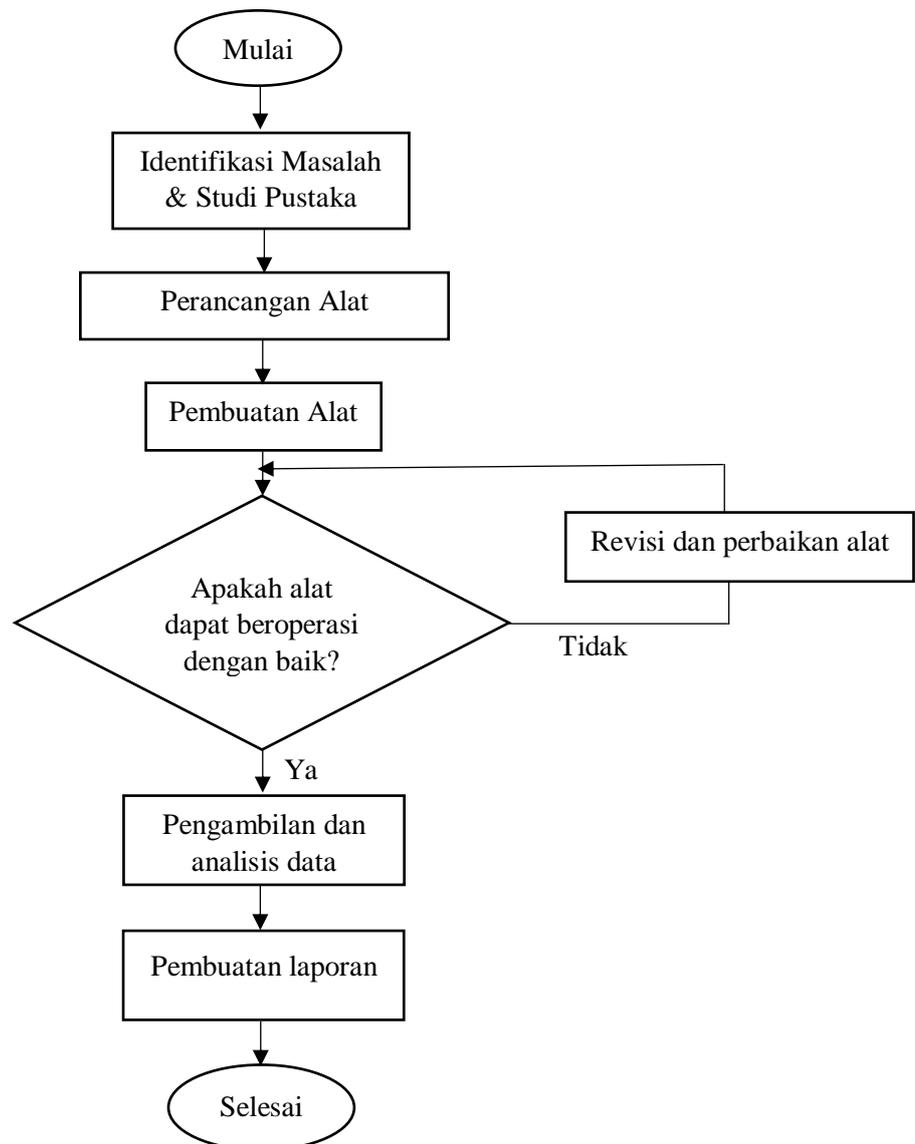


BAB III

METODE PENELITIAN

Metode pada penelitian ini menjelaskan mengenai prosedur dan teknik dalam pelaksanaan penelitian dengan tujuan untuk mendapatkan jawaban atas apa yang menjadi pertanyaan pada objek penelitian.

3.1. Flowchart Metode Penelitian



Gambar 3.1. Diagram Alir Metode Penelitian

Pada Gambar 3.1 merupakan flowchart atau diagram alir yang menunjukkan tahapan-tahapan yang dilakukan dalam penelitian diantaranya :

1. Identifikasi Masalah dan Belajar Literatur

Tahap ini merupakan tahap awal memulainya penelitian di mana mengidentifikasi masalah mengenai kontroler antena penjejak 2 axis dengan menggunakan sistem *control close loop* serta pengumpulan informasi dari berbagai literatur seperti buku, jurnal, karya ilmiah maupun dari internet yang akan digunakan sebagai penunjang dan pedoman dalam melakukan sebuah penelitian.

2. Perancangan Alat

Tahap untuk pembuatan suatu alat dan program yang pertama kali dilakukan adalah tahap perancangan. Tahap ini merupakan tahap yang paling penting dalam pembuatan alat, karena dengan merancang kita dapat mengetahui komponen apa yang digunakan dan program seperti apa yang harus dilakukan agar alat dapat berjalan sesuai yang diharapkan yaitu dapat menggerakkan motor ke arah vertikal dan horzontal.

3. Pembuatan Alat

Tahap ini merupakan tahap di mana alat kontroler antena mulai dikerjakan dan direalisasikan dari rancang bangun *software* dan rancang bangun *hardware* agar sesuai dengan perencanaan dan tujuan alat dibuat.

4. Pengujian Alat

Tahap ini merupakan pengujian terhadap kontroler antena untuk mengetahui kinerja dari alat yang telah dikerjakan, jika hasil yang di tampilkan pada LCD dan jika alat belum sesuai dengan rencana yang diharapkan yaitu dapat mengendalikan pergerakan motor ke arah horizontal dan vertikal, maka perlu adanya perbaikan hingga sesuai dengan rencana pembuatan alat.

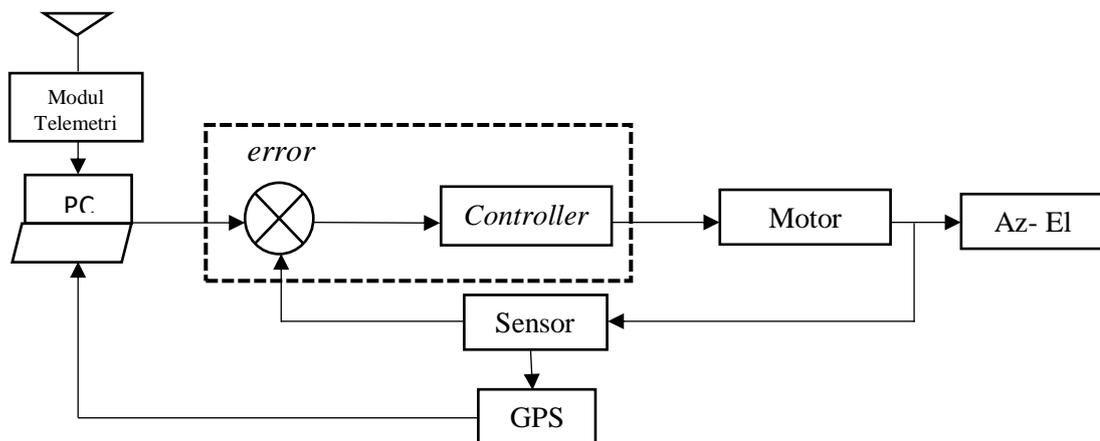
5. Pengambilan dan Analisis Data

Tahap pengambilan data dilakukan ketika pengujian dinyatakan berhasil. Data yang diambil pada penelitian ini yaitu data azimuth dan elevasi serta *error* yang dihasilkan dari perbandingan dua variabel inputan yaitu data yang berasal dari muatan dan data yang berasal dari sensor. Setelah itu dilakukan

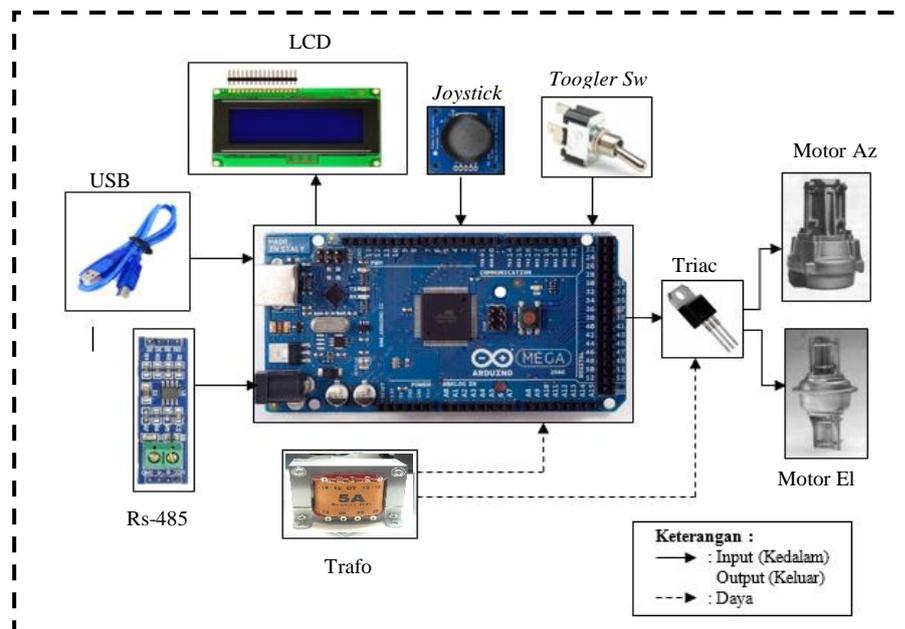
analisis terhadap pergerakan motor, seberapa besar penyimpangan yang dihasilkan antara pergerakan motor dan muatan.

3.2. Deskripsi Sistem

Sistem yang akan dirancang pada penelitian ini disajikan dalam bentuk blok diagram perancangan sistem.



Gambar 3.2. Blok Diagram Sistem Keseluruhan



Gambar 3.3. Blok Diagram Sistem Kontroler

Gambar 3.2 menjelaskan mengenai sistem kerja antenna penjejak secara keseluruhan, sedangkan pada Gambar 3.3 merupakan salah satu bagian dari sistem antenna penjejak yaitu kontroler. Penelitian ini dilakukan sebagai salah satu cara untuk menjaga komunikasi antara antenna pemancar dengan antenna pendeteksi/ *ground station*. Rancangan sistem ini dibagi menjadi 2 bagian yaitu sistem yang terdapat pada *radiosonde* dan sistem yang terdapat pada antenna pendeteksi/ *ground station*. Sistem yang terdapat pada *radiosonde* akan mengintegrasikan data-data dari sensor gps, akselerometer, barometer, dan lain-lain, namun yang akan diambil dan digunakan oleh sistem antenna pendeteksi berupa koordinat posisi dan ketinggian dari *radiosonde* yang kemudian akan dikirimkan dan ditangkap oleh modul telemetri yang terdapat pada *ground station*. Bagian sistem antenna pendeteksi juga terdapat sensor yang berfungsi untuk mendapatkan data-data terkait nilai azimuth dan elevasi, selain itu terdapat modul GPS sebagai penentu titik antenna pendeteksi berada. Data sensor GPS tidak diolah oleh mikrokontroler, melainkan langsung dikirim ke PC.

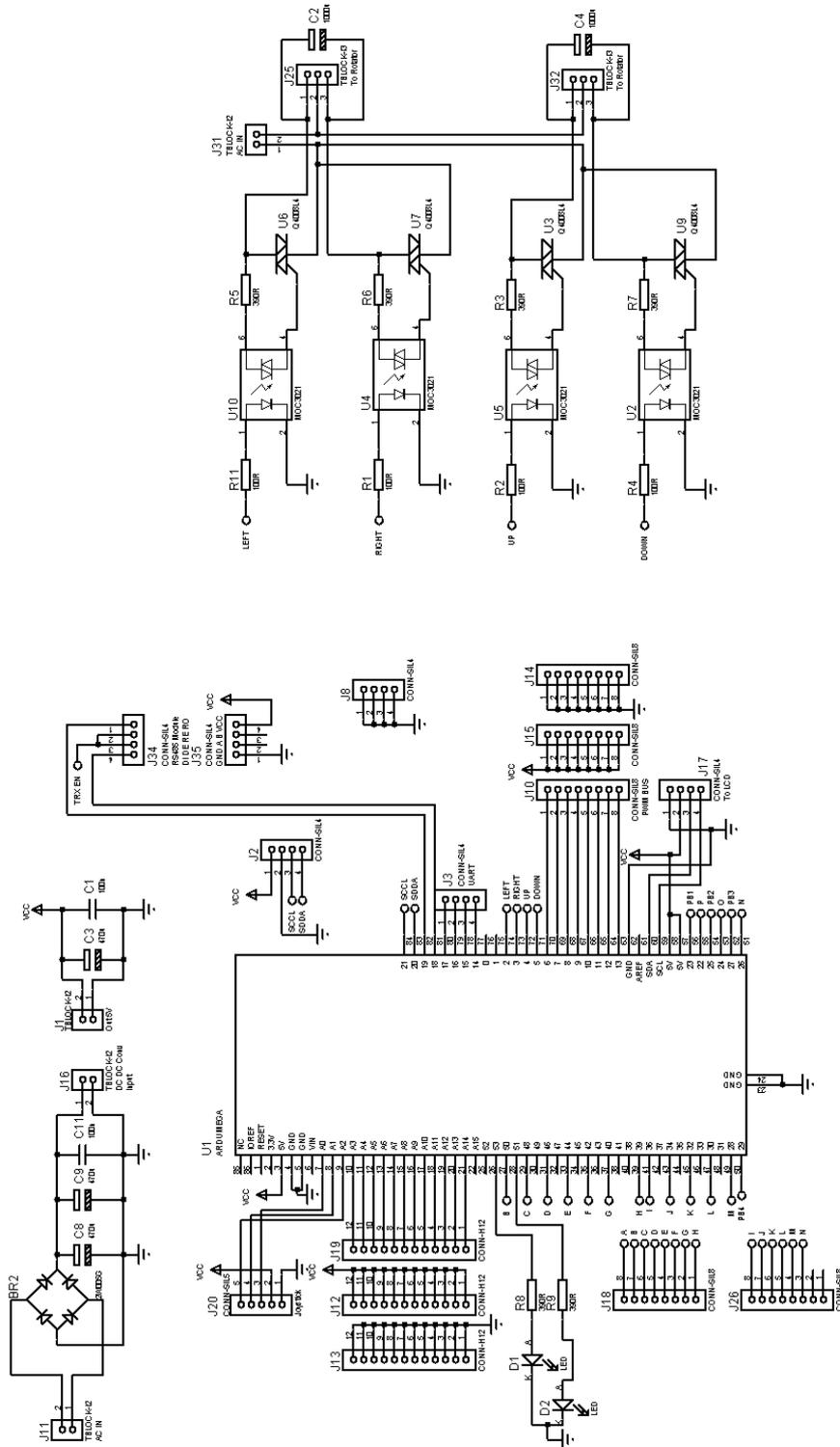
Sistem berawal dari data telah dikirimkan dari muatan menggunakan radio telemetri dan diterima oleh *ground station*, selanjutnya diproses oleh GUI (*Graphic User Interface*). Data yang diterima dari muatan pada mulanya masih berbentuk data *longitude*, *latitude*, dan *altitude*. Kemudian dilakukan pemrosesan data pada PC berupa perhitungan sudut azimuth dengan metode azimuth untuk mendapatkan sudut secara horizontal dan perhitungan sudut elevasi dengan metode *haversine* dan aturan sinus trigonometri untuk mendapatkan sudut secara vertikal. Metode azimuth adalah metode yang digunakan untuk memperoleh sudut yang terbentuk dari dua titik koordinat terhadap arah utara. Sudut yang diperoleh akan digunakan sebagai *setpoint* untuk gerak horizontal. Sedangkan untuk metode *haversine* adalah metode yang berfungsi untuk mencari jarak antara kedua titik koordinat yang di proses dengan aturan sinus trigonometri sehingga didapat sudut elevasi. Sudut elevasi akan digunakan *setpoint* untuk gerakan vertikal. Data keluaran yang berupa nilai azimuth dan elevasi tersebut akan masuk ke mikrokontroler untuk menggerakkan motor. Dimana sebelumnya, mikrokontroler akan melakukan *parsing* data dari sistem sistem tersebut, sehingga di dapat nilai keluaran azimuth dan elevasi secara terpisah.

Pada proses ini terdapat elemen umpan balik pada sistem, dimana nilai umpan balik tersebut adalah nilai yang berasal dari sensor. Nilai umpan balik akan mempengaruhi aksi pengontrolan. Nilai *setpoint* yang diperoleh dari PC akan dibandingkan dengan nilai yang berasal dari sensor pada antenna pendeteksi untuk melihat *error*, untuk melihat seberapa besar penyimpangan antenna pendeteksi terhadap muatan *radiosonde*. Apabila nilai yang dihasilkan pada sistem terdapat komparator, maka hasil penyimpangan tersebut kemudian diproses dan selanjutnya digunakan untuk menggerakkan motor sampai diperoleh kondisi posisi antenna yang stabil atau didapat keluaran sesuai dengan yang diinginkan. Sehingga antenna pendeteksi akan bergerak ke arah posisi aktual muatan. Kondisi ini dilakukan berulang-ulang selama sistem menerima data.

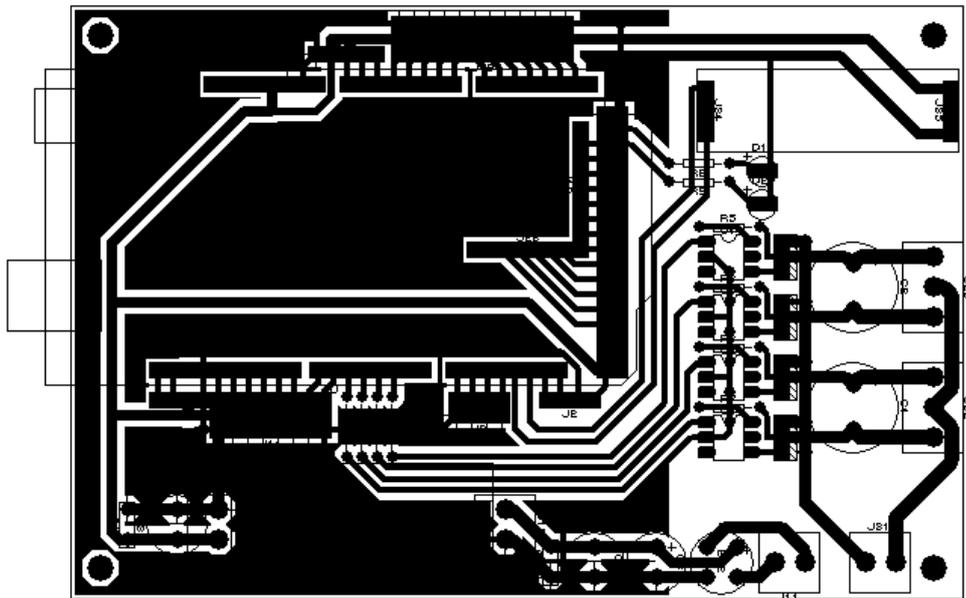
3.3. Rancangan Perangkat Keras

Pada perancangan ini meliputi perancangan bagian-bagian rangkaian dalam pembuatan rangkaian secara keseluruhan yang digunakan pada sistem kontroler antenna penjejak, beberapa komponen perangkat keras disusun untuk menghasilkan sebuah sistem kendali gerak pada antenna pendeteksi. Dalam pembuatan perangkat keras dibutuhkan *software* pendukung. *Software* yang digunakan adalah *proteus*, *software* ini digunakan untuk membuat skematik alat, desain PCB dan simulasi alat. Selain itu dapat membantu mempermudah untuk melakukan perancangan pada *hardware*. Perangkat keras secara keseluruhan terdiri dari beberapa sistem yang telah terhubung oleh mikrokontroler antara lain, catu daya sebagai penyuplai daya pada rangkaian elektronika dan motor, rangkaian arduino mega 2560 , rangkaian *driver triac* motor, *joystick*, LCD 20x4, *sw selector*, dan RS-485. Skema keseluruhan yang dibangun dalam perancangan ditunjukkan oleh Gambar 3.4 dan *layout* PCB rangkaian ditunjukkan oleh Gambar 3.5.

1. Pembuatan Skematik Alat

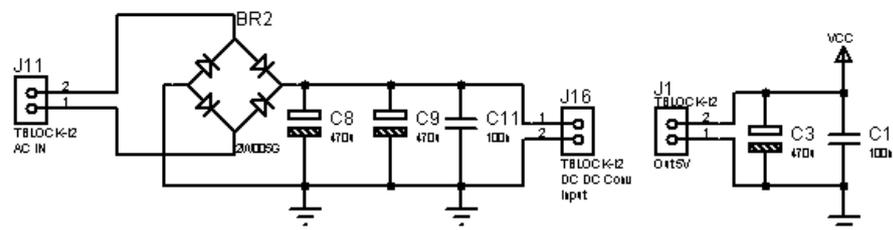


Gambar 3.4. Rangkaian Skematik Keseluruhan Sistem Kontroler Antena Penjejak



Gambar 3.5. Layout PCB Kontroler Antena Penjejak

3.3.1. Bagian Catu Daya Dari Sistem Kontroler Antena Penjejak



Gambar 3.6. Rangkaian Catu Daya Kontroler Antena Penjejak

Gambar 3.6 merupakan rangkaian *power supply* atau catu daya dalam sistem kontroler digunakan untuk memberikan sumber energi pada rangkaian. Sumber listrik yang digunakan pada rangkaian berasal dari tegangan 1 fasa yaitu 220V yang kemudian diturunkan oleh trafo *step down* untuk memberikan sumber tegangan pada motor dan rangkaian elektronika. Namun, pada rangkaian elektronika membutuhkan sumber tegangan DC untuk beroperasi, oleh karena itu dibutuhkan penyearah (*rectifier*) yang berguna untuk mengubah sumber AC ke DC. Terdapat beberapa tahapan pada catu daya dalam

Pada kontroler penggerak antena digunakan Arduino Mega 2560. Mikrokontroler yang terdapat pada Arduino mega 2560, berfungsi sebagai pengatur sistem secara keseluruhan. Unit pemroses dilakukan pada mikrokontroler ini, dimana data-data yang telah diakses dari PC dan sensor diolah, yang kemudian digunakan untuk sistem kendali pergerakan motor.

Rangkaian ini juga menunjukkan beberapa komponen elektronik yang terhubung atau terintegrasi dalam suatu unit pengolahan data atau mikrokontroler antara lain :

1. LCD 20 x 4

LCD (*Liquid Crystal Display*) terhubung oleh papan mikrokontroler yang terdapat pada pin 20 (SDA) dan pin 21 (SCL). LCD tersebut memiliki fungsi untuk menampilkan status kerja alat. Dimana pada sistem ini, LCD akan menampilkan status pergerakan motor saat keadaan vertikal maupun horizontal, kondisi sistem *tracking*, dan menampilkan nilai azimuth serta nilai elevasinya.

2. *Joystick*

Joystick pada rangkaian ini terhubung pada digital input yang terdapat pada papan mikrokontroler yaitu pada pin A2, sedangkan untuk sumbu x dan y terhubung pada pin analog input yaitu A0 dan A1. *Joystick* digunakan pada kondisi sistem *tracking* manual. *Joystick* ini dapat bergerak ke arah sumbu x dan y, dimana perangkat ini dapat digunakan sebagai penggerak motor pada posisi horizontal atau vertikal.

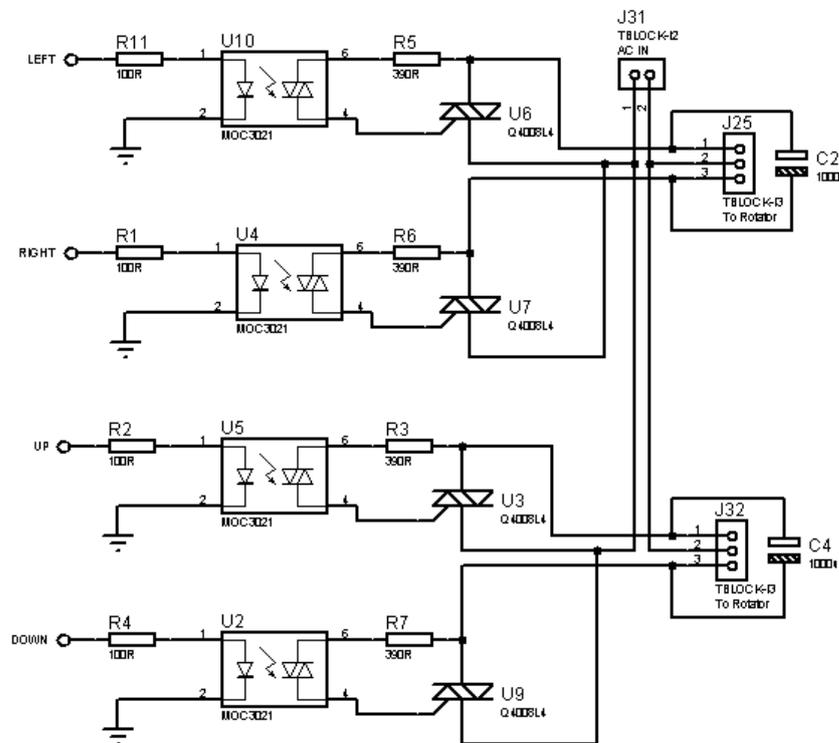
3. *Toogle Switch*

Toogle switch pada rangkaian ini terhubung oleh pin 12 dan pin 13. Dimana komponen ini dapat memutus rangkaian listrik maupun mengalihkan konduktor ke konduktor lain. *Switch* tersebut berfungsi untuk mengatur kondisi sistem *tracking* yaitu manual, otomatis dan *emerstop*.

4. RS-485

RS-485 juga terhubung oleh rangkaian mikrokontroler yaitu pada pin 18 (Tx) dan pin 19 (Rx). Modul ini digunakan sebagai modul komunikasi dua buah atau lebih mikrokontroler. Pada penelitian ini digunakan untuk komunikasi antara sistem kontroler dengan sistem *feedback* yang berupa data sensor posisi azimuth dan elevasi.

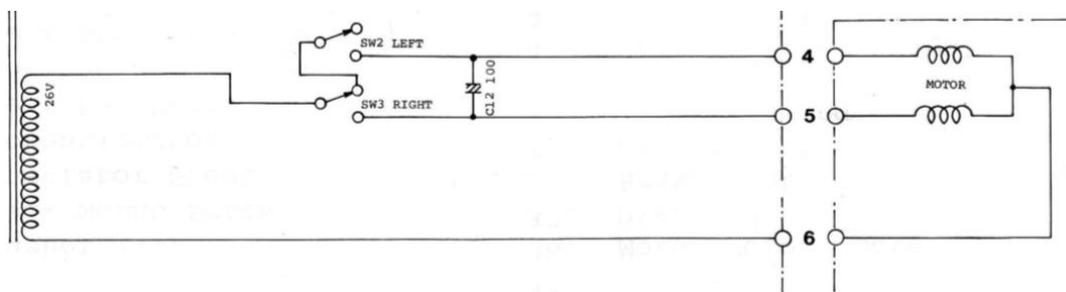
3.3.3. Bagian *Driver* Motor Dari Sistem Kontroler Antena Penjejak



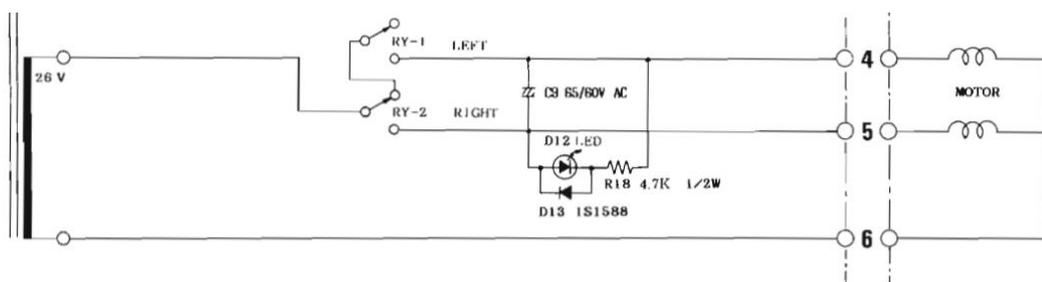
Gambar 3.8. Rangkaian *Driver* Motor Kontroler Antena Penjejak

Gambar 3.8 menunjukkan rangkaian *driver* motor pada kontroler antena penjejak. Pada rancang bangun sistem kontroler terdapat *driver* motor yang difungsikan sebagai penggerak motor, dimana kondisi *driver* akan dikendalikan oleh mikrokontroler sebagai pengatur arah putaran motor. Rangkaian *driver* ini terdiri dari beberapa komponen yaitu *triac* BTA12,

optocoupler MOC3020, resistor, dan kapasitor. Komponen yang digunakan memiliki fungsi diantaranya yaitu *triac* yang difungsikan sebagai *switching* motor. *Optocoupler* sebagai pemacu *triac* sekaligus sebagai jembatan pemisah antara tegangan rangkaian kontrol yang merupakan tegangan rendah DC terhadap rangkaian daya yang berupa tegangan tinggi AC untuk menghindari kerusakan pada mikrokontroler. Selanjutnya rangkaian *driver* dihubungkan oleh motor AC (azimut dan elevasi) yang rangkaiannya ditunjukkan oleh Gambar 3.9 dan Gambar 3.10.



Gambar 3.9. Skematik Rotator Kenpro 400RC



Gambar 3.10. Skematik Aktuator Yaesu G250

Cara Kerja Rotator dan Aktuator Sistem Penggerak Antena :

Pergerakan motor mengikuti *driver* motor, dimana dengan menggunakan *driver*, tegangan dan arus yang di *supply* ke motor dapat disesuaikan dengan kebutuhan dan spesifikasi masing-masing motor. *Driver* bekerja sesuai dengan intruksi yang diterima dari mikrokontroler melalui *port* yang terdapat 4 *triac*, yang digunakan untuk mengendalikan 2 motor dengan arah putaran yang berbeda. Sinyal

yang telah dikirimkan mikrokontroler kepada *driver* menentukan kapan kedua motor akan bergerak ke kiri dan kanan dan bergerak naik dan turun. Proses pergerakan motor diawali ketika *logic* dari *micro* berlogika HIGH, maka arus akan mengalir melewati resistor menuju *optocoupler*, ini menyebabkan *optocoupler* “on”. Saat *optocoupler* “on” maka terjadi pemicuan *gate* pada *triac*. Setelah *gate* pada *triac* mendapat arus pemicuan, maka gerbang A1 akan terhubung ke gerbang A2 sehingga dapat mengalirkan arus beban dimana terjadinya putaran pada motor induksi satu fasa. Sebaliknya saat *logic* dari *micro* “LOW”, maka arus dari VCC tidak akan masuk menuju *optocoupler* tetapi akan masuk menuju pin *micro*. Ini menyebabkan *optocoupler* dalam keadaan “off” begitu pula yang terjadi pada *triac* karena tidak mendapat arus pemicuan.

Output hasil keluaran dari mikrokontroler berubah-ubah sesuai dengan perintah. Arduino akan mengeluarkan *ouput* HIGH dan LOW yang dikirimkan ke masing-masing motor, sehingga arah yang dihasilkan dapat berubah. Setiap motor akan menerima 2 *output* dari arduino dimana *output* tersebut akan menentukan arah putaran. Berikut adalah tabel status pergerakan motor ketika setiap *port* diberi masukan yang berbeda.

Tabel 3.1. Tabel Status Pergerakan Motor Azimuth dan Elevasi

Pergerakan Motor		Port			
		2	3	4	5
Azimuth	CW	0	1	/	
	CCW	1	0		
	STOP	0	0		
Elevasi	↑ (Naik)	/		1	0
	↓ (Turun)			0	1
	STOP			0	0

Tabel 3.1 menunjukkan kondisi motor ketika *input* yang dikirimkan oleh mikrokontroler dan eksekusi oleh *driver* motor diberi nilai logika seperti diatas. Nilai logika 1 sama dengan pin *output* dalam keadaan HIGH sedangkan nilai logika

0 menunjukkan pin *output* dalam keadaan LOW. Motor azimuth mendapat masukan dari *port* 2 dan 3 sedangkan motor elevasi mendapatkan masukan dari *port* 4 dan 5. Ketika pin 3 diberi logika HIGH dan pin 2 diberi logika LOW maka motor akan berputar ke kanan, sebaliknya ketika pin 3 diberi logika LOW dan pin 2 diberi logika HIGH maka motor akan bergerak ke kiri. Begitu pula pada motor elevasi, ketika pin 5 diberi logika HIGH dan pin 4 diberi logika LOW maka motor akan bergerak ke atas, sebaliknya ketika pin 5 diberi logika LOW dan pin 4 diberi logika HIGH maka motor akan bergerak ke bawah. Kedua motor tidak akan melakukan pergerakan ketika pada kedua pin diberi logika LOW.

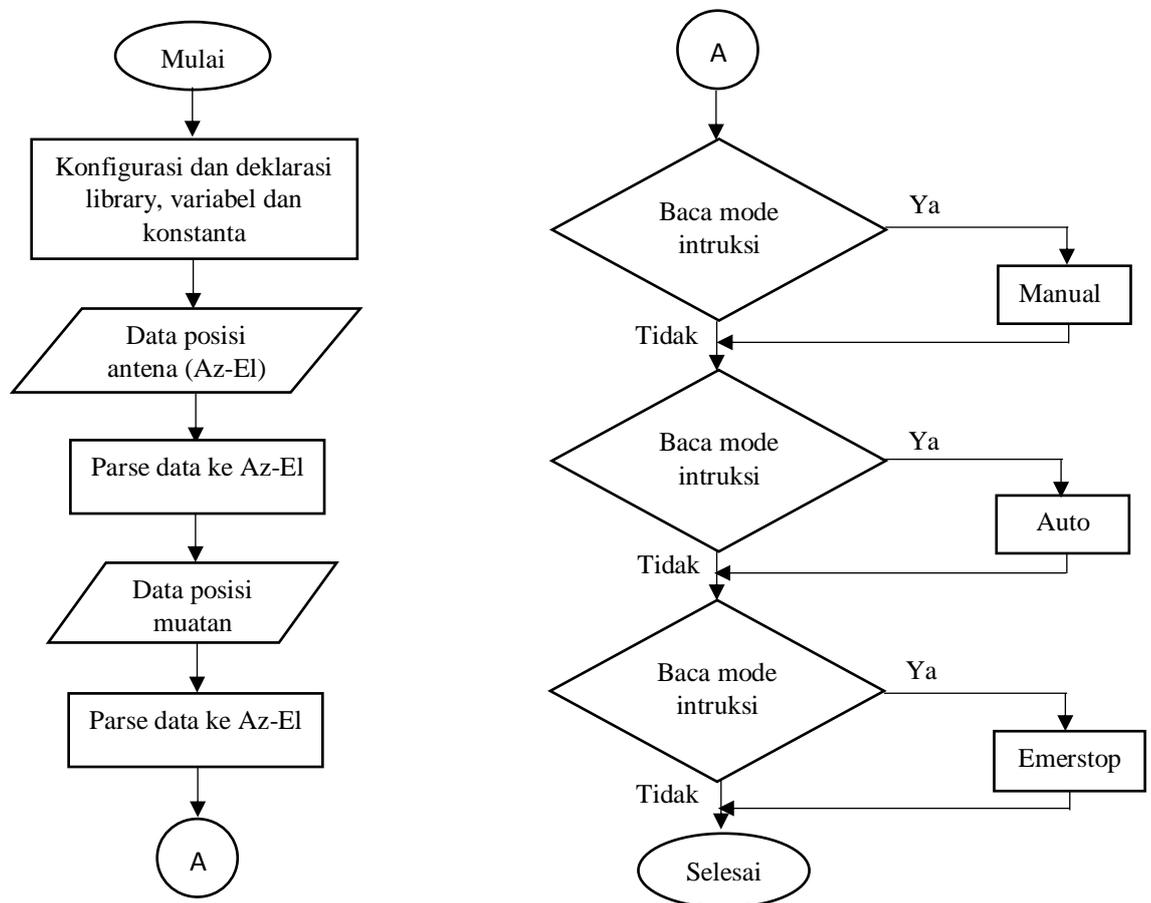
3.4. Rancangan Perangkat Lunak

Rancangan perangkat lunak sangatlah penting karena merancang secara keseluruhan program yang akan dijalankan. Perancangan perangkat lunak menggunakan program bahasa c dalam *Arduino IDE* yang digunakan. Perancangan ini dibuat untuk menjalankan sebuah mikrokontroler agar dapat bekerja sesuai dengan yang diinginkan. Selain itu agar nilai-nilai dari pengolahan data *Arduino*, dapat ditampilkan pada LCD dan dapat diketahui kondisi dan hasilnya secara *real time*.

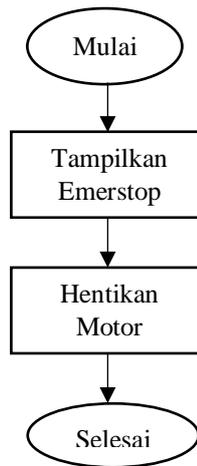
Pada pemrograman *Arduino* sendiri terdapat 3 struktur bagian utama yaitu yang pertama deklarasi awal, dimana digunakan untuk mendeskripsikan variabel-variabel yang akan digunakan pada program utama. Kemudian kedua bagian *setup* digunakan sebagai kode inisialisasi sistem *Arduino* dan dijalankan hanya sekali ketika memulai *start Arduino* atau ketika program direset. Ketiga adalah bagian loop yang merupakan inti utama pemrograman dan akan dijalankan berulang-ulang. Berikut adalah diagram alir program keseluruhan sistem kontroler antena penjejak yang ditunjukkan pada Gambar 3.11 hingga Gambar 3.14.

3.4.1. Diagram Alir Perangkat Lunak

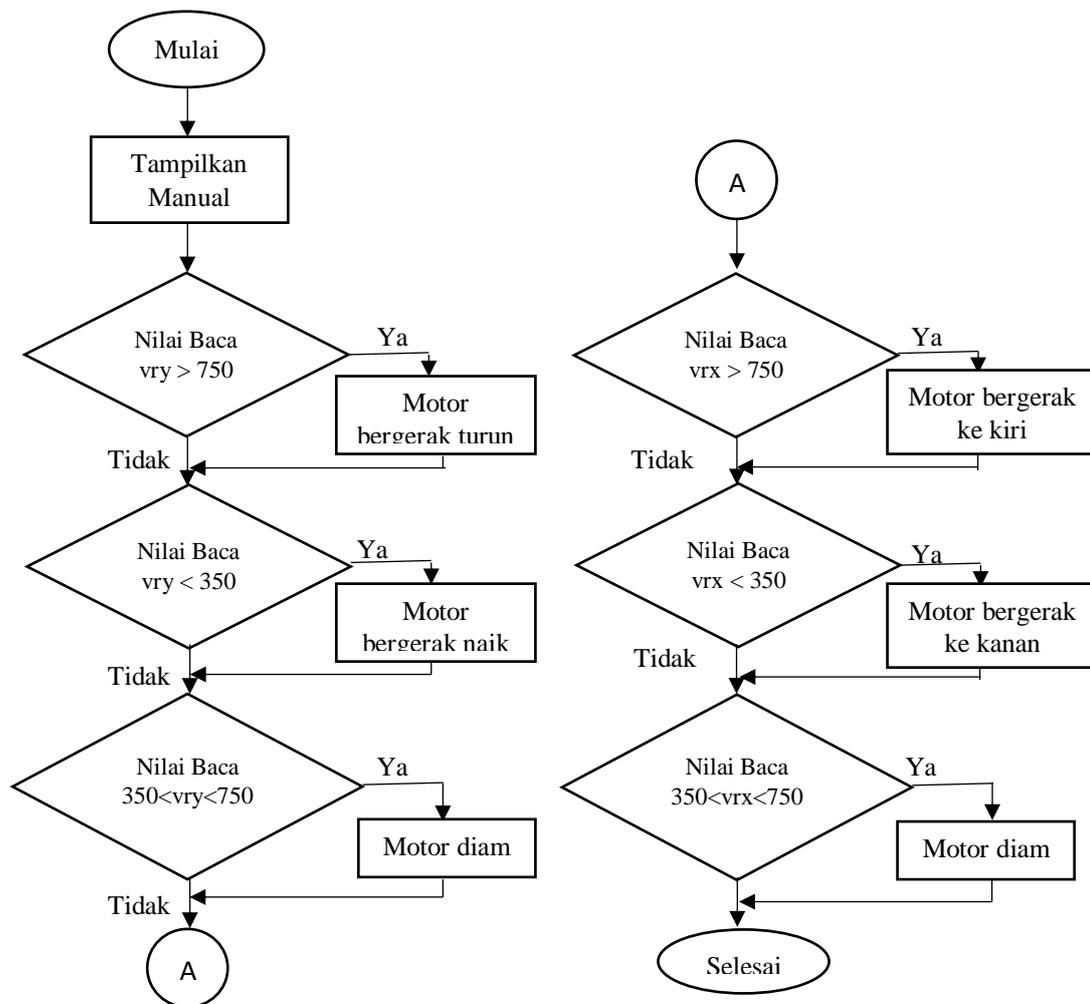
Pembuatan alat sistem kontroler antenna penjejak membutuhkan sistem perangkat lunak. Sistem perangkat lunak yang dibutuhkan adalah *software Arduino IDE* yang digunakan untuk membuat program. Program dibuat agar sistem dapat mengendalikan arah pergerakan motor. Adapun diagram alir program yang ditunjukkan pada Gambar 3.11.



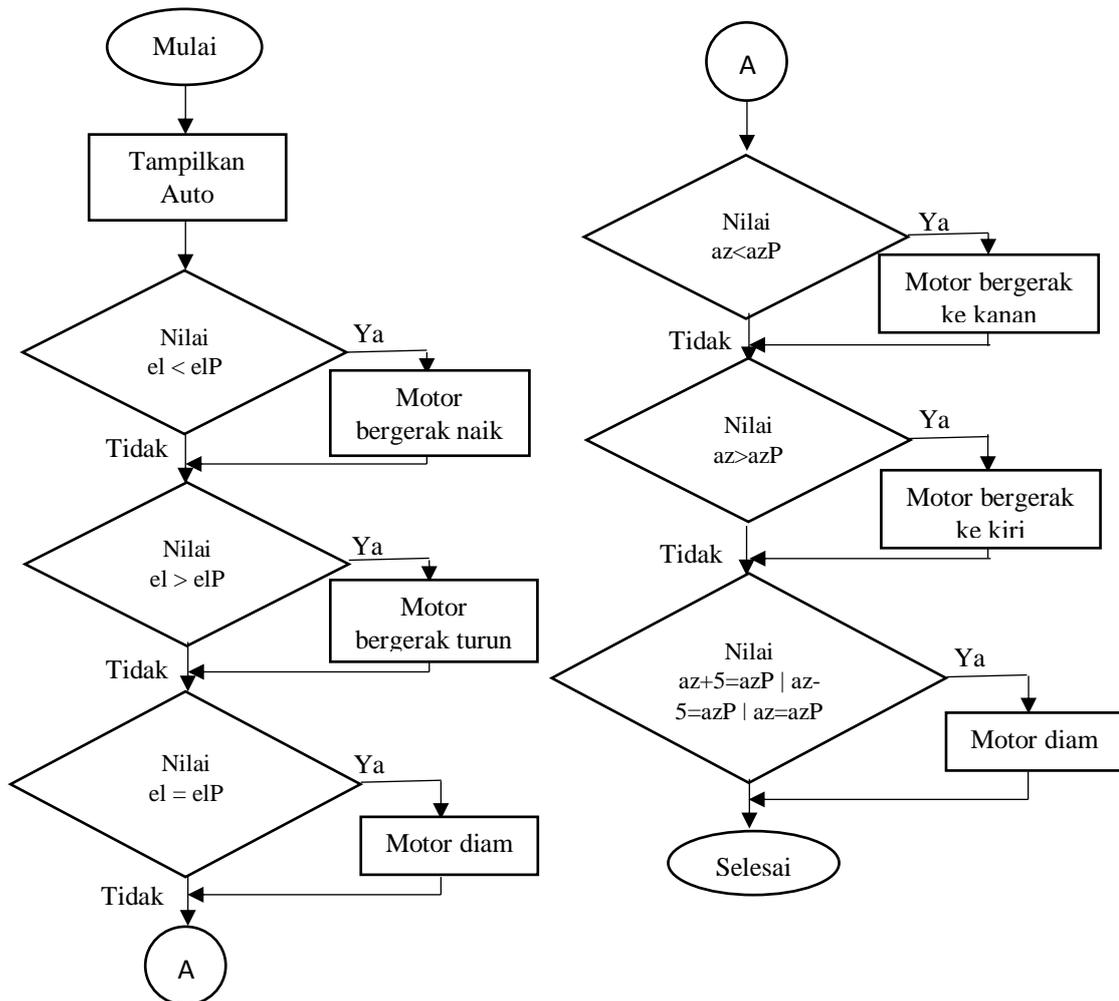
Gambar 3.11. Flowchart Program



Gambar 3.12. Flowchart Program Emerstop



Gambar 3.13. Flowchart Program Manual



Gambar 3.14. Flowchart Program Otomatis

Setelah bagian deklarasi awal dan bagian *setup* dari struktur bagian utama pada pemrograman dilakukan, masuk pada inti utama program dimana data posisi dari muatan *radiosonde* dan data dari antena diterima oleh *ground station* kemudian data yang telah masuk di pecah antara nilai azimut dan nilai elevasi sesuai dengan diagram alir yang telah disajikan. Pada program yang di tunjukan di tabel 3.2, terlihat terdapat dua prosedur program yang sama pada *void serialEvent1()* dan *void serialEvent()*, dimana program menunjukkan penerimaan data dari sensor dan PC. Ketika terjadi masukan, data langsung dieksekusi oleh program tersebut dimana pada *void serialEvent1()*, data yang diterima dari sensor masih berupa data gabungan antara nilai azimut dan elevasi, kemudian program segera memecah nilai

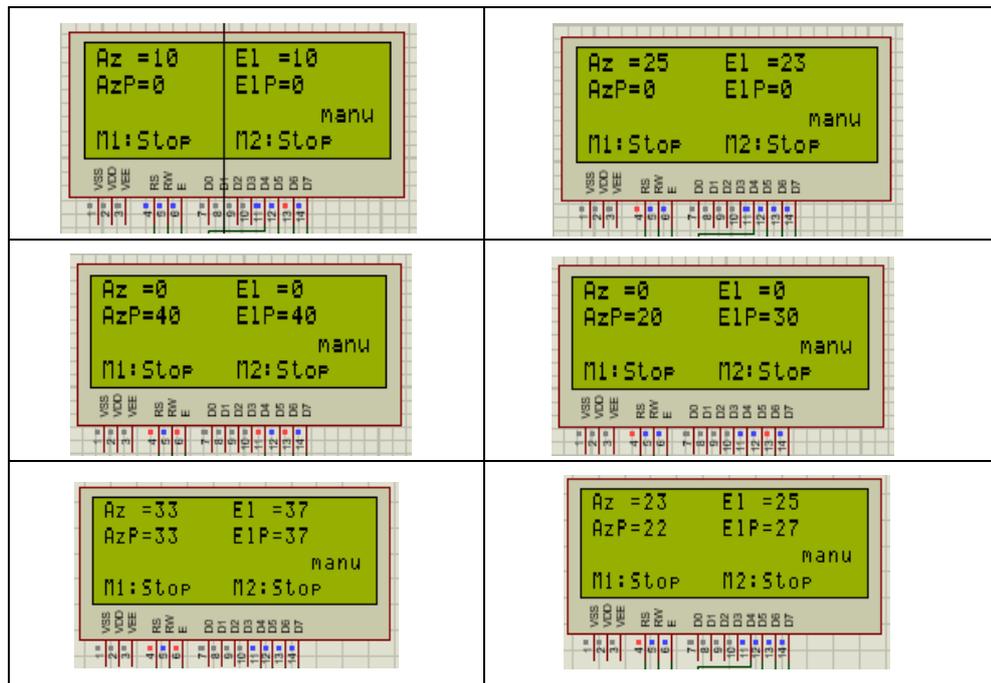
azimut dan elevasi sehingga dihasilkan 2 nilai keluaran. *Serial1.readStringUntil(',')*.toInt(); menyatakan pembacaan data azimut sampai tanda koma, sedangkan untuk *Serial1.readStringUntil('\n')*.toInt(); data elevasi dinyatakan pembacaan di mulai setelah tanda koma. Sama halnya dengan program sebelumnya, program *void serialEvent()* menerima data dari PC dimana nilai azimut dan elevasi juga di pecah.

Tabel 3.2. Tabel Program *Parsing* Data dari Sensor

95	void serialEvent1()
96	{
97	if (Serial1.available(>3)
98	{
99	az = Serial1.readStringUntil(',')
100	el = Serial1.readStringUntil('\n')
101	az = Serial1.readStringUntil(',')
102	el = Serial1.readStringUntil('\n')
103	}
104	}

Tabel 3.2. Tabel Program *Parsing* Data dari PC

105	void serialEvent()
106	{
107	if (Serial.available(>3)
108	{
109	azP = Serial.parseInt();
110	elP = Serial.parseInt();
111	}
112	}



Gambar 3.15. Hasil Program *Parsing* Data dari PC dan Sensor

Pada Gambar 3.15 menunjukkan hasil simulasi program arduino *parsing* data dengan menggunakan *software* proteus. Setelah data dipecah, kontroler akan membuat antenna berotasi sesuai dengan data yang diterima. Setelah itu kontroler membandingkan data input dari sensor dengan data input dari PC. Antena dapat digerakan secara manual maupun otomatis. Dapat dilihat pada tabel 3.3 merupakan program untuk menggerakkan antenna secara otomatis, ketika nilai perbandingan antara kedua inputan tidak sama atau dihasilkan *error*, maka motor akan berputar. Arah putaran motor ditunjukkan berdasarkan program yang telah dibuat. Pada *if*(*elNorm*<*elP*) *naik_vertikal()*; menunjukkan perubahan pergerakan motor bergerak naik atau ke atas, sedangkan *if*(*elNorm*>*elP*) *turun_vertikal()*; menunjukkan perubahan pergerakan motor turun atau ke bawah, dan ketika kedua nilai sama maka motor tidak akan terjadi perubahan atau diam. Berbeda dengan kondisi vertikal, pada kondisi horizontal ditambahkan nilai toleransi sudut yaitu $\pm 5^\circ$. Jika kondisi nilai inputan azimuth seperti pada program yaitu *if* ((*az* == *azP*) || (*az*+5 == *azP*) || (*az*-5 == *azP*)) {*diam_horizontal()*;*diam_horizontal()*;} maka kondisi motor

akan diam atau tidak terjadi perubahan. Selain dari itu, maka motor akan bergerak ke kanan dan kiri.

Tabel 3.3. Tabel Program Kontroler Antena Penjejak Otomatis

131	void otomatis()
132	{
133	lcd.setCursor(16,2);
134	lcd.print("auto");
139	if(elNorm<elP) naik_vertikal();
140	if(elNorm>elP) turun_vertikal();
141	if(elNorm == elP) diam_vertikal();
143	if ((az == azP) ((az+5 == azP) ((az-5 == azP)))
	{ diam_horizantal();diam_horizantal();}
144	else
145	{
146	if ((az > azP)&&(flag==1))
147	{
148	kiri_horizantal();flag=1;
149	if ((az == azP) ((az+5 == azP) ((az-5 == azP)))
	{ diam_horizantal();diam_horizantal();flag==0;}
150	}
151	if ((az < azP)&&(flag==1))
152	{
153	kanan_horizantal();flag=1;
154	if ((az == azP) ((az+5 == azP) ((az-5 == azP)))
	{ diam_horizantal();diam_horizantal();flag==0;}
155	}
156	}
157	}

Pada kondisi *void manual()*, antena digerakan secara manual dengan menggunakan *joystick* sehingga pemrograman dibuat seperti yang ditunjukkan pada tabel 3.4. Penggunaan *joystick* ini perlu pengamatan dari LCD mengenai arah putaran motor dan perbedaan sudut antara sensor dan PC, agar pergerakan motor antena menuju *set point* keakurasian sudutnya terpenuhi.

Tabel 3.4. Tabel Program Kontroler Antena Penjejak Manual

158	void manual()
159	{
160	lcd.setCursor(16,2);
161	lcd.print("manu");
162	vry = analogRead(A0);
163	vrx = analogRead(A1);
164	if(vry>750) turun_vertikal();
165	if(vry<350) naik_vertikal();
166	if(vry<750&&vry>350) diam_vertikal();
168	if(vrx>750) kiri_horizontal();
169	if(vrx<350) kanan_horizontal();
170	if(vrx<750&&vrx>350) diam_horizontal();
172	}

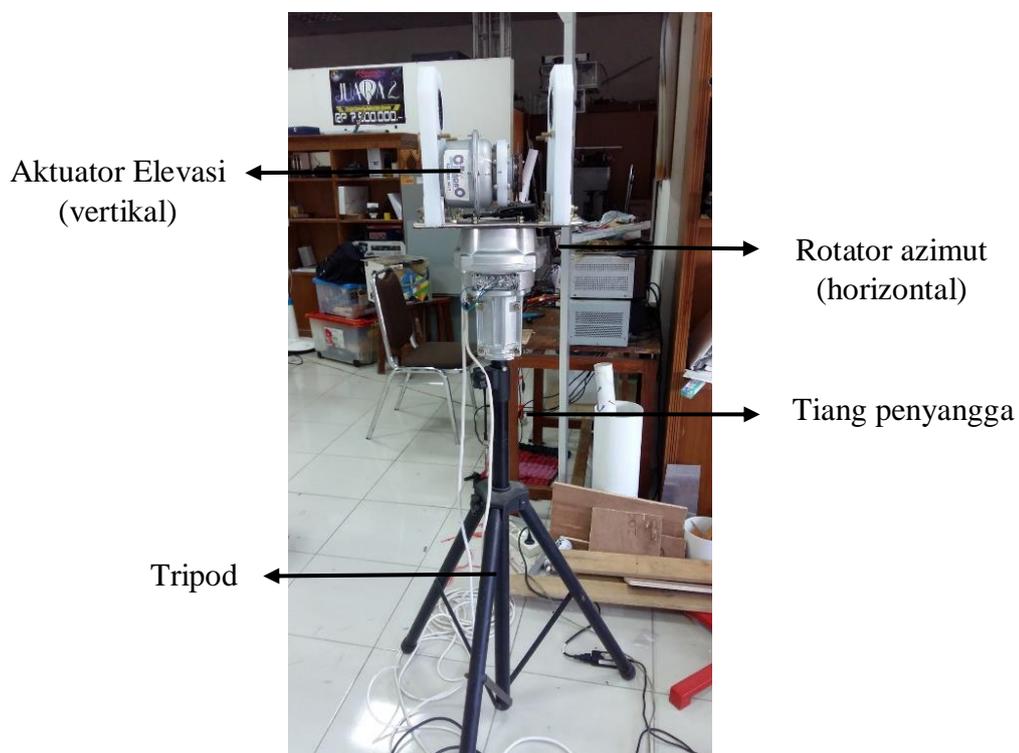
Tabel 3.5. Tabel Program Stop Kontroler Antena

116	void emerstop()
117	{
118	lcd.setCursor(16,2);
119	lcd.print("stop");
120	digitalWrite(2, LOW);
121	digitalWrite(3, LOW);
122	digitalWrite(4, LOW);
123	digitalWrite(5, LOW);
124	digitalWrite(51, LOW);
125	digitalWrite(53, LOW);
126	lcd.setCursor(0,3);
127	lcd.print("M1:Stop ");
128	lcd.setCursor(10,3);
129	lcd.print("M2:Stop ");
130	}

Pada program yang ditunjukkan pada tabel 3.5, semua pergerakan motor baik horizontal maupun vertikal akan terhenti dengan program *void emerstop()* baik saat keadaan motor sedang berotasi.

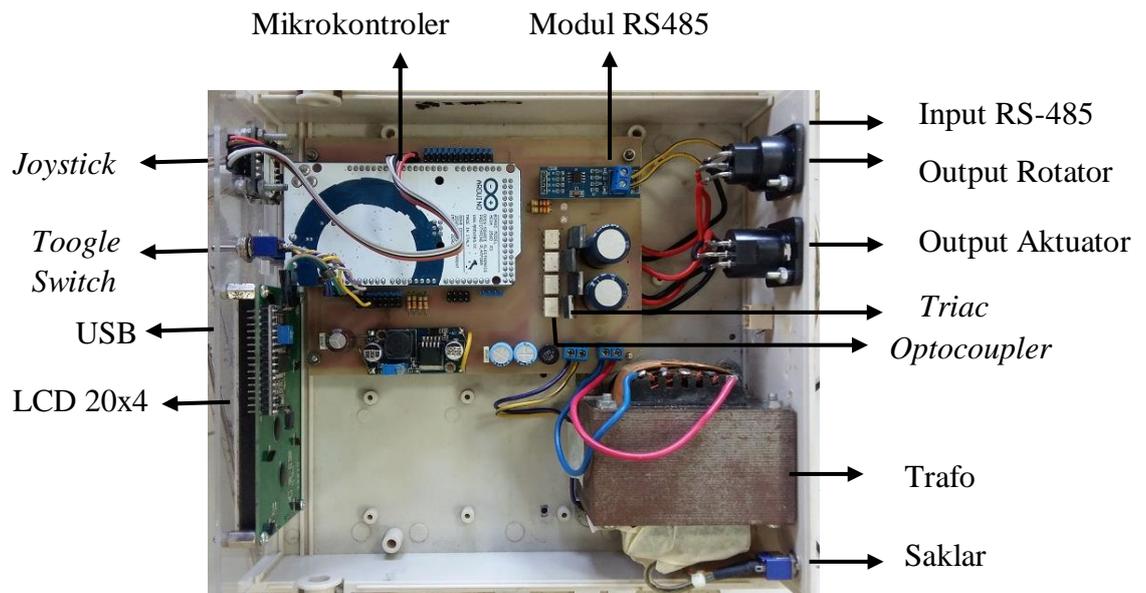
3.5. Rancangan Mekanik

Rancangan mekanik adalah rancangan desain sekaligus kerangka penggerak yang memiliki 2 derajat kebebasan yaitu dapat menyebabkan antena bergerak secara horizontal dan vertikal. Pada kerangka tersebut terdapat komponen pembangun untuk sistem penggerak seperti tiang penyangga antena, *tripod*, kabel, rotator, aktuator, dan antena yagi.



Gambar 3.16. Rangkaian Mekanik Antena Penjejak
(Sumber Pribadi)

Pada desain tersebut dapat dilihat rotator dan aktuator di topang oleh tiang penyangga dan *tripod*, selain itu tiang penyangga diusahakan dapat menyangga beban motor dan antena agar saat antena bergerak mengikuti pergerakan muatan, antena dalam kondisi pergerakan yang stabil. Kabel-kabel yang terpasang digunakan untuk menghubungkan antara rotator dan aktuator dengan rangkaian kontroler.



Gambar 3.17. Rangkaian Kontroler Antena Penjejak



Gambar 3.18. Kontroler Antena Penjejak, (a) tampak depan (b) tampak belakang

Dapat dilihat pada Gambar 3.17. terlihat bagian tampak atas yang berisi rangkaian elektronika dari kontroler antena penjejak. Sedangkan pada bagian kontroler sesuai yang ditunjukkan Gambar 3.18. (a) , terlihat bagian muka alat yang pada bagian tersebut terdapat *lcd* yang berukuran 20 x 4 karakter yang fungsinya untuk menampilkan data-data yang dikirimkan dari muatan dan sensor, yang diolah berdasarkan hasil perhitungan logika mikrokontroler. Selain itu pada sisi depan terdapat *joystick* yang digunakan untuk melakukan pergerakan terhadap motor secara manual dan terdapat *toggle switch* yang digunakan untuk mengatur kondisi pergerakan motor baik dilakukan secara manual, otomatis maupun berhenti. Pada Gambar 3.18 (b) merupakan sisi belakang pada bagian kontroler, yang disana terdapat saklar untuk catu daya untuk rangkaian elektronika dan motor 1 fasa. Selain itu terdapat satu masukan yang kecil yang terdapat di sisi kanan bawah,

digunakan untuk jalur RS-485 untuk menghubungkan antara kontroler dengan sistem *feedback*. Dan yang terakhir terdapat dua keluaran besar yang digunakan untuk kabel penghubung antara kontroler dan motor.

3.6. Tahapan Pengujian

Pada tahap pengujian dilakukan pengujian terhadap masing-masing alat yang telah tersedia dan telah dirancang.

3.6.1. Pengujian *Joystick* dengan Arduino

Pada pengujian ini menggunakan program sederhana untuk menguji apakah *joystick* dapat berkerja sesuai fungsionalnya. Pengujian ini dilakukan untuk mengetahui posisi harizontal dan vertikal yang terdapat pada *joystick*, karena *joystick* sendiri diketahui memiliki 2 axis yaitu axis x dan axis y. Untuk mengetahui posisi tersebut dapat dilihat dengan tampilan nilai ADC.

1. Pastikan seluruh alat dalam kondisi baik.
2. Pastikan *joystick* telah terhubung ke arduino.
 - GND – To Arduino GND
 - VCC – To Arduino 5V
 - X – To Arduino Analog In (A0)
 - Y – To Arduino Analog In (A1)
 - Key – To Arduino Digital In (A2)
3. Hubungkan mikrokontroler ke PC dengan menggunakan USB.
4. Buka aplikasi *Arduino IDE*.
5. Masukkan program untuk pengujian *joystick*.
6. *Verify* program, bila tidak terjadi kesalahan pada program maka akan menunjukkan *done compiling*.
7. Klik tombol *upload*, untuk mengunggah *sketch* yang telah dibuat dan dikompilasi *Arduino*.
8. Dengan bantuan LCD nilai ADC akan ditampilkan, selain itu dapat diamati lewat serial monitor yang terdapat pada *Arduino IDE*.
9. Gerakan *joystick* ke segala arah dan nilai *adc* akan berubah.
10. Amati, lalu analisis sesuai dengan hasil pengamatan.

3.6.2. Pengujian LCD dengan Arduino

Pengujian LCD dengan menggunakan *Arduino* dilakukan untuk mengetahui fungsi dari LCD yaitu sebagai media tampil dimana untuk menampilkan status kerja alat.

1. Pastikan seluruh alat dalam kondisi baik.
2. Pastikan LCD telah terhubung ke arduino.
GND – To Arduino GND
VCC – To Arduino 5V
SDA – To Arduino SDA (pin 20)
SCL – To Arduino SCL (pin 21)
3. Karena LCD yang digunakan menggunakan protokol pengiriman data dengan I2C, maka kita harus memiliki alamat tersebut dengan cara *men-download* di internet.
4. Hubungkan mikrokontroler ke PC dengan menggunakan USB.
5. Buka aplikasi *Arduino IDE*.
6. Masukkan program untuk pengujian LCD.
7. *Verify* program, bila tidak terjadi kesalahan pada program maka akan menunjukkan *done compiling*.
8. Klik tombol *upload*, untuk mengunggah *sketch* yang telah dibuat dan dikompilasi *Arduino*.
9. Kecerahan LCD juga dapat diatur melalui komponen *trimpot*.
10. Amati, lalu analisis sesuai dengan hasil pengamatan. Apakah LCD bekerja sesuai normal sesuai dengan yang diinginkan.

3.6.3. Pengujian Motor AC

Pengujian ini dilakukan untuk memastikan bahwa motor dapat bergerak sesuai dengan yang diinginkan ketika dihubungkan oleh kontroler.

Langkah pengujian motor :

1. Pastikan semua peralatan dalam kondisi baik.
2. Hubungkan kontroler dengan motor (azimut dan elevasi) dengan menggunakan kabel.

3. Hubungkan posokan daya untuk kontroler dengan saklar.
4. Tekan tombol *toggle* yang terdapat di belakang kontroler dalam posisi ON
5. Hubungkan PC dengan kontroler menggunakan USB
6. Buka aplikasi *Arduino IDE*.
7. Masukkan program untuk pengujian Motor.
8. *Verify* program, bila tidak terjadi kesalahan pada program maka akan menunjukkan *done compiling*.
9. Klik tombol *upload*, untuk mengunggah *sketch* yang telah dibuat dan dikompilasi *Arduino*.
10. Atur kondisi motor pada tombol *toggle* yang terdapat di bagian muka kontroler dalam keadaan manual.
11. Pengujian dilakukan dengan bantuan *joystick*.
12. Gerakan *joystick* ke kanan dan ke kiri untuk mengetahui kondisi motor azimuth.
13. Gerakan *joystick* ke atas dan ke bawah untuk mengetahui kondisi motor elevasi.
14. Amati pergerakan motor apakah sesuai dengan pengujian yang diinginkan lalu analisis. Catat waktu putaran motor pada saat berotasi secara penuh.

3.6.4. Pengujian Keseluruhan Sistem

Pengujian keseluruhan sistem disini adalah pengujian yang dilakukan terhadap penggerak antenna. Pengujian ini dilakukan dengan 2 tahap pengujian yaitu pengujian motor azimuth dengan sensor kompas dan pengujian motor elevasi dengan sensor sudut. Selain itu pengujian ini juga dapat mengetahui keakurasi pergerakan antenna.

Langkah pengujian yang dilakukan :

1. Pastikan semua peralatan dalam kondisi baik.
2. Hubungkan kontroler dengan motor (azimuth dan elevasi) dengan menggunakan kabel.
3. Hubungkan kontroler dengan sistem *feedback* menggunakan RS-485.

4. Hubungkan posokan daya untuk kontroler dengan saklar.
5. Atur kondisi sensor kompas dan sensor sudut pada keadaan 0°
6. Untuk motor azimut, posisi titik mati disamakan dengan titik 0° pada sensor kompas dan menjadi titik mula pergerakan motor.
7. Tekan tombol *toggle* yang terdapat di belakang kontroler dalam posisi ON
8. Hubungkan PC dengan kontroler menggunakan USB
9. Buka aplikasi *Arduino IDE*
10. Masukkan *listing* program untuk pengujian antena *tracker* baik pada sistem kontroler maupun sistem *feedback*
11. *Verify* program, bila tidak terjadi kesalahan pada program maka akan menunjukkan *done compiling*
12. Klik tombol *upload* pada *software Arduino IDE*
13. Pengujian dilakukan dengan memasukkan inputan berupa sudut sebagai pengganti *set point* muatan.
14. Buka serial monitor pada *Arduino IDE*
15. Pada saat pengujian dilakukan pada motor azimut, nilai elevasi disamakan dengan nilai yang terbaca pada sensor sudut agar motor elevasi tetap dalam kondisi diam. Begitu juga sebaliknya pada saat pengujian motor elevasi, nilai azimut disamakan dengan nilai yang terbaca pada sensor kompas agar motor azimut tetap dalam kondisi diam.
16. Masukkan *input*-an sesuai dengan tabel percobaan yang telah dibuat.
17. Apabila nilai *set point* azimut dan elevasi tidak sama dengan pembacaan sensor, maka motor akan bergerak. Amati pergerakan motor, apakah putaran motor sama dengan kondisi yang telah diberikan oleh program.
18. Bandingkan sudut yang dibentuk oleh motor dengan *set point* yang telah diberikan. Amati percobaan tersebut selanjutnya analisis.
19. Catat waktu putaran motor pada saat berotasi ketika menempuh sudut *set point* yang diberikan.