

BAB III

METODE PENELITIAN

Perancangan perangkat keras dan lunak dilakukan dengan metode penelitian yang didasarkan pada studi kepustakaan berupa data-data literatur dari masing-masing komponen, informasi dari *internet*, dan konsep-konsep teoretis dari buku-buku penunjang.

Dari data-data yang diperoleh maka dilakukan perencanaan rangkaian perangkat keras. Dalam perangkat keras ini, penulis akan melakukan pengujian perangkat keras dengan program-program yang telah dibuat, pembuatan perangkat lunak adalah tahap selanjutnya. Terakhir adalah penggabungan perangkat keras dengan kerja perangkat lunak yang telah selesai dibuat.

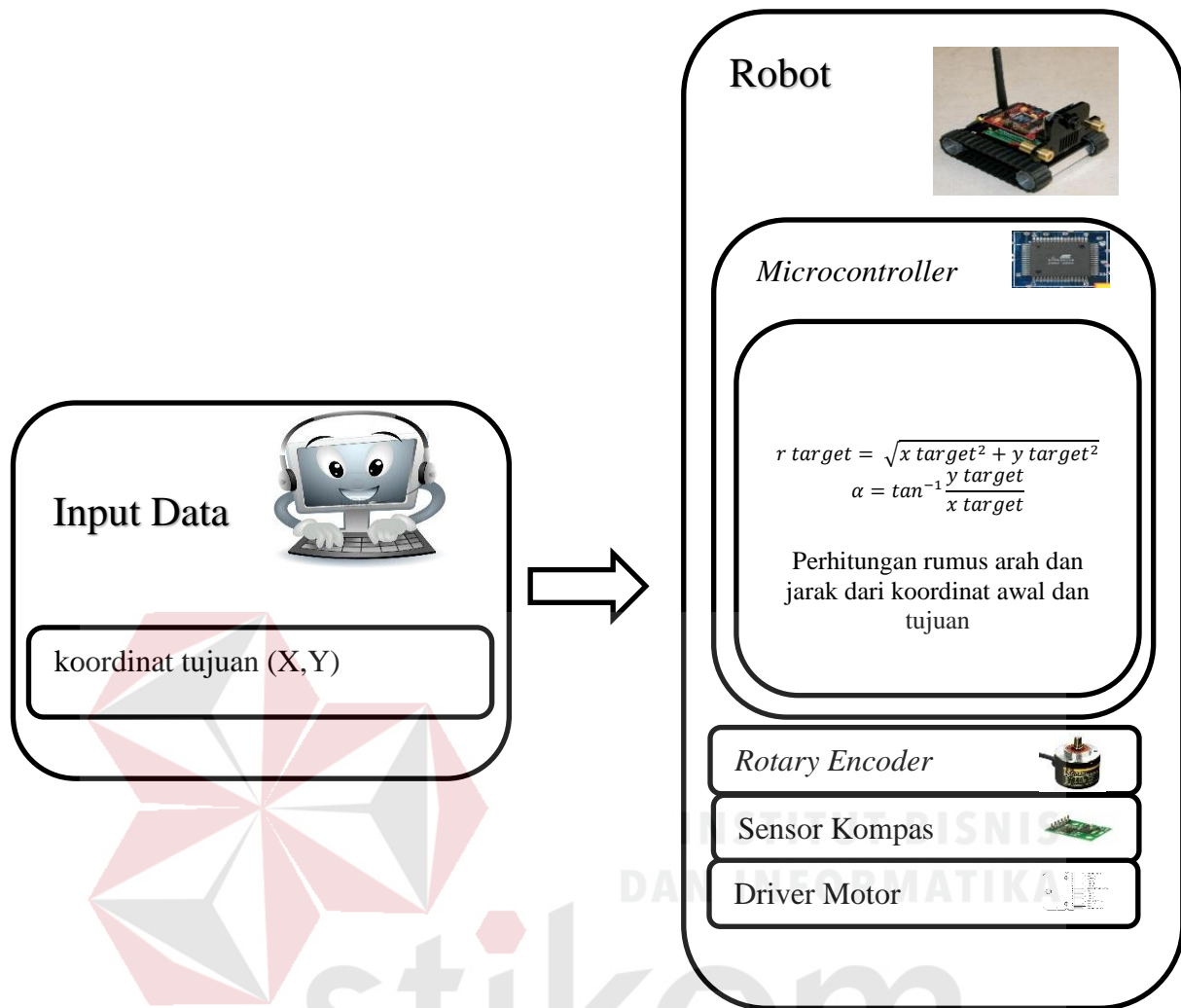
Pada bab ini akan dibahas mengenai masalah yang timbul dalam perencanaan dan pembuatan perangkat keras (*hardware*) maupun perangkat lunak (*software*). Dari kedua bagian tersebut akan dipadukan agar dapat bekerja sama untuk menjalankan sistem yang baik.

Perencanaan ini diperlukan sebelum proses pembuatan sistem tersebut, Perancangan ini berguna agar pengerjaan tahapan selanjutnya berjalan dengan lancar. Tahapan-tahapannya meliputi tahap pembuatan perangkat keras, perangkat lunak dan menggabungkan keduanya.

3.1 Blok Diagram Sistem

Dari penelitian ini terdapat dua proses utama yang dijalankan, yaitu proses penentuan arah hadap robot terhadap arah koordinat tujuan dan proses penentuan

jarak tempuh pada robot dalam menuju koordinat tujuan. Dimana pertama-tama *input* koordinat *download* kedalam mikrokontroler bersama dengan program untuk menjalankan *mobile robot*. Setelah mendapatkan *input* koordinat yang berupa koordinat mana saja yang harus ditempuh oleh robot, robot akan melakukan penentuan arah hadap robot terhadap koordinat tujuan. Proses penentuan arah ini dilakukan dengan mengambil data pada sensor kompas digital yang kemudian diteruskan dengan perhitungan rumus Trigonometri. Sehingga setelah proses ini dilakukan *mobile robot* akan mendapatkan data tentang arah tujuan robot dan mampu menyesuaikan arah hadap robot semula menjadi arah hadap robot terhadap koordinat tujuan. Setelah robot menghadap pada arah hadap yang benar, dilakukan proses penentuan jarak tempuh menuju koordinat tujuan. Proses ini dilakukan menggunakan perhitungan rumus Pythagoras dan melakukan *update* pada *rotary encoder*. Sehingga setelah proses ini dilakukan robot mampu mengetahui jarak yang harus di tempuh untuk menuju koordinat tujuan dan robot mempunyai kemampuan untuk berhenti saat jarak yang dilalui oleh robot sudah memenuhi jarak tempuh yang harus dilalui. Pada gambar 3.1 adalah diagram blok keseluruhan dari sistem ini.



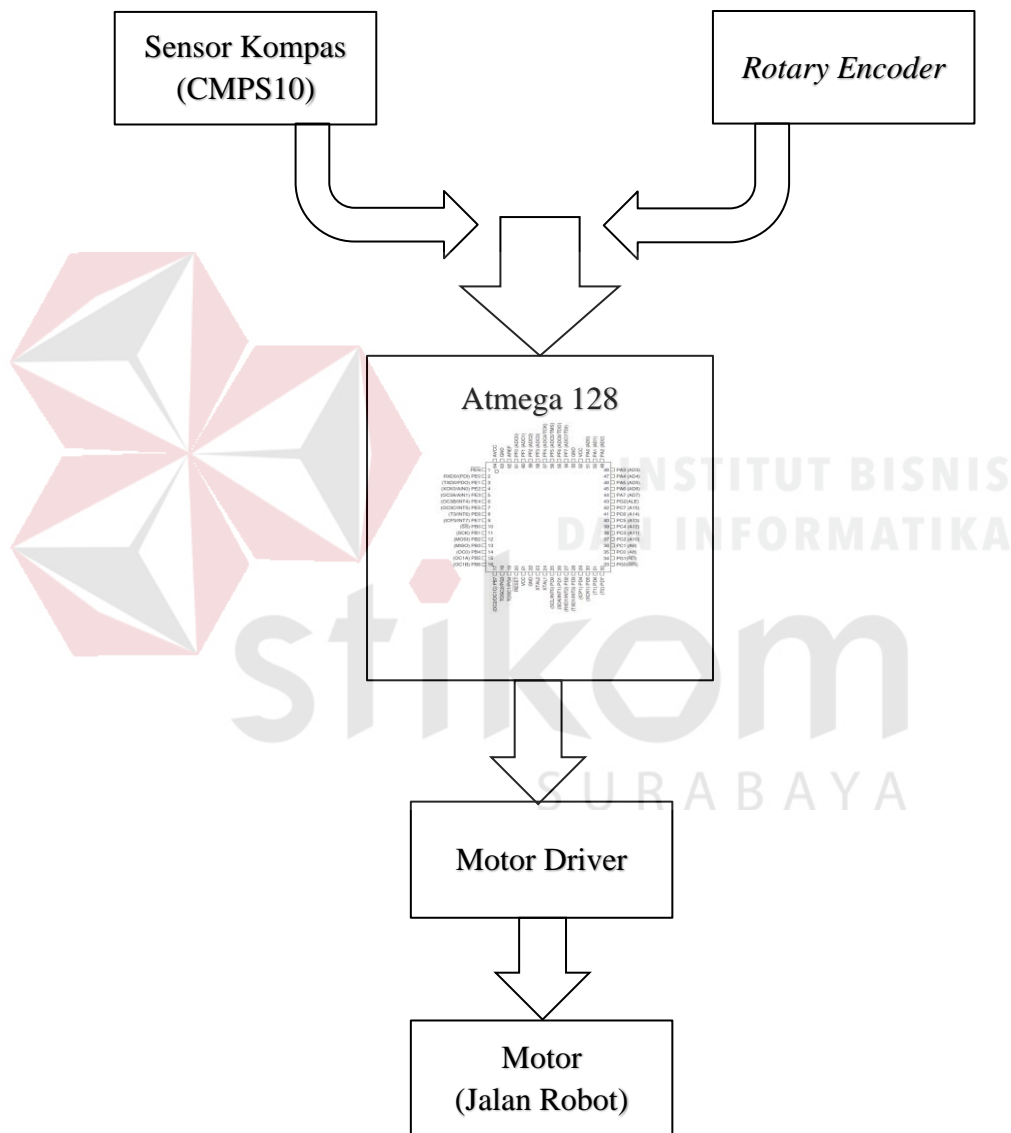
Gambar 3.1 Diagram blok keseluruhan dari sistem

3.2 Perancangan Perangkat Keras

Perancangan perangkat keras pada sistem ini dilakukan berdasarkan blok diagram sistem keseluruhan yang terdapat pada Gambar 3.1.

Dalam blok diagram pada gambar 3.1, mikrokontroler yang bertugas sebagai pemroses akan mendapatkan data *input* dari sensor kompas digital dan *rotary encoder*. Sensor kompas akan memberikan data berupa sudut arah mata angin dalam bentuk digital. Sedangkan rotary encoder akan memberikan data berupa *pulse* sebagai penghitung jarak tempuh robot. Kemudian mikrokontrol

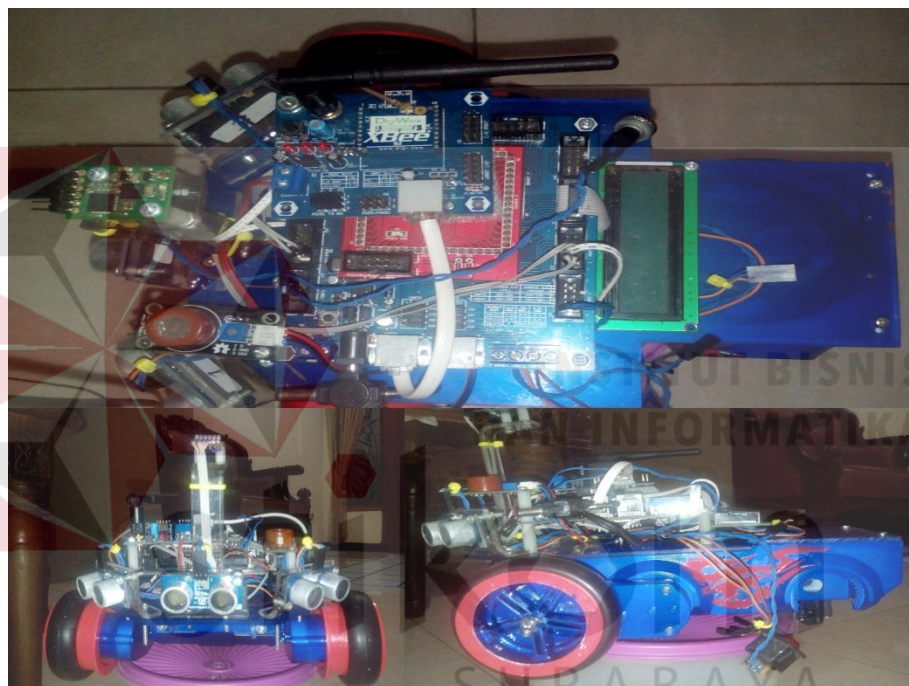
mengolah data tersebut, dalam hal ini pengolahan data pada mikrokontrol menggunakan rumus perhitungan arah dan jarak. Hasil dari rumus tersebut adalah agar robot dapat mengetahui arah dan jarak tempuh yang harus dilalui untuk mencapai koordinat tujuan. Berikut adalah gambar blok diagram perangkat keras secara keseluruhan :



Gambar 3.2 Blok diagram perangkat keras secara keseluruhan

3.2.1 Perancangan Mekanik Robot

Mekanik robot yang digunakan adalah Daguer 5 yang telah dimodifikasi. Daguer 5 adalah generasi baru *chassis* robot yang dirancang khusus untuk kepentingan penelitian dan hobi. Rangkaian elektronika mulai dari rangkaian *microcontroller*, *driver motor*, sensor kompas digital, dan *rotary encoder* akan diletakkan pada *chassis* robot ini.



Gambar 3.3 Tampilan keseluruhan robot

Berikut arsitektur secara detil dari gambar 3.3 :

Ukuran dimensi

Setelah semua komponen penunjang dari penelitian ini dipasangkan pada robot, maka dimensi robot menjadi:

Ukuran robot : 23cm (lebar) x 25cm (panjang) x 12cm (tinggi)

Struktur Material

Bahan material yang digunakan :

- a. Bagian rangka
 1. Dagu Rover 5
 2. Mur dan baut

- b. Bagian dari penggerak robot
 1. Motor DC 7,2 Volt
 2. Roda
 3. Roda bebas

3.2.2 Perancangan *Minimum System*

Pada proyek tugas akhir ini dibuat piranti pengendali menggunakan *microcontroller* keluaran AVR, yaitu ATmega128. Untuk mengaktifkan atau menjalankan *microcontroller* ini diperlukan rangkaian minimum sistem. Rangkaian minimum sistem tersebut terdiri rangkaian *reset* dan rangkaian osilator. Dalam perancangannya ini memerlukan beberapa komponen pendukung seperti kristal, resistor dan kapasitor.

Rangkaian minimum sistem dibuat untuk mendukung kerja dari *microcontroller* ATmega dimana *microcontroller* tidak bisa berdiri sendiri alias harus ada rangkaian dan komponen pendukung seperti halnya rangkaian catu daya, kristal dan lain sebagainya yang biasanya disebut minimum sistem .

Minimum sistem ini dirancang untuk Mikrokontroler ATmega128, dalam tugas akhir ini menggunakan minimum sistem keluaran *Innovative*

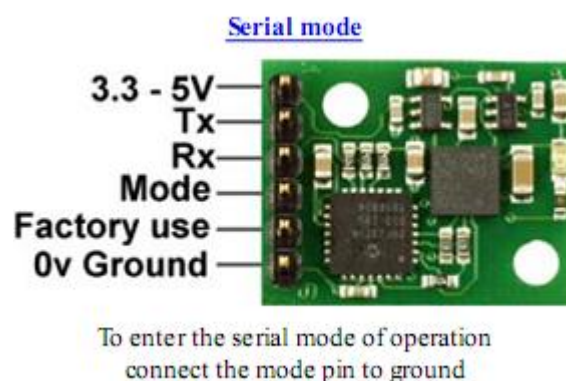
Electronics dengan tipe DT-AVR ATMEGA1280 CPU MODULE V1.0 yang di dalamnya terdapat komponen pendukung. Berikut adalah konfigurasi pin I/O yang digunakan:

Tabel 3.1 Konfigurasi pin I/O pada minimum sistem

Pin I/O	Fungsi
Vcc	Power 5 volt
Port C	LCD
Port D pin 2 (USART 1)	Komunikasi serial dengan kompas
Port D pin 3 (USART 1)	Komunikasi serial dengan kompas
Port D pin 0 (interrupt 0)	Rotary encoder
Port D pin 1 (interrupt 1)	Rotary encoder
Reset	Merereset program
Port E pin 2-5	Driver motor
Port B pin 5	OCR1A
Port B pin 6	OCR1B

3.2.3 Modul Kompas Digital CMPS10

Modul kompas digital ini berfungsi sebagai penunjuk sudut arah mata angin pada *mobile robot* dalam mencari koordinat tujuan yang akan dicapai. Mode yang digunakan pada untuk komunikasi antara kompas dengan mikrokontroller adalah komunikasi serial.



Gambar 3.4 Mode serial kompas digital CMPS10

3.2.4 Rotary Encoder

rotary encoder ini berfungsi sebagai penghitung jarak tempuh *mobile robot* dalam perjalanan menuju koordinat tujuan, sehingga *mobile robot* dapat berhenti tepat pada koordinat tujuan. Dalam tugas akhir ini menggunakan encoder quadrature optic dari *chassis* robot Dagru Rover 5 yang memberikan 1000 pulsa dari tiap 3 putaran dari poros output.

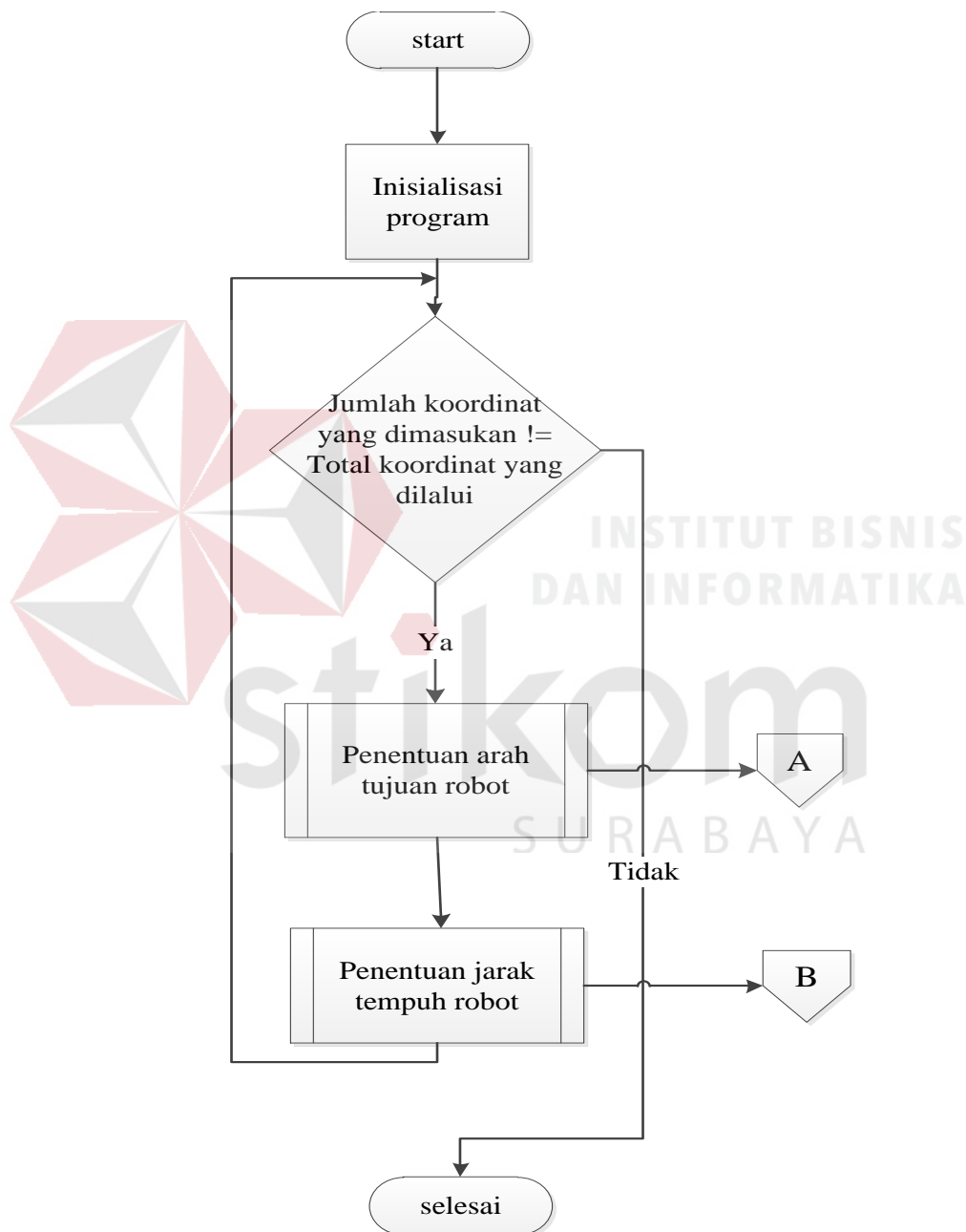


Gambar 3.5 Rotary encoder pada mobile robot

3.3 Perancangan Perangkat Lunak pada Mikrokontrol

Perancangan perangkat lunak bertujuan untuk mengirimkan data dari pembacaan sensor kompas ke minimum sistem melalui komunikasi serial dan juga melakukan pengolahan data dengan melalui proses perhitungan rumus arah dan jarak pada minimum sistem. Minimum sistem memperoleh data dari sensor kompas yaitu berupa data derajat arah mata angin pada *mobile robot* yang dapat digunakan untuk menentukan arah hadap robot terhadap koordinat tujuan robot dan dari *rotary encoder* yaitu berupa *pulse* yang dapat digunakan untuk penentuan jarak tempuh *mobile robot*, serta mengatur kecepatan putar roda melalui PWM

yang dikirimkan ke *driver motor*. Perancangan perangkat lunak terbagi dalam beberapa program antara lain : program motor DC, program membaca sensor, program penentuan arah tujuan robot, program penentuan jarak tempuh tujuan robot. Diagram alir perangkat lunak secara umum dapat dilihat pada Gambar 3.6.



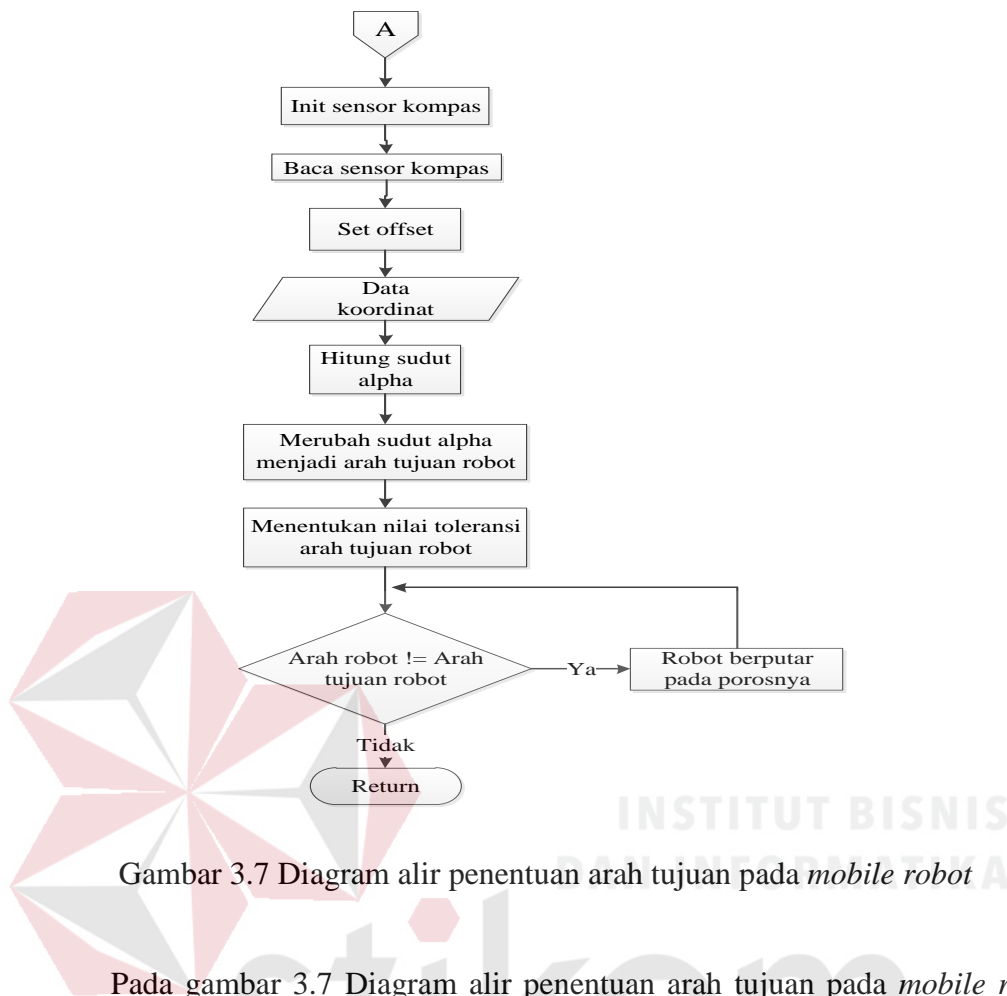
Gambar 3.6 Diagram alir program secara umum pada mikrokontrol

Pada gambar 3.6 diagram alir program secara umum pada mikrokontrol dimulai dengan inisialisasi program keseluruhan yang telah diunduh pada

mikrokontrol kemudian setelah itu melakukan pengecekan data koordinat tujuan yang akan dilalui oleh robot. Setelah dilakukan pengecekan koordinat tujuan maka robot akan memasuki subproses penentuan arah tujuan robot, pada subproses ini menggunakan implementasi dari rumus trigonometri yang berfungsi untuk menentukan arah hadap robot menuju koordinat tujuan yang akan dilalui robot. Setelah melakukan subproses penentuan penentuan arah, robot selanjutnya memasuki subproses penentuan jarak tempuh robot. Pada subproses ini menggunakan implementasi dari rumus phytagoras yang berfungsi untuk menentukan jarak tempuh yang akan dilalui robot dari titik koordinat awal menuju titik koordinat tujuan. Setelah kedua subproses tersebut dilakukan, robot akan berjalan menuju koordinat tujuan dan apabila telah sampai pada koordinat tujuan maka robot akan kembali pada proses pengecekan data koordinat tujuan untuk melakukan penghitungan ulang terhadap koordinat tujuan lain yang telah dimasukan. Proses pengecekan koordinat tersebut dilakukan berulang sampai jumlah koordinat yang telah dilalui sama dengan total koordinat yang dimasukan oleh pengguna.

3.3.1 Program Subproses Penentuan Arah Tujuan Pada *Mobile Robot*

Diagram alir untuk mengetahui penentuan arah tujuan pada *mobile robot* berdasarkan pembacaan sensor kompas digital terdapat pada Gambar 3.7.



Gambar 3.7 Diagram alir penentuan arah tujuan pada *mobile robot*

Pada gambar 3.7 Diagram alir penentuan arah tujuan pada *mobile robot*, penentuan arah tujuan pada *mobile robot* dimulai dengan inialisasi sensor kompas digital yang diteruskan dengan pembacaan sensor kompas agar dapat melakukan penentuan offset. Dalam program sebagai berikut:

```

unsigned int kompas_sekarang()
{
    unsigned int val;
    putchar1 (0x13);
    val = (unsigned int)getchar1() << 8;
    return ((val | (unsigned int)getchar1())/10);
}

offset = kompas_sekarang();

```

Nilai offset berguna sebagai derajat acuan robot atau bisa disebut sebagai 0 derajat robot pada saat robot melakukan navigasi. Setelah mendapatkan *input* data

koordinat, dilakukan perhitungan sudut alpha dengan menggunakan rumus Trigonometri sebagai berikut:

$$\alpha = \tan^{-1} \frac{y_{target}}{x_{target}} \quad (3.1)$$

Atau pada program ditulis sebagai berikut:

```
radian = atan(ytarget / xtarget);
alfa = (radian*180)/PI; // merubah radian menjadi degrees
```

Perhitungan sudut alpha berfungsi sebagai penentu arah hadap robot dari koordinat awal menuju koordinat tujuan. Kemudian diteruskan dengan merubah sudut alpha menjadi arah tujuan robot, dalam program dapat ditulis sebagai berikut:

```
real = (sudut_alfa + offset)%360;
```

Pada proses ini hasil perhitungan sudut alpha akan dijumlahkan dengan nilai offset yang telah didapatkan pada awal proses, sehingga robot dapat mengetahui sudut yang diharapkan agar dapat bernavigasi menuju koordinat tujuan. Setelah mendapatkan arah tujuan robot, dilakukan penentuan nilai toleransi arah tujuan robot. Dalam program sebagai berikut:

```
max = (real + toleransi)%360 ;
if (real >= toleransi)
    min = real-toleransi;
else
    min = real+360-toleransi;
```

Penentuan nilai toleransi ini dilakukan untuk memberikan toleransi terhadap arah tujuan yang akan ditempuh robot dalam bernavigasi. Setelah mendapatkan nilai-nilai diatas dilakukan penyesuaian arah hadap *mobile robot* terhadap arah tujuan robot. Dalam program sebagai berikut:

```
do{
    arah = kompas_sekarang();
    if (arah >= min && arah <= max) break;
    motor_kiri(200,1);
    motor_kanan(200,2);
```

```

ftoa(arah,0,a_rah);
lcd_gotoxy(8,0);
lcd_puts(a_rah);
lcd_putsf("    ");
}while (1);

motor_kiri (0,0);
motor_kanan (0,0);
lcd_clear();

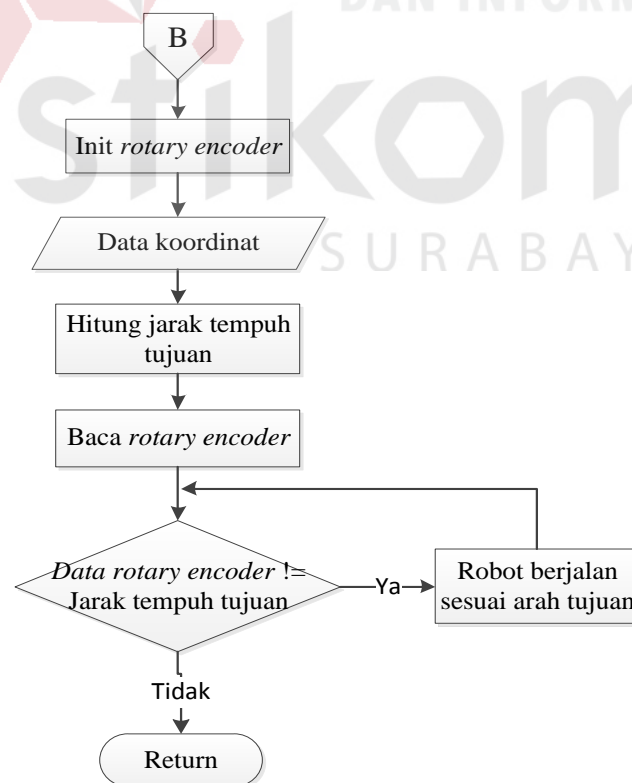
```

Pada proses ini robot akan berputar menuju sudut tujuan robot yang diharapkan untuk melakukan navigasi menuju koordinat tujuan dan akan berhenti apabila telah mencapai sudut yang diharapkan.

3.3.2 Program Subproses Penentuan Jarak Tempuh Tujuan Pada *Mobile Robot*

Robot

Diagram alir untuk mengetahui penentuan jarak tempuh tujuan berdasarkan pembacaan *rotary encoder* terdapat pada Gambar 3.8.



Gambar 3.8 Diagram alir penentuan jarak tempuh pada *mobile robot*

Pada gambar 3.8 Diagram alir penentuan jarak tempuh tujuan pada *mobile robot*, penentuan jarak tempuh tujuan pada *mobile robot* dimulai dengan inisialisasi *rotary encoder* yang diteruskan dengan pengambilan data koordinat. Kemudian dilakukan perhitungan jarak tempuh dengan menggunakan rumus Phytagoras sebagai berikut:

$$r\ target = \sqrt{x\ target^2 + y\ target^2} \quad (3.2)$$

Atau pada program ditulis sebagai berikut:

```
akar = (sqrt(pow(xtarget,2) + pow(ytarget,2))) ;
```

Perhitungan jarak tempuh ini berfungsi untuk mengetahui jarak yang harus dilalui robot untuk dapat bernavigasi menuju koordinat tujuan, hasil perhitungan jarak tempuh tersebut kemudian dikonversikan menjadi pulse sehingga memudahkan dalam proses pembandingan jarak tempuh yang diharapkan dan jarak tempuh yang sedang dilalui. Setelah mendapatkan nilai jarak tempuh, dilakukan pembacaan *rotary encoder* dan pembandingan nilai antara *output rotary encoder* dan nilai jarak tempuh yang didapat untuk menentukan pergerakan *mobile robot*.

Dalam program sebagai berikut:

```
while (rotaryka < rtarget1 && rotaryki < rtarget1)
{
    itoa (rotaryka,rotary);
    lcd_gotoxy (0,0);
    lcd_puts ("r =");
    lcd_gotoxy (5,0);
    lcd_puts (rotary);
    motor_kanan (255,1);
    motor_kiri (255,1);
    flag_c = 0;
}
flag_c = 1;
motor_kanan (0,0);
motor_kiri (0,0);
rotaryka = rotaryki = 0;
```

Pada Proses ini robot akan terus berjalan sampai perbandingan pembacaan data pada *rotary encoder* telah sama dengan data perhitungan jarak tempuh yang diharapkan untuk mencapai koordinat tujuan.

3.4 Metode Pengujian dan Evaluasi Sistem

Dalam pengujian sistem ini pengujian akan dilakukan pada perangkat keras dan perangkat lunak yang telah dibuat. Pengujian yang dilakukan dimulai dari pengujian minimum sistem, pengujian sensor kompas digital, pengujian *rotary encoder*, pengujian motor DC, pengujian program penentuan arah tujuan pada *mobile robot*, pengujian program penentuan jarak tempuh tujuan pada *mobile robot*, dan yang terakhir adalah pengujian sistem secara keseluruhan yaitu *mobile robot* dapat menuju koordinat tujuan yang telah di tentukan.

3.4.1 Pengujian dan Evaluasi Minimum Sistem

Pengujian minimum sistem ini bertujuan untuk mengetahui apakah minimum sistem dapat melakukan proses *signature* dan *download* program ke mikrokontroller dengan baik. Pengujian ini dilakukan dengan cara mengaktifkan *power supply* dan hubungkan dengan minimum sistem. Sambungkan minimum sistem dengan komputer menggunakan kabel *downloader* dan jalankan *compiler CodeVisionAVR*. Setelah itu lakukan proses *chip signature* untuk mengetahui apakah mikrokontroler sudah terkoneksi dengan baik. Jika proses *chip signature* berhasil maka akan keluar informasi tentang mikrokontroler yang terhubung. Kemudian lakukan proses *download* pada mikrokontroler, jika proses *download*

berhasil maka akan *CodeVisionAVR* akan mengeluarkan tampilan proses *download*.

3.4.2 Pengujian dan Evaluasi Sensor Kompas Digital

Pengujian sensor kompas digital ini bertujuan untuk mengetahui informasi derajat arah mata angin pada robot. Dalam hal ini pengujian dilakukan dengan memberikan tegangan pada sensor kompas digital dan melakukan koneksi antara sensor kompas digital dengan minimum sistem. Setelah itu proses dilanjutkan dengan menggunakan program pembacaan sensor kompas digital pada *compiler CodeVisionAVR* yang akan *download* pada minimum sistem dan akan ditampilkan pada LCD. Jika proses pengujian tersebut berhasil maka LCD akan menampilkan derajat arah mata angin yang sesuai dengan kompas sebenarnya.

3.4.3 Pengujian dan Evaluasi Program Penentuan Arah Tujuan Pada *Mobile Robot*

Pengujian ini dilakukan untuk menguji apakah program penentuan arah tujuan pada *mobile robot* yang dibuat telah berjalan sesuai dengan keinginan dan kebutuhan pada tugas akhir ini. Dalam hal ini pengujian dilakukan dengan melakukan koneksi antara semua perangkat keras yang dibutuhkan untuk melakukan proses ini. Setelah itu proses dilanjutkan dengan melakukan proses *download* program penentuan arah tujuan yang telah dibuat sesuai dengan diagram alir pada gambar 3.10. Pengujian ini dinyatakan benar apabila hasil dari penentuan ini menunjukkan bahwa posisi arah hadap robot sesuai dengan posisi

arah hadap tujuan robot pada koordinat yang telah ditentukan pada lapangan pengujian yang telah disiapkan.

3.4.4 Pengujian dan Evaluasi Program Penentuan Jarak Tempuh Tujuan Pada *Mobile Robot*

Pengujian ini dilakukan untuk menguji apakah program penentuan jarak tempuh tujuan pada *mobile robot* yang dibuat telah berjalan sesuai dengan keinginan dan kebutuhan pada tugas akhir ini. Dalam hal ini pengujian dilakukan dengan melakukan koneksi antara semua perangkat keras yang dibutuhkan untuk melakukan proses ini. Setelah itu proses dilanjutkan dengan melakukan proses *download* program penentuan jarak tempuh tujuan yang telah dibuat sesuai dengan diagram alir pada gambar 3.11. Pengujian ini dinyatakan benar apabila hasil dari penentuan ini menunjukkan bahwa jarak yang ditempuh robot terhadap koordinat tujuan sesuai dengan jarak yang harus ditempuh robot terhadap koordinat tujuan dan robot dapat berhenti pada koordinat tujuan pada lapangan pengujian yang telah disiapkan.

3.4.5 Pengujian dan Evaluasi Keseluruhan Sistem Navigasi Pada *Mobile Robot*.

Pengujian ini dilakukan untuk menguji apakah keseluruhan sistem navigasi ini sudah sesuai dengan keinginan dan kebutuhan pada tugas akhir ini. Pengujian ini meliputi pengujian keseluruhan perangkat keras yang digunakan pada *mobile robot* dan pengujian keseluruhan pada program yang mendukung sistem navigasi ini. Dalam hal ini pengujian dilakukan dengan melakukan koneksi

antara semua perangkat keras dan sinkronisasi pada semua program yang telah dibuat untuk melakukan proses ini. Setelah itu proses dilanjutkan dengan melakukan proses *download* program yang telah disinkronisasi kedalam robot yang sudah siap untuk melakukan proses navigasi atau dengan kata lain robot sudah terintegrasi dengan semua sensor yang digunakan. Pengujian ini dinyatakan benar apabila hasil dari sistem navigasi ini sesuai dengan apa yang diinginkan dan dibutuhkan pada tugas akhir ini atau *mobile robot* berhasil melakukan navigasi dari koordinat awal robot menuju koordinat tujuan robot dan mampu melakukan perpindahan titik secara berulang atau menuju koordinat tujuan yang lebih dari satu tujuan pada lapangan pengujian yang telah disiapkan.

