

BAB III

KONSEP PERANCANGAN

Dalam perancangan Alat *Prototype Autopilot Pada Mobil Menggunakan Kendali GPS*, secara urut dimulai dari identifikasi kebutuhan yang diperlukan.

A. Identifikasi Kebutuhan

Dalam merealisasikan pembuatan alat ini diperlukan kebutuhan, adapun beberapa kebutuhan untuk dapat menunjang pengerjaan proyek akhir ini. Adapun identifikasi bagian-bagian yang dibutuhkan sebagai berikut :

1. Baterai lippo untuk mensuplai tegangan agar bekerja dengan baik.
2. Mikrokontroller sebagai sistem pengendali yang mengendalikan sistem *autopilot* pada prototipe.
3. Mikrokontroller sebagai sistem pengendali yang mengendalikan sensor jarak dan relay.
4. *Driver* untuk menggerakkan motor DC.
5. Motor DC sebagai *actuator* gerak *Prototype Autopilot*.
6. Motor Servo sebagai bagian untuk membelok alat kearah kanan dan kearah kiri.
7. Sensor yang dapat membaca jarak benda.
8. Relay sebagai kontrol *ESC*.
9. *GPS* sebagai alat untuk menentukan posisi dari satelit
10. Pengendalian jarak jauh untuk mengendalikan *Prototype Autopilot*.
11. Kotak Persegi Panjang sebagai tempat dudukan *Prototype Autopilot* untuk memudahkan dalam menyetting *Prototype Autopilot*.

12. *Software* untuk membantu penulisan program dengan bahasa pemrograman yang sesuai.
13. *Software* untuk membantu menampilkan navigasi dari *GPS*.

B. Analisa Kebutuhan

Berdasarkan identifikasi kebutuhan diatas, maka diperoleh beberapa analisa kebutuhan pada sistem yang akan dirancang, sebagai berikut :

1. Baterai lippo

Baterai lippo yang digunakan baterai TURNIGY 11.1V, 1500 mAh, baterai TURNIGY 7,4V, 850 mAh dan 1 buah modul UBEC 5V. Baterai TURNIGY 11.1V, 1500 mAh digunakan untuk mensuplai Arduino Mega 2560 dan Arduino Nano. Baterai TURNIGY 7,4V, 850 mAh digunakan untuk mensupali ESC. Modul UBEC 5V/6V digunakan untuk menurunkan tegangan 11.1 V menjadi 6 V ke Arduino Mega.

2. Mikrokontroller Arduino Mega 2560

Mikrokontroller Arduino Mega 2560 sebagai sistem pengendali yang mengendalikan sistem *autopilot* pada prototipe. Arduino Mega 2560 merupakan mikrokontroller yang berbasis Arduino dengan menggunakan *chip* ATmega2560. Arduino Mega 2560 dipilih sebagai pengendali utama karena mempunyai memori sebesar 256KB dan memiliki sebuah *port* USB, dan tombol *reset*.

3. Mikrokontroller Arduino Nano

Mikrokontroller Arduino Nano sebagai sistem pengendali yang mengendalikan sensor jarak dan relay. Arduino Nano merupakan mikrokontroller yang berbasis Arduino dengan menggunakan *chip* ATmega328. Arduino Nano

dipilih sebagai pengendali utama karena ukuran pcb yang kecil dibandingkan arduino lain dan memiliki konfigurasi 14 pin I/O (*input/output*) digital, sebuah *port* USB, dan tombol *reset*.

4. ESC

Driver untuk menggerakkan motor DC. RC ESC 20A Brush Motor Speed Controller 1/18 scale RC Car dipilih karena tegangan 3.0V-9.4V, *input* Li-Po 2S, dan fungsinya maju, mundur dan rem.

5. Motor DC

Motor DC sebagai *actuator* gerak *Prototype Autopilot*. Motor DC menggunakan Motor DC dari mobil RC Rock Rover 1 : 18 *scale*. Mobil RC yang digunakan hanyalah kerangka bodinya dan 2 motor DC. Tegangan motor DCnya sendiri 5V sesuai dengan tegangan pada *ESC*.

6. Motor Servo

Sebagai bagian untuk belok kanan atau kiri, untuk jalannya *Prototype Autopilot*. Tower pro MG-90 Metal Servo dipilih karena dapat menahan beban sampai dengan 2.4kg/cm (6V), dan tegangan yang digunakan 4.8 sampai 6 Volt.

7. Sensor ultrasonik

Sensor ultrasonik digunakan untuk mendeteksi benda dengan jarak 20cm. Sensor yang digunakan HCSR04 dan US-015 dikarenakan sensor ini dapat mendeteksi dari 2cm hingga 400cm.

8. Relay

Relay digunakan untuk kontrol *ESC* pada saat sensor ultrasonik mendeteksi benda kurang dari 20 cm maka *ESC* akan mati, bila mendeteksi benda lebih dari

20cm maka *ESC* akan hidup. Relay ini dipilih karena kesesuaiannya sebagai saklar yang bisa otomatis dikendalikan oleh arduino, dan membutuhkan 5V.

9. *GPS* Neo 7

GPS sebagai alat untuk menentukan posisi dari satelit. *GPS* berfungsi sebagai *penerima GPS (Global Positioning System Receiver)* yang dapat mendeteksi lokasi dengan menangkap dan memproses sinyal dari satelit navigasi. *GPS* Neo 7 sendiri dipilih karena tegangannya 5v, Akurasi penetapan lokasi *GPS* secara horisontal: 2,5 meter (SBAS = 2m).

10. 3DR Radio V2

3DR Radio V2 merupakan perangkat radio V2 telemetri yang memungkinkan untuk menghubungkan ke pengontrol penerbangan seperti APM Pixhawk, atau multiwii AIO, perangkat ini juga dilengkapi USB atau UART seperti komputer, laptop, tablet, atau android yang mendukung koneksi USB (OTG).

11. Kotak Persegi Panjang akrilik ukuran panjang 10 cm, lebar 8 cm dan tinggi 9,6 cm.

12. *Software* Arduino IDE digunakan untuk membantu penulisan program dengan bahasa C.

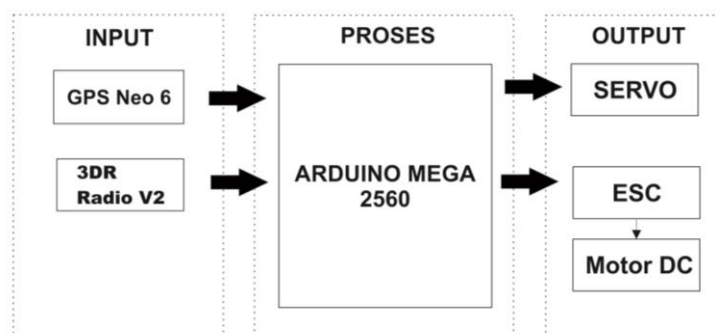
13. *Mission Planner* untuk membantu menampilkan navigasi dari *GPS*.

C. Perancangan Alat

Perancangan *Prototype Autopilot Pada Mobil Menggunakan Kendali GPS* terdiri dari *hardware* dan *software*.

1. Rancangan Blok Diagram *Microcontroller* Arduino Mega 2560

Rangkaian Blok diagram *Prototype Autopilot Pada Mobil Menggunakan Kendali GPS* dapat dilihat pada gambar dibawah ini.



Gambar 15. Diagram Blok Sistem Arduino Mega 2560

Sebelum menuju pada bagian analisis terdapat sebuah gambar 15 diagram blok sistem dari alat. Diagram tersebut terdiri dari *input*, *proses*, dan *output*. Pada bagian *input* dibutuhkan beberapa komponen yaitu sensor. Bagian *input* berguna untuk memberi masukan dan data yang akan diolah pada proses. Pada bagian proses diperlukan sebuah komponen yaitu Arduino Mega 2560. Pada bagian proses berguna untuk mengendalikan seluruh kinerja sistem alat. Selanjutnya bagian *output* yang terdiri dari Motor DC, *ESC*, Servo dan 3DR Radio V2.

Penjelasan bagian-bagian blok pada gambar 15 diatas sebagai berikut :

a. Blok *Input*

Block *input* terdiri dari *GPS Neo 7*, *3DR Radio V2*. *GPS* digunakan untuk menentukan posisi dari satelit. *3DR Radio V2* digunakan perangkat radio *V2 Telemetry* yang memungkinkan untuk menghubungkan ke pengontrol penerbangan.

b. Blok Proses

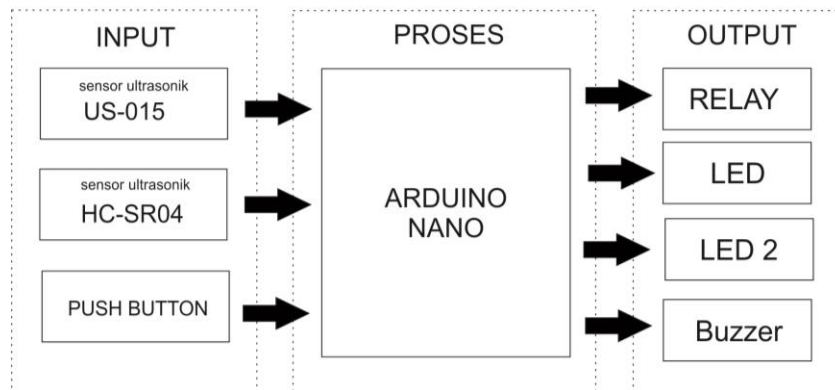
Block proses merupakan bagian untuk mengendalikan seluruh kinerja sistem alat. *Arduino Mega 2560* merupakan mikrokontroler yang berbasis *Arduino* dengan menggunakan chip *ATmega2560*. *Arduino Mega 2560* mempunyai memori sebesar *256KB* dan memiliki sebuah *port USB*, dan tombol *reset*.

c. Blok *Output*

Block *output* terdiri dari *Motor DC*, *ESC*, dan *Servo*. *Motor DC* digunakan sebagai *actuator* gerak *Prototype Autopilot*. *ESC* digunakan untuk menggerakkan motor *DC*. *RC ESC 20A Brush Motor Speed Controller 1/18 Scale RC Car*, fungsinya maju, mundur dan rem. *Servo* digunakan sebagai bagian untuk membelokan alat kearah kanan dan kearah kiri.

2. Rancangan Blok Diagram *Microcontroller* Arduino Nano

Rangkaian Blok diagram *Prototype Autopilot Pada Mobil Menggunakan Kendali GPS* dapat dilihat pada gambar dibawah ini.



Gambar 16. Diagram Blok Sistem Arduino Nano

Sebelum menuju pada bagian analisis terdapat sebuah gambar 16 diagram blok sistem dari alat. Diagram tersebut terdiri dari *input*, *proses*, dan *output*. Pada bagian *input* dibutuhkan beberapa komponen yaitu sensor. Bagian *input* berguna untuk memberi masukan dan data yang akan diolah pada proses. Pada bagian proses diperlukan sebuah komponen yaitu Arduino Nano. Pada bagian proses berguna untuk mengendalikan seluruh kinerja sistem alat. Selanjutnya bagian *output* yang terdiri dari relay, LED, LED 2 dan buzzer.

Penjelasan bagian-bagian blok pada gambar 16 diatas sebagai berikut :

a. Blok *Input*

Block *input* terdiri dari 2 sensor ultrasonik , dan 1 *push button*. Sensor ultrasonik digunakan untuk mendeteksi benda dengan jarak 20cm, jika terdeteksi kurang dari jarak tersebut maka relay akan memutus arus *ESC*. *Push button* sebagai

tombol, digunakan jika sensor ultrasonik mendeteksi benda kurang dari jarak 20 cm, berfungsi untuk menghidupkan relay sehingga relay menyalakan *ESC*.

b. Blok Proses

Block proses merupakan bagian untuk mengendalikan seluruh kinerja sistem alat. Mikrokontroller Arduino Nano sebagai sistem pengendali yang mengendalikan sensor jarak dan relay. Arduino Nano merupakan mikrokontroller yang berbasis Arduino dengan menggunakan chip ATmega328. Arduino Nano memiliki konfigurasi 14 pin I/O (*input/output*) digital, sebuah *port* USB, dan tombol *reset*.

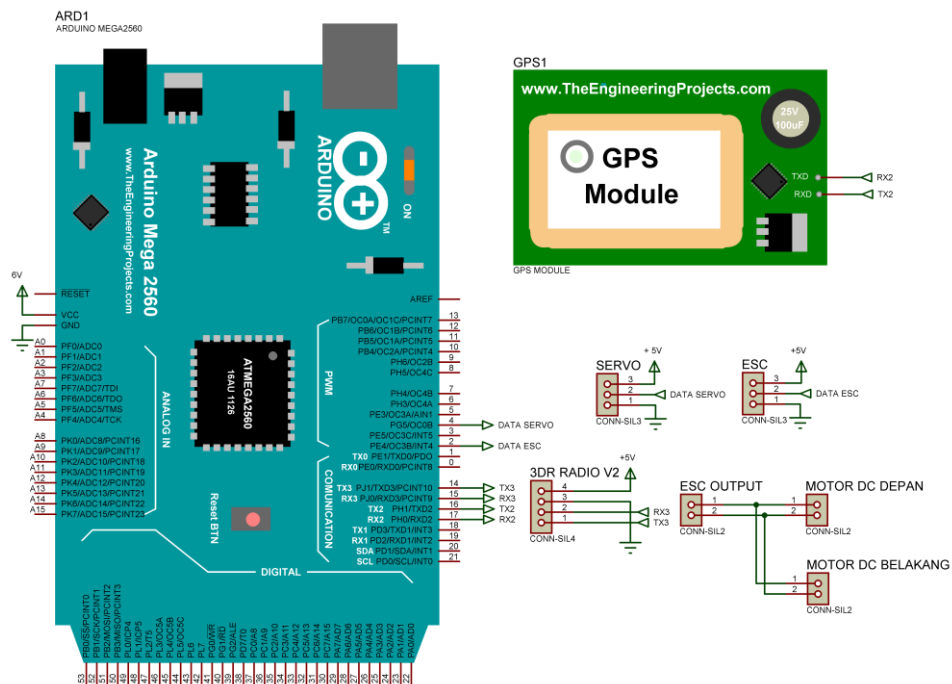
c. Blok *Output*

Block *output* terdiri dari relay, LED, LED 2 dan buzzer. Relay digunakan untuk kontrol ESC pada saat sensor ultrasonik mendeteksi benda kurang dari 20 cm maka ESC akan mati, bila mendeteksi benda lebih dari 20cm maka ESC akan hidup. LED sebagai indikator untuk mengetahui bila sensor ultrasonik mendeteksi benda kurang dari 20 cm maka LED akan menyala. LED 2 berfungsi sebagai penerangan pada malam hari. Buzzer digunakan sebagai indikator suara, berbunyi bila sensor ultasonik mendeteksi benda kurang dari 20 cm.

3. Hardware

Dalam pembuatan *Prototype Autopilot Pada Mobil Menggunakan Kendali GPS* memerlukan perancangan komponen elektronika sebagai pendukung fungsi dari pengendalian alat sebagai berikut:

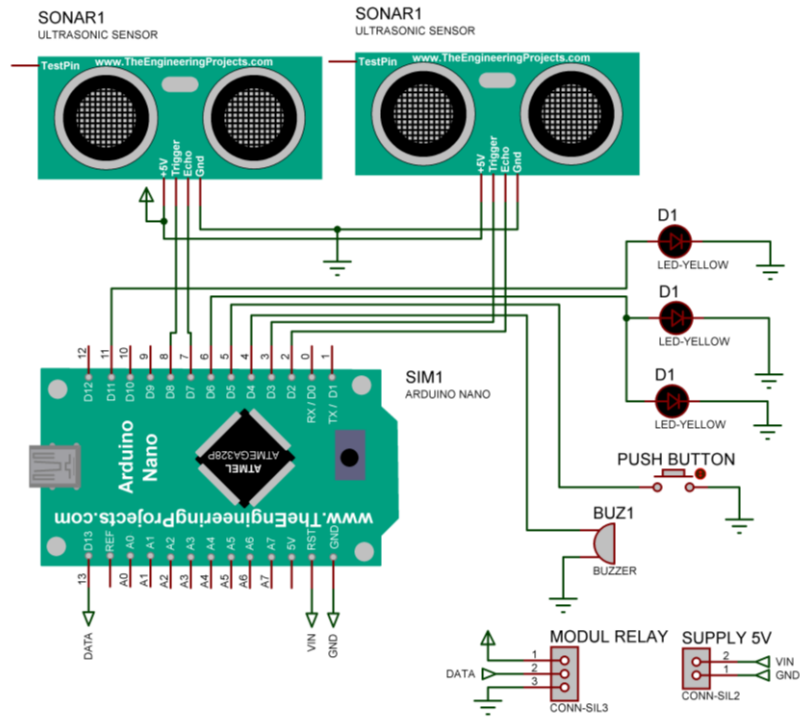
a. Rangkaian keseluruhan *Microcontroller* Arduino Mega 2560



Gambar 17. Rangkaian keseluruhan *Microcontroller* Arduino Mega 2560

Gambar 17 merupakan gambar rangkaian keseluruhan yang berupa konfigurasi dari komponen terhadap Arduino Mega 2560. Pin 4 disambungkan ke servo. Pin 2 disambungkan ke ESC. Pin 16,17 disambungkan ke *GPS* Neo 7. Pin 14,15 disambungkan ke 3D Radio V2. ESC *output* disambungkan ke motor DC depan dan motor DC belakang.

b. Rangkaian keseluruhan *Microcontroller* Arduino Nano



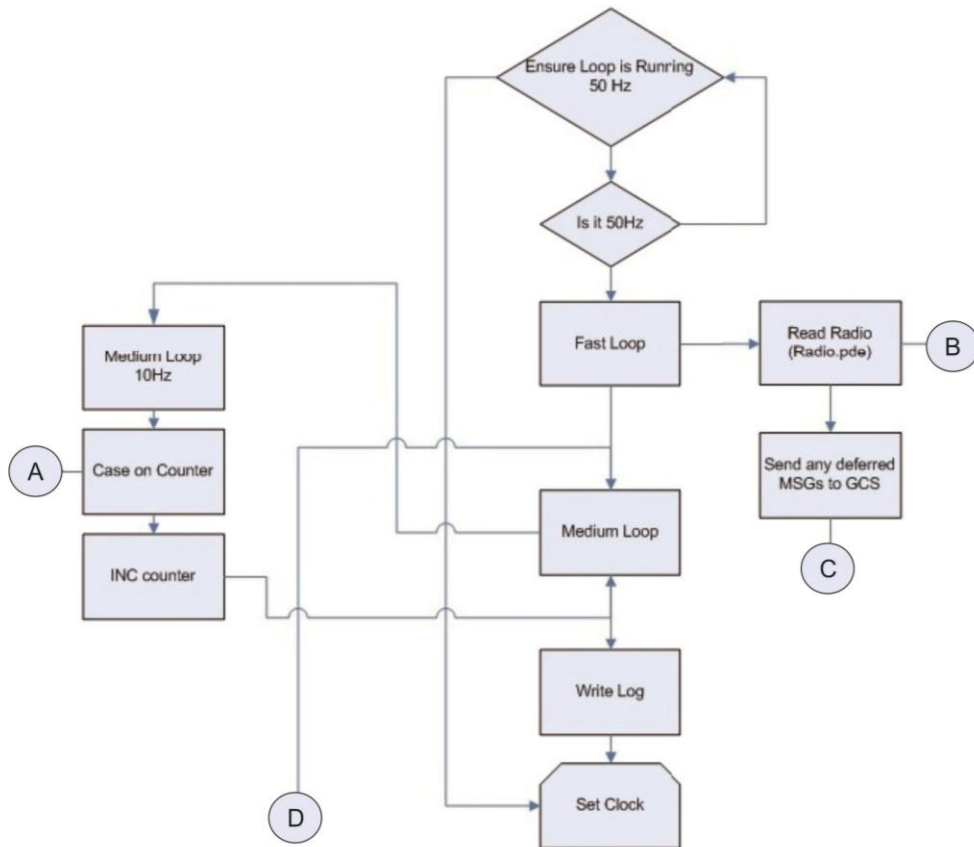
Gambar 18. Rangkaian keseluruhan *Microcontroller* Arduino Nano

Gambar 18 merupakan gambar rangkaian keseluruhan yang berupa konfigurasi dari komponen terhadap Arduino Nano. Pada pin 2,3 dan 7,8 disambungkan ke sensor ultasonik. Pin 11 disambungkan Ke LED. Pin 6 disambungkan ke 2 LED. Pin 5 disambungkan ke *Push Button*. Pin 4 disambungkan ke *Buzzer*. Pin 13 disambungkan ke Relay.

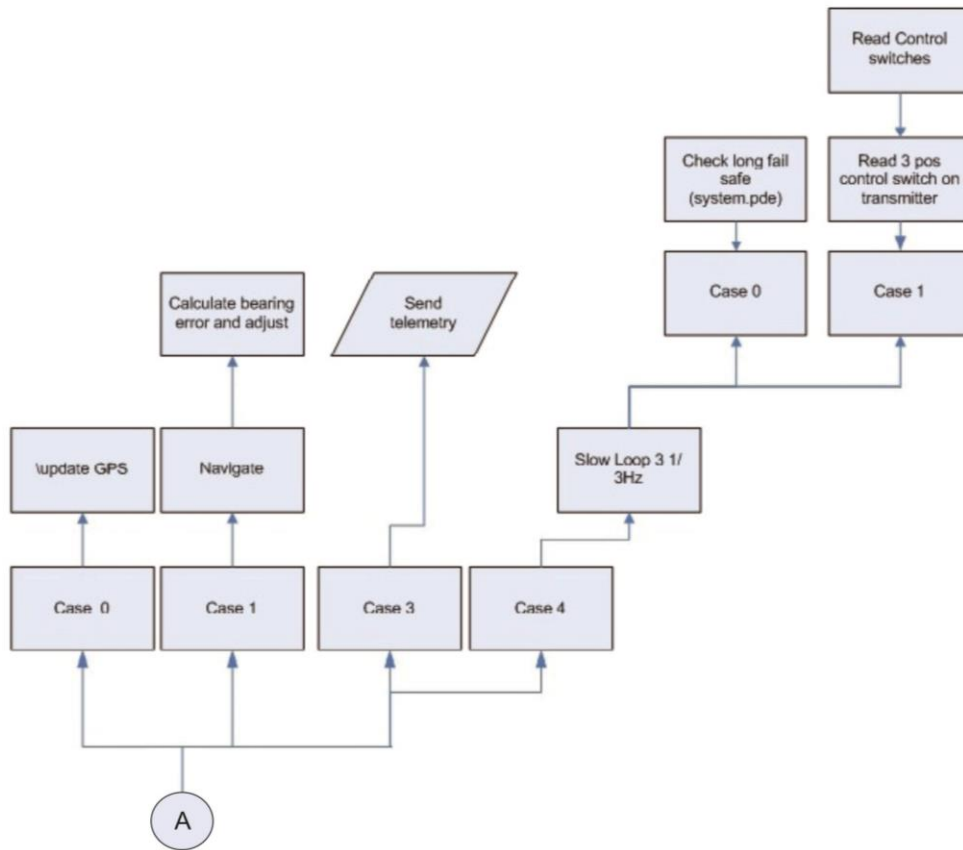
4. Software

a. Diagram alur (*Flowchart*)

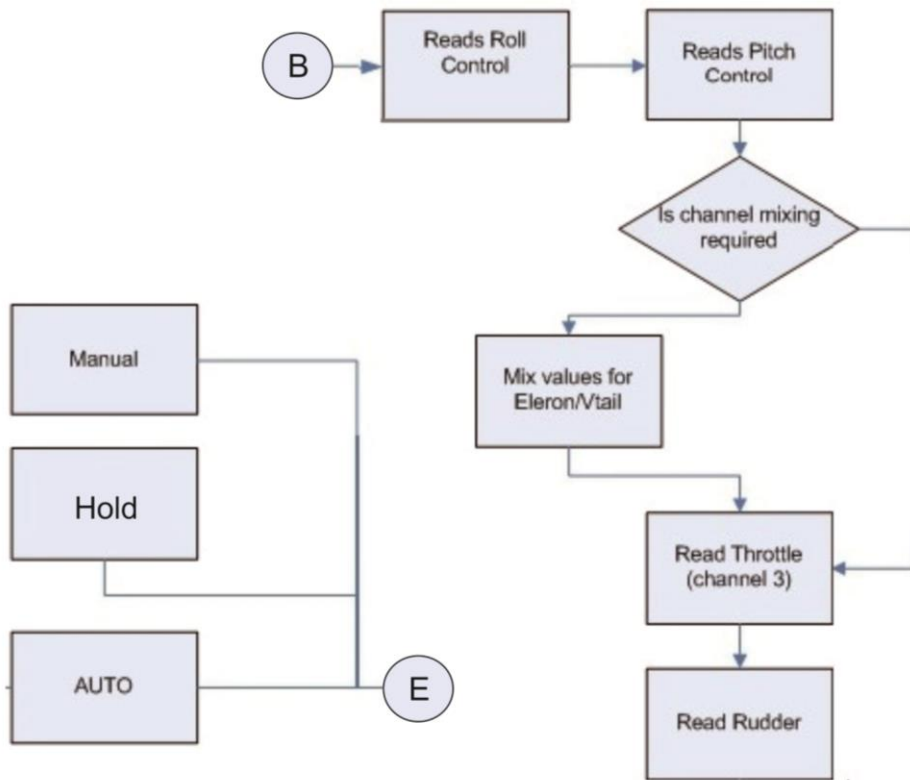
Perancangan *software* perangkat lunak ini, membutuhkan sistematika pembuatan yang baik. Berikut ini merupakan *flowchart*:



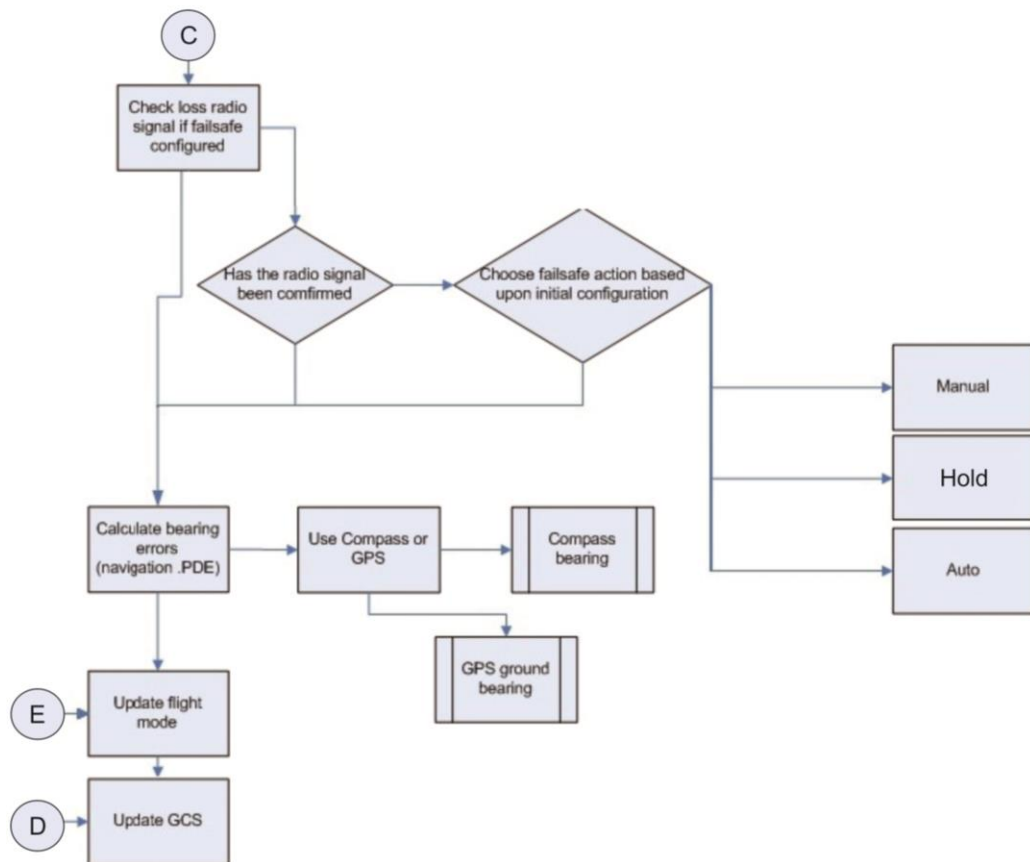
Gambar 19. Flowchart Program Prototype Autopilot 1



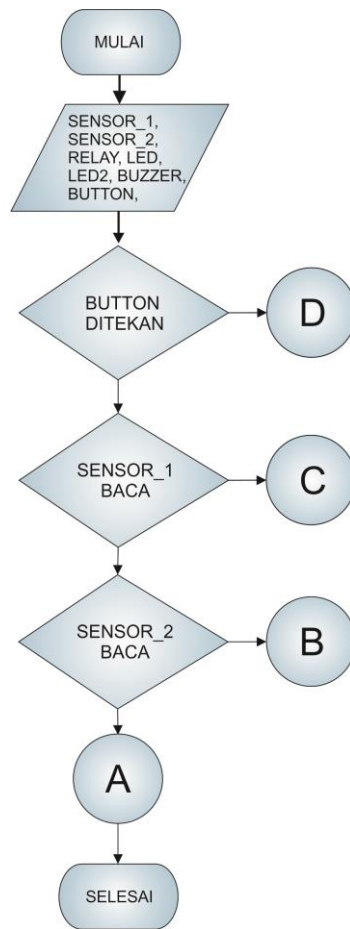
Gambar 20. Flowchart Program Prototype Autopilot 2



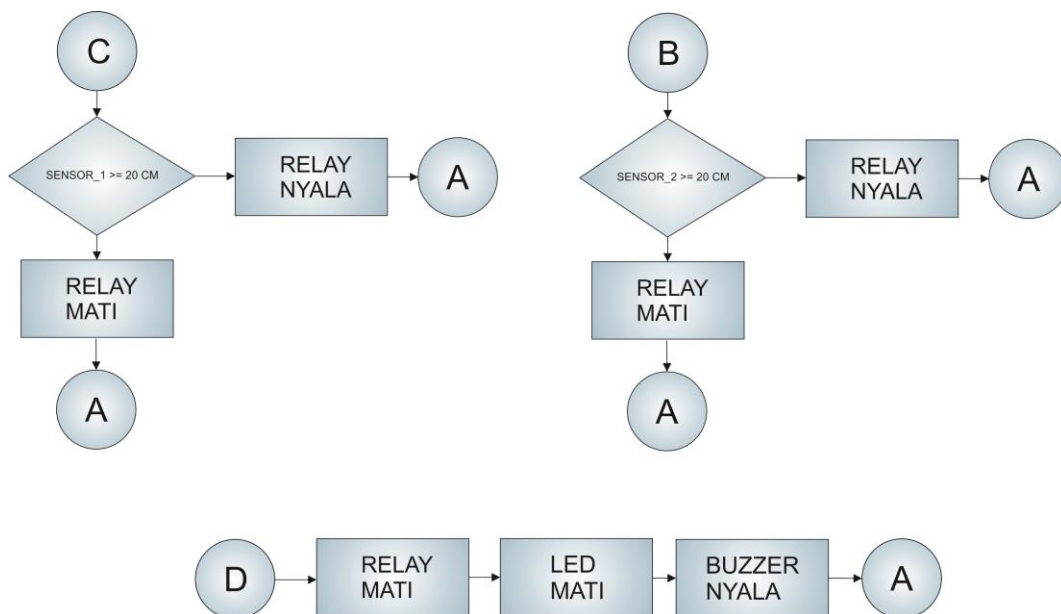
Gambar 21. Flowchart Program Prototype Autopilot 3



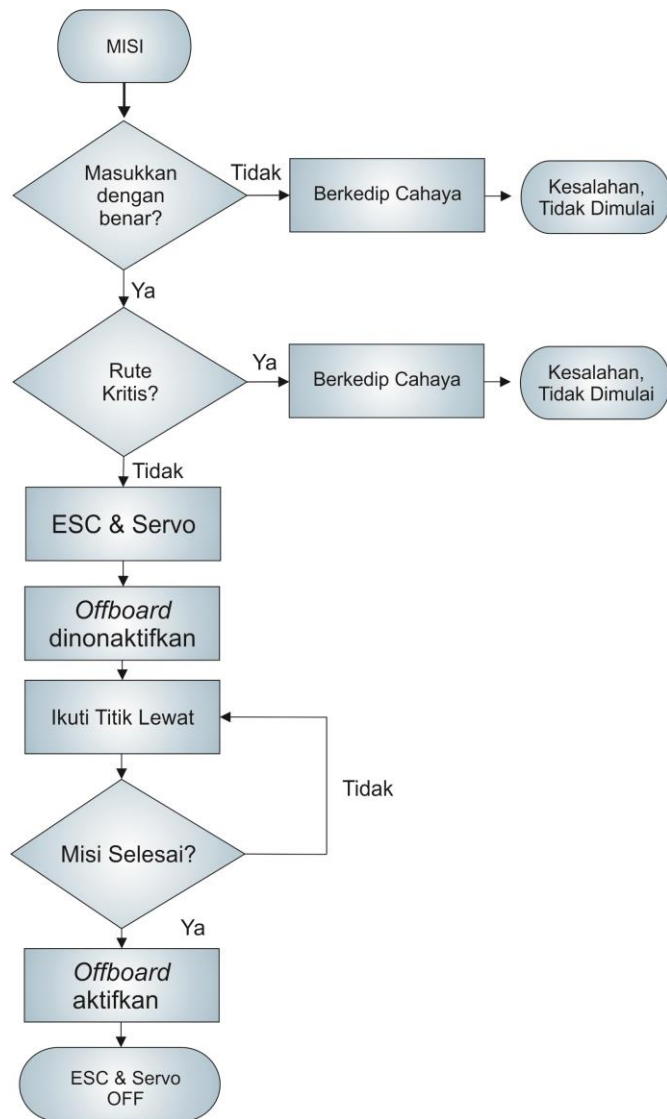
Gambar 22. Flowchart Program Prototype Autopilot 4



Gambar 23. Flowchart Program Sensor Anti Nabrak 1



Gambar 24. Flowchart Program Sensor Anti Nabrak 2



Gambar 25. Flowchart Program dari Mission Planner

b. Alat dan bahan

1) Alat

Kebutuhan alat yang diperlukan dalam pembuatan alat seperti pada tabel 1.

Tabel 1. Tabel Peralatan yang diperlukan

| No | Nama Alat | Jumlah |
|----|----------------|--------|
| 1 | Bor mini | 3 buah |
| 2 | Bor duduk | 1 buah |
| 3 | Gerinda Potong | 1 buah |
| 4 | Multimeter | 1 buah |
| 5 | Obeng + | 1 buah |
| 6 | Solder | 1 buah |
| 7 | Atraktor | 1 buah |
| 8 | Penggaris | 1 buah |
| 9 | Tang cucut | 1 buah |
| 10 | Tang potong | 1 buah |
| 11 | Cutter | 1 buah |
| 12 | Gunting | 1 buah |
| 13 | Laptop | 1 buah |

2) Bahan

Bahan yang diperlukan dalam pembuatan alat seperti pada tabel 2.

Tabel 2. Bahan yang diperlukan

| No | Nama Barang | Jumlah |
|----|-------------------|--------|
| 1 | Arduino Mega 2560 | 1 buah |
| 2 | Arduino Nano | 1 buah |
| 3 | Sensor Ultrasonik | 2 buah |
| 4 | Servo | 1 buah |
| 5 | 3DR Radio V2 | 1 buah |
| 6 | GPS Neo 7 | 1 buah |
| 7 | ESC | 1 buah |
| 8 | UBEC 5V @3A | 1 buah |
| 9 | Mobil-mobilan | 1 buah |
| 10 | LED | 1 buah |
| 11 | Relay | 1 buah |
| 12 | Batterai Lippo | 1 buah |

3) Jadwal Kegiatan

Proses pengerjaan karya tugas akhir mulai dari awal persiapan hingga pembuatan alat selesai direncanakan sebagai berikut:

Tabel 3. Jadwal Kegiatan Pengerjaan Alat

| Kegiatan | Bulan | | | | | | | | | | | |
|--|-------|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | | | | 2 | | | | 3 | | | |
| | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 |
| Pencarian ide | ■ | ■ | ■ | ■ | | | | | | | | |
| Pengumpulan informasi dan materi | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | | | | |
| Pembuatan rangkaian dan <i>hardware</i> | | | | | ■ | ■ | ■ | ■ | | | | |
| Pembuatan <i>software</i> | | | | | ■ | ■ | ■ | ■ | | | | |
| Uji coba <i>software</i> dan <i>hardware</i> | | | | | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ |
| Pembuatan laporan | | | | | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ |
| Ujian | | | | | | | | | | | | ■ |

D. Pengembangan

Pengembangan dalam pengerjaan proyek akhir *Prototype Autopilot* pada Mobil Menggunakan Kendali *GPS* merupakan pengembangan dari perancangan yang telah dibuat ke bentuk *hardware* dan *software* yang siap untuk diimplementasikan. Berikut merupakan tahapan dari proses pengembangan:

1. *Hardware*

a. Elektronik

- 1) Menyiapkan alat dan bahan yang dibutuhkan dengan menganalisis kebutuhan.
- 2) Merealisasikan *shiled* Arduino Mega, menggunakan PCB.
- 3) Merealisasikan *shiled* Arduino Nano, menggunakan PCB.

- 4) Menguji unjuk kerja tiap komponen elektronik yang telah dibuat.

2. *Software*

a. Arduino IDE

- 1) Menyiapkan alat dan bahan yang dibutuhkan dengan menganalisis kebutuhan.
- 2) Merealisasikan *flowchart* yang telah dirancang ke bentuk program pada Arduino IDE.
- 3) Menguji program Arduino IDE ke *hardware* yang telah dibuat.

b. Mission Planner

- 1) Menyiapkan alat dan bahan yang dibutuhkan.
- 2) Mencari lokasi yang luas seperti lapangan sepak bola, atau tanah kosong yang lebih luas dari lapangan sepak bola.
- 3) Mencoba alat tersebut sesuai dengan misi yang diinginkan
- 4) Menguji alat dari Mission Planner ke gerak prototipe mobil, ataupun menampilkan hasil misi sesuai dengan arah jalan prototipe mobil.

E. Implementasi

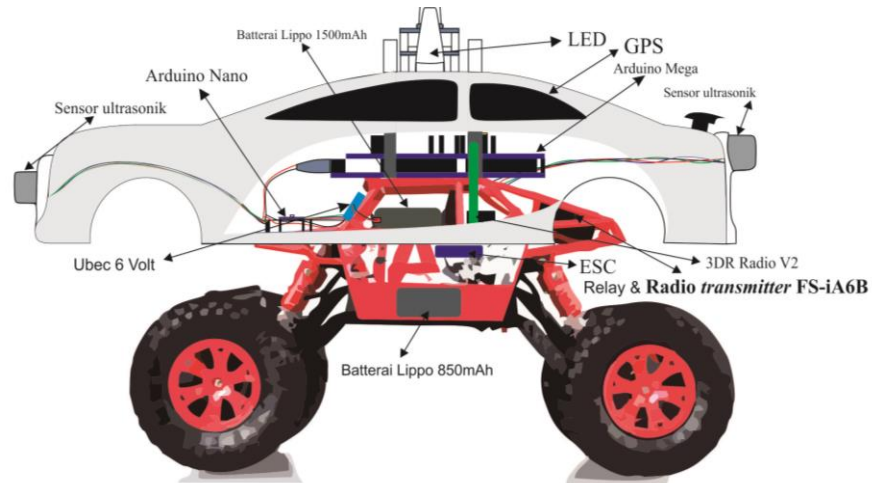
Implementasi merupakan tahapan merealisasikan tahapan pengembangan yang telah dibuat. Berikut merupakan hasil dari tahapan implementasi dari perancangan yang terdiri dari desain *hardware* dan *software*.

1. *Hardware*

a. Prototipe Mobil

Pada tugas akhir ini memakai mobil RC 1/18 *scale* yang menempatkan beberapa komponen didalamnya. Berikut ini merupakan desain prototipe mobil

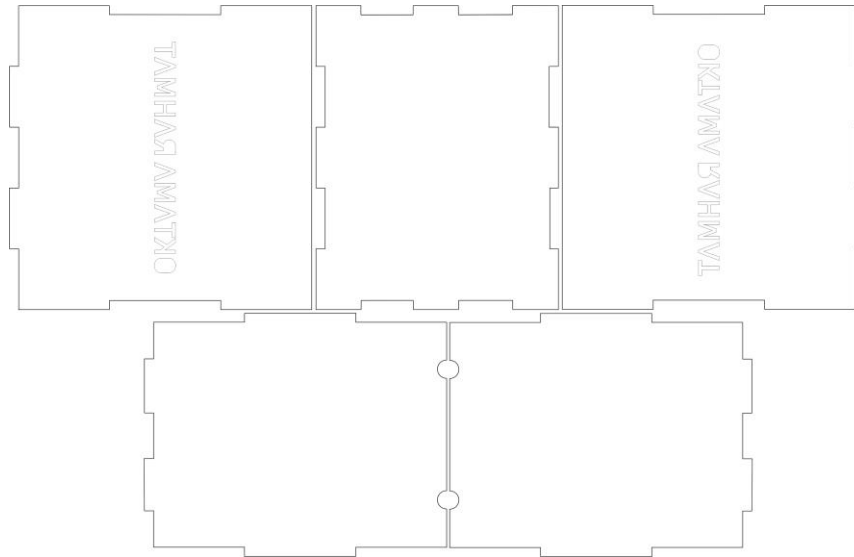
yang tampak dari samping dan tata letak penempatan komponen pada prototipe mobil.



Gambar 26. Desain Prototype Tampak Samping

Gambar 26 merupakan detail tata letak komponen pada prototipe mobil. sensor ultrasonik yang ditempatkan didepan dan dibelakang. Baterai lippo dibawah dan didalam. ESC, relay, Arduino Mega 2560 dan Arduino Nano dan UBEC yang diletakan didalam bodi mobil. *GPS*, *LED*, *Buzzer* dan *Push Button* yang diletakan melekat diatas bodi mobil.

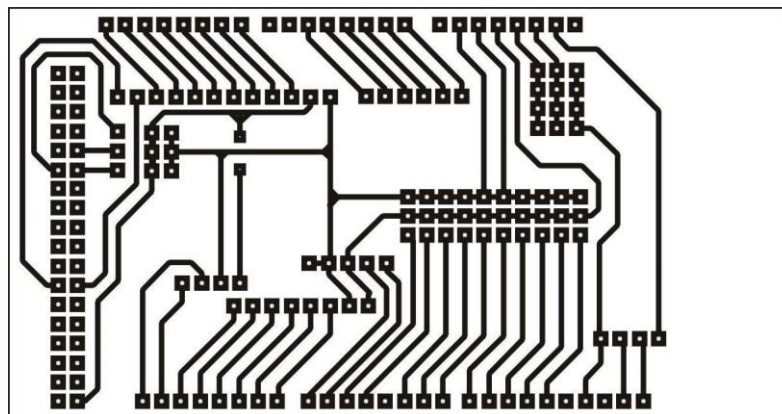
b. Kotak Persegi Panjang (Dudukan *prototype* mobil)



Gambar 27. Desain Dudukan *prototype* mobil

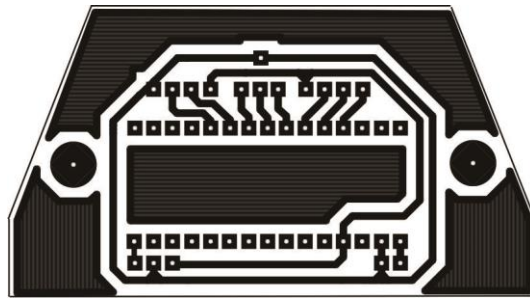
Gambar 27 merupakan desain kotak akrilik persegi panjang. Desain kotak akrilik persegi panjang ini digunakan untuk memudahkan dalam mengatur sebelum melakukan misi, bila terjadi hal-hal yang tidak diinginkan seperti mobil berjalan sendiri.

c. Elektronik



Gambar 28. *Layout* PCB *Shield* Arduino Mega

Gambar 28 merupakan *layout* PCB *shield* mikrokontroler dari Arduino mega. Desain *layout* telah disesuaikan dengan kebutuhan I/O yang dibutuhkan agar mempermudah pemasangan perkabelan.



Gambar 29. *Layout* PCB *Shield* Arduino Nano

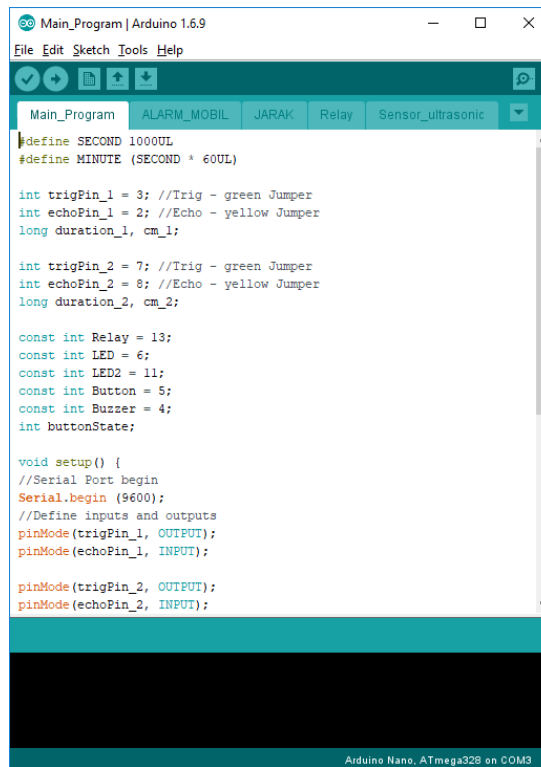
Gambar 29 merupakan *layout* PCB *shield* mikrokontroler dari Arduino Nano. Desain *layout* telah disesuaikan dengan kebutuhan I/O yang dibutuhkan agar mempermudah pemasangan perkabelan.

2. *Software*

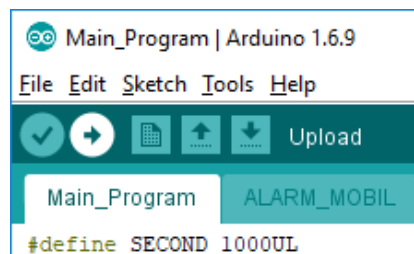
Software direalisasikan pada alat dengan cara mengupload program dari Arduino IDE dan menjalankan Mission Planner. Lalu alat dijalankan sesuai dengan pengoprasian alat.

a. *Software* arduino IDE

Perancangan *software*, pada bagian pemrograman mikrokontroler dengan menggunakan *software* Arduino IDE. Pemrograman Arduino IDE menggunakan bahasa pemrograman bahasa C. Program Arduino sendiri dimulai dengan menginialisasi pin-pin dari mikrokontroler arduino yang akan digunakan oleh sistem dan setelah itu pin-pin tersebut dikontrol sesuai dengan *flow chart* program. Gambar 30 merupakan potongan dari program yang digunakan.

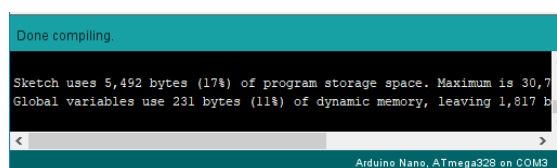


Gambar 30. Program Arduino IDE



Gambar 31. Arduino IDE *Upload* Program

Setelah program selesai dibuat kemudian *upload* program ke Arduino, seperti gambar 31. Apabila program sudah benar maka terdapat pemberitahuan *done compiling* yang berarti program yang telah dibuat sudah benar seperti gambar 32.



Gambar 32. *Done* Program

F. Pengujian Alat

Pengujian alat dilakukan untuk mendapatkan data penelitian. Pengujian alat ini dilakukan dengan dua pengujian, yaitu:

1. Uji Fungsional

Pengujian alat dilakukan dengan cara menguji setiap bagian-bagian berdasarkan karakteristik dan fungsi masing-masing. Pengujian ini dilakukan untuk mengetahui apakah setiap bagian dari perangkat telah dapat bekerja sesuai dengan fungsi dan keinginan yang akan dibutuhkan.

2. Uji Unjuk Kerja

Pengujian unjuk kerja alat sangat dibutuhkan dengan tujuan agar dapat mengetahui unjuk kerja alat. Beberapa hal yang perlu diamati antara lain: rangkaian sensor, rangkaian mikrokontroler. Sehingga apa yang diuji dapat diketahui bagaimana kinerja masing-masing bagian.

G. Spesifikasi Alat

Pada pembuatan proyek akhir *Prototype Autopilot Pada Mobil Menggunakan Kendali GPS* memiliki spesifikasi sebagai berikut:

1. Sumber tegangan yang digunakan adalah 6V.
2. Kendali sistem menggunakan Arduino Nano dan Arduino Mega 2560.
3. GPS yang digunakan *GPS Neo 7*.
4. *Driver motor* yang digunakan *ESC 20A Brush Motor Speed Controller 1/18 Scale RC*
5. Sensor ultrasonik yang digunakan HCSR04 dan US-015.
6. Motor Servo yang digunakan Tower pro MG-90 Metal Servo.

7. Baterai Lippo 1500mAh dan 850 mAh.
8. 3DR Radio V2.
9. Buzzer 5VDC.
10. Prototipe Mobil
11. Bahasa Pemrograman yang digunakan Bahasa C.
12. *Software* yang digunakan Arduino IDE dan Mission Planner.

H. Tabel Hasil Uji

1. Pengujian Tegangan UBEC (*Stepdown*)

Tabel 4. Pengukuran Tegangan UBEC (*Stepdown*)

| No | V-Out (Volt) | V output (VDC) | | error% | |
|----|--------------|----------------|--------------|-------------|--------------|
| | | Tanpa Beban | Dengan Beban | Tanpa Beban | Dengan Beban |
| 1 | | | | | |
| 2 | | | | | |
| 3 | | | | | |
| 4 | | | | | |

2. Pengujian Tegangan Mikrokontroler

Tabel 5. Tegangan Mikrokontroler

| Mikrokontroler | Pegujian ke- | V-Out (Volt) | V output (VDC) | | error% | |
|-------------------|--------------|--------------|----------------|--------------|-------------|--------------|
| | | | Tanpa Beban | Dengan Beban | Tanpa Beban | Dengan Beban |
| Arduino Mega 2560 | 1 | | | | | |
| | 2 | | | | | |
| | 3 | | | | | |
| | 4 | | | | | |
| Arduino Nano | 1 | | | | | |
| | 2 | | | | | |
| | 3 | | | | | |
| | 4 | | | | | |

3. Pengujian Motor Servo

Tabel 6. Pengujian Motor Servo

| No | Kondisi Motor | Tegangan (V) |
|----|---------------|--------------|
| 1 | 0 | |
| | 1 | |
| 2 | 0 | |
| | 1 | |
| 3 | 0 | |
| | 1 | |
| 4 | 0 | |
| | 1 | |

4. Pengujian Motor DC

Tabel 7. Pengujian Motor DC

| No | Kondisi Motor | Tegangan (V) |
|----|---------------|--------------|
| 1 | 0 | |
| | 1 | |
| 2 | 0 | |
| | 1 | |
| 3 | 0 | |
| | 1 | |
| 4 | 0 | |
| | 1 | |

5. Pengujian ESC

Tabel 8. Pengujian ESC

| No | ESC | Kondisi ESC | Tegangan (V) |
|----|-------|-------------|--------------|
| 1 | Kanan | 0 | |
| | | 1 | |
| 2 | Kiri | 0 | |
| | | 1 | |

6. Pengujian Sensor Ultrasonik

Tabel 9. Pengujian Tegangan Sensor Ultrasonik

| No | Sensor | Pengukuran ke- | Jarak Asli (Cm) | Jarak pada Sensor (Cm) | Keterangan |
|----|----------|----------------|-----------------|------------------------|------------|
| 1 | Sensor 1 | 1 | | | |
| | | 2 | | | |
| | | 3 | | | |
| | | 4 | | | |
| | | 5 | | | |
| 2 | Sensor 2 | 1 | | | |
| | | 2 | | | |
| | | 3 | | | |
| | | 4 | | | |
| | | 5 | | | |

7. Pengujian Relay

Tabel 10. Pengujian Relay

| No | Relay | Tegangan Arduino (5V) | Kondisi Relay | Kondisi ESC | Keterangan |
|----|-------|-----------------------|---------------|-------------|------------|
| 1 | Relay | <i>HIGH</i> | | | |
| | | <i>LOW</i> | | | |

8. Pengujian GPS

Tabel 11. Pengujian GPS

| No | Koordinat Asli | Koordinat Alat | Keterangan |
|----|----------------|----------------|------------|
| 1 | | | |
| 2 | | | |
| 3 | | | |

9. Pengujian unjuk kerja bagian Sensor Anti Nabrak

Tabel 12. Pengujian Unjuk Kerja bagian Sensor Anti Nabrak

| No | Percobaan | Percobaan ke- | kondisi | Buzzer | LED | Relay |
|----|---------------------|---------------|------------------|--------|-----|-------|
| 1 | Sensor ultrasonik 1 | 1 | Lebih dari 20cm | | | |
| | | | Kurang dari 20cm | | | |
| | | 2 | Lebih dari 20cm | | | |
| | | | Kurang dari 20cm | | | |
| | | 3 | Lebih dari 20cm | | | |
| | | | Kurang dari 20cm | | | |
| 2 | Sensor ultrasonik 2 | 1 | Lebih dari 20cm | | | |
| | | | Kurang dari 20cm | | | |
| | | 2 | Lebih dari 20cm | | | |
| | | | Kurang dari 20cm | | | |
| | | 3 | Lebih dari 20cm | | | |
| | | | Kurang dari 20cm | | | |

10. Pengujian unjuk kerja bagian misi

Tabel 13. Pengujian Alat

| Percobaan | Koordinat Misi | Koordinat Mobil | Keterangan |
|-----------|----------------|-----------------|------------|
| 1 | | | |
| 2 | | | |
| 3 | | | |
| 4 | | | |
| 5 | | | |
| 6 | | | |
| 7 | | | |
| 8 | | | |
| 9 | | | |
| 10 | | | |
| 11 | | | |

I. Pengoperasian Alat

Pengoperasian alat ini dapat dilakukan dengan cara sebagai berikut:

1. Memasang Baterai 7,4 Volt (2 cell) dan menekan saklar pada prototipe mobil untuk menghidupkan prototipe mobil.
2. Menghidupkan laptop dan membuka aplikasi mission planner.



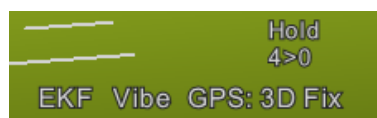
Gambar 33. Bagian *ToolBox* Mission planner

3. Klik COM5 ganti sesuaikan dengan COM *port* telemetry, Klik menustrip pada angka 57600 lalu ganti dengan 57600 dan bila sudah sesuai, Klik *CONNECT* pada *ToolBox* seperti gambar 33.



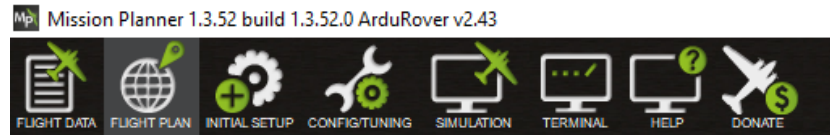
Gambar 34. Tampilan Mission Planner Setelah *Connect*

4. Setelah klik *CONNECT*, tunggu beberapa menit sampai tampilan muncul seperti gambar 34.

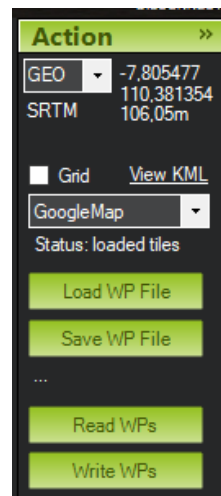


Gambar 35. Kondisi Prototipe Mobil

- Pastikan prototipe mobil pada kondisi *hold* seperti gambar 35, untuk melakukan settingan mode *autopilot*.

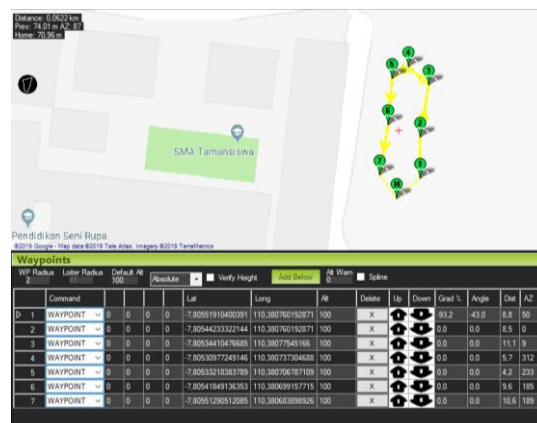


Gambar 36. Bagian Menu *ToolBox* Mission planner



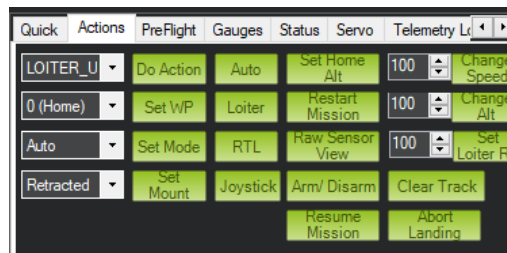
Gambar 37. Bagian Menu *ToolBox Action* pada *flight plan*

- Klik *FLIGHT PLAN* seperti gambar 36, lalu Klik GoogleMap ganti sesuai kebutuhan seperti gambar 37.



Gambar 38. Sample Jalur *Autopilot* pada Prototipe Mobil

7. Bikin jalur sesuai dengan keinginan, seperti contohnya pada gambar 38, setelah selesai *setting comand* jalur dari pertama sampai akhir dengan *waypoint*.
8. Klik *Write WPs* seperti gambar 37, dan setelah diklik pastikan kembali dengan klik *Read WPs* apakah sudah sesuai dengan jalur setelah mengklik tombol *Write WPs*.
9. Setelah selesai menyetting jalur *autopilot* pada mobil lalu klik tombol *FLIGHT DATA* seperti gambar 36.



Gambar 39. Bagian Menu *ToolBox Actions* pada *flight data*

10. Setelah selesai mengsetting mission planner letakan mobil pada posisi *home* (H), lalu klik *Actions* pada *flight data* seperti gambar 39. Setelah itu ganti *hold* menjadi *auto* bila sudah siap klik tombol *Set mode* maka otomatis mobil akan berjalan sesuai dengan koordinat yang sudah diatur.