

BAB III

METODE PENELITIAN

3.1. Model Pengembangan

Tujuan dari tugas akhir ini yaitu akan membuat sebuah *mobile Robot* yang mampu membantu manusia dalam mendeteksi kebocoran gas. Robot ini berperan sebagai perantara antara manusia dengan bau gas. Robot akan dikendalikan secara manual menggunakan *joystick Robot Bluetooth* yang sudah dirancang untuk mengirim perintah *user*, Robot akan berjalan menuju ke tempat yang diinginkan *user* untuk memeriksa apakah ada kebocoran gas.

Robot ini mempunyai lengan yang berfungsi untuk perantara antara Robot dengan benda yang didalamnya terdapat bau gas yang berbahaya. Dibagian ujung lengan Robot terdapat sensor gas yang berfungsi sebagai pencium bau gas yang mengandung unsur gas metana dan butane. Sedangkan sensor jarak akan mendeteksi suatu benda di depannya, agar Robot tidak menabrak benda tersebut karena benda tersebut sangat berbahaya dan supaya tidak terjadi hal-hal yang tidak diinginkan.

3.2. Prosedur Penelitian

Prosedur penelitian yang dipakai dalam pengerjaan tugas akhir ini adalah:

1. Studi literatur

Pada penelitian ini terdapat dua perancangan yang akan dilakukan yaitu, perancangan perangkat keras dan perangkat lunak. Adapun metode

penelitian yang dilakukan antara lain: Pencarian data-data literatur untuk perangkat keras dari masing-masing komponen, informasi dari internet dan konsep teoritis dari buku-buku penunjang tugas akhir ini, serta materi-materi perkuliahan yang telah didapatkan dan perancangan perangkat lunak yaitu menggunakan *CodeVisionAVR* melalui pencarian dari internet, dan konsep-konsep teoritis dari buku-buku penunjang tersebut. Dari kedua bagian tersebut akan dipadukan agar dapat bekerja sama untuk menjalankan sistem dengan baik.

2. Tahap perancangan dan pengembangan sistem

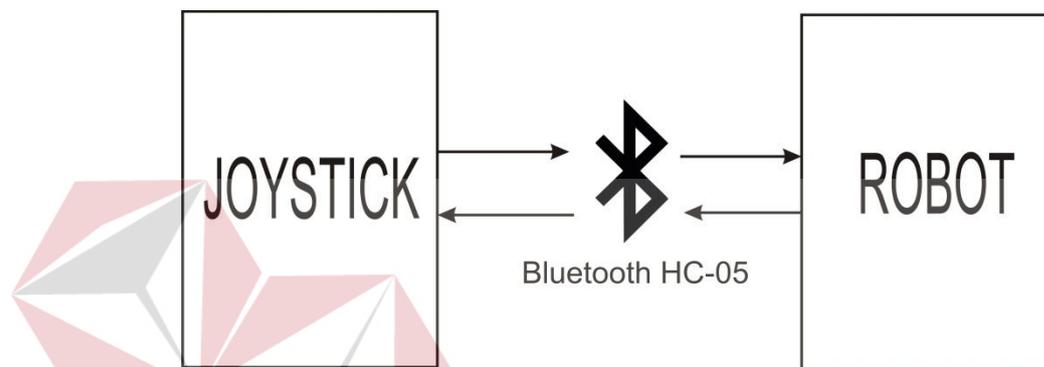
Dalam membuat pengembangan sistem, terdapat beberapa langkah rancangan sistem yang diambil antara lain:

- a. Membuat *flowchart* pada proses sistem secara keseluruhan
- b. Melakukan perancangan perangkat keras yang meliputi:
 1. Merancang rangkaian elektronik yang digunakan pada penelitian ini
 2. Melakukan percobaan tentang cara penggunaan sensor dan *device* yang digunakan pada penelitian ini
 3. Merancang mekanik untuk *mobile Robot*
- c. Melakukan perancangan perangkat lunak yang meliputi:
 1. Mengatur protokol pengiriman antara PC dan *microcontroller*.
 2. Membuat program pada *microcontroller* untuk mendeteksi benda yang didalamnya terdapat bau gas.
 3. Membuat program pengiriman dari *joystick Bluetooth* ke *microcontroller* pada Robot.
 4. Membuat program untuk menggerakkan lengan Robot.

5. Membuat program tampilan ke LCD.

3.3. Diagram Blok Sistem

Dari penelitian ini terdapat suatu proses dimana *joystick* dapat terhubung dengan Robot melalui komunikasi *Bluetooth* HC-05. Pada Gambar 3.1 adalah diagram blok keseluruhan sistem ini.

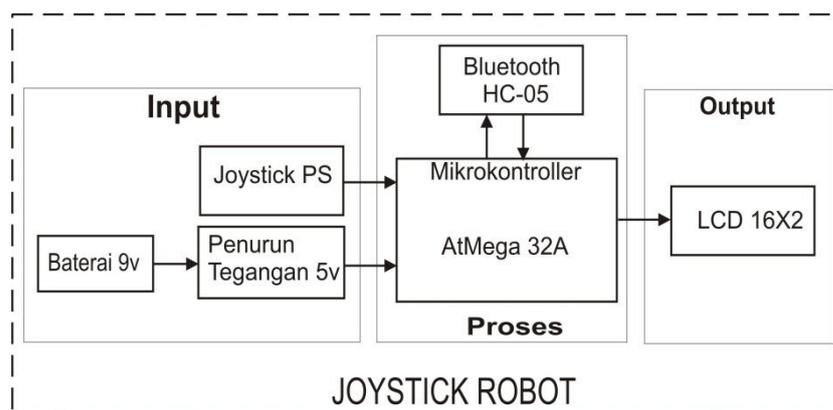


Gambar 3.1 Diagram blok keseluruhan dari sistem

Pada Gambar 3.1 dapat dijelaskan bahwa *Bluetooth* HC-05 berguna sebagai perantara antara *joystick* dengan Robot. *Joystick* dapat mengirim data ke Robot dan sebaliknya Robot juga dapat mengirim data ke *joystick* melalui komunikasi *Bluetooth* HC-05. Didalam diagram blok keseluruhan sistem ini dapat dibagi menjadi dua bagian diagram blok yaitu diagram blok pada *joystick* dan diagram blok pada Robot. Berikut penjelasan tentang diagram blok *joystick* dan diagram blok Robot.

3.3.1 Diagram Blok *Joystick*

Didalam diagram blok *joystick* terdapat beberapa komponen pendukung yaitu *microcontroller* ATmega32A, *Bluetooth* HC-05, LCD, Baterai 9 Volt, dan Penurun Tegangan 5 Volt. Berikut adalah gambar diagram blok *joystick* terdapat pada Gambar 3.2

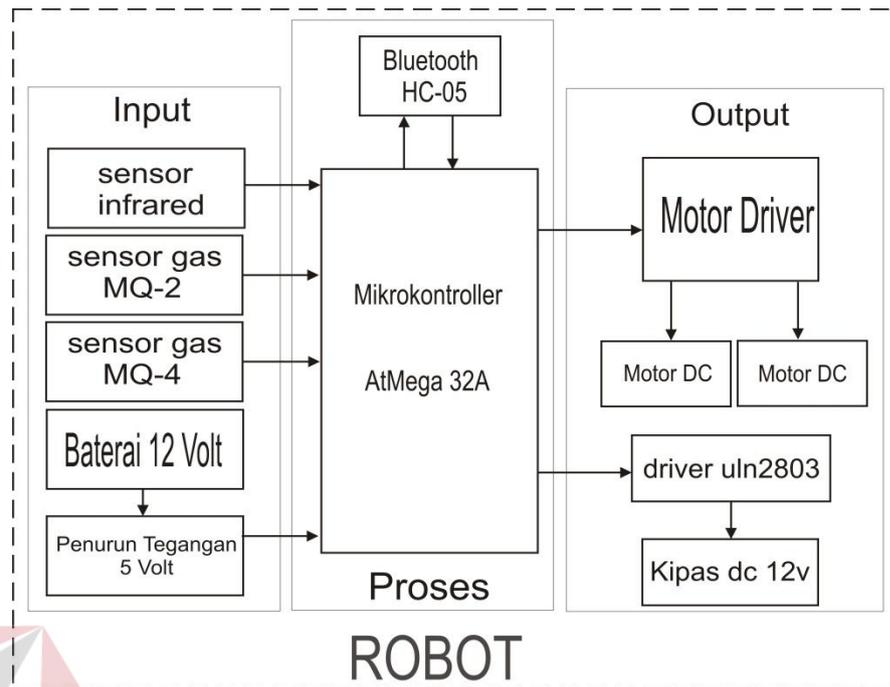


Gambar 3.2 Diagram Blok *Joystick*

Pada Gambar 3.2 dijelaskan bahwa didalam diagram blok *joystick* terdapat suatu sistem yaitu *input*, *proses*, dan *output*. Didalam *input* terdapat 3 komponen yaitu *joystick*, baterai 9 volt, dan penurun tegangan yang berguna sebagai masukan ke suatu proses. Sedangkan pada proses terdapat 2 komponen yaitu *microcontroller* ATmega32A yang berguna untuk memproses data masukan, dan *Bluetooth* HC-05 digunakan sebagai alat komunikasi antara *microcontroller joystick* dengan *microcontroller* Robot. Pada *output* terdapat LCD yang digunakan untuk menampilkan data yang sudah diproses oleh *microcontroller* ATmega32A.

3.3.2 Diagram Blok Robot

Didalam diagram blok pada Robot terdapat beberapa komponen pendukung yaitu *microcontroller* ATmega32A, Sensor *Infrared*, Sensor Gas MQ2, Sensor Gas MQ4, Baterai 12 volt, Penurun tegangan, *Motor Driver*, Driver ULN2803, Motor DC, dan Kipas DC 12 volt. Berikut adalah gambar Diagram Blok Robot terdapat pada Gambar 3.3



Gambar 3.3 Diagram Blok Robot

Pada Gambar 3.3 dijelaskan bahwa didalam diagram blok Robot terdapat suatu sistem yaitu *input*, *proses*, dan *output*. Didalam *input* terdapat beberapa komponen yaitu:

1. Sensor *Infrared* yang berfungsi sebagai alat untuk mendeteksi adanya suatu wadah yang didalamnya terdapat gas.
2. Sensor Gas MQ2 dan Sensor Gas MQ4 yang berfungsi sebagai alat untuk mendeteksi bau gas metana dan butane.
3. Baterai 12 volt sebagai power supply.
4. Penurun tegangan sebagai penurun tegangan baterai 9 volt menjadi 5 volt.

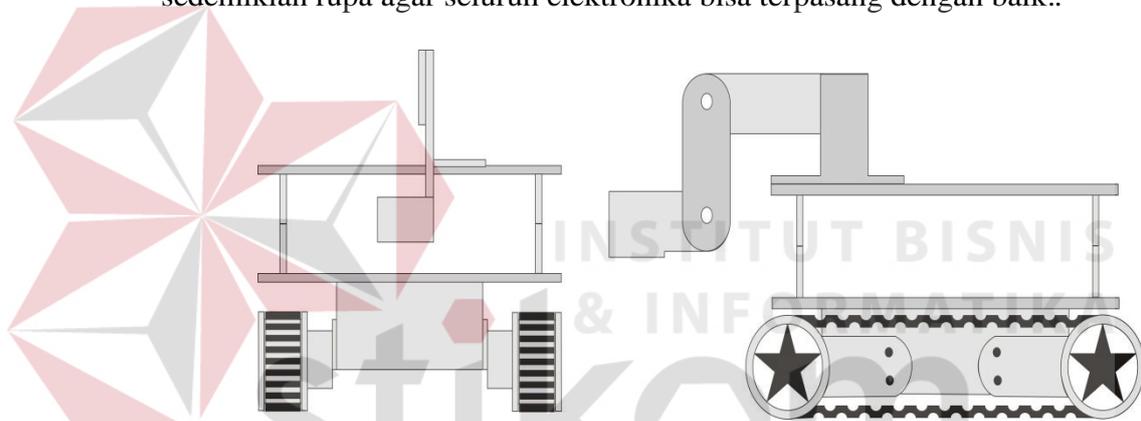
Didalam proses terdapat *Microcontroller* ATMega32A yang berfungsi sebagai otak yang memproses data masukan dari tiap sensor dan data masukan dari *Bluetooth* HC-05. Sedangkan *Bluetooth* HC-05 berfungsi sebagai alat komunikasi antara *microcontroller* pada Robot dengan *microcontroller* pada *joystick*. Pada bagian *output* terdapat motor driver yang berfungsi sebagai

kontrol pergerakan pada Motor DC dan driver ULN2803 sebagai penggerak pada kipas dc 12 volt.

3.4 Perancangan Mekanik Robot

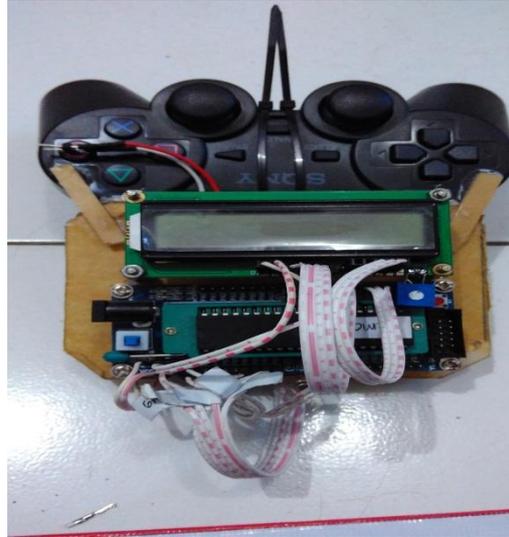
Didalam perancangan mekanik Robot ini terdapat dua pengerjaan mekanik, yaitu mekanik Robot dan mekanik joystick.

- a. Mekanik Robot yang di gunakan adalah Dagu Rover 5 pada bagian base Robot dan didesain menyerupai bentuk roda tank. Robot ini didesain sedemikian rupa agar seluruh elektronika bisa terpasang dengan baik..



Gambar 3.4. Mekanik Robot

- b. Mekanik *joystick* yang digukan adalah *joystick* PS dan didesain sedemikian rupa agar seluruh elektronika terpasang dengan baik pada *joystick* tersebut, mulai dari *microcontroller*, baterai, LCD, penurun tegangan, dan *Bluetooth* HC-05.



Gambar 3.5. Tampilan *Joystick*

3.4.1. Ukuran dimensi *Mobile Robot*

Setelah semua komponen tambahan dari penelitian ini dipasangkan ukuran dimensi dari Robot:

1. Ukuran Robot : 22cm (lebar) x 22cm (panjang) x 15,5cm (tinggi)
2. Ukuran lengan Robot : 20cm (panjang) x 3cm (lebar) x 25cm(tinggi)

3.4.2. Struktur Material *Mobile Robot*

Bahan material yang digunakan dalam penelitian ini menggunakan beberapa bahan diantaranya sebagai berikut :

- a. Bagian rangka
 1. *Dagu Rover 5*
 2. Mur dan baut
 3. Akrilik
- b. Bagian dari penggerak Robot
 1. *Motor dc 7,2 Volt*
 2. Roda tank

c. Bagian dari *Joystick*

1. *Joystick* PS
2. Akrilik
3. Mur dan baut

3.4.3. Perancangan *Minimum sistem ATmega32A*

Pada tugas akhir ini dibuat beberapa buah pengendali menggunakan *microcontroller* keluaran pada perangkat lunak AVR, yaitu ATmega32A. Untuk menjalankan *microcontroller* ini diperlukan rangkaian *minimum sistem*. Rangkaian *minimum sistem* tersebut terdiri rangkaian *reset* dan rangkaian osilator. Dalam perancangan *minimum sistem* memerlukan beberapa komponen pendukung seperti kristal, resistor dan kapasitor.

Rangkaian *minimum sistem* dibuat untuk mendukung kerja dari *microcontroller* ATmega dimana *microcontroller* sebagai otak pada Robot dan tidak bisa berdiri sendiri atau harus ada rangkaian dan komponen pendukung seperti halnya rangkaian catu daya dan lain sebagainya yang biasanya disebut *minimum sistem*. Pada rangkaian *minimum sistem* ATmega32A ini terdapat konfigurasi pin *input* dan *output* sebagai berikut:

Tabel 3.1 Konfigurasi pin I atau O pada *minimum sistem*

Pin I atau O	Fungsi
Vcc	Power 5 volt
Port D-0	RX
Port D-1	TX
Port D-2	Direction 1.1
Port D-3	Direction 1.2
Port D-4	PWM 1
Port D-5	PWM 2
Port D-6	Direction 2.1

<i>Port D-7</i>	Direction 2.2
<i>Port A-1</i>	Sensor <i>Infrared</i>
<i>Port A-6</i>	Sensor MQ2
<i>Port A-7</i>	Sensor MQ4
<i>Port C-1</i>	<i>Input Kipas DC</i>
<i>Port B-9</i>	Mereset Program

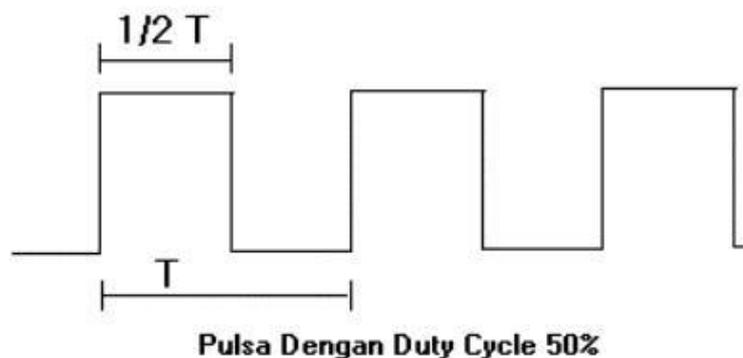
3.4.4 Driver Motor DC 24 V (EMS 2A Dual H-Bridge)

Embedded Module Series (EMS) 2A Dual H-Bridge didesain untuk menghasilkan *drive* 2 arah dengan arus kontinyu sampai dengan 2 A pada tegangan 5,5 Volt sampai 46 Volt Modul ini dilengkapi dengan rangkaian sensor arus beban yang dapat digunakan sebagai umpan balik ke pengendali. Secara garis besar, fungsi modul pengendali motor ini adalah untuk mengendalikan arah dan kecepatan putaran *motor dc* sesuai instruksi kendali dari *minimum sistem ATmega32A*.

Kecepatan putar motor dapat dikendalikan dengan mengatur besar – kecilnya tegangan yang di masukkan, atau dapat juga dengan menggunakan teknik PWM (*Pulse Width Modulation*).

Dengan menggunakan PWM kita dapat mengatur kecepatan yang diinginkan dengan mudah. Teknik PWM untuk pengaturan kecepatan motor adalah dengan cara merubah besarnya *duty cycle* pulsa. Pulsa yang berubah ubah *duty cycle*-nya inilah yang menentukan kecepatan motor. Penulis menggunakan *mode fast* PWM. Dalam *mode fast* PWM sifat cacahan *register* pencacah TCNT1 mencacah dari *bottom* (0x0000) terus mencacah naik (*counting-up*) hingga mencapai *top* (nilai maksimal yang ditentukan sesuai resolusi, misal resolusinya 10 bit maka nilai *top* = 0x01ff), kemudian

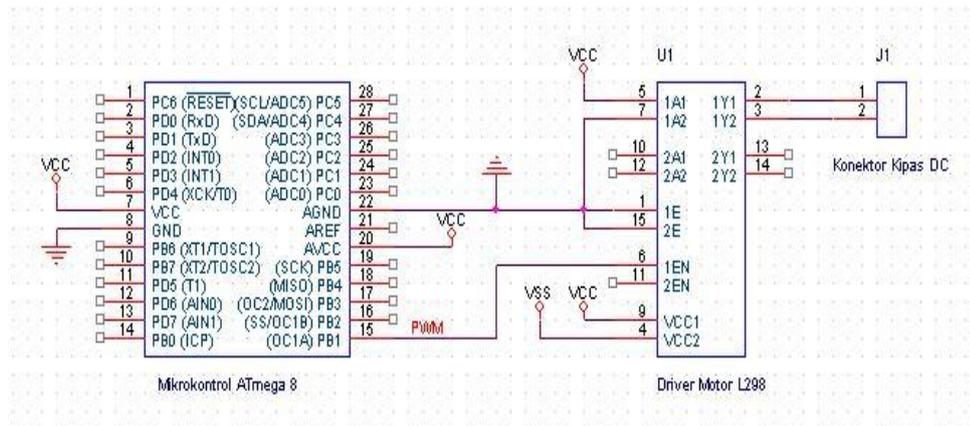
mulai dari *bottom* lagi dan begitu seterusnya atau yang dinamakan *single slope* (satu arah cacahan) (Ardi Winoto, 2008). Berikut adalah Gambar 3.3 untuk pulsa dengan *duty cycle* 50% :



Gambar 3.6 Pulsa dengan *duty cycle* 50%

Besarnya amplitudo dan frekuensi pulsa adalah tetap, sedangkan besarnya *duty cycle* berubah-ubah sesuai dengan kecepatan yang diinginkan, semakin besar *duty cycle* maka semakin cepat pula kecepatan motor. Sebagai contoh bentuk pulsa yang dikirimkan adalah seperti pada Gambar 3.3, pulsa kotak dengan *duty cycle* pulsa 50%. Semakin besar *duty cycle* pulsa kotak, maka semakin lama pula posisi logika *high*. Jika motor diatur agar berjalan ketika diberi logika *high*, maka jika memberi pulsa seperti pada Gambar 3.4, maka motor akan berada pada kondisi “hidup-mati-hidup-mati” sesuai dengan bentuk pulsa tersebut. Semakin lama motor berada pada kondisi “hidup” maka semakin cepat pula kecepatan motor tersebut. Motor akan berputar dengan kecepatan maksimum jika mendapat pulsa dengan *duty cycle* 100%.

Microcontroller akan mengirimkan gelombang pulsa ke *driver motor* untuk mengatur kecepatan motor. Rangkaian *driver motor dc* 24V dapat dilihat pada Gambar 3.4.



Gambar 3.7 Rangkaian *driver motor dc 24V L298N*

3.4.5 Perancangan *Joystick Bluetooth*

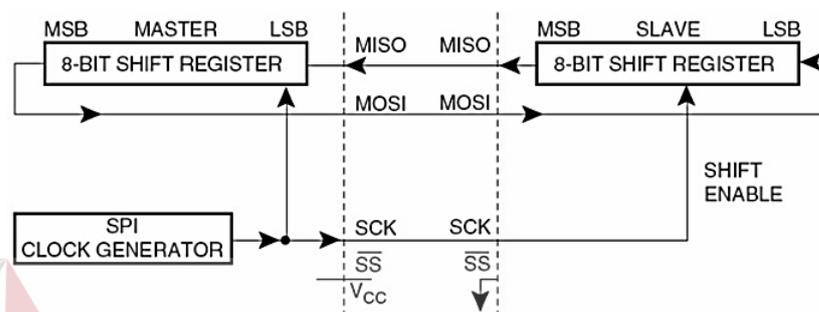
A. *Serial Peripheral Interface (SPI)*

SPI merupakan hubungan data serial yang standar untuk mikroprosesor, *microcontroller* dan peripheral yang dikeluarkan oleh perusahaan Motorola. Hubungan dalam SPI merupakan hubungan data serial yang *full-duplex, synchronous*. SPI dipakai untuk menyediakan komunikasi antara kontroler dengan piranti peripheral. Komunikasi antara mikroprosesor dan peripheral atau *inter-processor* dapat dilakukan dengan SPI. Piranti peripheral SPI tersedia dari *shift register* sederhana untuk ADC, DAC dan *chip* memori. Kontroler yang terintegrasi dengan *port* SPI menyediakan hubungan ke piranti peripheral dengan *port* SPI.

Sistem SPI cukup fleksibel sebagai antarmuka secara langsung dengan banyak peripheral yang tersedia. *Port* SPI memiliki sinyal sebagai berikut:

Ketika dikonfigurasi sebagai *slave*, antarmuka SPI akan menjadi status *sleep* dengan jalur MISO ke kondisi tri state (*high impedance*) selama pin SS dibawa ke logika tinggi. Pada kondisi ini, perangkat lunak mungkin memperbarui isi dari SPI Data register (SPDR), tetapi data tidak akan digeser

keluar oleh pulsa *clock* yang diterima pada pin SCK sampai pin SS dibuat rendah. Jika bit *SPI Interrupt Enable* (SPIE) pada register diset, sebuah interupsi diminta. *Slave* mungkin melanjutkan untuk data baru dikirim ke SPDR sebelum membaca data yang diterima. *Byte* terakhir yang diterima akan dijaga dalam *Buffer Register* untuk digunakan selanjutnya.



Gambar 3.8 Interkoneksi SPI *master-slave*

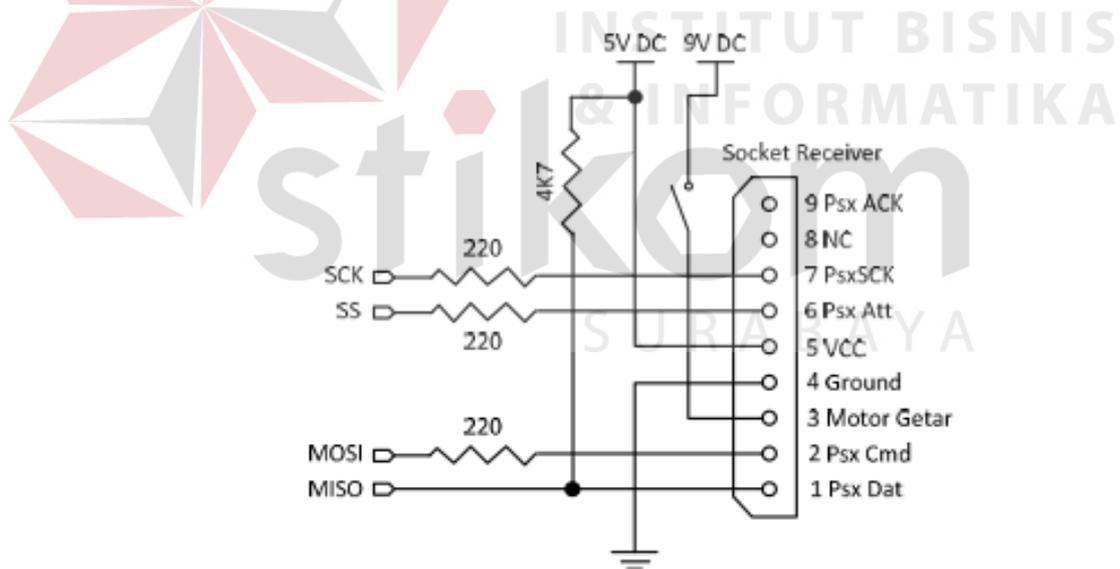
B. Joystick Playstation

Stik PS2 terdiri dari dua modul, yaitu modul *transmitter* dan modul *receiver*. Modul *transmitter* berfungsi sebagai data *input* dan mengirim data *input* tersebut ke modul *receiver*. Sedangkan modul *receiver* berfungsi sebagai penerima data yang dikirim dari modul *transmitter*. Pada setiap Stik PS (*joystick Playstation*) terdapat kontroler yang bertugas untuk berkomunikasi dengan *console playstation*. Komunikasi yang digunakan adalah serial sinkron, yaitu data dikirim satu per satu melalui jalur data. Untuk mengkoordinasikan antara pengirim dan penerima terdapat satu jalur *clock*. Hal inilah yang membedakan serial sinkron dengan serial asinkron (UART atau RS232) yang dapat bekerja tanpa jalur *clock* karena masing-masing pengirim dan penerima mempunyai *clock*.



Gambar. 3.9 Konfigurasi pin Stik PS2

Beberapa pin Stik PS2 tersebut harus dikoneksikan ke *microcontroller* supaya dapat berkomunikasi dengan *microcontroller* tersebut. Adapun pengkoneksiannya dalam bentuk rangkaian skematik diperlihatkan pada Gambar 3.8.



Gambar. 3.10 Skematik koneksi *receiver* Stik PS2.

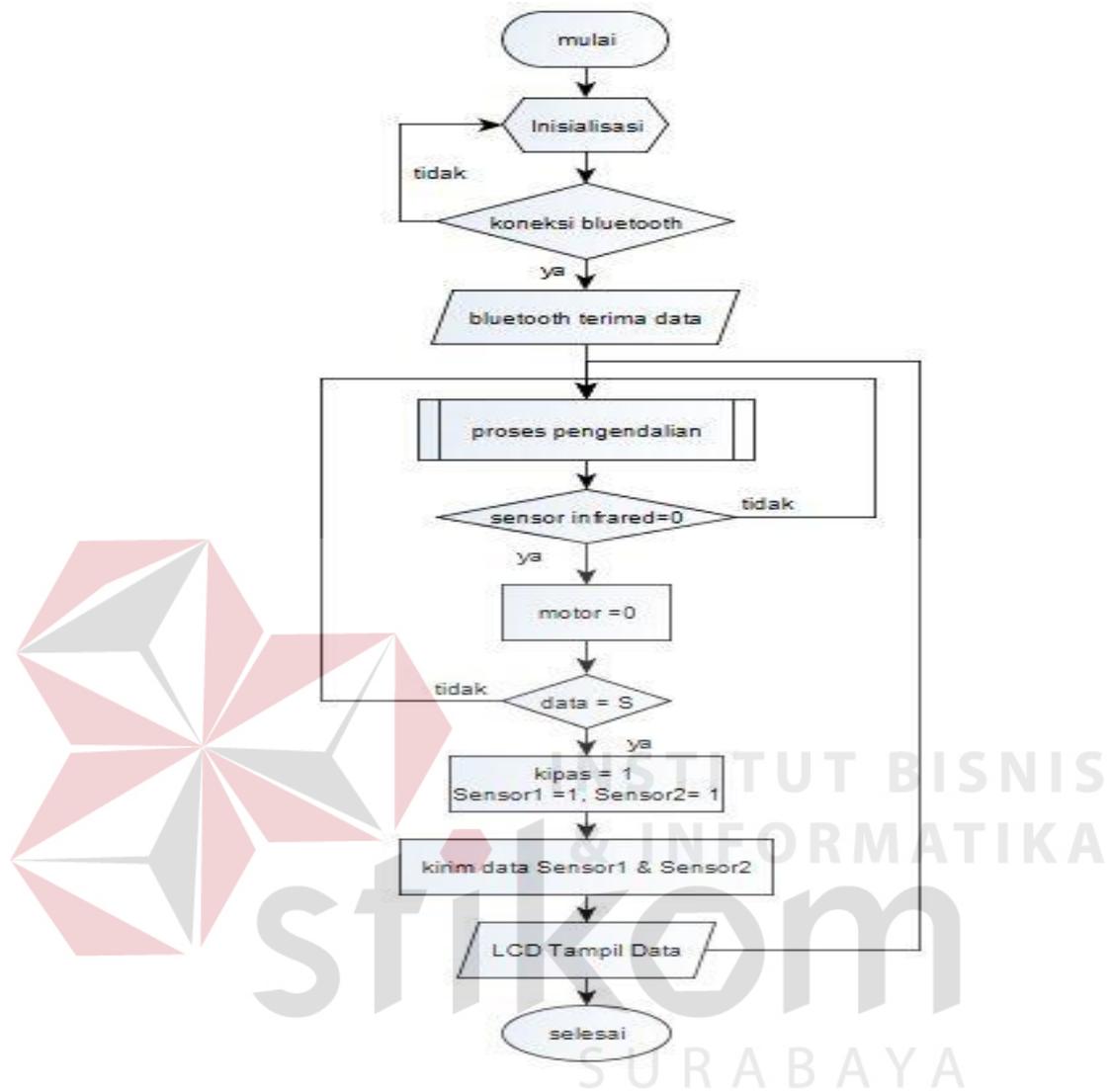
Tabel 3.2 Data *joystick Playstation*

Byte	Psx CMD	Psx Data	Keterangan							
01	0x01	-								
02	0x42	0x41								
03	-	0x5A	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
04	-	Digital 1	←	↓	→	↑	<i>Start</i>	<i>Joy R</i>	<i>Joy L</i>	<i>Select</i>
05	-	Digital 2	□	X	O	Δ	R1	R2	L1	L2
06	-	Analog 1X	<i>Joystick analog kanan sumbu X 128 center</i>							
07	-	Analog 1 Y	<i>Joystick analog kanan sumbu Y 128 center</i>							
08	-	Analog 2X	<i>Joystick analog kiri sumbu X 128 center</i>							
09	-	Analog 2 Y	<i>Joystick analog kiri sumbu Y 128 center</i>							

3.5. Rancangan Perangkat Lunak

Perancangan perangkat lunak bertujuan untuk mengetahui alur kendali Robot sampai pada tahap pembacaan sensor terhadap gas dan proses pengiriman data pada *Joystick*. Perancangan perangkat lunak ini di bahas menggunakan *flowchart* seperti Gambar 3.11 dan Gambar 3.12.

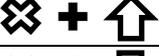
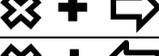
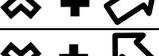
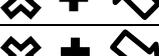
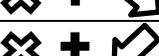
a. Diagram alir keseluruhan sistem pada Robot



Gambar 3.11 Diagram alir keseluruhan sistem pada Robot

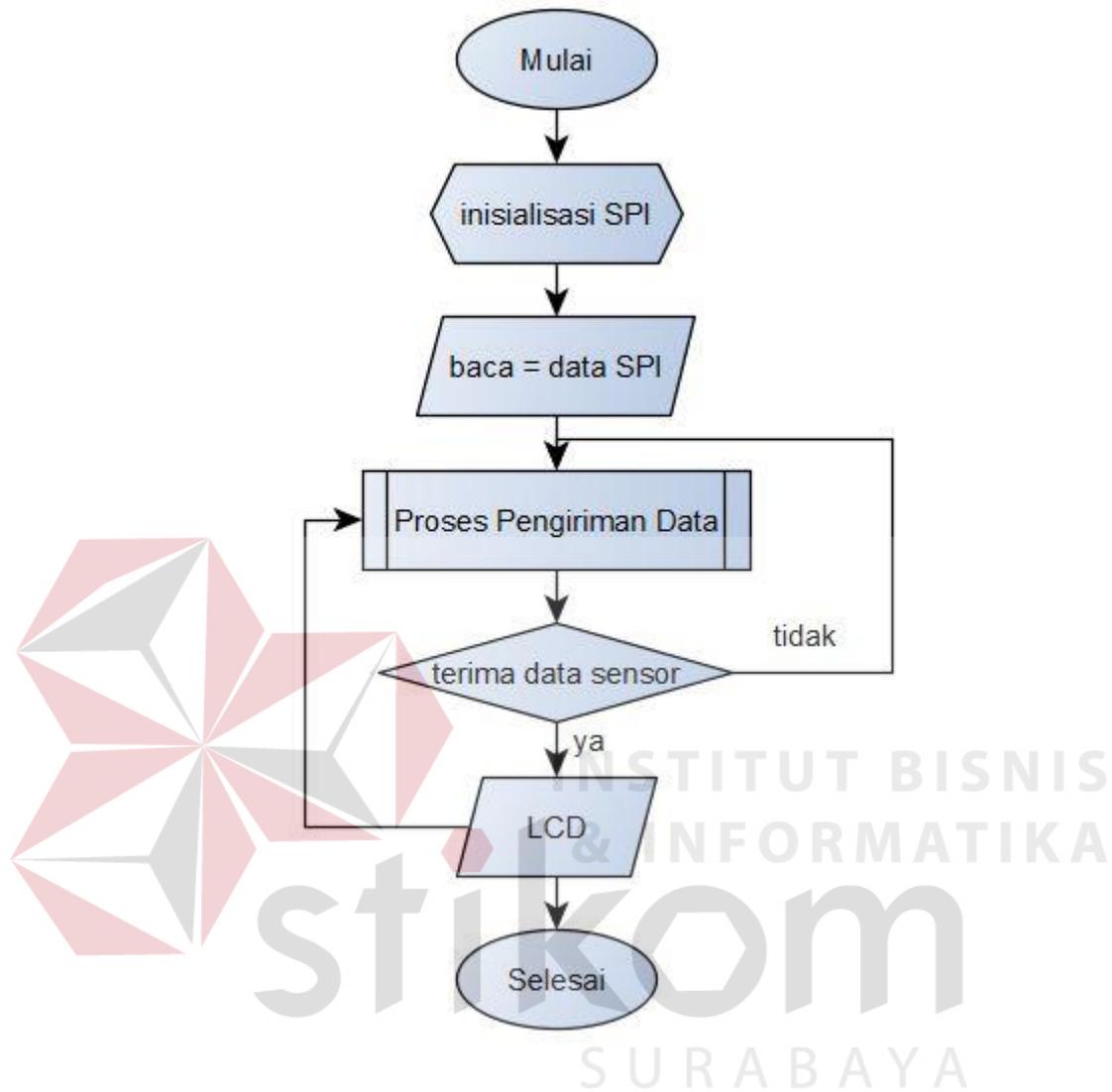
Tabel 3.3 Proses Pengendalian Robot

No.	Tombol Joystick	Data yang diterima	Pergerakan Robot
1	↑	A	Maju
2	↓	B	Mundur
3	↘	C	Belok Kanan
4	↙	D	Belok Kiri
5	↗	E	Maju Serong Kanan

6		F	Maju Serong Kiri
7		G	Mundur Serong Kanan
8		H	Mundur Serong Kiri
9		I	Kecepatan + Maju
10		J	Kecepatan + Mundur
11		K	Kecepatan + Kanan
12		L	Kecepatan + Kiri
13		M	Kecepatan + Maju Serong Kanan
14		N	Kecepatan + Maju Serong Kiri
15		O	Kecepatan + Mundur Serong Kanan
16		P	Kecepatan + Mundur Serong Kiri
17		S	Mengaktifkan Kipas Dan Sensor Gas
18	-	Z	Berhenti (tidak ada <i>input</i>)

Pada Gambar 3.11 dijelaskan bahwa proses berjalannya Robot dimulai dari inisialisasi. Setelah itu menunggu *Bluetooth* pada Robot terkoneksi dengan *Bluetooth* pada *joystick*, jika terkoneksi maka *Bluetooth* pada Robot sudah siap menerima data dari *input joystick* berupa data karakter. Selanjutnya menuju ke proses pengendalian Robot. Didalam proses pengendalian Robot terdapat beberapa perintah untuk mengendalikan Robot, perintah tersebut dapat dilihat pada Tabel 3.3. Setelah proses pengendalian Robot maka Robot akan berjalan menyusuri ruangan untuk mencari benda dan jika sensor *infrared* aktif maka Robot berhasil mendeteksi adanya benda dan Robot akan berhenti. Proses selanjutnya yaitu *user* menginputkan perintah untuk mengaktifkan kipas dan mengaktifkan sensor gas. Data masukan dari sensor gas akan dikirim ke *joystick* dan ditampilkan ke LCD *joystick*.

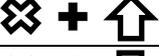
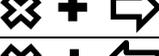
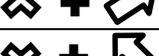
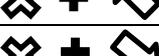
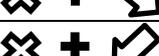
b. Diagram alir keseluruhan sistem pada *joystick*



Gambar 3.12 Diagram Alir Keseluruhan Sistem Pada *Joystick*

Tabel 3.4 Proses Pengiriman Data

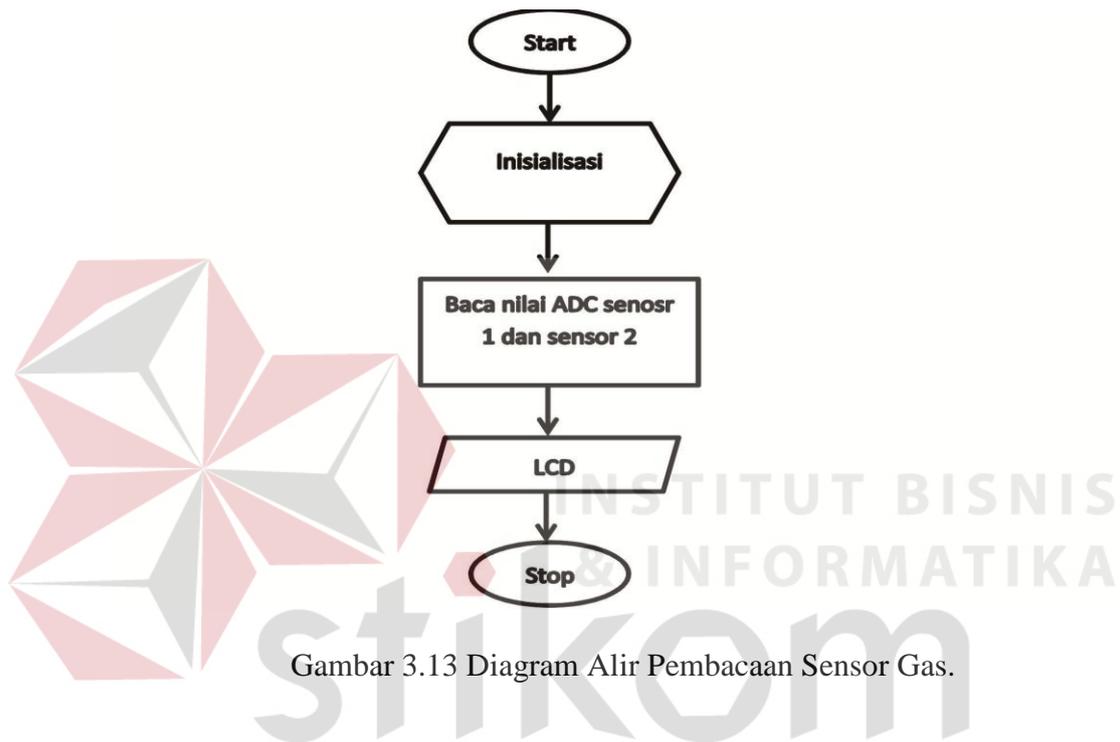
No.	Tombol <i>Joystick</i>	Data yang dikirim	Pergerakan Robot
1	↑	A	Maju
2	↓	B	Mundur
3	→	C	Belok Kanan
4	←	D	Belok Kiri
5	↘	E	Maju Serong Kanan

6		F	Maju Serong Kiri
7		G	Mundur Serong Kanan
8		H	Mundur Serong Kiri
9		I	Kecepatan + Maju
10		J	Kecepatan + Mundur
11		K	Kecepatan + Kanan
12		L	Kecepatan + Kiri
13		M	Kecepatan + Maju Serong Kanan
14		N	Kecepatan + Maju Serong Kiri
15		O	Kecepatan + Mundur Serong Kanan
16		P	Kecepatan + Mundur Serong Kiri
17		S	Mengaktifkan Kipas Dan Sensor Gas
18	-	Z	Berhenti (tidak ada <i>input</i>)

Pada Gambar 3.12 dijelaskan bahwa proses pengiriman data pada *joystick* dimulai dari inialisasi SPI yaitu proses terhubungnya SPI *joystick* dengan *microcontroller*. Proses selanjutnya yaitu pembacaan data pada SPI dan dilanjutkan dengan proses pengiriman data. Proses pengiriman data dapat dilihat pada Tabel 3.4. Setelah proses pengiriman data maka *joystick* menunggu data masukan dari sensor gas. Jika tidak ada data masukan dari sensor gas maka prosesnya akan kembali ke proses pengiriman data. Jika sudah menerima data dari sensor gas, maka proses selanjutnya data dari sensor gas tersebut akan ditampilkan di LCD.

3.5.1 Program Membaca Sensor Gas dan Menampilkan Pada LCD

Diagram alir untuk mengetahui nilai ADC (*Analog to Digital Converter*) dari bensin maupun LPG terhadap *mobile Robot* berdasarkan pembacaan sensor gas (MQ2 dan MQ4) dan ditampilkan ke LCD terdapat pada Gambar 3.13.



Gambar 3.13 Diagram Alir Pembacaan Sensor Gas.

Pada Gambar 3.13 diagram alir pembacaan nilai ADC bensin maupun LPG pada sensor gas MQ2 dan MQ4, dimulai dengan inisialisasi sensor gas MQ2 dan MQ4, kemudian melakukan penghitungan lebar pulsa dan data sensor gas MQ2 dan MQ4 tersebut disimpan pada sebuah variabel dan data tersebut dimasukkan dalam rumus untuk mengubah pulsa menjadi nilai digital, Setelah itu ditampilkan ke LCD. Berikut potongan program pembacaan sensor gas MQ2 dan MQ4 serta menampilkan ke Komputer :

```

void baca_sensor()
{

```

```

kipas = 1;
kanan_stop;
kiri_stop;

atauataufomat pengiriman
atauatauP(data1)S(data2)K%
delay_ms(5000);
atauataumemulai kirim data sensor
putchar('P');

if(sensor1 < 10) printf("00%dS",sensor1);
else if(sensor1 >= 10 && sensor1 < 100)
printf("0%dS",sensor1);
else printf("%dS",sensor1);

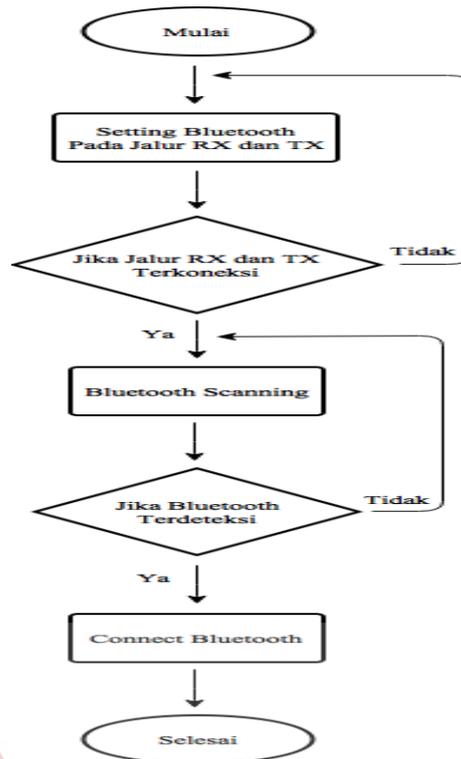
if(sensor2 < 10) printf("00%dK",sensor2);
else if(sensor2 >= 10 && sensor2 < 100)
printf("0%dK",sensor2);
else printf("%dK",sensor2);

delay_ms(1);
}

```

3.5.2 Program Komunikasi *Bluetooth* HC-05

Pada pengujian untuk komunikasi *Bluetooth* terdapat *flowchart* dimana *flowchart* tersebut merupakan alur dari cara modul *Bluetooth* untuk saling terkoneksi. Dibawah ini pada Gambar 3.14 merupakan diagram alir dari komunikasi *Bluetooth*.

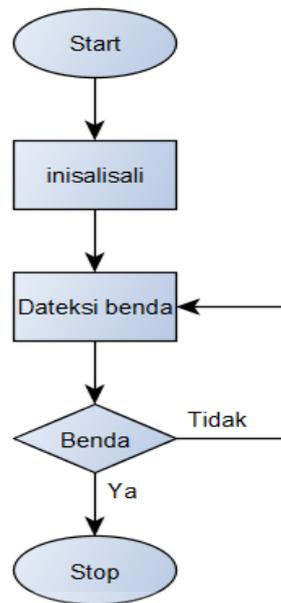


Gambar 3.14 Diagram alir komunikasi *Bluetooth*.

Pada Gambar 3.14 diagram komunikasi *bluetooth* dimulai dengan mensetting *Bluetooth* pada jalur RX dan TX. Setelah itu jika jalur RX dan TX terkoneksi maka *Bluetooth* akan melakukan proses *scanning* dan jika tidak terkoneksi maka *Bluetooth* pada jalur RX dan TX harus disetting lagi. Setelah melakukan proses *scanning*, jika *Bluetooth* terdeteksi maka *Bluetooth* berhasil terhubung, dan jika belum terdeteksi maka *Bluetooth* tetap melakukan *scanning* sampai *bluetooth* berhasil terhubung.

3.5.3 Program Membaca Sensor *Infrared*

Diagram alir untuk mengetahui jarak benda terhadap *mobile Robot* berdasarkan pembacaan sensor *infrared* dan ditampilkan ke LCD terdapat pada Gambar 3.15.



Gambar 3.15 Diagram alir pembacaan sensor *infrared*

Pada diagram alir pembacaan sensor *infrared* terhadap benda dimulai dengan inisialisasi sensor, setelah itu sensor proses mendeteksi benda, jika benda tidak ada maka kembali ke proses pendeteksi benda, jika benda ada maka proses berhenti.

3.6 Metode Pengujian dan Evaluasi Sistem

Dalam pengujian sistem ini pengujian akan dilakukan pada perangkat keras serta perangkat lunak yang telah dibuat. Pengujian yang telah dilakukan dimulai dari pengujian minimum sistem, pengujian *motor dc*, pengujian komunikasi *Bluetooth*, pengujian sensor gas MQ-2 dan MQ-4, serta pengujian sensor *Infrared*.

3.6.1 Pengujian dan Evaluasi Minimum Sistem (ATMega32A)

Pengujian minimum sistem ini bertujuan untuk mengetahui apakah minimum sistem dalam *mobile Robot* dapat melakukan proses *signature* dan *download* program ke *microcontroller* dengan baik. Pengujian ini dilakukan

dengan cara mengaktifkan *power supply* dan hubungkannya dengan minimum sistem. Sambungkan minimum sistem dengan komputer menggunakan kabel *downloader* lalu jalankan *compiler CodeVisionAVR* pada komputer tersebut. Setelah itu lakukan proses *chip signature* untuk mengetahui apakah sudah terkoneksi dengan baik dengan *microcontroller*. Jika proses *chip signature* berhasil maka akan keluar informasi tentang *microcontroller* yang terhubung antara komputer dengan *microcontroller*. Kemudian lakukan proses *download* pada *microcontroller*, jika proses *download* berhasil maka akan *CodeVisionAVR* akan mengeluarkan tampilan proses *download*.

3.6.2 Pengujian dan Evaluasi *Driver Motor* dan *Motor DC*

Pengujian *motor dc* ini bertujuan untuk mengetahui apakah *motor dc* bergerak atau berputar sesuai dengan yang diharapkan dan berputar dengan *pwm* tertentu sesuai dengan program yang telah di tentukan. Dalam hal ini pengujian dilakukan dengan memberikan tegangan pada driver *motor dc* dan melakukan koneksi antara *driver motor dc* dengan minimum sistem. Setelah itu proses dilanjutkan dengan menggunakan program untuk menggerakkan *motor dc* pada *compiler CodeVisionAVR* yang di *download* pada minimum sistem. Jika proses pengujian tersebut berhasil maka *motor dc* akan bergerak sesuai dengan yang diperintahkan dalam program yang terdapat pada *microcontroller*.

3.6.3 Pengujian dan Evaluasi Sensor Gas (MQ2 dan MQ4)

Pengujian sensor gas ini bertujuan untuk mengetahui informasi bau gas yang akan dideteksi. Dalam hal ini pengujian dilakukan dengan memberikan tegangan pada sensor gas dan melakukan koneksi antara sensor gas dengan

minimum sistem pada Robot. Setelah itu proses dilanjutkan dengan membaca nilai ADC pada *compiler CodeVisionAVR* yang akan di *download* pada *minimum sistem* dan akan ditampilkan pada LCD. Jika proses pengujian tersebut berhasil maka LCD akan menampilkan data berupa nilai kadar gas dari bensin maupun LPG.

3.6.4 Pengujian dan Evaluasi Sensor *Infrared*

Pengujian sensor *infrared* ini bertujuan untuk mengetahui informasi jarak antara benda dengan *mobile Robot*. Dalam hal ini pengujian dilakukan dengan memberikan tegangan pada sensor *infrared* dan melakukan koneksi antara sensor *infrared* dengan *minimum sistem*. Setelah itu proses dilanjutkan dengan menggunakan program membaca nilai “0” dan “1” dari sensor *infrared* pada *compiler CodeVisionAVR* yang akan di *download* pada *minimum sistem ATMega32*.

3.6.5 Pengujian dan Evaluasi Komunikasi *Bluetooth HC-05*

Pengujian ini bertujuan untuk mengetahui apakah proses pengiriman data pada *Bluetooth* berjalan sesuai yang diharapkan. Didalam pengujian ini *Bluetooth* pada joystick akan mengirimkan data yang berupa karakter ke *Bluetooth* pada Robot dan peneliti dapat mengetahui apakah data yang diterima oleh *Bluetooth* sesuai dengan data yang dikirim.