BAB III
METODE PENILITIAN

Untuk menyelesaikan rumusan masalah dan merealisasikan tujuan penelitian maka diperlukan langkah-langkah metode untuk menyelesaikan masalah tersebut. Adapun langkah-langkah yang perlu dilakukan untuk merealisasikan sistem yang dirancang adalah penentuan studi literatur, spesifikasi alat, perancangan dan pembuatan alat, pengujian alat, dan pengambilan kesimpulan.

Sebelum menyusun langkah-langkah pembuatan alat, ada baiknya terlebih dahulu menentukan target sistem yang akan dibuat, sehingga alat yang akan dibuat jelas sistem dan fungsinya, berikut merupakan target sistem yang akan dibuat :

1. Robot dapat bergerak ke segela arah dengan menggunakan sistem *Three-Omni Directional*.
2. Robot dapat mengetahui arah hadap robot yang sebenarnya menggunakan sensor *Accelerometer* dan *Magnetometer* yang terdapat pada modul CMPS-11.
3. Menggunakan aplikasi android untuk memberikan *Set point* dan memantau arah hadap robot.

3.1. Penentuan Spesifikasi Alat

Spesifikasi alat secara keseluruhan ditetapkan terlebih dahulu sebagai acuan dalam perancangan selanjutnya. Spesifikasi alat yang direncanakan adalah sebagai berikut:

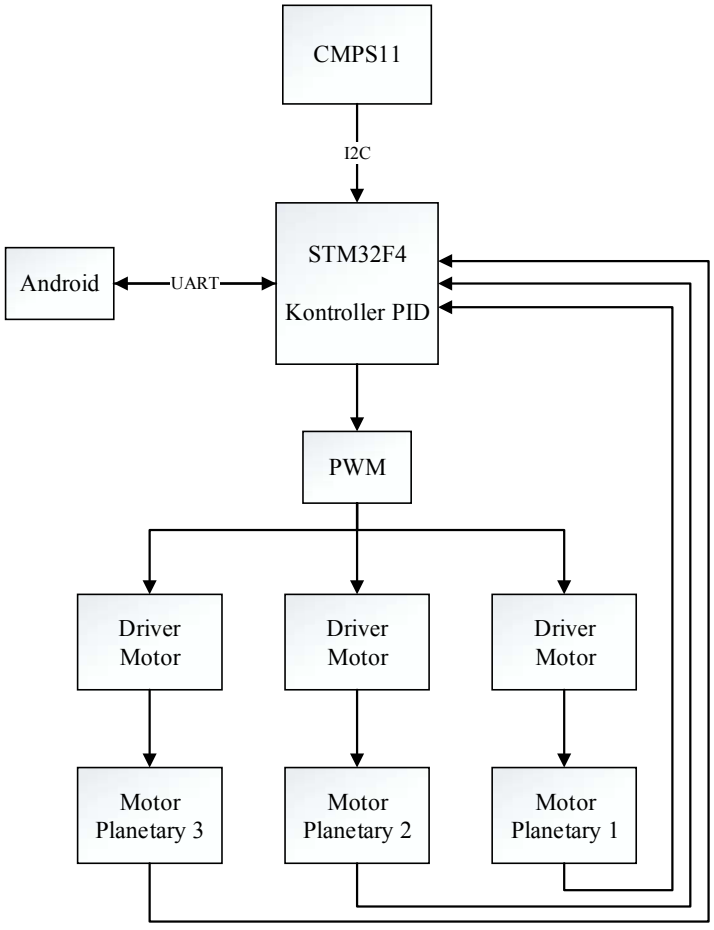
1. Mekanika robot berbahan dasar PCB dan mika *acrylic.*
2. Dimensi robot berbentuk lingkaran dengan diameter 25cm.
3. Menggunakan 3 buah motor DC sebagai actuator.
4. Menggunakan 3 buah roda *Omni* dengan diameter 5cm.
5. Menggunakan IMU CMPS-11 untuk mendapatkan nilai heading yang sebenarnya.
6. Menggunakan sistem *omni-kinematics* untuk menggerakan robot.
7. Driver motor yang digunakan adalah sebuah L298N H-bridge untuk menggerakkan dua buah motor DC.
8. Menggunakan mikrokontroller STM32F4.
9. Menggunakan modul *Bluetooth* HC-05 untuk komunikasi robot dengan android.
10. Menggunakan 1 sumber tegangan 12V dari baterai lippo.
11. Menggunakan modul regulator switching LM2576 *simple switcher* untuk memberikan tegangan 5V ke STM32F4, HC-05, dan CMPS-11.

3.2. Perancangan dan Pembuatan Alat

Dalam perancangan kali ini, yang pertama dilakukan penulis adalah membuat sebuah blok diagram sistem kesulurhan, setelah itu membuat spesifikasi mekanik robotnya. Dan yang terakhir adalah perancangan bagian *software.*

3.2.1. Perancangan blok diagram kesuluruhan

Pada perancangan alat diperlukan perencanaan blok diagram sistem yang dapat menjelaskan sistem secara garis besar dan diharapkan alat dapat bekerja sesuai dengan rencana



Gambar 3.1 Diagram Blok Sistem

Fungsi dari masing-masing bagian dalam blok ini adalah sebagai berikut :

1. CMPS-11 yang digunakan untuk memantau *heading* robot.
2. Pada perancangan ini, perancang menggunakan android sebagai kontrolernya. Yang di mana pengiriman data dari android ke mikrokontroler STM32F4 menggunakan koneksi *Bluetooth.* Data yang dikirim berupa arah gerak robot yang diingnkan. Dan mikrokontroler juga akan mengirim data ke android berupa arah gerak robot sebenarnya yang didapat dari CMPS-11.

3. STM32F4 sebagai mikrokontroler utama digunakan untuk mendapatkan data dari CMPS-11 dan data arah gerak dari *user* serta mengolahnya dengan fungsi kontroler PID untuk dijadikan PWM, yang di mana PWM akan digunakan untuk menggerakan motor.
4. Android digunakan sebagai media *user* untuk mengendalikan robot serta memonitoring arah gerak robot.
5. Motor digunakan sebagai actuator utama robot.
6. Dari motor *planetary* akan menghasilkan keluaran yang berupa data dari *rotary encoder* yang akan menjadi *feedback* untuk mikrokontroler.

Prinsip kerja sistem ini adalah awalnya robot akan menerima data dari user yang berupa arah gerak robot. Lalu pada mikrokontroler data tersebut akan diolah dan dijadikan PWM untuk masing-masing motor sehingga robot akan bergerak ke arah yang sudah ditentukan sebelumnya. Saat robot bergerak, mikrokontroler juga menerima data dari CMPS-11, pada mikrokontroler data dari IMU akan diolah dan dijadikan nilai *heading* aktual robot, lalu nilai *heading* akan dikirim ke *user* untuk *monitoring* serta diolah menggunakan sistem PID untuk menjaga arah gerakan robot sesuai dengan arah robot yang diinginkan *user*. Selain menerima data dari IMU, mikrokontroler juga menerima data dari *rotary encoder,* yang di mana data tersebut dapat diolah sehingga menghasilkan nilai kecepatan putaran roda dalam satuan *Radian/minute* (RPM). Lalu data RPM juga akan di kontrol dengan PID sehingga kecepatan putaran motor akan setabil.

3.2.2. Perancangan Mekanika Robot

Sistem mekanika yang baik, akan mendukung pergerakan robot menjadi lebih baik pula, oleh sebab itu perancangan mekanik dalam hal ini bodi dan rangka robot haruslah proporsional dengan berat, panjang dan lebar serta tinggi dari robot. Agar robot memiliki beban yang tidak terlalu berat maka untuk perancangan dan pembuatan perangkat keras mekanika dan elektrik di desain menjadi satu kesatuan.

Untuk rincian dimensi robot adalah sebagai berikut :

- Diameter robot = 25cm
- Tinggi robot maksimum = 15cm.

3.2.3. Perancangan Perangkat Lunak

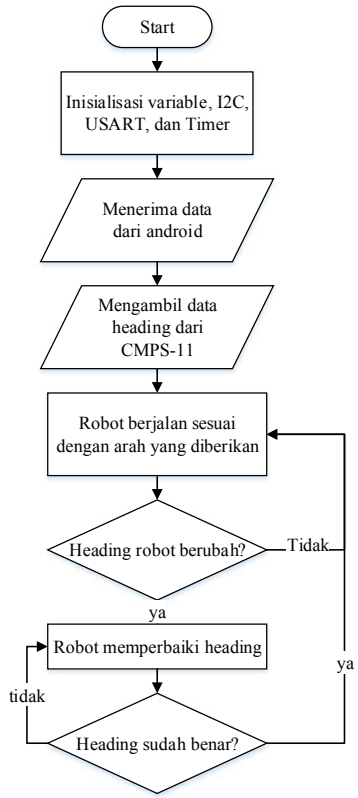
Perancangan *Software* dilakukan dengan merancang diagram alir (*flowchart*) terlebih dahulu. Diagram alir ini berfungsi sebagai alur kerja untuk setiap perangkat keras yang dikendalikan oleh mikrokontroler STM32 Nucleo serta proses-proses perhitungan yang dikerjakan oleh mikrokontroler. Bahasa pemrograman yang digunakan dalam program utama adalah bahasa C dengan menggunakan *compiler CooCox*.

3.2.3.1. *Flowchart* sistem kesuluruhan

Secara umum, saat robot dinyalakan, hal yang pertama dilakukan adalah mengkonfigurasi fungsi I2C, PWM, serta pin-pin yang digunakan. Setelah konfigurasi

selesai, Selanjutnya adalah menerima data dari android/*user* yang berupa arah gerak robot yang diinginkan, lalu data yang diterima tersebut diolah di mikrokontroler dan dijadikan PWM untuk mengendalikan kecepatan motor serta mengontrol dua pin keluaran untuk mengendalikan arah putaran motornya.

Selagi robot berjalan, mikrokontroler akan menerima data dari modul CMPS-11 dan data yang dihasilkan oleh *rotary encoder*. Data dari CMPS-11 akan diolah untuk menjaga *heading* robot, sedangkan data yang dihasilkan oleh *rotary encoder* akan diolah menjadi kecepatan putaran roda dan digunakan untuk menjaga kestabilan kecepatan putaran roda. Untuk lebih jenisnya dapat dilihat pada diagram alir di bawah ini.



Gambar 3.2 Diagram alir keseluruhan pada mikrokontroler

3.2.3.2. Perancangan Rangkaian Antarmuka Mikorkontroler Utama

Pada perancangan alat ini, digunakan STM32F4 sebagai mikrokontroler utama, berikut perancangan pin-pin yang digunakan

PIN A.0 = dihubungkan dengan pin ch.a *rotary encoder* motor1

PIN A.1 = dihubungkan dengan pin ch.a *rotary encoder* motor2

PIN A.2 = dihubungkan dengan pin ch.a *rotary encoder* motor3

PIN B.0 = dihubungkan dengan pin ch.b *rotary encoder* motor1

PIN B.1 = dihubungkan dengan pin ch.b *rotary encoder* motor2

PIN B.3 = dihubungkan dengan *input 4 driver* motor L298N pertama

PIN B.4 = dihubungkan dengan pin ch.b *rotary encoder* motor3

PIN B.6 = dihubungkan dengan pin SCL CMPS-11

PIN B.7 = dihubungkan dengan pin SDA CMPS-11

PIN B.10 = dihubungkan dengan pin TX HC-05

PIN B.11 = dihubungkan dengan pin RX HC-05

PIN B.15 = dihubungkan dengan *input 1 driver* motor L298N pertama

PIN D.1 = dihubungkan dengan *input 1 driver* motor L298N kedua

PIN D.3 = dihubungkan dengan *input 2 driver* motor L298N kedua

PIN D.7 = dihubungkan dengan *input 3 driver* motor L298N pertama

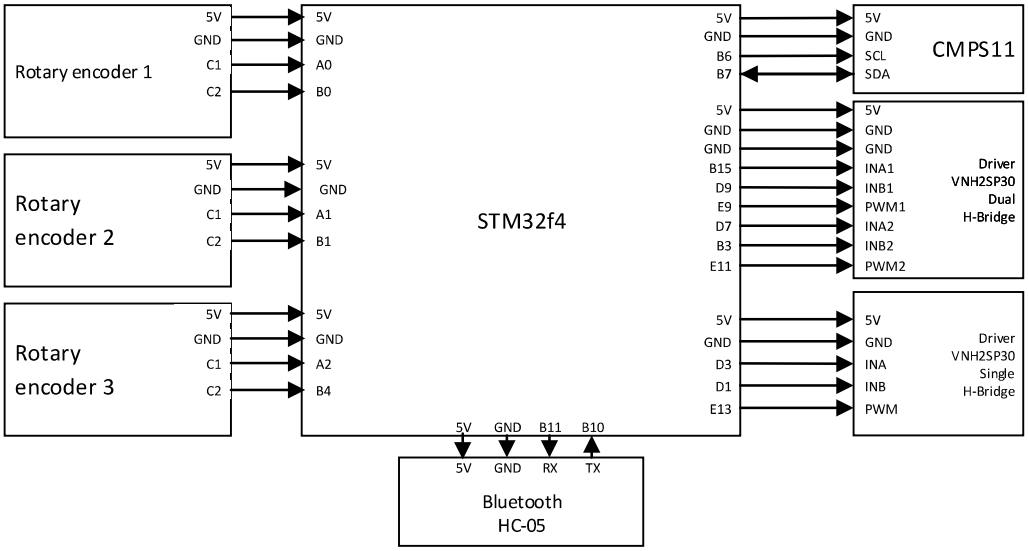
PIN D.9 = dihubungkan dengan *input 2 driver* motor L298N pertama

PIN E.9 = dihubungkan dengan *ENABLE* A *driver* L298N pertama

PIN E.11 = dihubungkan dengan *ENABLE* B *driver* L298N pertama

PIN E.13 = dihubungkan dengan *ENABLE* A *driver* L298N kedua

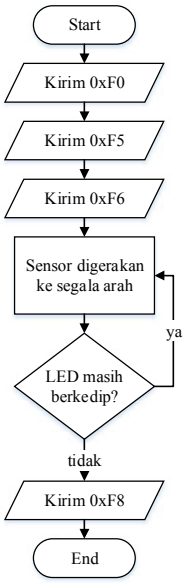
Adapun diagram antarmuka mikorkontroler utama secara kesuluruhan dalam gambar 13.



Gambar 3.3 Diagram Blok Antarmuka Mikrokontroler Utama Secara Keseluruhan

3.2.3.3. Kalibrasi Sensor Kompas CMPS-11

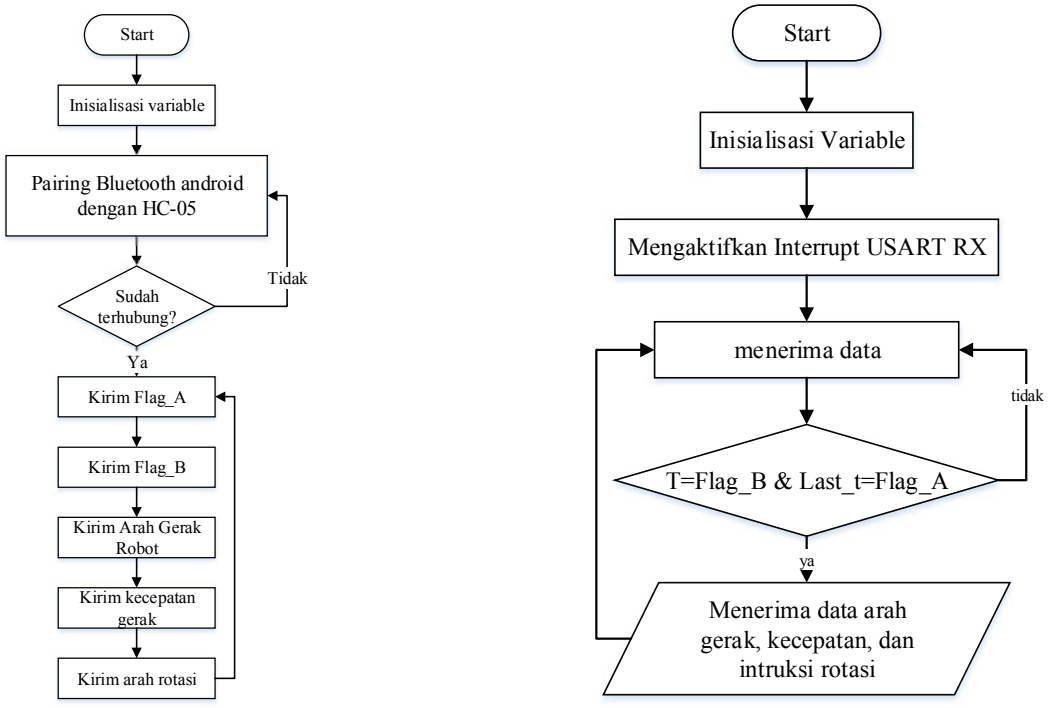
Sebelum sensor digunakan, sensor ini harus terlebih dahulu di kalibrasi untuk mengatur ulang penguatan sensor dan offset dari *magnetometer* dan *accelerometer*. Pertama-tama hal yang harus dilakukan adalah masuk ke mode kalibrasi sensor dengan mengirimkan urutan 3-*byte* yang terdiri dari 0xF0, 0xF5, dan 0xF6, setelah itu sensor digerakan ke segala arah, jika LED sudah tidak berkedip lagi, itu menandakan proses pengambilan data kalibrasi sudah selesai, dan selanjutnya adalah keluar dari mode kalibrasi dengan mengirim data 0xF8. Agar lebih jelasnya dapat dilihat pada gambar 14 di bawah ini yang merupakan diagram alir kalibrasi sensor kompas CMPS-11



Gambar 3.4 Diagram alir kalibrasi sensor kompas CMPS-11

3.2.3.4. Perancangan Komunikasi Android dengan STM32F4

Proses komunikasi antara android dengan STM32F4 terbagi menjadi dua proses, yaitu proses pengiriman data pada android, dan proses penerimaan data pada STM32F4. Gambar 14 (a) dan (b) di bawah ini merupakan diagram alir proses pengiriman data pada android dan penerimaan data pada STM32F4



(a) proses pengiriman data (b) proses penerimaan data

Gambar 3.5 Proses pengiriman (a) dan penerimaan (b) data

dari gambar 14 di atas, diketahui data yang pertama dikirim oleh android adalah data FLAG_A dan disusul oleh FLAG_B, setelah kedua data tersebut dikirim, barulah data

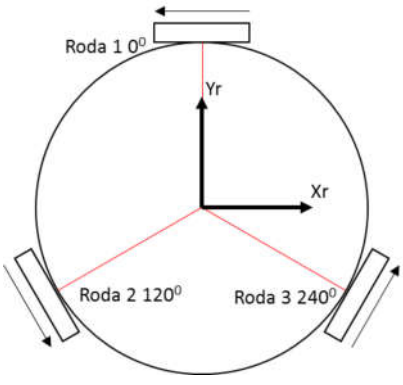
bergerakan robot dikirm. Pada STM32F4, proses penerimaan data merupakan sebuah looping yang akan terus berjalan selama STM32F4 menerima data. Pada siklus pertama, *variable* T akan menerima data FLAG_A dan *variable* LAST_T akan menyimpan data T yang nilainya adalah FLAG_A, lalu pada siklus kedua, nilai T akan berubah menjadi FLAG_B, saat T=FLAG_B dan LAST_T=FLAG_A, maka proses penerimaan data untuk pergerakan data dimulai, cara ini digunakan agar tidak ada pergeseran data untuk pergerakan robot. Gambar 15 di bawah ini menunjukan *source code* untuk proses penerimaan data

```
void USART3_IRQHandler(void){
    if( USART_GetITStatus(USART3, USART_IT_RXNE) ){
        char t = USART3->DR; // the character from the USART3 data register is saved in t
        char static last_data;
        Pin(GPIOD,TO,P_15);
        tanda_BT2++;
        switch(angka1)
        {
            case 0 : vxy=USART3->DR; angka1+=1; break;
            case 1 : GetSudut_v=USART3->DR; angka1+=1; break;
            case 2 : kec_w=USART3->DR; angka1+=1; break;
            case 3 : SetAksel_v=USART3->DR; angka1+=1; break;
            case 4 : SetAksel_v2=USART3->DR; angka1+=1; break;
            case 5 : re_center_flg=USART3->DR; angka1=100; break;
        }
        if(t==170&&last_data==85)angka1=0;
        last_data=t;
        if(kec_w == 40)kec_w= 100;
        else if(kec_w == 216)kec_w= -100;

        GetAksel_v = SetAksel_v+SetAksel_v2;
    }

}
```

Gambar 3.5 *Source Code* proses penerimaan data pada STM32F4

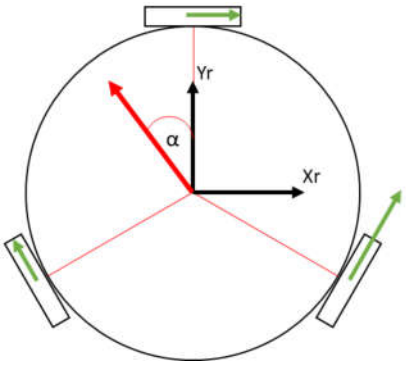
3.2.3.5.Persamaan untuk Memberi Nilai Kecepatan dan Arah Putaran Motor

Untuk memberi nilai kecepatan putaran dan arah putaran motor pada robot *omni* beroda tiga, pada perancangan ini digunakan rumus yang bersumber dari jurnal "*Three Omni-Directional Wheels Control On A Mobile Robot*" yang dibuat oleh F. Riberio dkk. Pada jurnal tersebut, setiap roda diletakan sedemikian rupa yang dimana titik rotasinya atau pusat roda menghadap ke titik tengah robot, dan terdapat sudut 120^0 antara roda. Untuk lebih jelasnya dapat dilihat pada gambar 3.6 dibawah ini



Gambar 3.6 Penempatan motor dan roda robot

Dari gambar 3.6 roda satu, dua, dan tiga pada posisi 0^0, 120^0, dan 240^0. Lalu diasumsikan bahwa arah depan robot adalah 0^0. Dan arah panah menunjukan arah positif setiap motor.



Gambar 3.7 Arah yang diinginkan serta perputaran motor

Gambar 3.7 menunjukan bagaimana respon robot jika diberi pergerakan yang diinginkan (anak panah merah) dengan α sebagai arah sudut, dan panjang anak panah merah merupakan kecepatan yang diinginkan. Lalu garis hijau adalah hasil dari masukan yang diberikan (arah dan kecepatan).

Karena arah depan robot diwakili Yr, masing-masing kecepatan motor terdiri dari sinus sudut α (arah yang diinginkan) diproyeksikan pada posisi setiap roda, dan dikalikan dengan kecepatan yang diinginkan. Maka dari itu didapatkan persamaan

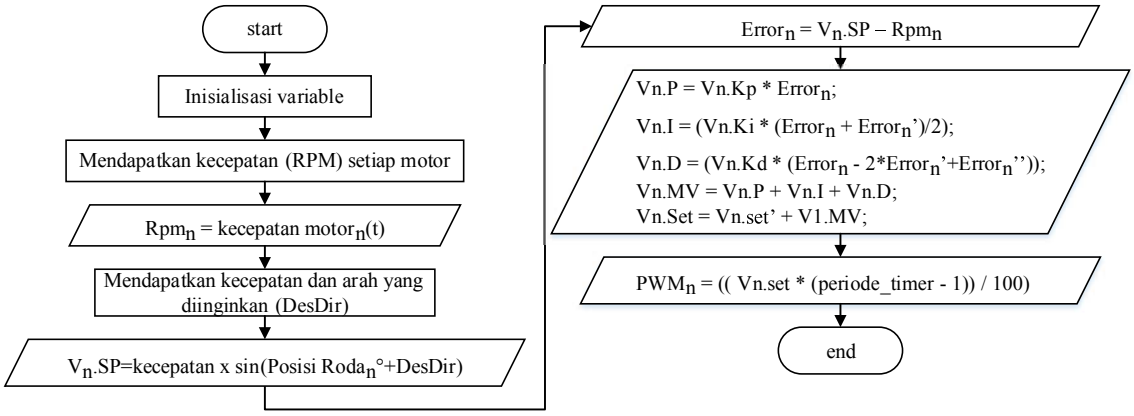
Vn.SP=kecepatan x sin(posisi roda° + α) .....(1)

Dengan posisi setiap roda adalah 0^0, 120^0, dan 240^0, persamaan untuk masing-masing roda adalah sebagai berikut

V1.SP=kecepatan x sin(0°+α) .....(2)
V2.SP=kecepatan x sin(120°+ α) .....(3)
V3.SP=kecepatan x sin(240°+ α) .....(4)
ket :
Vn.SP : kecepatan roda ke-n(RPM)
α: arah gerak yang ingin dituju(derajat).

3.2.3.6.Perancangan Pemberian *Duty Cycle* Sinyal PWM untuk *Driver* L298N

Untuk memberi nilai atau mengatur besar *duty cycle* PWM yang akan menjadi masukan *driver* L298N, digunakan persamana *invers kinematic* yang sudah dijelaskan pada sub-bab sebelumnya yang masukannya adalah kecepatan roda yang diinginkan dan arah yang diinginkan dalam bentuk sudut. Lalu dari persamaan ini, didapatkan kecepatan setiap roda yang dibutuhkan agar dapat bergerakan ke arah yang diinginkan serta arah putaran rodanya. Pada Gambar 3.7 dapat dilihat bagaimana proses pemberian nilai *duty cycle* PWM untuk *driver* L298N.



Gambar 3.8 Diagram Alir Menghasilkan Nilai *Duty Cycle* PWM

Keterangan :
n : 1,2, dan 3.
Rpm_n : Kecepatan Motor ke-n aktual(RPM),
Kecepatan : Kecepatan gerak robot yang diinginkan (RPM),
α : arah gerak robot yang diinginkan (derajat),
Vn.SP : kecepatan motor ke-n sesuai dengan posisi penempatan roda ke-n(RPM),
$Error_n$: kesalahan kecepatan motor ke-n (RPM),
Vn.P : hasil kontroler *proportional* untuk motor ke-n.
Vn.I : hasil kontroler *Integral* untuk motor ke-n.
Vn.D : hasil kontroler *Derivative* untuk motor ke-n.
Vn.Set : keluaran kontroler PID untuk motor ke-n (%).
PWMn : nilai untuk *register timer* PWM motor ke-n.

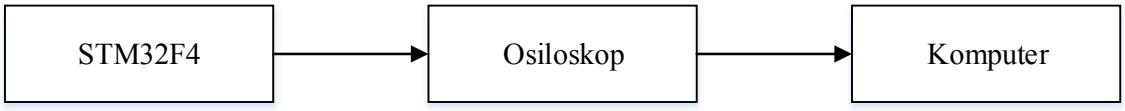
3.3. Pengujian Alat

Untuk mengetahui kinerja alat apakah sudah sesuai dengan yang direncanakan maka dilakukanlah pengujian alat. Pengujian dilakukan pada masing-masing bagian dan kemudian secara kesuluruhan sistem. Secara garis besar pengujian dilakukan adalah sebagai berikut :

3.3.1. Pengujian Keluaran *Duty Cycle* PWM STM32F4

Pengujian ini dilakukan untuk mengetahui apakah keluaran *duty cycle* PWM dari STM32F4 sudah sesuai yang diharapkan atau tidak. Pengujian dilakukan dengan menggunakan osiloskop digital dan PC untuk menampilkan hasilnya.

Prosedur pengujian dilakukan dengan cara menghubungkan STM32F4 dengan osiloskop, lalu osiloskop dihubungkan ke perangkat PC untuk dipantau hasilnya. PWM yang akan diuji sebesar 0%, 25%, 75%, dan 100%. Gambar 17 di bawah ini menunjukan skema pengujian *duty cycle* PWM yang dilakukan

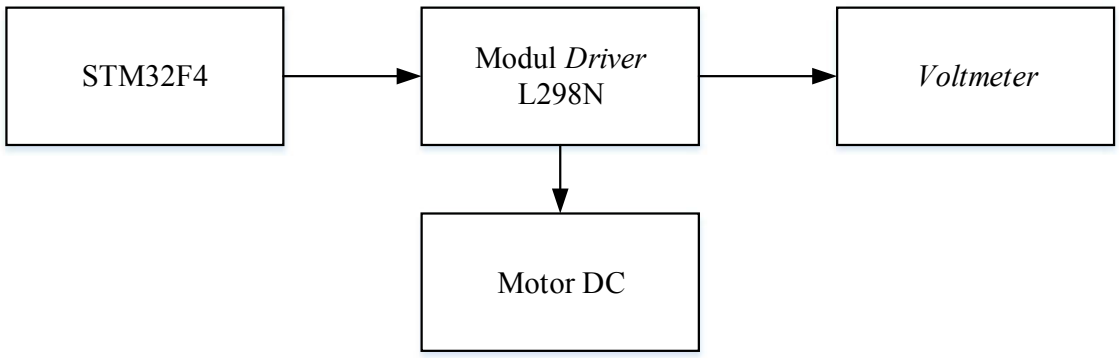


Gambar 3.9 Skema pengujian *duty cycle* PWM STM32F4

3.3.2. Pengujian Modul *Driver* Motor L298N

Pengujian ini dilakukan untuk mengetahui bagaimana keluaran tegangan dari masing masing *driver* motor saat diberikan tegangan masukan yang sama dan *duty cycle* PWM yang sama.

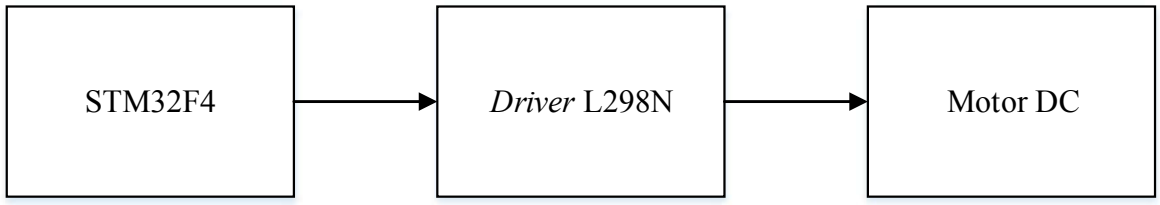
Prosedur pengujian dilakukan dengan menghubungkan STM32F4 dan modul *driver* motor L298N, pada pengujian ini motor DC juga dihubungkan ke keluaran dari L298N. lalu tegangan di cek menggunakan *voltmeter* seperti pada gambar 18 di bawah ini.



Gambar 3.10 Skema pengujian tegangan keluaran L298N

3.3.3. Pengujian Arah Putaran Motor

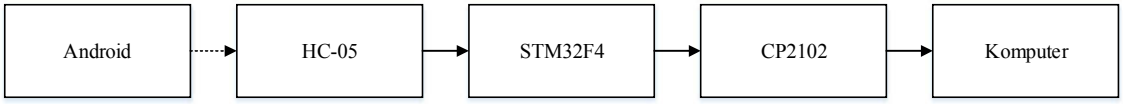
Pengujian ini dilakukan untuk mengetahui apakah arah putaran motor sesuai dengan masukan logika *driver* L298N. Alat yang digunakan dalam pengujian ini adalah STM32F4, *driver* L298N, motor DC, dan catu daya 5V dan 12V. Prosedur pengujian dilakukan dengan menghubungkan keluaran logika STM32F4 dengan masukan logika *driver* L298N, lalu keluaran tegangannya dihubungkan dengan motor DC sesaui pada gambar 19 d bawah ini



Gambar 3.11 Skema pengujian arah putaran motor

3.3.4. Pengujian Transmisi Data *Bluetooth* HC-05

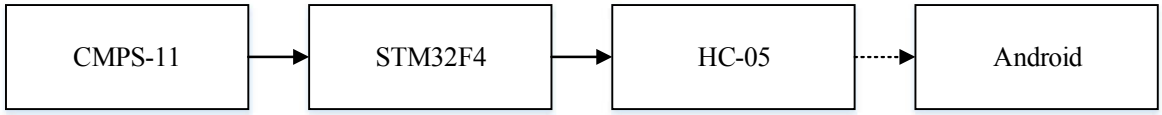
Pengujian ini dilakukan untuk mengetahui apakah data yang dikirim oleh android sudah diterima dengan baik di STM32F4. Prosdur pengujian ini dilakukan dengan mengatur HC-05 agar dapat terkoneksi dengan alamat *device Bluetooth* yang ada di sekitarnya dan mengatur *baudrate*nya menjadi 9600. Setelah itu dilakukan *pairing* antara *Bluetooth* android dengan HC-05, dan selanjutnya data yang diterima STM32F4 dapat dicek pada PC menggunakan modul CP2102. Untuk lebih jelasnya dapat dilihat pada gambar 20 di bawah ini



Gambar 3.12 Skema pengujian transmisi data

3.3.5. Pengujian Nilai *Heading* dari CMPS-11

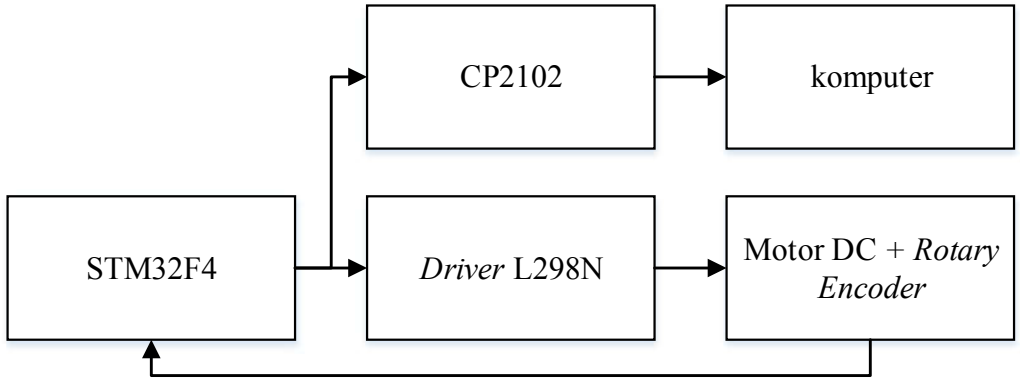
Pengujian ini dilakukan untuk mengetahui apakah data *heading* yang diterima dari sensor kompas CMPS-11 sudah sesuai dengan perputaran robot. Alat-alat yang diperlukan dalam pengujian ini antara lain sensor kompas CMPS-11, STM32F4, HC-05, android, serta catu 5V. Pengujian dilakukan dengan cara memutar robot setiap 45^0 lalu dibandingkan dengan nilai *heading* kompas CMPS-11, nilai *heading* akan dimonitoring melalui aplikasi android dengan *Bluetooth* HC-05 sebagai media pengirimannya dan hasilnya dicatat untuk dibuat persamaan jika nilai *heading* CMPS-11 tidak sesuai dengan perputaran robot yang sebenanya. Gambar 21 dibawah ini menunjukan skema pengujian CMPS-11



Gambar 3.13 Skema pengujian CMPS-11

3.3.6. Pengujian Kecepatan Motor

Pengujian ini dilakukan untuk memastikan kecepatan motor yang dihasilkan sesuai dengan kecepatan yang diinginkan. Alat-alat yang digunakan dalam pengujian ini adalah STM32F4, *driver* motor L298N, motor DC, CP2102, catu daya 5V dan 12V. Pengujian ini dilakukan dengan cara yang sama seperti pengujian arah putaran motor, hanya saja pada pengujian ini nilai yang ditetapkan adalah kecepatan motor dalam satuan RPM dan juga terdapat *feedback rotary encoder* untuk mengetahui kecepatan motor yang sesungguhnya, serta pada pengujian kali ini digunakan kontroller PID untuk mengontrol motor. Gambar 22 berikut menunjukan skema pengujian kecepatan motor



Gambar 3.14 Skema Pengujian Kecepatan Motor

3.3.7. Pengujian Arah Gerak Robot

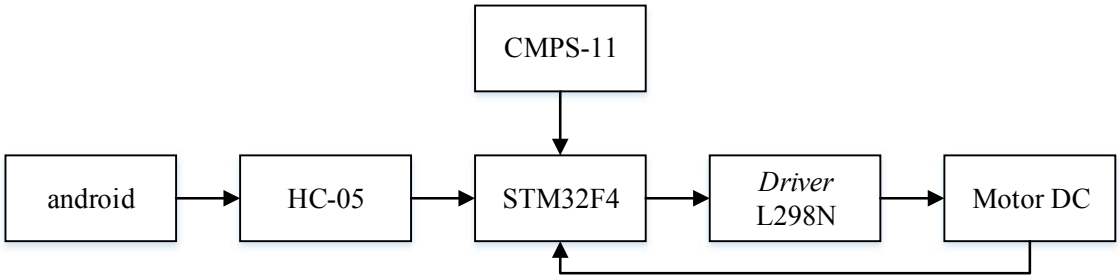
Pengujian ini dilakukan untuk mengetahui apakah persamaan yang digunakan sudah benar atau masih memiliki kesalahan. Pengujian ini menggunakan robot yang

sudah jadi sepenuhnya. Lalu pengujian dilakukan dengan cara memberikan arah sudut yang ingin dituju, lalu respon robot diamati dan dicatat. Alat-alat yang dibutuhkan dalam pengujian ini antara lain, android, hc-05, STM32F4, *driver* motor L298N, motor DC, catu tegangan 5V dan 12V dan gambar 23 di bawah ini menunjukan skema pengujian

Gambar 3.15 Skema pengujian arah gerak robot

3.3.8. Pengujian Keseluruhan

Pengujian ini dilakukan untuk mengetahui apakah robot sudah bekerja dengan hasil yang diharapkan atau tidak. Pengujian ini dilakukan sama seperti pengujian arah gerak robot, akan tetapi pada pengujian kali ini digunakan CMPS-11 yang keluarannya merupakan *feedback heading* robot yang sesungguhnya. Pada pengujian kali ini akan dibandingkan pergerakan robot saat tidak menggunakan CMPS-11 dan saat menggunakan CMPS-11, yang dimana saat tidak menggunakan CMPS-11, robot tidak menggunakan fungsi untuk memperbaiki *heading*nya, dan saat menggunakan CMPS-11 robot memiliki fungsi yang diharapkan dapat memperbaiki *heading*nya secara otomatis jika *heading* robot sewaktu-waktu berubah. Gambar 24 di bawah ini menunjukan skema pengujian keseluruhan



Gamabr 3.16 Skema pengujian keseluruhan robot