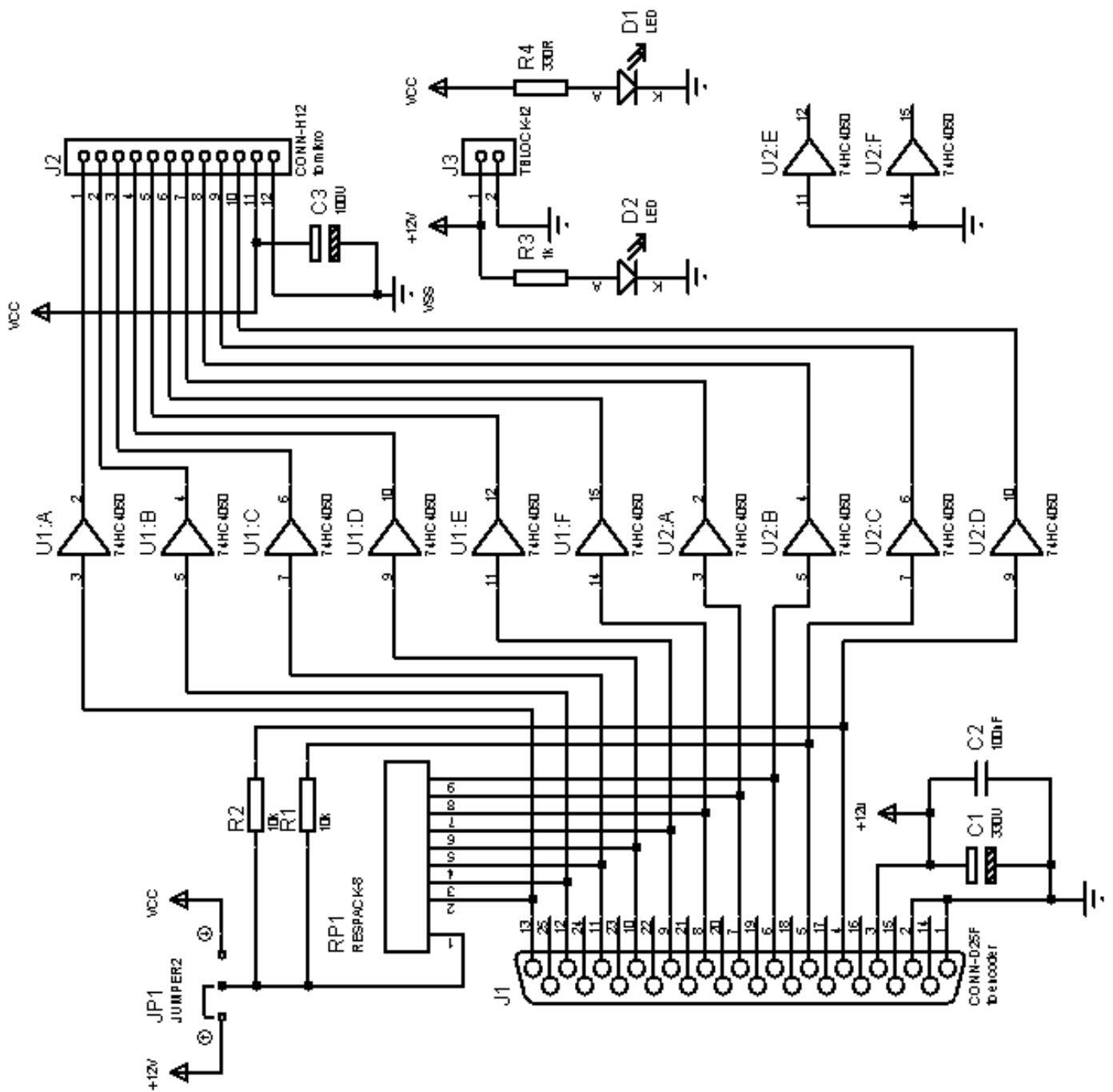
Gambar 3. 11 Simulasi Sensor Autonic menggunakan Logic Probe

Pembuatan simulasi program merupakan salah satu hal yang penting dalam tahap penelitian. Pembuatan simulasi berfungsi untuk mengetahui rencana perancangan suatu sistem dapat direalisasikan atau tidak. Pada tahap ini, dilakukan pembuatan simulasi untuk sensor. Sensor ini merupakan sensor yang akan digunakan untuk menunjukkan sudut elevasi antena. Sensor ini memiliki 10 *output* yang berupa nilai biner. Simulasi ini bertujuan untuk mengetahui keberhasilan dalam konversi *output* biner menjadi nilai desimal.

Pada gambar 3.11, pembuatan simulasi menggunakan komponen *logic probe*, terminal dan arduino. Penggunaan 10 *logic probe* disebabkan oleh keluaran komponen tersebut berupa nilai biner sehingga *logic probe* diumpamakan sebagai sensor *rotary encoder* yang memiliki 10 *output* biner.

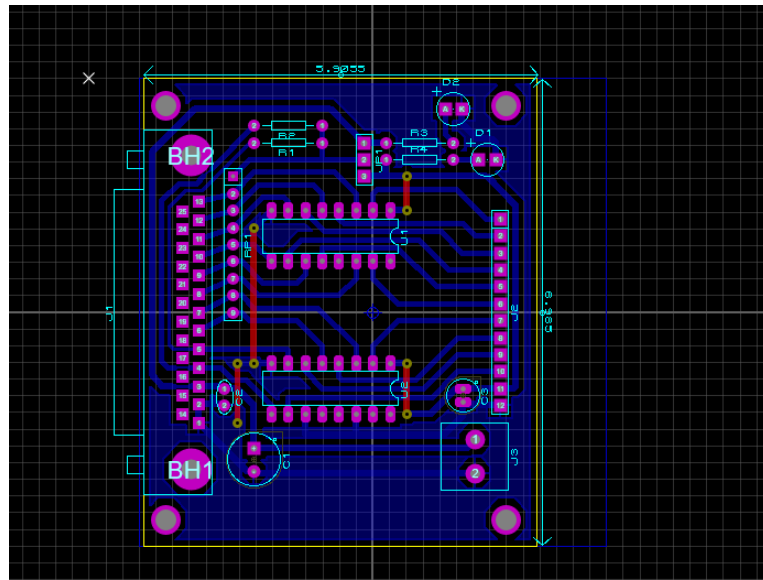
#### 4. Pembuatan simulasi dan layout PCB Rangkaian *level shifter* sensor *rotary encoder*.

sensor autonics EP50S8-1024-2F-N-24 merupakan sensor yang digunakan untuk menentukan sudut elevasi antena penjejak. Sensor ini memiliki tegangan *input* sebesar 12-24v. Tegangan yang cukup besar tersebut akan dihubungkan dengan mikrokontroler arduino mega yang memiliki tegangan operasi sebesar 5V. Pada kondisi ini dibutuhkan rangkaian *level shifter* yang dapat mengkonversi tegangan tanpa merubah gelombang tegangan. Penggunaan *logic probe* diumpamakan sebagai *output* biner pada sensor. Hal ini dikarenakan keterbatasan *library* pada proteus. Selain itu terdapat mikrokontroler aduino mega 2560 yang berguna untuk memproses perangkat lunak atau program sensor. Terminal dan LCD berguna untuk menampilkan nilai hasil konversi dari biner ke desimal.



Gambar 3. 12 Simulasi Rangkaian Level Shifter

Gambar 3.12 menunjukkan pembuatan simulasi rangkaian level *shifter*. Rangkaian ini menggunakan IC CD4050 yang memiliki 6 pin *input* dan *output*, 1 *vdd*, 1 *vss* dan 2 *Nc*. Ditambahkan juga kapasitor sebagai penyetabil gelombang tegangan, resistor sebagai pengamanan dan komponen lain sesuai fungsinya.



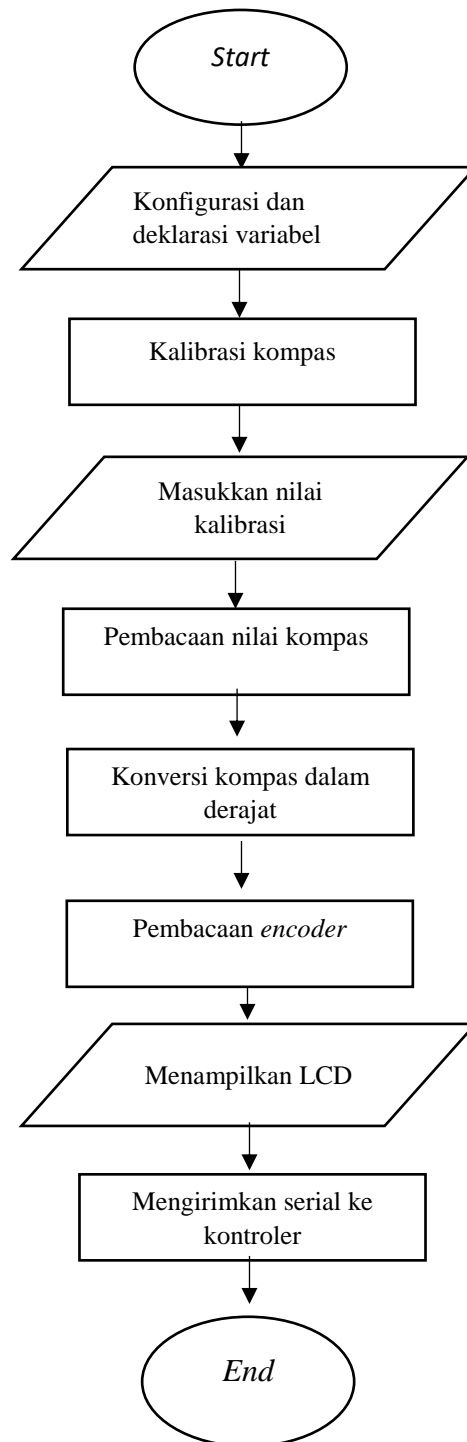
Gambar 3. 13 PCB Rangkaian Shifter sensor Autonics

Setelah pembuatan skematik simulasi berjalan dengan baik, langkah selanjutnya adalah pembuatan *layout* PCB. *Layout* PCB diatas pada gambar 3.13 dibuat dengan ukuran 5.9x6.9 cm. Pembuatan *layout* PCB komponen harus disusun dengan rapi dan rapat. Hindari desain jalur antar komponen yang saling bersilangan.

### 3.4.3 Perancangan perangkat lunak

Pembuatan alat sistem umpan balik kendali *close loop* membutuhkan sistem perangkat lunak. Sistem perangkat lunak yang dibutuhkan adalah *software arduino IDE* yang digunakan untuk membuat program. Program dibuat agar dapat mengakses sensor yang akan digunakan. Adapun diagram alir program adalah





Gambar 3. 14 Diagram alir perangkat lunak

Diagram alir pada gambar 3.14 merupakan diagram alir pembuatan program sensor pendeteksi posisi elevasi dan azimuth. Tahap utama dalam pembuatan program adalah dengan memasukkan library dan mendeklarasikan variabel. Pada tahap utama dilakukan juga *setting* konfigurasi pada program. Selanjutnya kompas akan melakukan kalibrasi. Nilai kalibrasi tersebut akan dimasukkan kedalam program agar sensor dapat bekerja dengan baik. Sensor kompas yang telah dikalibrasi, akan membaca arah medan magnet dengan satuan mG. Nilai tersebut dikonversikan menjadi nilai sudut dalam satuan derajat.

Setelah pembacaan nilai kompas, nilai *sensor rotary encoder* dibaca. Sensor memiliki nilai output berupa nilai biner. Nilai biner tersebut akan dikonversikan menjadi nilai sudut dalam satuan derajat. Kedua nilai sensor yang telah dikonversi akan ditampilkan pada LCD dan serial monitor. Data kedua sensor akan dikirimkan ke kontroler secara serial. Program ini akan bekerja secara berulang.

- Program sensor kompas digital

Sensor kompas digital merupakan sensor yang digunakan untuk menentukan nilai azimuth pada *kontroler* antena. Sensor kompas digital yang digunakan adalah HMC 5883L . Sensor ini menggunakan I<sup>2</sup>C sebagai protokol komunikasi data. Sensor ini akan menampilkan tiga Axis X,Y, dan Z.

Program yang digunakan pada sensor digital didapatkan dari sumber internet

<https://github.com/helscream/HMC5883L-Header-Arduino-Auto-calibration/tree/master/Core/Compass-header-example-ver-0.2> . Program ini hanya bisa digunakan pada sensor kompas HMC5883L. Program ini dipilih untuk digunakan karena terdapat program kalibrasi sensor yang dapat meminimalisir terjadinya *error* pada kompas.

Tabel 3. 2 Compass Header

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

Header ini hanya dapat digunakan oleh sensor HMC5883L magnetometer dengan alamat I<sup>2</sup>C 0x1E. `compass.h`, membutuhkan `#include <Wire.h>` sebelum `#include "compass.h"`. Selain itu, tugas lainnya adalah untuk menginisialisasi *port* I<sup>2</sup>C. Selain itu terdapat beberapa variabel global pada program diantaranya adalah *bearing*, *compass x,y,dan z scaled*, *offset* dan *gain error*.

Tabel 3. 3 Variabel Global

float bearing	# Sudut magnetic north dalam derajat
float compass_x_scaled	#Medan magnet sumbu x
float compass_y_scaled	# Medan magnet sumbu y
float compass_z_scaled	#Medan magnet sumbu z
float compass_x_offset	#pengukuran nilai offset sumbu X yang digunakan untuk kalibrasi
float compass_y_offset	#pengukuran nilai offset sumbu Y yang digunakan untuk kalibrasi
float compass_z_offset	#pengukuran nilai offset sumbu Y yang digunakan untuk kalibrasi
floatcompass_x_gainError	#kesalahan <i>gain</i> medan magnet disumbu X yang digunakan dalam kalibrasi
float compass_y_gainError	# kesalahan <i>gain</i> medan magnet disumbu Y yang digunakan dalam kalibrasi
float compass_z_gainError	# kesalahan <i>gain</i> medan magnet disumbu Z yang digunakan dalam kalibrasi

Pada tabel 3.3 terdapat beberapa variabel global yang digunakan. Variabel tersebut akan digunakan pada perintah fungsi pada program. `compass_offset_calibration(3)` merupakan fungsi yang terdapat pada program. Fungsi ini bertugas untuk melakukan kalibrasi pada program. Angka 3 dalam

fungsi *compass\_offset\_calibration(3)* mengartikan nilai *int select*. *Int select* akan bekerja sebagai mana mestinya . Adapun penjelasan *int select* adalah

Tabel 3. 4 int select pada *compass\_offset\_calibration*

1	Pengkalibrasian berfungsi untuk mendapatkan nilai error pada gain. Kesalahan pengukuran gain akan didapatkan pada variabel <i>compass_x_gainError</i> , <i>compass_y_gainError</i> dan <i>compass_z_gainError</i> diperbarui.
2	Magnetometer dikalibrasi untuk mendapatkan nilai <i>error offset</i> dalam pengukuran. Nilai <i>error offset variabel</i> yang didapatkan adalah variabel <i>compass_x_offset</i> , <i>compass_y_offset</i> dan <i>compass_z_offset</i> diperbarui.
3	Pengkalibrasian berfungsi untuk mendapatkan nilai gain dan offset dalam pengukuran.
Lainnya	Tidak dapat terkalibrasi

Tabel 3.4 menjelaskan tentang fungsi dari *int\_select* yang ada. Program tersebut akan bekerja secara berkelanjutan sehingga untuk melakukan kalibrasi dengan benar nilai *int\_select* yang dimasukkan adalah 3. Apabila bila nilai yang dimasukkan tidak terdapat pada tabel diatas maka nilai pengkalibrasian yang diinginkan tidak akan bisa di dapatkan.

```
compass_x_offset = 38.86;
compass_y_offset =141.66;
compass_z_offset =116.84;
compass_x_gainError = 1.03;
compass_y_gainError = 1.05;
compass_z_gainError = 0.98;
```

Gambar 3. 15 Memasukan nilai offset

Nilai hasil kalibrasi yang didapatkan berupa nilai *gain* dan *offset* x,y, dan z akan dimasukkan kedalam program seperti gambar 3.15. Nilai tersebut akan digunakan sebagai nilai kalibrasi kompas sehingga dapat meminimalisir derajat sudut kompas *error*. Ketika ingin menjalankan sensor kompas, masukkan nilai *int\_select* selain 1, 2, dan 3. Apabila nilai tersebut tidak diubah maka nilai kompas akan terkalibrasi ulang.

Fungsi lainnya yang terdapat pada program adalah *compass\_init(int gain)*. Nilai *int\_gain* berfungsi untuk mengatur penguatan medan magnet pada kompas sehingga nilai ini akan mempengaruhi pembacaan nilai sumbu X,Y, dan Z. Ketika ingin mengatur nilai *gain* pada kompas, masukan salah satu nilai *gain* 0 hingga 7. Nilai *int\_gain* 0 hingga 7 berada pada tabel dibawah ini

Tabel 3. 5 Nilai *input* 0-7 *int\_gain*

<b><i>Int_gain</i></b>	<b>Nilai <i>Gain</i></b>
0	0.73 gauss
1	0.92 gauss
2	1.22 gauss
3	1.52 gauss
4	2.27 gauss
5	2.56 gauss
6	3.03 gauss
7	4.35 gauss

Pada tabel 3.5 dijelaskan bahwa nilai *int\_gain* 0-7 memiliki nilai *gain* yang berbeda-beda. Semakin tinggi nilai *int\_gain* yang dipilih maka nilai gain dalam satuan gauss semakin besar. X,Y dan Z.

Setelah pengaturan gain dan kalibrasi, maka program X,Y,Z *scalled reading* akan dilakukan. Pembacaan nilai dapat dilakukan dengan memanggil fungsi *compass\_scalled\_reading()*. Kompas akan membaca nilai X,Y dan Z dengan nilai pengukuran terakhir sesuai dengan letak kompas. Lalu, terdapat fungsi *compass\_heading()* akan selalu memperbarui nilai X,Y,Z *scalled* dan *bearing*. Nilai *bearing* merupakan nilai kompas yang sudah dikonversi menjadi *Degree*.

- Pembuatan program sensor autonics EP50S8-1024-2F-N-24

Sensor Autonics EP50S8-1024-2F-N-24 merupakan sensor *rotary encoder* yang digunakan untuk menentukan posisi. Penelitian ini menggunakan sensor *rotary encoder* untuk menentukan sudut elevasi. pemrograman untuk sensor ini menggunakan software arduino IDE. Keluaran dari sensor berupa nilai biner sebanyak 10 bit. Hasil keluaran sensor yang berupa nilai biner akan dikonversi menjadi desimal. Nilai desimal tersebut akan diolah menjadi nilai derajat dengan memanfaatkan nilai putaran dan resolusi pada sensor. Ketelitian pada sensor *rotary encoder* memanfaatkan nilai resolusi pada sensor tersebut. Semakin kecil nilai resolusi atau mendekati nol maka semakin baik pula ketelitian yang didapatkan.

```
#include <LiquidCrystal.h> //library
LiquidCrystal lcd(41,39,37,35,33,31); //pin dari penampil lcd rs e d4-d7

boolean pin_state[8]; //tipe data 1 &0 dengan pin state masukan 8 jalur
byte input_pin[]={5,6,7,8,9,10,11,12}; //tipe data bit dengan pinnya
int dec_position= 0; //type data bilangan bulat untuk nilai dec_pos=0
float z= 1.41; //resolusi nilai sensor
int elevasi; // nilai elevasi dalam bilangan bulat
void setup() // sebagai inisialisasi
{
  Serial.begin(9600); // serial dengan baud rate 9600
  lcd.begin(16,2); // lcd digunakan 16*2
  for(byte i=0;i<8;i=i+1) //jenis data byte yang digunakan dengan nilai i=0 dan kurang dari 8(0-7)
    pinMode (input_pin[i],INPUT); // dijadiakan sebagai input
}
```

Gambar 3. 16 Bagian Pertama Program Sensor *Rotary Encoder*

Berdasarkan kode pemrograman pada gambar 3.16 pembuatan kode program sensor *rotary encoder* menggunakan *library Liquid Crystal* yang berupa library untuk menampilkan data pada LCD 16x2. Pada inialisai tipe data diatas terdapat tiga jenis tipe data yang berbeda. Tipe data yang pertama adalah *boolean* yang bernilai 1 dan 0. Tipe data ini adalah tipe data yang digunakan karena nilai output pada sensor yang bernilai 1 dan 0. Tipe data yang kedua adalah *int* yang merupakan bilangan bulat atau *integer*. Tipe data *int* digunakan untuk nilai *dec pos* yang merupakan konversi bilangan bulat dari *output* biner. Tipe data yang ketiga adalah *float*. Tipe data ini merupakan tipe data yang digunakan untuk elevasi dan inialisasi nilai "z." Tipe data ini digunakan untuk menampilkan bilangan desimal pada nilai konversi ke elevasi. Selain itu program akan bekerja dengan *array byte* yang berjumlah 8 byte.

Pada perintah *void setup* berfungsi untuk menginisialisasi variabel program yang akan digunakan. Variabel yang akan digunakan salah satunya adalah memunculkan nilai serial dengan aliran kecepatan data sebesar 9600 bps. Setelah itu terdapat inialisasi LCD dengan jenis 16x2 yang akan digunakan.

Pada baris bawah inialisasi variabel LCD, terdapat inialisasi *type data byte* dengan nilai *i=0* dan kurang dari 8. Jenis data *byte* ini akan terus berulang mulai dari 0 hingga 7 dengan nilai pin input berasal dari *input\_pin[i]*.

```
void loop () //program eksekusi
{
  baca_enc(); // program dijelaskan di bawah void baca enc
  lcd.setCursor (0,0); // lcd disetting dengan posisi 0,0
  lcd.print ("pos:"); // menampilkan tulisan pos:
  lcd.print(dec_position); //nilai yang ditampilkan diambil dari hasil dec position
  lcd.print ("");

  lcd.setCursor (1,1); // lcd disetting dengan posisi 0,0
  lcd.print ("el:"); // menampilkan tulisan pos:
  lcd.print(elevasi); //nilai yang ditampilkan diambil dari hasil dec position
  lcd.print ("");
}
```

<sup>1</sup> Gambar 3. 17 Bagian kedua Program Sensor *Rotary Encoder*

Pada bagian kedua pemrograman *rotary encoder* seperti gambar 3.17 merupakan program yang harus dieksekusi. Program diatas memerintahkan untuk membaca nilai sensor *rotary encoder* dan menampilkan nilai pos dan nilai elevasi. Nilai pos didapat dari nilai pembacaan *output* sensor *rotary encoder*

sedangkan nilai elevasi didapatkan dari perhitungan nilai `dec_pos` dan resolusi sensor ( $z$ ).

Tabel 3. 6 Rumus Perhitungan Resolusi Sensor & Elevasi

Perhitungan Nilai	Perhitungan Sudut
Resolusi Sensor	Elevasi
$Z = 255 / 360$	Elevasi = <code>dec_pos*z</code>

Tabel 3.6 menjelaskan tentang rumus yang digunakan untuk mendapatkan nilai sudut elevasi. Rumus resolusi sensor didapatkan dari nilai *output* pada sensor *rotary encoder*. Nilai *output* pada sensor merupakan nilai biner dengan 8 bit *output* sehingga resolusi sensor adalah  $2^8$  adalah 255. Nilai resolusi sensor akan dibagi dengan satu putaran penuh sudut elevasi sebesar 360 sehingga dihasilkan nilai  $z$  adalah 1.41.

Perhitungan sudut elevasi pada program didapatkan dengan perhitungan sesuai tabel 3.2. perhitungan sudut elevasi dilakukan dengan menghitung nilai `dec_pos` dan  $z$ . Nilai `dec_pos` didapatkan dari *output* biner sensor *rotary* yang diterjemahkan kedalam nilai *integer*.

```
void baca_enc()
{
  for(byte i=0;i<8; i++) pin_state [i] =digitalRead (input_pin[i]); //jenis data byte dimulai dari i=0 hingga kurang dari 1
  dec_position=0; //awalmula posisi dec adalah 0
  for (int i=0;i<8;i++)
  {
    dec_position = dec_position | (pin_state[i]<<i);
    Serial.print("i=");
    Serial.print(i);
    Serial.print("dec pos value");
    Serial.println(dec_position);
    lcd.setCursor(i+5,1);
    lcd.print(pin_state[i]);
    delay(100);
    elevasi= z*(dec_position);
  }
}
```

Gambar 3. 18 Bagian Ketiga Program Sensor *Rotary Encoder*

Gambar 3.18 menjelaskan tentang rumus pembacaan *rotary encoder*. Pembacaan *sensor rotary encoder* membutuhkan nilai `dec_pos` dan `pin_state` yang didapat dari pembacaan nilai `input_pin[i]`.



Rumus pembacaan encoder ini, memanfaatkan perulangan *for* dengan tipe data *byte*. Tipe data *byte* ini akan dimulai dari  $i=0$  hingga  $i<8$ . Data akan berulang sebanyak 8 kali dengan data pembacaan berasal dari  $pin\_input[i]$ . Data pada  $dec\_pos$  juga memanfaatkan perulangan *for* sebanyak 8 kali dengan tipe data *int* yang merupakan bilangan bulat.

Dalam pembacaan nilai sensor, rumus yang digunakan adalah  $dec\_position = dec\_position |(pin\_state[i] \ll i)$ . Rumus ini menggeser nilai *pin state* sebanyak nilai *input* dan dilakukan perhitungan *OR* menggunakan nilai  $dec\_pos$  yang mulanya adalah 0. Perhatikan contoh perhitungan nilai *output* sensor yang dikonversi menjadi bilangan bulat .

Nilai *output* sensor (*pinstate*) :000 1010 =10

$dec\_pos$  awal =0

Rumus pada program:  $dec\_position = dec\_position |(pin\_state[i] \ll i)$ ;

$I=0$

$dec\_pos = dec\_pos\ awal |(pin\_state[i] \ll i)$

$= 0 | 0 \ll 0$

$= 0$

Hasil Biner 0000 0000

$I=1$

$dec\_pos = \text{hasil } dec\_pos \text{ sebelumnya } |(pin\_state[i] \ll i)$

$= 0 | 1 \ll 1$

$= 2$

Perhitungan biner

$dec\_pos = dec\_pos \text{ sebelumnya } |(pin\_state[i] \ll i)$

$= 0000\ 0000 | 0000\ 0001 \ll 1$   $\longrightarrow$

$= 0000\ 0010$   $\longrightarrow$

0001 digeser 1  
kali menjadi  
0010

$I=2$

$dec\_pos = \text{hasil } dec\_pos \text{ sebelumnya } |(pin\_state[i] \ll i)$

$= 2 | 0 \ll 2$

$= 2$

0010 di OR  
dengan  $dec\_pos$   
sebelumnya  
0000

Perhitungan biner

$\text{dec\_pos} = \text{dec\_pos sebelumnya} | (\text{pin\_state}[i] \ll i)$   
 $= 0000\ 0010 | 0000\ 0000 \ll 2$  → 0000 digeser 2 kali menjadi  
 $= 0000\ 0010$  → 0000 di OR dengan dec pos sebelumnya 0010

I=3

$\text{dec\_pos} = \text{hasil dec\_pos sebelumnya} | (\text{pin\_state}[i] \ll i)$   
 $= 2 | 1 \ll 3$   
 $= 10$

Perhitungan biner

$\text{dec\_pos} = \text{dec\_pos sebelumnya} | (\text{pin\_state}[i] \ll i)$   
 $= 0000\ 0010 | 0000\ 0001 \ll 2$  → 0000 digeser 3 kali menjadi  
 $= 0000\ 1010$  → 0010 di OR dengan dec pos sebelumnya 1000

Pada contoh diatas, dengan menggunakan rumus tersebut nilai perhitungan biner menunjukkan hasil yang sama dengan nilai keluaran sensor.

Tabel 3. 7 Program Lengkap Sensor Autonics EP50S8-1024-2F-N-24

```

#include <LiquidCrystal.h> //library
LiquidCrystal lcd(41,39,37,35,33,31); //pin dari penampil lcd rs e
                                     d4-d7
boolean pin_state[8]; //tipe data 1 &0 dengan pin state masukan 8
                                     jalur
byte input_pin[]={5,6,7,8,9,10,11,12}; //tipe data bit dengan
pinnya
int dec_position= 0; //tipe data bilangan bulat untuk nilai
dec_pos=0
float z= 1.4; //resolusi nilai sensor
int elevasi; // nilai elevasi dalam bilangan bulat

void setup() // sebagai inisialisasi

```

```

{
  Serial.begin(9600); // serial dengan baud rate 9600
  lcd.begin(16,2); // lcd digunakan 16*2
  for(byte i=0;i<8;i=i+1) //jenis data byte yang digunakan dengan
  nilai i=0 dan kurang dari 8(0-7)
  pinMode (input_pin[i],INPUT); // dijadikan sebagai input
}

void loop () //program eksekusi
{
  baca_enc(); // program dijelaskan di bawah void baca enc
  lcd.setCursor (0,0); // lcd disetting dengan posisi 0,0
  lcd.print ("pos:"); // menampilkan tulisan pos:
  lcd.print(dec_position); //nilai yang ditampilkan diambil dari
                          hasil dec position

  lcd.print (" ");
  lcd.setCursor (8,0); // lcd disetting dengan posisi 8,0
  lcd.print ("el:"); // menampilkan tulisan pos
  lcd.print(elevasi); //nilai yang ditampilkan diambil dari hasil dec
                      position
  lcd.print ((char)223); //menampilkan derajat
  lcd.print (" ");
}
void baca_enc()
{
  for(byte i=0;i<8; i++)
  pin_state [i] =digitalRead (input_pin[i]); //jenis data byte dimulai
                                              dari i=0 hingga kurang
                                              dari 10 (0-9)
                                              menunjukkan pin
                                              inputan. pembacaan
                                              nilai biner akan
                                              diambil dari inputan pin
                                              i

  dec_position=0; //awalmula posisi dec adalah 0
  for (int i=0;i<8;i++) //perulangan
  {
    dec_position = dec_position |(pin_state[i]<<i); //rumus konversi
    lcd.setCursor(i+5,1); //posisi pada lcd
    lcd.print(pin_state[i]); //print nilai biner pada lcd
    delay(100);
    elevasi= z*(dec_position); //mendapatkan nilai elevasi
  }
}

```

Tabel 3.7 diatas merupakan penjelasan program secara lengkap yang digunakan oleh sensor autonics EP50S8-1024-2F-N-24 untuk menentukan sudut elevasi.

### 3.5 Tahapan Perilaku Pengujian

Tahapan pengujian merupakan tahap melakukan percobaan terhadap komponen-komponen yang digunakan dalam merancang alat.

#### 3.5.1 Perilaku Pengujian LCD dan Mikrokontroler

AT mega 2560 adalah jenis mikrokontroler yang digunakan dalam pembuatan alat penelitian ini. Pengujian yang dilakukan pada mikrokontroler ini dilakukan dengan menghubungkan arduino terhadap komputer menggunakan kabel USB. Setelah dihubungkan, cek *device manager* pada komputer dan perhatikan *port* USB yang terdeteksi. Apabila port USB arduino terdeteksi, cobalah untuk mengupload program yang akan ditampilkan pada LCD.

Pengujian LCD berfungsi untuk mengetahui kelayakan LCD dalam menampilkan teks. LCD yang digunakan dalam penelitian ini adalah LCD berukuran 16x2 dengan posisi  $x=0$  hingga 16 dan  $y=0$  dan 1. Pengujian LCD ini dilakukan dengan memasukan program yang telah ditulis pada arduino IDE ke dalam mikrokontroler. Pengujian kondisi pada LCD dapat dilakukan dengan mengikuti langkah dibawah ini.

1. Hubungkan pin *out* LCD dengan pin *input* mikrokontroler.
2. Hubungkan komputer dan mikrokontroler menggunakan kabel USB.
3. Buatlah program pada arduino IDE dengan menggunakan Library `#LiquidCrystal` dan tampilkan pin yang terhubung dengan arduino seperti `LiquidCrystal lcd (41,39,37,35,33,31);`
4. Gunakan perintah `lcd.begin(16,2);` didalam fungsi void `setup()`
5. Gunakan perintah `lcd.setCursor` untuk menentukan letak teks yang diinginkan seperti `lcd.setCursor (0,0);`.

6. Gunakan perintah `lcd.print` untuk memunculkan teks pada LCD seperti  
`lcd.print ("pos:");`.

7 *Upload* program pada arduino.

Setelah dilakukan tahapan pengujian di atas, amatilah tampilan pada LCD. Apabila terdapat teks dengan tulisan “pos”, maka dapat dikatakan LCD dapat bekerja dengan baik begitu sebaliknya.

### 3.5.2 Perilaku Pengujian Regulator Step Down

Regulator merupakan komponen yang digunakan untuk menurunkan tegangan sesuai dengan tegangan operasi arduino. Baterai yang digunakan pada penelitian ini merupakan baterai bertipe 2 sel dengan tegangan 7,2V sedangkan kebutuhan tegangan operasi arduino adalah 5V.

Pengujian pada regulator dilakukan dengan mengukur tegangan *input* dan *ouput* pada komponen regulator. Pengukuran dilakukan dengan menggunakan dua multimeter. Putarlah skala penunjuk multimeter pada skala tegangan dc dan letakkan pada bagian *input* dan *output*. Amatilah tegangan pada sisi masukan dan keluaran . Apabila nilai *output* belum sesuai dengan yang diinginkan, putarlah pengatur tegangan yang berupa baut pada sisi tengah komponen.

### 3.5.3 Perilaku Pengujian *DC-DC Converter step up* XL6009

Komponen *DC-DC converter step up* berfungsi untuk menaikkan tegangan agar dapat menyuplai tegangan yang lebih besar. Baterai yang digunakan pada penelitian ini merupakan baterai bertipe 2 sel dengan tegangan 7,2V sedangkan kebutuhan tegangan pada sensor *rotary encoder* adalah minimal 12V.

Pengujian pada *DC-DC converter* sangat diperlukan. Apabila tegangan yang disuplay pada sensor berlebihan atau kekurangan, hal ini dapat mempengaruhi kinerja dari sensor. Pengujian dilakukan dengan mengukur tegangan *input* dan *ouput* pada komponen *DC-DC Converter*. Pengukuran dilakukan dengan menggunakan dua multimeter. Putarlah skala penunjuk multimeter pada skala tegangan dc dan letakkan pada bagian *input* dan *output*. Amatilah tegangan pada sisi *input* dan *output*. Apabila nilai *output* belum sesuai

dengan yang diinginkan, putarlah pengatur tegangan yang berupa baut pada sisi tengah komponen.

#### 3.5.4 Perilaku Pengujian Rangkaian *Level Shifter*

Rangkaian *level shifter* merupakan rangkaian yang dibuat untuk menerjemahkan gelombang dengan tegangan 12V menjadi tegangan 5V. Fungsi dari rangkaian ini adalah untuk melindungi mikrokontroler yang tidak dapat menerima tegangan dengan jumlah besar tanpa mengubah gelombang informasi dari sensor *rotary encoder*. Pengujian rangkaian ini harus dilakukan dengan hati-hati dan teliti karena kesalahan pengujian dapat mengakibatkan hal buruk terhadap sensor maupun mikrokontroler. Pengujian rangkaian *level shifter* dapat dilakukan dengan mengikuti langkah-langkah pengujian dibawah ini

1. Gunakanlah AFG ntuk membangkitkan sinyal dengan masukkan gelombang kotak frekuensi 1KHz.
2. Masukkan tegangan 12V pada sumber masukan dan 5 Volt pada pin keluaran sensor.
3. Perhatikan komponen yang dipasang pada rangkaian *level shifter*. Komponen yang digunakan harus memiliki nilai sesuai dengan rancangan pembuatan rangkaian.
4. Cek posisi pemasangan komponen sesuai dengan skematik perancangan rangkaian tersebut. Pastikan bahwa kaki komponen tidak terbalik.
5. Gunakanlah *multimeter* untuk mengecek jalur pada PCB. Hal ini untuk mengetahui jalur yang saling berhubungan.
6. Gunakan *voltmeter* untuk melihat tegangan *input* dan *output* pada rangkaian ketika diberi tegangan masukan sebesar 12V.
7. Cek gelombang masukan dan keluaran secara bergantian menggunakan osiloskop. Apabila rangkaian bekerja dengan baik maka gelombang yang ditampilkan akan memiliki nilai yang berbeda yaitu gelombang *input* sebesar 12V dan gelombang *output* sebesar 5V.

### 3.5.5 Perilaku Pengujian Sensor Autonics EP50S8-1024-2F-N-24 .

Sensor Autonics EP50S8-1024-2F-N-24 merupakan sensor yang digunakan untuk menentukan sudut elevasi pada alat. Pengujian kondisi sensor dilakukan menggunakan alat *logic analyzer*. Penggunaan alat ini disebabkan karena sensor *rotary encoder* memiliki kabel *output* dalam jumlah yang banyak dan dapat menampilkan gelombang secara bersamaan. Penggunaan *logic analyzer* ini didukung oleh *software* GLA TYPE 1016 dengan sistem operasi *windows 7*. *Software* ini berfungsi sebagai penampil 10 gelombang secara bersamaan. Gunakanlah AFG ntuk membangkitkan sinyal dengan masukkan gelombang kotak frekuensi 1KHz. Ketika seluruh kabel data telah disambungkan dengan *logic analyzer*, tekanlah tombol run pada *software* GLA-Insteak dan putarlah *shaft* sensor dengan cepat. Kemudian amatilah gelombang yang keluar pada layar monitor. Gelombang yang keluar pada monitor akan menunjukkan kondisi sensor pada saat itu.

Pengujian yang kedua adalah pengujian sensor terhadap busur derajat. Pengujian ini bertujuan untuk melihat keakuratan sudut yang dihasilkan oleh putaran sensor. Dalam melakukan pengujian ini, titik *offset* pada busur tidak dibenarkan untuk berubah-ubah . Perubahan titik *offset* akan menyebabkan nilai selisih menjadi lebih besar. Pengujian sudut elevasi dapat dilakukan dengan mengikuti langkah-langkah dibawah ini

1. Masukkan program sensor Autonics EP50S8-1024-2F-N-24 pada arduino yang telah terhubung dengan sensor.
2. Lubangi titik tengah busur sebesar 4.5 mm dengan bor. Pembuatan lubang ini berfungsi agar titik *offset* busur tidak berpindah.
3. Pasangkan busur yang telah dilubangi kepada shaft sensor *rotary encoder*.
4. Pada posisi diatas sensor dan tepatnya dibawah busur 360°, letakkan jarum/lidi dengan lem yang cukup kuat.
5. Putarlah busur dengan kelipatan 5°.

6. Amatilah sudut putaran yang dihasilkan sensor *rotary encoder* pada LCD.

### 3.5.6 Perilaku Pengujian Sensor HMC5833 L

Sensor HMC5883L merupakan sensor yang akan digunakan untuk mendapatkan sudut azimuth. Pengujian kondisi sensor bertujuan agar mengetahui kondisi sensor masih mampu berkerja atau tidak. Adapun tahap-tahap pengujian kondisi sensor adalah

1. Tahap pertama dalam pengujian ini adalah dengan menghubungkan sensor dengan mikrokontroler.
2. Setelah terhubung, pengecekan pertama dilakukan untuk mengetahui bahwa kaki VCC,GND,SCL dan SDA sudah terhubung dengan benar pada pin di mikrokontroler.
3. Sambungkan mikrokontroler pada komputer menggunakan kabel USB.
4. Masukkan program kompas ke dalam mikrokontroler
5. Amatilah yang terjadi pada monitor PCB

Setelah melakukan pengujian kondisi sensor, tahap selanjutnya adalah pengujian terhadap nilai sensor. Apabila tahap diatas dilaksanakan dengan baik dan sensor menunjukkan kondisi yang positif maka pengujian terhadap nilai sensor dilakukan. Pengujian terhadap nilai sensor akan menggunakan dua alat pembanding yaitu kompas handphone dan busur derajat. Tahap-tahap pengujian azimuth sensor adalah

1. Dalam melakukan pengujian nilai sensor, dibutuhkan dudukan sensor agar nilai *offset* sensor tidak berubah. Pada penelitian ini menggunakan bagian belakang mangkok yang berbentuk bundar untuk dimanfaatkan sebagai dudukan sensor. Selain itu dibawah sensor ditempelkan plastik dudukan yang dialaskan kardus berbentuk bulat agar ketika sensor diputar nilai *offset* tidak bergeser.



2. Tempelkan busur pada bagian bawah sensor untuk mengetahui pergeseran derajat pada busur.
3. Jauhi barang-barang yang terbuat dari besi dan logam. Hal ini dikarenakan nilai kompas menjadi tidak stabil.
4. Gunakan kompas digital untuk mengetahui arah utara pada sensor.
5. Sebelum mengukur nilai sensor, lakukan kalibrasi pada sensor.
6. Masukkan program kalibrasi sensor pada mikrokontroler, dan putarlah sensor sesuai dengan waktu yang telah ditentukan.
7. Catatlah nilai  $x,y$  dan  $z$  *offset* serta  $x,y$  dan  $z$  *gain offset*. Masukkan data tersebut kedalam program sensor kompas.
8. Masukan ulang program kompas pada mikrokontroler, serta putarlah sensor dengan kelipatan  $5^\circ$ .
9. Amatilah nilai sudut kompas pada LCD dan busur.

### 3.5.7 Pengujian pengiriman serial RS485

RS485 merupakan komponen yang berfungsi untuk mengirimkan data serial. Data serial yang dikirimkan oleh RS485 berupa data GPS serta sudut elevasi dan azimuth. Pengiriman data GPS dan data sensor dikirimkan dengan dua RS485 yang berbeda. Pengujian pengiriman data serial didukung oleh *software Real term* dan *U center*. Pada bagian pengiriman data sensor, Tahap-tahap pengujiannya adalah

1. Gunakan perintah `Serial.print` untuk mengirimkan data serial
2. *Upload* program gabungan sensor pada mikrokontroler
3. Hubungkan RS485 (kabel berwarna merah) dengan USB RS485 yang dihubungkan pada komputer
4. Bukalah *device manager* untuk melihat *port* USB RS485
5. Bukalah aplikasi *Real Term* pada komputer dan aturlah *baudrate* sesuai dengan pengaturan pada program.
6. Ubahlah *port* pada aplikasi sesuai dengan *port* pada *device manager*
7. Data mentah akan muncul pada layar dan klik *change* untuk merubah format data menjadi angka desimal.

8. Pindahkan kabel dari USB RS485 menuju kontroler antenna.
9. Amatilah data pada LCD kontroler.

Pada bagian kedua merupakan tahap pengujian pengiriman data serial GPS. Data serial GPS tidak dikirimkan ke kontroler antenna. Data serial GPS dikirimkan pada USB serial komputer. Hal ini dikarenakan data GPS hanya dibutuhkan untuk mengetahui titik pada *ground control station*. Adapun tahap pengujian pengiriman serial GPS adalah

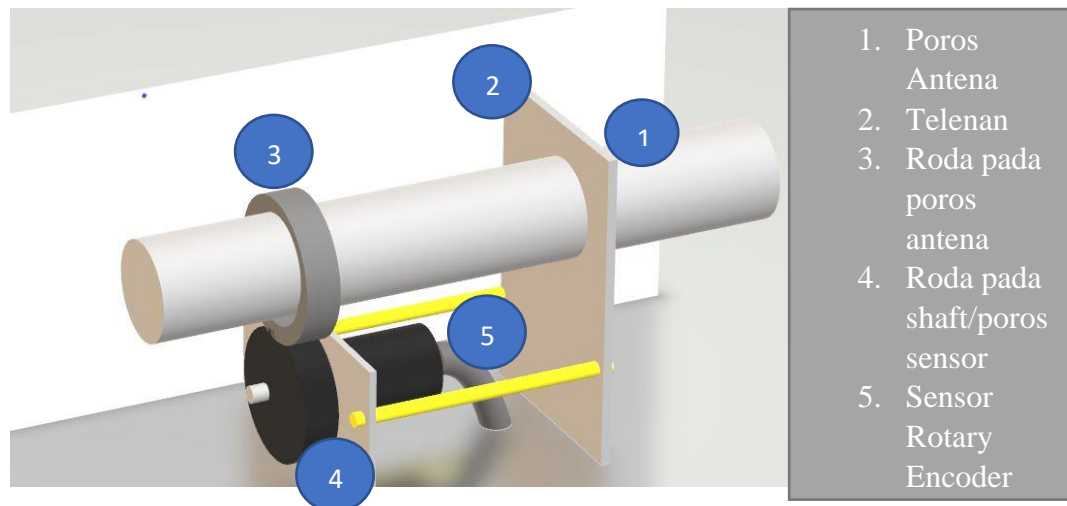
1. Hubungkan RS485 (kabel berwarna hitam) dengan USB RS485 yang dihubungkan pada komputer
2. Bukalah *device manager* untuk melihat *port* USB RS485
3. Ubahlah *port* pada aplikasi sesuai dengan *port* pada *device manager*
4. Data mentah akan muncul pada layar dan klik *change* untuk merubah format data menjadi angka desimal.
5. Bukalah aplikasi *U center* untuk mengetahui titik *longitude* , *altitude*, dan *latitude*.
6. Ubahlah pengaturan *port* sesuai dengan *port* pada *device manager*.
7. Ketika aplikasi telah terhubung, akan terdapat indikator lampu hijau pada bagian koneksi.
8. Titik *longitude*, *altitude* dan *latitude* pada GPS akan muncul pada aplikasi *U blox center*.

### **3.6 Pemasangan Sistem Umpan Balik pada Antena Controller Dua-Axis**

Setelah dilakukan pembuatan alat sistem umpan balik maka tahap selanjutnya adalah peletakan alat tersebut pada antenna penjejak. Peletakan alat dimulai dari perencanaan peletakkan elemen umpan balik. Peletakkan elemen umpan balik yang pertama adalah sensor elevasi. Perencanaan peletakkan sensor elevasi adalah dengan menyentuh sensor dengan poros pada antenna. Shaft/ poros pada sensor dan antenna akan dipasangkan roda dengan diameter yang sama. Kedua roda pada sensor dan poros antenna akan saling bersentuhan. Disentuhkannya kedua roda akan

menggerakkan roda pada sensor yang menyebabkan *shaft*/poros sensor akan berputar. Perhatikan gambar berikut

Gambar 3. 19 Perencanaan Pemasangan Sensor Elevasi

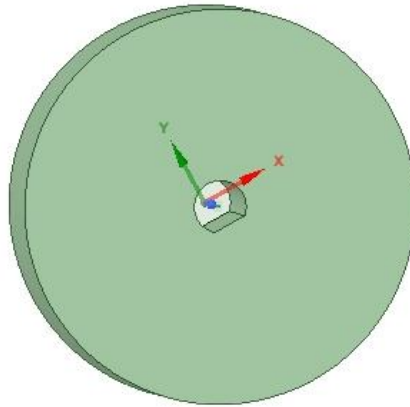


Gambar 3.19 adalah rencana pemasangan sensor elevasi.

Gambar pemasangan sensor di desain menggunakan *software 3D paint* pada *windows 10*. Pemasangan dilakukan dengan penggunaan baut dengan Panjang 15cm dan diameter 8 mm yang dipasangkan pada akrilik yang diapit oleh sensor dan roda. Setelah itu baut akan dipasangkan pada telenan antena kontroler sehingga sensor dapat terpasang tepat dibawah poros antena. Ketika poros antena diputar menggunakan kontroler, sensor akan berputar sesuai dengan sudut putar pada poros. Hal ini disebabkan oleh ukuran roda yang saling bersentuhan memiliki perbandingan ukuran 1:1 sehingga perputaran akan menghasilkan nilai yang sama.

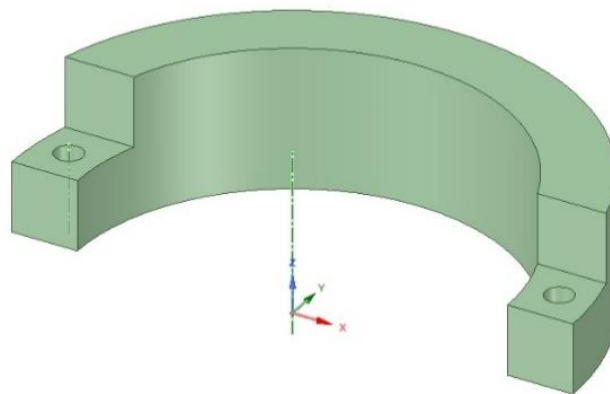
Gambar 3.20 adalah pembuatan roda yang akan dipasangkan pada poros sensor. Desain roda sensor menggunakan *software Design Spark Mechanical* versi 4. Roda akan dibuat menggunakan 3D printer dengan ukuran jari-jari *shaft* adalah 4 mm, dengan diameter permukaan adalah 60 mm dan ketebalan roda adalah 250 mm. Setelah roda berhasil di cetak maka sisi roda akan dibalut dengan amplas dengan

nomor 1000. Penggunaan amplas berguna untuk memperkasar sisi roda sehingga tidak terjadi *slip* pada saat roda poros dan roda sensor saling bersentuhan.



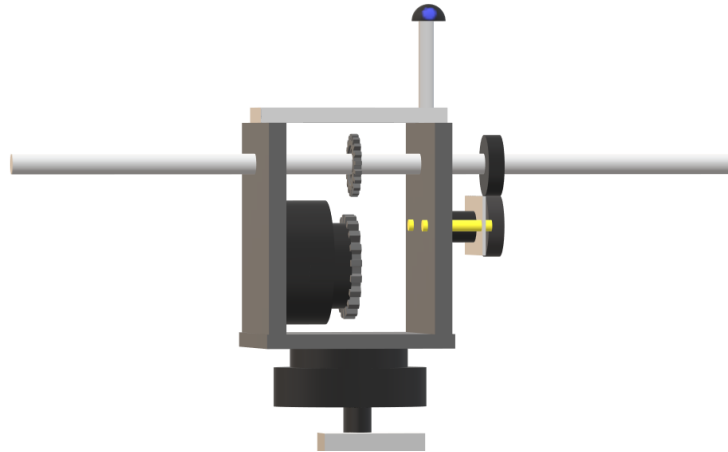
*Gambar 3. 20 Pembuatan Roda pada shaft sensor*

Gambar 3.21 adalah desain pembuatan roda pada poros antenna. Pembuatan roda dilakukan dengan dua buah roda diatas. Hal ini dikarenakan untuk mempermudah pemasangan roda pada poros antenna yang panjang. Ukuran poros pada antenna adalah 45 mm sehingga ukuran diameter permukaanroda adalah 15 mm. ketebalan roda pada gambar diatas memiliki ketebalan yang sama dengan roda pada shaft sensor yaitu 250 mm. Ukuran roda pada poros antenna memiliki nilai yang sama dengan roda pada poros agar mencapai perbandingan ukuran 1:1.



**Gambar 3. 21 Pembuatan Roda pada Poros Sensor**

Peletakan sensor selanjutnya adalah kompas dan GPS. Perencanaan peletakan sensor adalah dengan meletakkan sensor kedalam dudukan GPS. Setelah itu, sensor akan dipasangkan dengan posisi lebih tinggi diatas motor antena. Perhatikan gambar berikut



Gambar 3. 22 Peletakan Sensor Kompas

Pada gambar 3.22 terlihat bahwa pemasangan sensor kompas lebih tinggi dibandingkan motor AC. Sensor akan dipasang 30 cm di atas motor. Hal ini disebabkan sensor kompas rentan terhadap benda yang memiliki gaya kemagnetan seperti *paramagnetik* dan *feromagnetik*. Apabila sensor kompas mendekati jenis benda tersebut maka kompas akan sangat mudah terdistorsi dan nilai menjadi *error*.

Selanjutnya adalah peletakan *Box Sensor Processing*. Pada kotak tersebut terdapat rangkaian mikrokontroler dan rangkaian sensor. Selain itu terdapat LCD yang dapat menampilkan nilai azimuth dan elevasi. Perhatikan gambar 3.23. Gambar 3.23 terlihat bahwa *Box* diletakkan di bawah motor *antenna tracker*. Hal ini disebabkan agar rangkaian pada sensor dan elemen umpan balik memiliki jarak yang lebih dekat sehingga tidak memerlukan kabel yang lebih panjang. Selain itu LCD pada *Box* yang menampilkan data azimuth dan elevasi dapat dilihat dengan mudah dengan keadaan diam.



Gambar 3. 23 Peletakan *Box Sensor Processing*