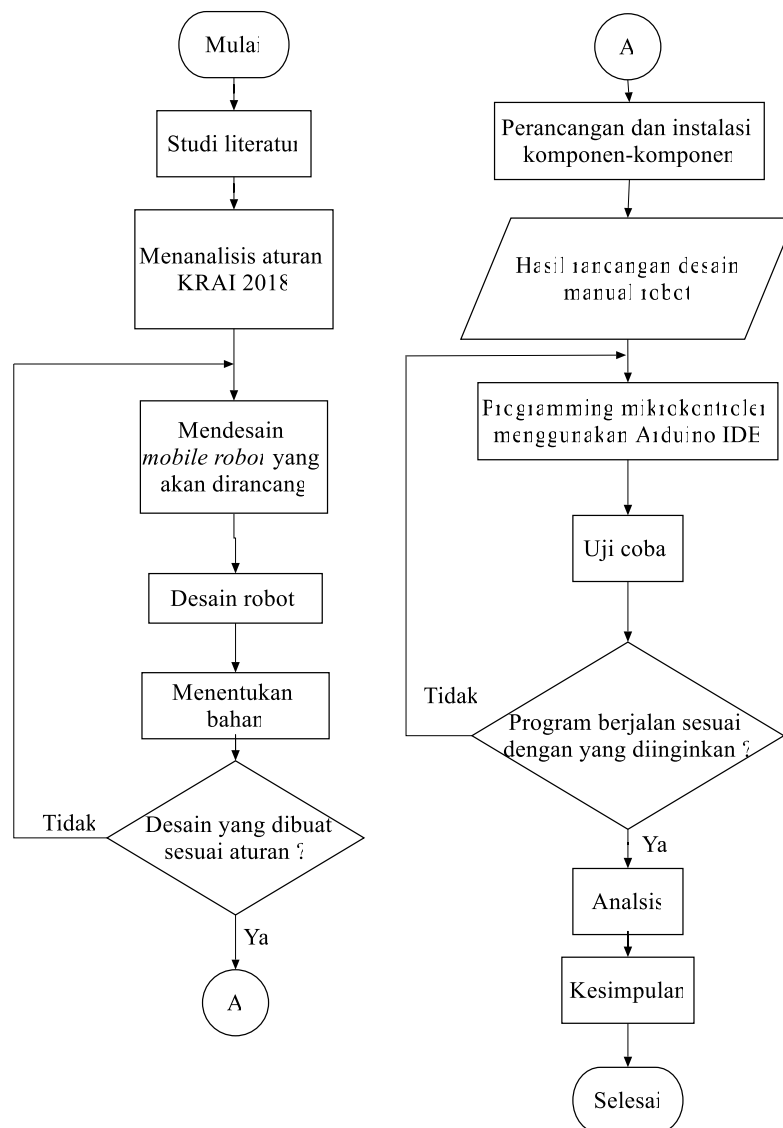


**BAB III**  
**METODE PERANCANGAN DAN PEMBUATAN**

**3.1 Diagram Alir Perancangan**

Dalam perancangan ini memiliki beberapa tahap yang perlu disiapkan. Mulai dari persiapan, proses perancangan, serta proses pengujian hasil perancangan yang bertujuan mengetahui apakah robot manual yang dirancang bekerja dengan baik. Tahapan perancangan dapat dilihat dalam diagram alir dibawah ini.



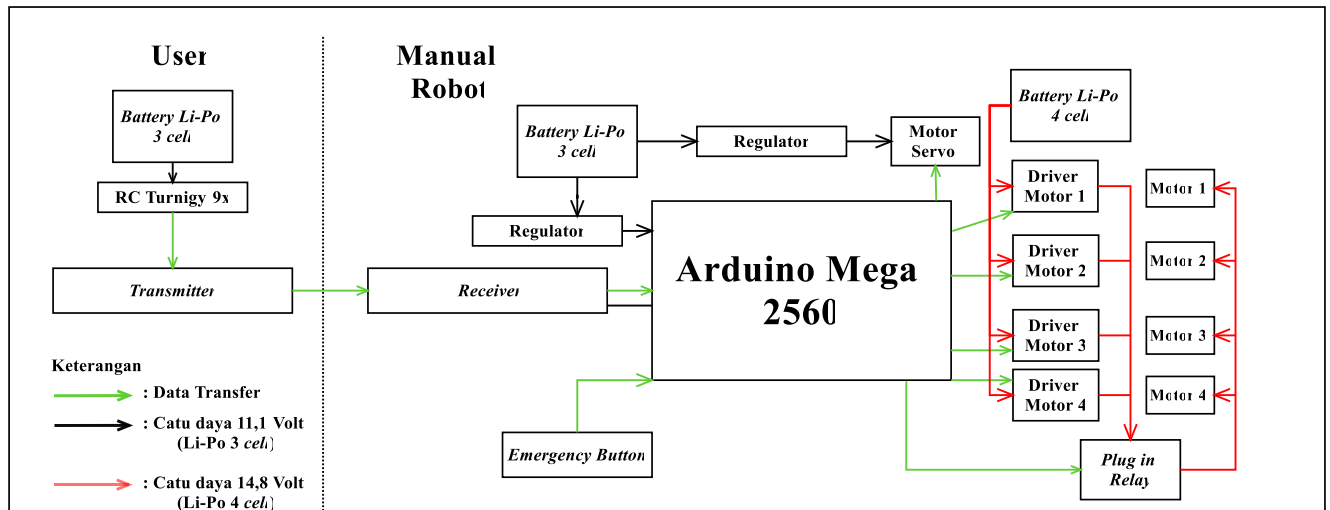
**Gambar 3.1** Diagram Alir Perancangan

Pada gambar 3.1 menjelaskan diagram mengenai alur-alur perancangan yang akan dilakukan hingga tahap analisis dan kesimpulan. Diawali dengan studi literatur yang diambil pada jurnal-jurnal dan buku mengenai perancangan *mobile robot*. Tahap berikutnya ialah menganalisis peraturan yang berkaitan dengan Kontes Robot ABU Indonesia (KRAI) tahun 2018. Setelah peraturan dipahami, desain dilakukan untuk menentukan kerangka robot yang didasarkan tidak melebihi dimensi ukuran yang ada pada peraturan. Tahap berikutnya adalah mendesain robot menggunakan perangkat lunak *SketchUp*. Dimensi pada robot tentunya mengacu pada peraturan Kontes Robot ABU Indonesia (KRAI) tahun 2018 karena dimensi pada robot merupakan tahap awal ketika akan melakukan kontes apakah robot tersebut memenuhi standar atau tidak, bila tidak robot tidak bisa mengikuti kontes. Penentuan bahan perlu dilakukan agar robot dapat bergerak dengan baik dan pemilihan bahan yang tepat adalah menggunakan bahan yang ringan tetapi memiliki struktur kuat. Bila desain robot dan bahan yang digunakan sudah sesuai maka dilanjutkan ke tahap berikutnya dan bila belum memenuhi persyaratan, maka akan dilakukan desain ulang. Tahap selanjutnya adalah merancang dan instalasi tiap komponen yang akan digunakan. Setelah semua perancangan dilakukan maka bentuk fisik robot manual sudah terbentuk.

Setelah komponen dan bentuk robot manual sudah jadi, maka akan dilakukan memprogram perintah yang dilakukan pada komputer. Programming atau kegiatan membuat program dilakukan menggunakan Arduino IDE. Setelah kode perintah sudah jadi maka perintah akan diunggah pada *board* Arduino tersebut. Setelah selesai maka bisa dicoba menggunakan *remote control* yang berperan sebagai pengendali robot manual tersebut, bila pergerakan sesuai maka perancangan selesai dan dapat dilakukan analisis pada robot manual tersebut dan bila pergerakan atau hasil tidak sesuai, maka akan dilakukan programming ulang. Bila robot sudah dirasa cukup dalam perancangan dan pemrograman cukup baik, maka tahap selanjutnya adalah melakukan analisis terhadap fungsi tiap komponen yang digunakan dan fungsi robot itu sendiri. Tahap terakhir adalah menarik kesimpulan apa yang sudah dirancang.

### 3.2 Blok Diagram Komponen

Pada perancangan sebuah robot, diperlukan gambaran blok diagram komponen yang berlaku pada robot tersebut. Pada robot manual yang dirancang memiliki diagram antar komponen sebagai berikut:



Gambar 3.2 Diagram Blok Robot Manual

Pada robot manual yang dirancang menggunakan 2 catu daya berupa Baterai Li-Po dengan sel yang berbeda. Untuk catu daya mikrokontroler menggunakan catu daya baterai Li-Po sebesar 2 sel sebesar 7,4 volt (1 sel baterai Li-Po sebesar 3,7 volt) cukup untuk catu daya sebuah mikrokontroler. Catu daya untuk menghidupkan keempat motor PG-45 adalah baterai sebesar 4 sel yaitu sebesar 14,8 volt. Switch ON/OFF digunakan untuk memudahkan dalam menyalakan dan memutus catu daya dengan komponennya.

Mikrokontroler berperan sebagai otak pada robot manual yang dirancang. Perintah pergerakan robot akan dikirim pada driver motor tersebut dan untuk perintah pengereman akan dilakukan oleh *plug in relay* dengan memutuskan-menghubungkan catu daya yang berasal dari baterai. Driver motor akan mengirim perintah yang dikirim dari mikrokontroler kepada motor DC PG-45 yang akan menggerakkan arah dan kecepatan robot manual tersebut.

### 3.3 Lokasi dan Waktu Perancangan

Lokasi yang digunakan untuk perancangan robot ini akan dilakukan di Laboratorium Mikrokontroler dan Robotika Teknik Elektro Universitas Muhammadiyah Yogyakarta. Beralamatkan di Kampus Terpadu Universitas Muhammadiyah Yogyakarta Jl. Brawijaya, Geblagan, Desa Tamantirto, Kecamatan Kasihan, Kabupaten Bantul, Daerah Istimewa Yogyakarta kode pos: 55183. Waktu perancangan ini akan dilakukan pada rentang waktu antara bulan Desember 2018 hingga Maret 2019.

### 3.4 Tahap Perencanaan Perancangan

Pada tahap ini, merencanakan sebuah robot memang sangat perlu persiapan yang baik karena banyak yang perlu diperhatikan dalam perancangannya. Seperti robot apa yang akan kita rancang dan apa fungsi robot tersebut. Hal tersebut perlu ditentukan dalam beberapa tahap di bawah ini, diantaranya adalah :

1. **Dimensi**, yaitu panjang, lebar, tinggi serta perkiraan massa pada robot tersebut. Aturan yang ada pada KRAI 2018 perlu diperhatikan agar robot dapat melewati saat ada penilaian dimensi robot.
2. **Struktur Material**, ialah bahan yang akan digunakan dalam perancangan robot tersebut. Apakah akan dibuat dari besi, aluminium, plastic, atau bahan lainnya.
3. **Cara Kerja Robot**, yang berisikan dari bagian-bagian dan fungsi pada robot tersebut. Contohnya yaitu seperti fungsi lengan robot yang digunakan untuk menjepit suatu objek.
4. **Mekanisme Robot**, yaitu bagaimana sistem mekanik dari robot tersebut agar dapat menyelesaikan perintah dengan baik.
5. **Metode Pengontrolan**, yaitu mengenai bagaimana robot tersebut dapat dikontrol dan bergerak sesuai perintah yang diberikan serta alat apa yang akan digunakan untuk mengontrol robot tersebut.

6. **Strategi**, dalam suatu kontes robot perlu strategi untuk memenangkan pertandingannya. Karena strategi yang baik sangat perlu ditentukan supaya robot dapat memenangkan kontes dengan baik.

### 3.5 Tahap Realisasi Perancangan

Pada tahap ini ada 3 bidang yang harus dilakukan, yaitu dalam pembuatan mekanik robotnya, elektroniknya, dan memprogram robot tersebut. Seharusnya masing-masing pada bidang tersebut membutuhkan orang dengan keterampilan di bidang masing-masing yaitu :

1. Bidang Mekanik: pada ilmu bidang ini yang cocok ialah Teknik mesin.
2. Bidang Elektroknik: pada ilmu bidang ini yang cocok ialah Teknik Elektro.
3. Bidang Programming: pada ilmu bidang ini yang cocok ialah Teknik Informatika dan Teknik elektro (walau tidak terlalu mendalam).

Memang dalam sebuah tim robotic perlu adanya anggota yang berasal dari berbagai bidang tidak hanya dari Teknik elektro atau Teknik mesin supaya dalam perencanaan dapat lebih terukur dan tertata serta *deadline* waktu bisa terpenuhi.

#### 3.5.1 Merancang Desain dan Komponen

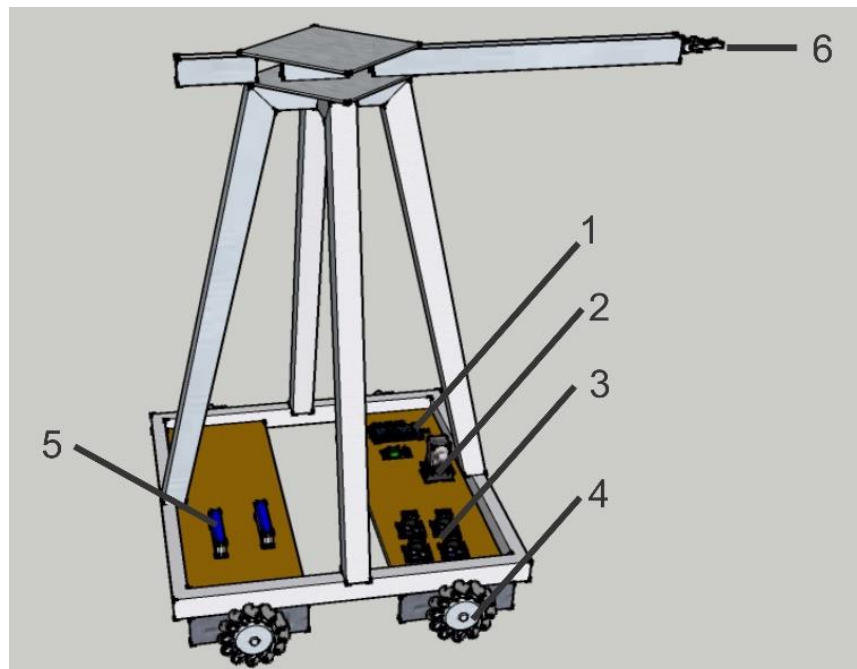
Sebelum memulai sebuah perancangan, diperlukan membuat gambaran atau design robot yang akan dirancang dan pemilihan serta peletakan komponen. Tujuan pembuatan ini tidak lain untuk memudahkan dan perancangan lebih terarah dengan baik karena perancangan robot manual Kontes Robot ABU Indonesia (KRAI) tahun 2018, memiliki peraturan dalam perancangan robot yang akan diikuti sertakan dalam kontes. Perangkat lunak yang digunakan dalam mendesign perancangan adalah Google Sketchup. Tahap-tahap ini akan dijelaskan sebagai berikut :

1. Instal terlebih dahulu perangkat lunak *SketchUp*.
2. Buka *software* lalu pilih *new project*.
3. Terdapat beberapa pilihan template pada *software* tersebut dan pilih nilai pengukuran menggunakan *centimeters*.

4. Buat dimensi robot terlebih dahulu lalu design robot tersebut yang tidak melebihi dimensi yang sudah diterapkan dalam aturan kontes.
5. Bila design sudah cukup, tambahkan beberapa bentuk komponen yang akan digunakan.

### 3.5.2 Perancangan Desain Robot

Perancangan desain robot merupakan perancangan yang ada pada bidang mekanik. perancangan berupa pembuatan desain rancangan robot yang akan dirancang. Perancangan memperhatikan ukuran dimensi yang sesuai dengan aturan pada Kontes Robot ABU Indonesia (KRAI) Tahun 2018. Pada perancangan ini menggunakan perangkat lunak CAD yaitu *SketchUp*. Perancangan desain berupa rancangan kerangka robot dan gambaran awal pemasangan dan posisi komponen robot yang akan digunakan.

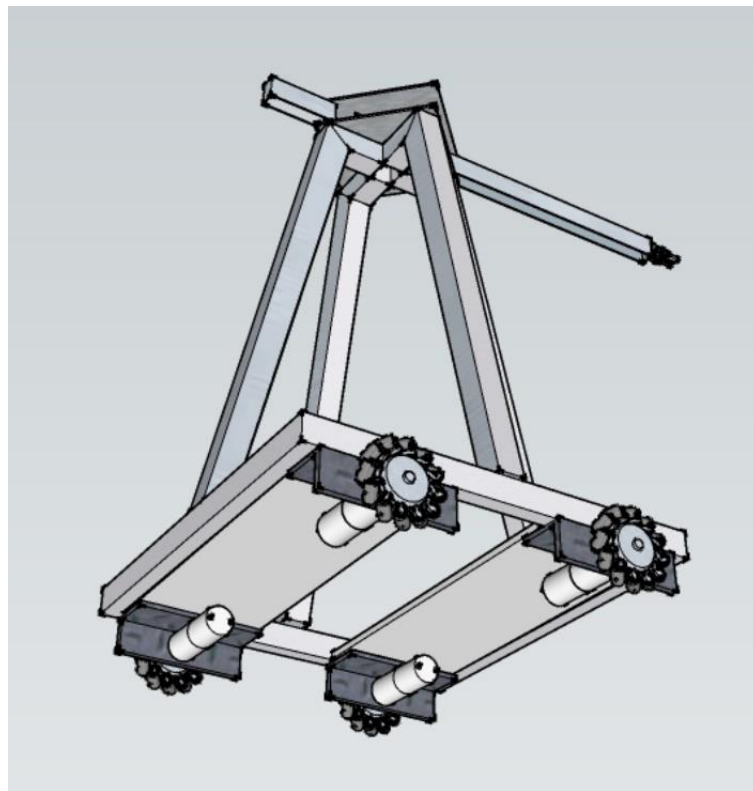


**Gambar 3.3** Desain Rancangan Robot Manual

Robot yang di desain memiliki komponen-komponen diantaranya seperti yang ditunjuk oleh gambar 3.3 :

1. *Board* PCB yang terdiri dari mikrokontroler dan beberapa komponen lainnya seperti *receiver*

2. Plug in Relay yang berfungsi sebagai pengereman pada robot.
3. Motor driver jenis BTS7960.
4. Roda mekanum.
5. Baterai Li-Po 2 buah yang digunakan untuk catu daya motor dan mikrokontroler.
6. Motor servo dan regulator *step down*-nya.



**Gambar 3.4** Bagian Bawah Robot

Pada bagian bawah robot manual yang dirancang dapat dilihat terdiri dari 4 roda mekanum beserta dudukan motor dan motor DC PG45. Dudukan motor terbuat dari plat besi yang dapat mendinginkan motor DC PG45 dan dudukan tersebut cukup kuat dan tebal atau biasa disebut dengan *heatsink*. Pengaruh dari roda mekanum yang bekerja dengan menekan gaya kebawah cukup tinggi maka dudukan perlu dibuat cukup kuat untuk menahan serta mendinginkan motor DC saat bekerja.

Robot manual yang dirancang dibuat seringan mungkin dan ramping. Robot memiliki berat kurang lebih 15kg dan dibuat ramping dikarenakan kecepatan saat kontes sangat diperlukan, bila robot terlalu berat maka akan mengganggu pergerakan robot itu sendiri. Peraturan yang dibuatpun diharuskan robot tidak boleh memiliki berat lebih dari 25kg. Robot manual yang dirancang menggunakan bahan plat alumunium yang ringan serta kuat. Untuk menyambungkan satu dengan yang lainnya menggunakan lempengan besi yang memiliki ketebalan sekitar 0,5cm – 1 cm dan dibuat sedemikian mungkin untuk pemasangan lempengan tersebut.

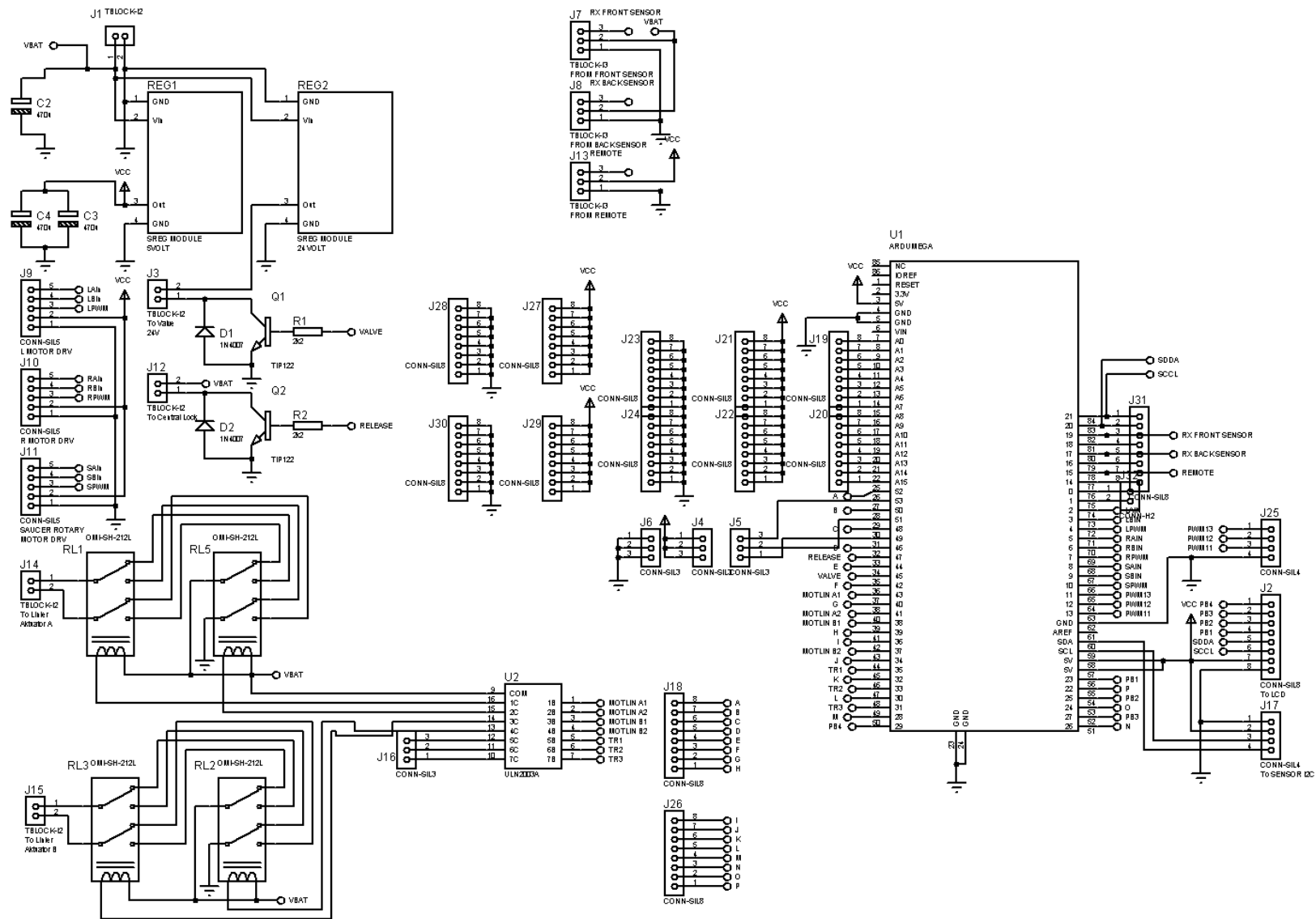
Motor servo pada robot digunakan untuk menarik bola yang ada pada rak bola. Bola yang dibuat memiliki berat kurang lebih 15g. Motor servo yang dirancang harus dipastikan dapat menarik “bola berkah” (nama istilah bola yang digunakan dalam kontes) dan bola tersebut diberikan kepada robot *automatic* untuk dilemparkan melewati tiang-tiang yang ada pada arena.

### **3.5.3 Perancangan Rangkaian**

Sebelum memulai perancangan rangkaian, diperlukan gambaran kasar yang akan terapkan pada rangkaian. Perancangan dibuat menggunakan perangkat lunak Proteus 8.0 yang dapat membuat rancangan serta mensimulasikan rangkaian yang dibuat. Perangkat lunak Proteus 8.0 sudah cukup kompleks dalam membuat suatu perancangan sebuah rangkaian yang akan digunakan. Perancangan rangkaian dan pembuatan sebelum melakukan perancangan, diperlukan mendesain rangkaian yang akan digunakan pada *board* PCB. Desain yang dibuat meliputi rangkaian mikrokontroler yang terhubung dengan beberapa komponen seperti relay, kapasitor, dudukan regulator *step up*, dan beberapa resistor, transistor.

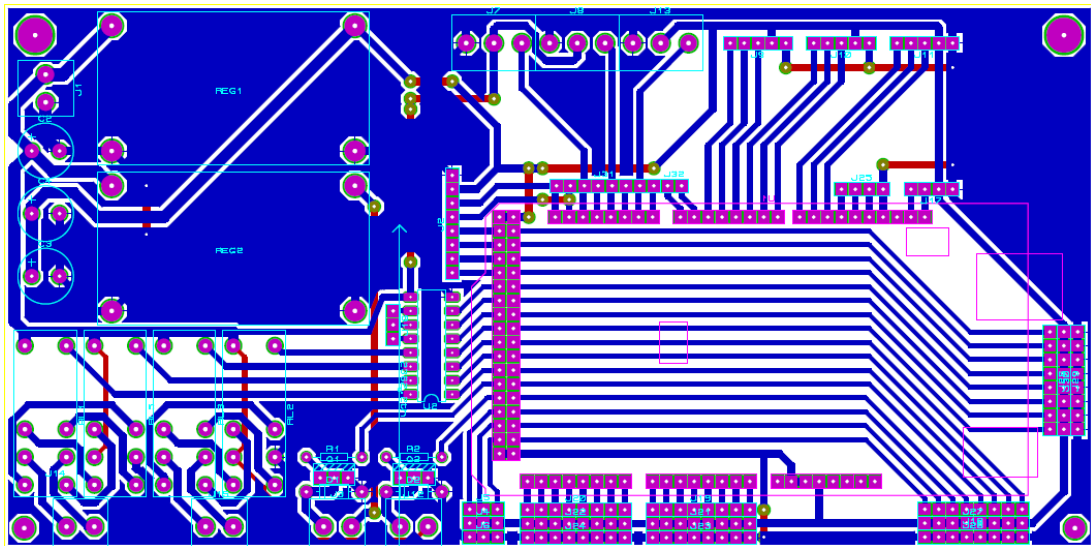
Perancangan rangkaian ini berguna untuk mengurangi dan menghindari eror atau kesalahan yang dapat disebabkan oleh kabel-kabel *jumper*. Karena komponen yang akan digunakan dalam pembuatan *board* PCB cukup banyak dan terhubung satu sama lain, maka sangat diperlukan pembuatan *board* PCB untuk mengurangi nilai eror yang dapat timbul karena kesalahan pada sistem transmisi data atau catu daya.





Gambar 3.5 Rangkaian Mikrokontroler

Pada gambar 3.5 rangkaian di atas akan digunakan untuk desain papan PCB. Rangkaian meliputi mikrokontroler, relay, pin untuk driver motor, dan lainnya. Papan PCB merupakan salah satu komponen penunjang yang dapat digunakan untuk memasang komponen dan sebagai rangkaian yang digunakan untuk dudukan mikrokontroler. Papan PCB perlu didesain terlebih dahulu menggunakan salah satu perangkat lunak yaitu Proteus 8. Perangkat lunak ini sudah cukup kompleks karena dapat membuat rangkaian skematik untuk diaplikasikan pada board PCB. Desain yang dibuat perlu menunjang apa saja yang diperlukan dalam papan PCB dan dibuat sangat terperinci karena papan PCB yang akan dirancang dalam perancangan ini akan menjadi *board* utama dalam perancangan robot manual Kontes Robot ABU Indonesia (KRAI) tahun 2018, maka desain yang digunakan cukup kompleks dan terperinci.



**Gambar 3.6** Rangkaian Skematik Mikrokontroler

Pada gambar 3.6 adalah rangkaian yang didesain dan digunakan untuk *board* utama pada robot yang akan dirancang. *Board* diatas terdiri dari tempat dudukan mikrokontroler, 2 regulator *step down*, 4 relay, dan beberapa komponen lainnya. Setelah desain didapatkan lalu desain tersebut akan dicetak pada board PCB.

### 3.5.4 Pembuatan *Board* PCB

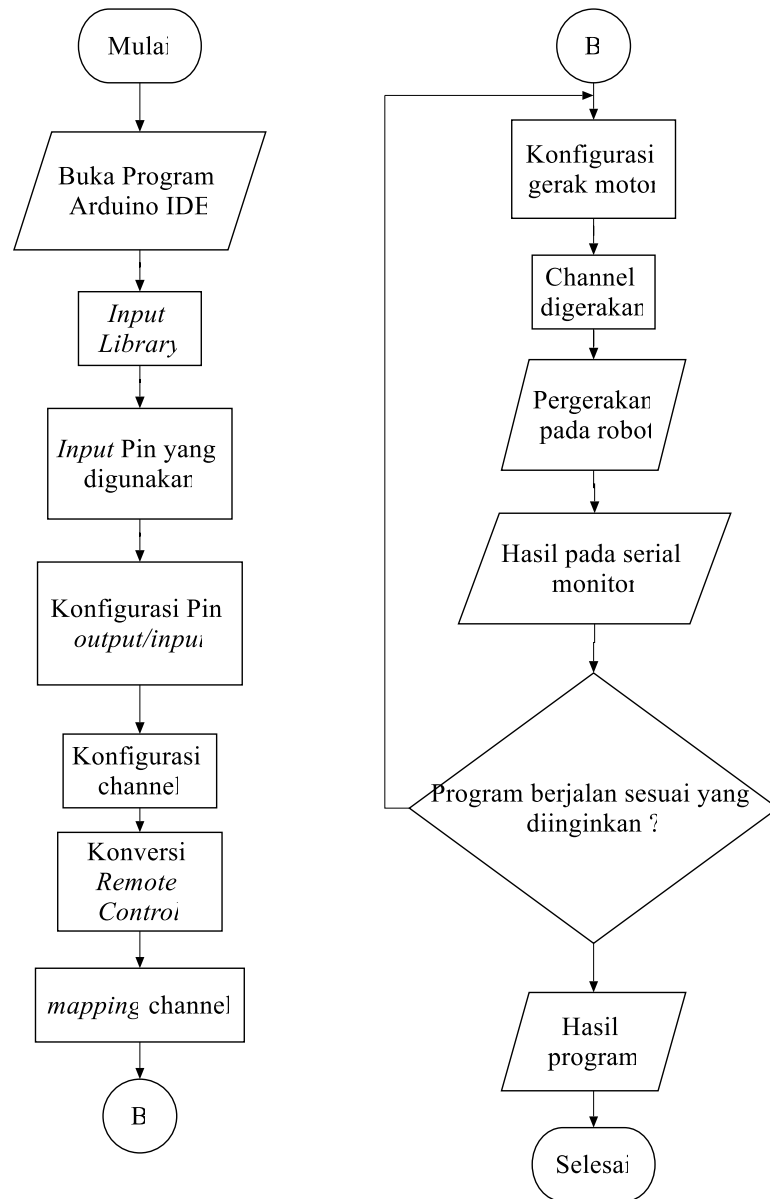
Perancangan sebuah *Board* PCB termasuk perancangan mekanik dan perancangan elektronik juga. Perancangan mekanik berupa tahap-tahap dalam pembuatan *board* PCB dan perancangan elektronik berupa instalasi tiap komponen yang saling terhubung. Langkah-langkah pembuatan *board* PCB akan dijelaskan sebagai berikut :

1. Buat desain rangkaian skematik terlebih dahulu menggunakan perangkat lunak Proteus 8.
2. Cetak design pada kertas HVS.
3. Fotocopy design tersebut pada kertas glossy.
4. Siapkan papan PCB polos.
5. Gunakan setrika listrik untuk memindahkan rangkaian skematik yang ada pada kertas glossy pada PCB polos dan dinginkan setelahnya.
6. Rangkaian yang sudah menempel pada papan PCB, diperlukan melarutkan lapisan tembaga yang tidak diperlukan menggunakan cairan kimia ferriclorit( $\text{FeCl}_3$ ).
7. Air yang sudah bercampur dengan pelarut dapat dimasukan PCB yang sudah disablon sebelumnya, goyangkan nampan lakukan proses ini 10 hingga 15 menit.
8. Setelah lapisan tembaga yang tidak diperlukan larut, bersihkan tinta yang masih menempel dan agar lapisan permukaan PCB tidak mudah korosi atau kotor, lapisi permukaan menggunakan cairan pelapis PCB.

### 3.5.5 Rancangan Perangkat Lunak

Rancangan yang dibuat adalah rancangan *script code* yang akan di eksekusi oleh perangkat lunak Arduino IDE. *Script code* ini digunakan untuk memprogram dan memberikan perintah yang akan dijalankan oleh mikrokontroler. Mikrokontroler akan memberikan sinyal atau gelombang yang dikirim melalui pin-pin yang nantinya akan menghasilkan suatu perubahan nilai PWM ataupun

tegangan pada komponen yang terlibat. *Script code* terdiri dari konfigurasi pin, pengaturan driver motor, pengaturan *receiver*, serta beberapa pengaturan lainnya. Adapun diagram alir perancangan perangkat lunak adalah sebagai berikut :



**Gambar 3.7** Diagram Alir Perancangan Perangkat Lunak

Pada gambar 3.7 merupakan diagram alir perancangan untuk perangkat lunak yang digunakan dalam perancangan robot manual. Perancangan ini berupa gambaran hasil dari *script code* yang dirancang untuk menjalankan robot manual. Diagram alir diawali dengan membuka program perangkat lunak Arduino IDE yang

digunakan untuk membuat *script program*. Masukan *library* yang akan digunakan perancangan seperti *servo.h* merupakan *library* untuk motor servo. Selanjutnya merupakan *input* pin yang digunakan dan mengkonfigurasi pin tersebut.

```

/*
const int Channel1 =26;
const int Channel2 =28;
const int Channel4 =30;
const int Channel5 =24;

pinMode(Channel1, INPUT);
pinMode(Channel2, INPUT);
pinMode(Channel4, INPUT);
*\

```

Potongan *Script code* diatas adalah salah satu konfigurasi pin yang digunakan pada pin mikrokontroler. Pin di atas digunakan untuk mengkonfigurasi channel yang digunakan bahwa channel masuk ke pin 26,28,30, dan 24 serta semua sebagai *input* untuk mikrokontroler.

```

/*
if(mot1>0) //FORWARD
{
digitalWrite(5,LOW);
digitalWrite(6,HIGH);
analogWrite(7,mot1);
}
else if(mot1<0)//BACKWARD
{
digitalWrite(5,HIGH);
digitalWrite(6,LOW);
analogWrite(7,abs(mot1));
}

```

```

else    //BRAKE
{
    digitalWrite(5,LOW);
    digitalWrite(6,LOW);
    analogWrite(7,0); //kiri
}
if(mot2>0) //FORWARD
{
    digitalWrite(2,HIGH);
    digitalWrite(3,LOW);
    analogWrite(4,mot2); //kanan
}
*\

```

Potongan *Script code* di atas merupakan salah satu konfigurasi untuk motor. Motor yang dikonfigurasi ada 4 buah motor tetapi disini hanya menampilkan 1 konfigurasi motor. Program diatas digunakan untuk memerintahkan motor bergerak ke arah jarum jam atau berlawanan arah jarum jam. Pin 5 merupakan pin R\_EN pada *driver* motor BTS7960 dan pin 6 merupakan pin L\_EN-nya. Untuk pin 7 merupakan pin RPWM dimana pin tersebut untuk *output* PWM pada *driver* motor tersebut.

Konversi *remote control* terhadap nilai channel yang terlibat diperlukan untuk mengubah sinyal yang berada pada *remote control* untuk terbaca oleh mikrokontroler. Konversi terdiri dari penggunaan *deadband* dan proses *mapping*. Proses ini sangat penting agar *remote control* dapat mengontrol robot sesuai keinginan pada hasil akhir.

```

/*
int rcmidUpper=rcmid+70; //dead band
int rcmidLower=rcmid-70;
*\

```

Dua *script* di atas merupakan potongan untuk *deadband* pada *remote control*. *Deadband* merupakan jumlah sinyal yang diizinkan untuk mengubah tanpa mempengaruhi nilai keluaran sebagai batas toleransi kesalahan pada nilai keluaran itu sendiri. Sinyal yang berasal dari *remote control* selalu tidak tepat pada angka 0 maka diperlukan nilai *deadband*.

```
/*
    int rcmin=980;
    int rcmid=1480;
    int rcmax=1999;
    int vmotor=0;
    int vmotormax=255;
    int vmotormin=0;
    if(ch2>rcmidUpper)
    {vfw=map(ch2,rcmidUpper,rcmax,vmotormin,vmotormax);
    digitalWrite(47,LOW);robotmove(vfw,vfw,vfw,vfw);}
    else if(ch2<rcmidLower) {vfw=map(ch2,rcmidLower,rcmin,-vmotormin,-
    vmotormax);digitalWrite(47,LOW);robotmove(vfw,vfw,vfw,vfw);}
    else {vfw=0;}
*\
```

*Script* di atas merupakan proses *mapping*. Proses *mapping* merupakan proses yang mengkonversikan suatu bilangan pada suatu rentang ke rentang lain. Proses konversi ini mengubah nilai yang awalnya memiliki rentang 980 – 2000 menjadi 0-255 karena sinyal PWM memiliki rentang 0-255. Bila proses *mapping* tidak dilakukan terlebih dahulu, maka sinyal keluaran akan sulit terbaca oleh mikrokontroler dan motor tersebut tidak akan mengeksekusi perintah yang ada.

## 1.6 Perlakuan Pengujian

Beberapa perlakuan yang dilakukan pada suatu perancangan perlu diinformasikan sebagai tahap-tahap yang dilakukan pada perancangan tersebut. Perlakuan perancangan dibuat secara berurutan dari apa yang pertama harus dilakukan hingga akhir dalam suatu pengujian. Proses yang dilakukan memiliki

tujuan memberikan informasi dan memudahkan dalam memahami dalam perancangan Tugas Akhir ini. Beberapa tahapan-tahapan diantaranya adalah sebagai berikut :

### **3.6.1 Menghubungkan Receiver dengan Transmitter**

*Receiver* yang belum terhubung memerlukan proses *binding* dengan *transmitter*-nya. Proses *binding* merupakan proses mengikat antara *receiver* dan *transmitter*. *Receiver* yang bukan bawaan dari *remote controller/transmitter*, diperlukan proses *binding* agar terikat satu sama lain.

*Receiver* yang digunakan adalah jenis V8FR-II Rx dan *transmitter* menggunakan FrSky Telemetry Module DJT dengan *remote control* berjenis Turnigy 9x. Beberapa tahapan yang diperlukan dalam proses mengikat diantaranya adalah :

1. Atur *switch* pada module DJT dalam keadaan OFF.
2. Koneksikan CH1-CH2 dengan *jumper* pada modul *receiver*.
3. Berikan sumber tegangan pada *receiver* (5V DC).
4. Atur pada *transmitter* dalam keadaan *bind mode*. Keadaan *bind mode* akan terlihat pada modul dengan tanda lampu led berkedip setiap 1 detik.
5. *Receiver* akan menerima sinyal dari *transmitter* dengan tanda lampu led pada *receiver* akan berkedip berwarna merah dengan cepat.
6. Lepaskan *jumper* yang terpasang pada CH1-CH2.
7. Lepaskan sumber tegangan pada *receiver* lalu masukan kembali, bila proses *binding* berhasil maka lampu led pada *receiver* perlahan-lahan akan berubah warna menjadi warna hijau. Menandakan *transmitter* dan *receiver* sudah terhubung satu sama lain.
8. Analisis hasil yang didapat.

### **3.6.2 Pengecekan Fungsi Receiver**

Pengecekan fungsi antara channel-channel yang ada pada *receiver* perlu dilakukan terlebih dahulu. Tujuan dari pengecekan ini adalah mengetahui channel-



channel yang ada pada *receiver* dapat digunakan sesuai kebutuhan dan tidak terjadi error ataupun rusaknya salah satu channel.

Pengecekan dilakukan melalui beberapa tahap, diantaranya adalah :

1. Pilih channel *receiver* yang akan dilakukan untuk pengujian.
2. Hubungkan dengan *remote control/transmitter* yang sudah tersambung dengan *receiver*.
3. Ambil beberapa kabel untuk digunakan sebagai media transmisi ke mikrokontroler dan osiloskop.
4. Masukkan perintah program pada mikrokontroler agar data dapat tertampil dan dapat mengetahui perubahan nilai PWM-nya.
5. Gelombang PWM pada pengujian channel akan terbaca pada alat osiloskop.
6. Analisis hasil yang didapat.

### 3.6.3 Pengecekan Fungsi Motor

Motor DC yang akan digunakan adalah motor DC PG45 berjumlah 4 buah motor. Motor DC PG45 memiliki internal encoder pada *gearbox* yang sudah terpasang dengan motor tersebut dan dapat melakukan test uji sinyal PWM menggunakan alat osciloscop. Pengecekan fungsi dan sistem dari keempat motor dilakukan dengan tujuan apakah keempat motor tersebut dapat digunakan sesuai fungsi sebelum dirancang menjadi *actuator* utama penggerak dari robot yang akan dirancang. Pengecekan motor dijelaskan dalam tahap-tahap sebagai berikut :

1. Siapkan motor DC PG45 serta beberapa kabel *jumper*.
2. Hubungkan catu daya motor dengan *driver* motor yang sudah terhubung dengan mikrokontroler, catu daya yang digunakan berupa baterai Li-Po 3 *cell*.
3. Masukkan *script code* pada mikrokontroler agar perintah dapat dieksekusi oleh mikrokontroler.
4. Program yang sesuai akan menghasilkan putaran dan arah gerak motor (*clockwise* atau *counterclockwise*).

5. Perubahan kecepatan dipengaruhi oleh nilai PWM yang dieksekusi oleh program, hasil dari perubahan PWM dapat dilihat pada gelombang yang tertampil pada osciloscop.
6. Bila semua sudah sesuai dan berfungsi seperti yang diperlukan maka motor dapat digunakan.
7. Analisis hasil data yang didapat.

#### 3.6.4 Pengaturan Motor Servo

Motor servo adalah motor yang memiliki *output* berupa pergerakan yang sesuai sudut. Sudut yang dihasilkan dapat diatur menggunakan eksekusi program pada mikrokontroler. Motor servo yang digunakan adalah motor servo berjenis MG996R. Motor servo ini memiliki tegangan operasi pada 4,8V – 7V. Karena sumber tegangan baterai yang digunakan adalah baterai Li-Po 3 *cell*, maka tegangan perlu diturunkan menggunakan regulator *stepdown*. Tahapan-tahapan pengaturan motor servo sehingga dapat digunakan dengan aman adalah sebagai berikut :

1. Atur regulator *stepdown* terlebih dahulu dengan menggunakan bantuan *power supply*.
2. Hubungkan *power supply* pada *input* regulator dan hubungkan multimeter pada *output* regulator.
3. Atur tegangan *output power supply* pada 12V.
4. Putar resistor variable pada regulator *stepdown* hingga tegangan *output* pada regulator berada pada 5V.
5. Setelah tegangan pada *output* regulator berada pada 5V, maka servo dapat digunakan sesuai spesifikasi motor.
6. Hubungkan pin data motor servo dengan pin digital yang ada pada mikrokontroler yang berfungsi untuk mengatur perubahan sudut pada motor servo.
7. Analisis hasil data yang didapat.

### 3.6.5 Pengukuran Kecepatan

Pengujian kecepatan dilakukan di lobby G5 dengan menggunakan jarak pandang 7 meter ke depan. Pengujian dilakukan sebanyak 10 kali diantaranya 5 kali pengujian maju dan 5 kali pengujian mundur.



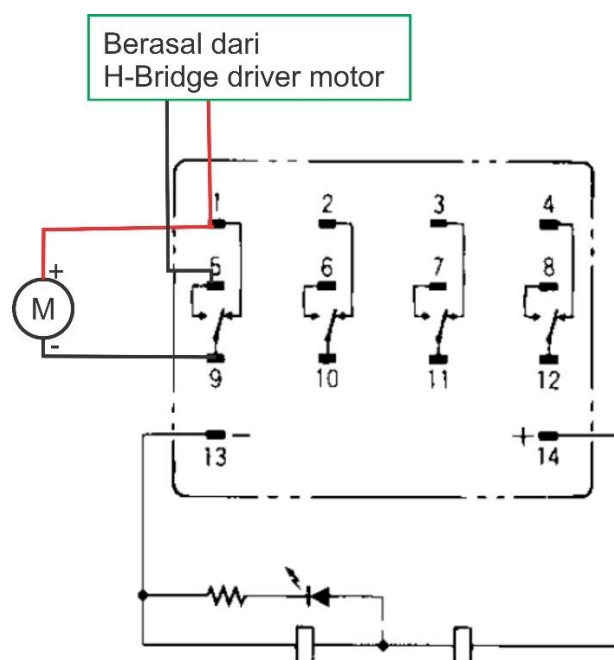
**Gambar 3.8** Aplikasi *Speedometer*

Gambar 3.8 merupakan aplikasi yang ada pada *smartphone* yang dapat diunduh. Aplikasi tersebut bernama *speedometer gps* yang menghitung kecepatan menggunakan sistem perubahan posisi yang tertampil pada GPS (*Global Positioning System*). Tahap-tahap pengukurannya akan dijelaskan sebagai berikut :

1. Buka aplikasi tersebut pada perangkat *smartphone* yang sudah diunduh sebelumnya.
2. Pasangkan *smartphone* pada robot.
3. Tetapkan jarak yang akan ditentukan, misalkan 7 meter dalam sekali jalan.
4. Jalankan robot dan ulangi sebanyak 10 kali percobaan agar mendapat hasil yang baik.
5. Bandingkan nilai yang didapat dalam setiap percobaan.
6. Hitung kecepatan rata-rata yang didapat.
7. Analisis setiap data yang didapat.

### 3.6.5 Sistem Pengereman Robot

Dalam pergerakan, harus ada sistem bagaimana robot tersebut dapat berhenti dengan baik. Dalam perancangan Tugas Akhir ini, robot menggunakan sistem pengereman yang menggunakan komponen listrik relay. Relay yang digunakan berjenis OMRON LY4N dimana memiliki 14 pin. Relay ini digunakan karena dapat bekerja pada tegangan 24 VDC yang sama digunakan pada motor DC PG45 yang digunakan.



**Gambar 3.9** Rangkaian Skematik LY4N

Pada gambar 3.9 merupakan rangkaian skematik yang ada pada relay LY4N. dimana pin 1 dihubungkan dengan kutub positif motor dan kutub dari H-Bridge yang ada pada *driver motor* BTS7960. Pin 5 dihubungkan dengan kutub H-Bridge lagi. Pin 9 dihubungkan dengan kutub negatif motor. Pada saat bergerak dan terjadi pengereman, maka switch pada relay akan berpindah mengarah pada sisi kutub lain dari H-bridge. Dalam H-bridge tersebut, kutub motor akan bertemu dan akan mengalami pengereman pada motor. Begitu juga dengan yang lainnya karena motor yang digunakan ada 4 buah motor dan 4 buah *driver motor*, maka selanjutnya sistemnya sama dan pin-pin yang dimasukkan menyesuaikan dengan apa yang sudah dijelaskan di awal.