

BAB III

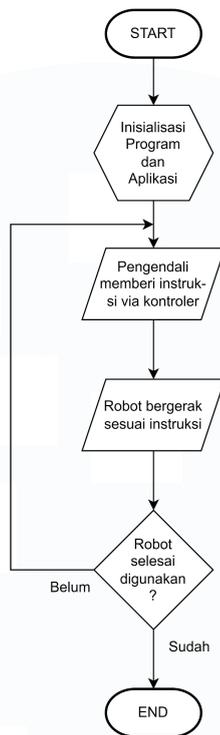
RANCANGAN DAN IMPLEMENTASI

3.1 Perancangan Pemrograman Robot Pengumpan

Robot diprogram menggunakan bahasa pemrograman C++. Robot menggunakan mikrokontroler Arduino Mega, sehingga pemrograman dilakukan pada aplikasi Arduino IDE. Berdasarkan peraturan perlombaan, robot dapat dikendalikan secara nirkabel atau beroperasi secara otonom. Metode yang dipilih adalah pengendalian secara nirkabel dengan alasan kemudahan dan komponen yang digunakan. Pengendalian nirkabel menawarkan kemudahan dari sisi pemrograman dan navigasi robot. Komponen yang digunakan dengan pengendalian nirkabel juga lebih sedikit dibanding robot otonom. Robot otonom membutuhkan sensor-sensor pendukung lainnya, seperti sensor ultrasonik, sensor *heading angle*, dan *encoder*. Sensor yang dibutuhkan dengan metode pengendalian nirkabel hanya Bluetooth *transceiver*.

Program yang dibuat memiliki spesifikasi dapat menerima instruksi dari pengendali secara nirkabel untuk menggerakkan roda dan lengan robot. *Flowchart* dari interaksi pengendali dan robot dapat dilihat pada Gambar 3.1.





Gambar 3.1 *Flowchart* Interaksi Pengendali dengan Robot yang Dikendalikan

Untuk mewujudkan spesifikasi tersebut, diperlukan sebuah program untuk mengirim instruksi dari pengendali serta program untuk menerima instruksi dan menjalankan fungsi tertentu pada robot. Fungsi yang diperlukan antara lain fungsi untuk bergerak maju, mundur, belok kiri, belok kanan, mengangkat tempat sampah, menuang isi tempat sampah, dan menaruh tempat sampah.

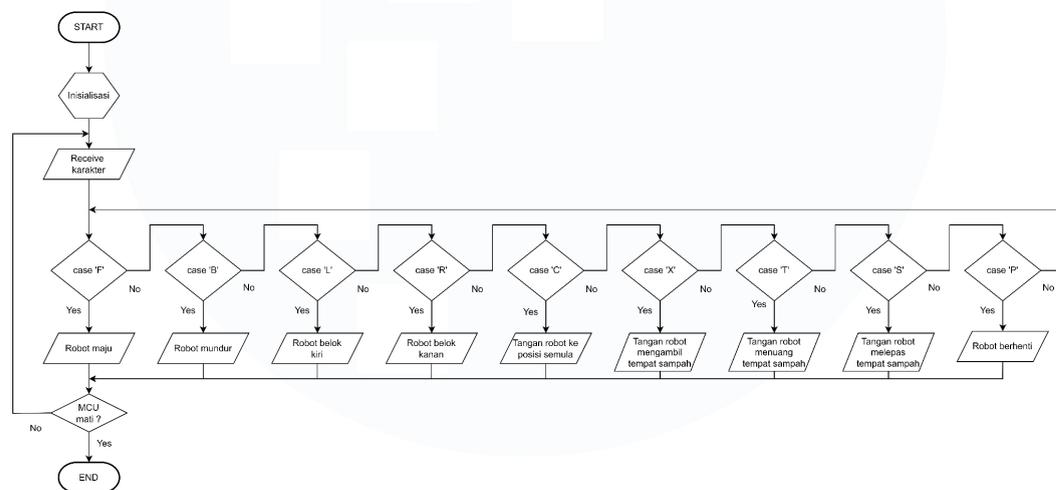
Pengendalian robot menggunakan protokol komunikasi Bluetooth. Kontroler (*transmitter*) menggunakan aplikasi “Arduino Bluetooth Controller”. Instruksi dikirimkan dalam bentuk karakter. Daftar karakter yang dapat digunakan dapat dilihat pada Tabel 3.1.

Tabel 3.1 Daftar Karakter Berdasarkan Instruksi

Tombol	Karakter Ketika Tombol Ditekan	Karakter Ketika Tombol Dilepas
Panah atas	‘F’	‘P’
Panah kanan	‘R’	‘P’
Panah bawah	‘B’	‘P’

Panah kiri	'L'	'P'
Segitiga	'T'	'P'
Bulat	'C'	'P'
Silang	'X'	'P'
Kotak	'S'	'P'

Flowchart rancangan dari program sisi robot (*receiver*) yang dibuat dapat dilihat pada Gambar 3.2.



Gambar 3.2 Flowchart Rancangan pada Sisi Receiver

3.2 Sensor dan Aktuator

Robot memiliki sensor dan aktuator sebagai perantara antara dunia luar dengan sistem. Sensor dan aktuator yang sesuai dengan fungsi pada masing-masing robot dijabarkan pada Tabel 3.2.

Tabel 3.2 Sensor dan Aktuator Robot Pemilah dan Pengumpan Sampah

Robot Pemilah Sampah	Robot Pengumpan Sampah
Sensor: <ol style="list-style-type: none"> HuskyLens Sensor induktif Sensor <i>infrared</i> <i>Rotary encoder</i> 	Sensor: <ol style="list-style-type: none"> Bluetooth <i>transceiver</i>
Aktuator:	Aktuator:

<ol style="list-style-type: none"> 1. Servo 2. Motor DC 3. Motor pompa 	<ol style="list-style-type: none"> 1. Servo 2. Motor DC
---	---

3.3 Pengujian Sensor

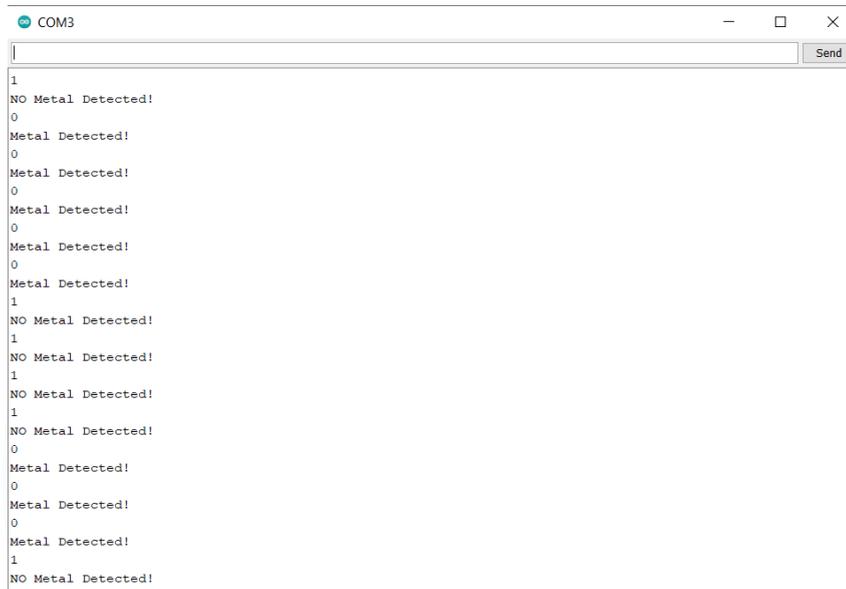
Semua sensor yang digunakan perlu diprogram terlebih dahulu agar dapat berfungsi sebagaimana mestinya. Pemrograman dan pengujian sensor dijabarkan sebagai berikut.

3.3.1 Pengujian Sensor Induktif

Sensor induktif digunakan untuk mendeteksi sampah logam. Sensor induktif dipilih karena waktu respon yang cepat dalam mendeteksi logam. Kegunaan sensor induktif yang lainnya adalah mengurangi kerja *image processing* dari HuskyLens. HuskyLens tidak perlu melakukan *image processing* bila menghadapi sampah logam, sehingga HuskyLens tidak perlu bekerja terlalu berat. Tipe sensor induktif yang digunakan adalah LJ12A3-4-Z yang biasanya digunakan untuk skala industri. Wujud sensor dapat dilihat pada Gambar 3.3. Walaupun biasa digunakan dalam skala industri, sensor ini masih dapat digunakan untuk Arduino. Sensor menghasilkan output sinyal *HIGH or LOW*. Sensor menghasilkan output *LOW* bila mendeteksi logam dan menghasilkan output *HIGH* bila tidak mendeteksi logam, seperti pada Gambar 3.4 dan Tabel 3.3.



Gambar 3.3 Pengujian Sensor Induktif dengan Sampah Logam



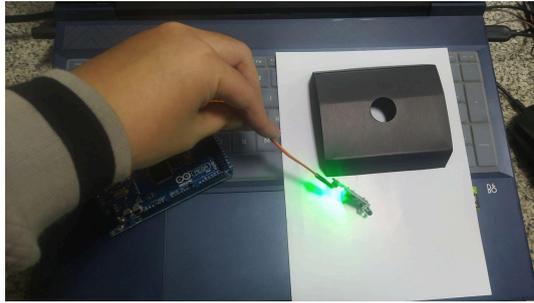
Gambar 3.4 Hasil Pembacaan Sensor Induktif

Tabel 3.3 Hasil Pengujian Sensor Induktif

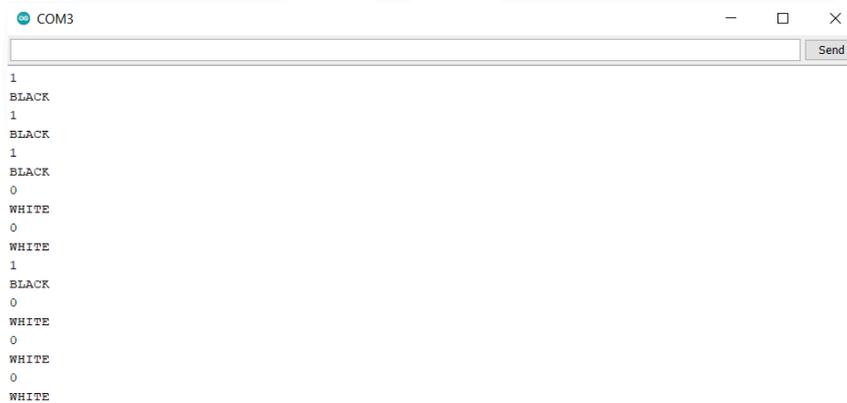
Jenis Sampah	Hasil Deteksi
Logam feromagnetik	Logam terdeteksi (<i>LOW</i>)
Logam non-feromagnetik	Logam terdeteksi (<i>LOW</i>)
Daun	Logam tidak terdeteksi (<i>HIGH</i>)
Plastik	Logam tidak terdeteksi (<i>HIGH</i>)
Kertas	Logam tidak terdeteksi (<i>HIGH</i>)

3.3.2 Pengujian Sensor *Infrared*

Sensor *infrared* digunakan untuk mendeteksi garis jalur robot pada arena. Tipe sensor *infrared* yang digunakan adalah FC-51 yang umum digunakan untuk sistem *line follower* dengan Arduino. Wujud sensor dapat dilihat pada Gambar 3.5. Sensor menghasilkan output sinyal *HIGH or LOW*. Sensor menghasilkan output *LOW* bila mendeteksi adanya pantulan sinar *infrared* (mengenai permukaan berwarna terang) dan menghasilkan output *HIGH* bila tidak mendeteksi adanya pantulan sinar *infrared* (mengenai permukaan berwarna gelap), seperti pada Gambar 3.6 dan Tabel 3.4.



Gambar 3.5 Pengujian Sensor *Infrared* dengan Permukaan Berwarna Hitam dan Putih



Gambar 3.6 Hasil Pembacaan Sensor *Infrared*

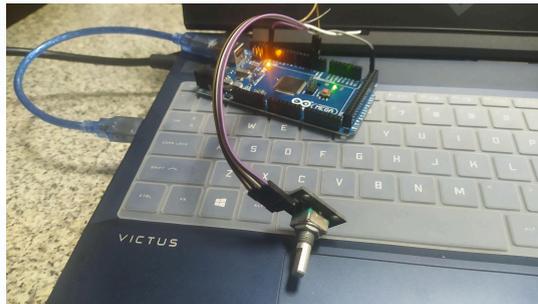
Tabel 3.4 Hasil Pengujian Sensor *Infrared*

Warna	Hasil Deteksi
Hitam	Hitam (<i>HIGH</i>)
Putih	Putih (<i>LOW</i>)

3.3.3 Pengujian *Rotary Encoder*

Rotary encoder digunakan untuk mendeteksi jumlah dan arah perputaran roda. Data jumlah perputaran roda dapat dikonversi menjadi data jarak tempuh robot, dan data arah perputaran roda dapat digunakan untuk penentuan posisi. Data *rotary encoder* dapat dilihat pada Gambar 3.8 dan Tabel 3.5. Kegunaan *rotary encoder* adalah untuk mengetahui posisi relatif robot pemilah pada saat proses pemilahan sampah di samping konveyor. Untuk mengatasi keterbatasan DOF lengan robot, robot perlu bergerak menyamping untuk dapat mengambil

sampah atau membuang sampah ke tempat sampah yang ditentukan secara akurat. *Rotary encoder* digunakan agar robot tidak bergerak secara acak ke kiri atau kanan yang dapat menyebabkan malfungsi robot. Tipe *rotary encoder* yang digunakan adalah KY-40 yang umum digunakan untuk Arduino. Wujud *encoder* dapat dilihat pada Gambar 3.7.



Gambar 3.7 Pengujian *Rotary Encoder*

```

COM3
Send
Direction: CCW | Counter: -14
Direction: CCW | Counter: -15
Direction: CW | Counter: -14
Direction: CW | Counter: -13
Direction: CW | Counter: -12
Direction: CW | Counter: -11
Direction: CW | Counter: -10
Direction: CW | Counter: -9
Direction: CW | Counter: -8
Direction: CW | Counter: -7
Direction: CW | Counter: -6
Direction: CW | Counter: -5
Direction: CW | Counter: -4
Direction: CW | Counter: -3
Direction: CW | Counter: -2
Direction: CW | Counter: -1
Direction: CW | Counter: 0
Direction: CW | Counter: 1
Direction: CW | Counter: 2
Direction: CW | Counter: 3
Direction: CW | Counter: 4
Direction: CW | Counter: 5
Direction: CW | Counter: 6
Direction: CW | Counter: 7
Direction: CW | Counter: 8
Direction: CW | Counter: 9
Direction: CCW | Counter: 8
Direction: CCW | Counter: 7
    
```

Gambar 3.8 Hasil Pembacaan *Rotary Encoder*

Tabel 3.5 Hasil Pengujian *Rotary Encoder*

Perlakuan	Hasil Deteksi
Putar searah jarum jam	Putaran searah jarum jam, nilai bertambah (<i>increment</i>)
Putar berlawanan arah jarum jam	Putaran berlawanan arah jarum jam,

nilai berkurang (<i>decrement</i>)

3.3.4 Pengujian Bluetooth *Transceiver*

Bluetooth *transceiver* (*transmitter and receiver*) digunakan untuk mengendalikan robot pengumpan sampah secara nirkabel. Tipe Bluetooth *transceiver* yang digunakan adalah USB Host Shield 2.0 untuk Arduino. Wujud Bluetooth *transceiver* dapat dilihat pada Gambar 3.9. *Dongle* Bluetooth dihubungkan dengan USB *port*, lalu protokol Bluetooth dihubungkan dengan kontroler PS4. Pengujian ini menghasilkan sebuah kegagalan. Kontroler PS4 tidak tersambung dengan Bluetooth dari *dongle*. Penyebab kegagalan kemungkinan besar karena protokol Bluetooth *dongle* yang digunakan tidak cocok dengan protokol Bluetooth dari kontroler PS4.

Pengujian dilanjutkan dengan mengganti kontroler PS4 menjadi *wireless keyboard*. *Wireless keyboard* memiliki *dongle* pasangan, sehingga protokol Bluetooth yang digunakan pasti cocok. Namun, pengujian ini juga berujung kegagalan, dimana *wireless keyboard* tidak dapat terhubung dengan *dongle* yang dapat dilihat pada Gambar 3.10 dan Gambar 3.11. Kegagalan kemungkinan besar disebabkan dari USB Host Shield 2.0 yang digunakan. Kelemahan dari USB Host Shield 2.0 ini adalah proses *debugging* yang tidak bisa dilakukan, sehingga bila USB Host Shield 2.0 melakukan inisiasi program, *programmer* tidak dapat mengamati apa yang terjadi pada program, apakah program berhasil diinisialisasi atau tidak.



Gambar 3.9 Pengujian Bluetooth *Transceiver* dengan *Wireless Keyboard*



Gambar 3.10 *Wireless Keyboard* Tidak Tersambung dengan Protokol Bluetooth



Gambar 3.11 Input *Wireless Keyboard* Tidak Terbaca

Pengujian dilanjutkan dengan mengganti Bluetooth *transceiver* menjadi NRF24L01. NRF24L01 dapat digunakan sebagai *transmitter* dan *receiver*. Sisi *transmitter* dihubungkan dengan kontroler. Kontroler menggunakan *joystick* KY-023 dan *push button* untuk mengirimkan sinyal, selanjutnya kontroler pada umumnya. Wujud *transceiver* dan kontroler dapat dilihat pada Gambar 3.12. Sisi *receiver* dihubungkan dengan aktuator-aktuator robot pengumpan sampah. Pengujian Bluetooth *transceiver* dapat dilihat pada Gambar 3.13 dan Tabel 3.6.

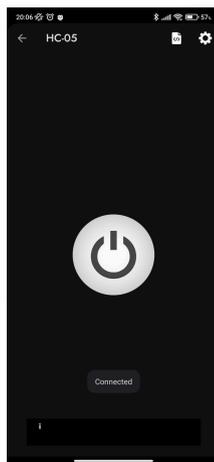
UNIVERSITAS
MULTIMEDIA
NUSANTARA

Sensor NRF24L01 yang digunakan mengalami kerusakan. Hal ini dikarenakan tegangan suplai yang berlebih (5 V). Merujuk *datasheet*, NRF24L01 menggunakan tegangan suplai 3,3 V. Namun, saat pertama kali dicoba, tegangan suplai yang diberikan adalah 3,3 V. Sensor NRF24L01 tidak bekerja. Sensor NRF24L01 baru bekerja pada tegangan suplai 5 V. Bila diberi tegangan suplai sebesar 5 V, sensor NRF24L01 mampu berfungsi selama kurang lebih 8 jam.

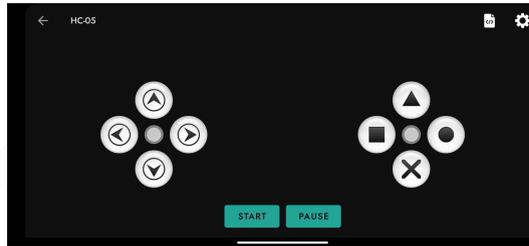
Bluetooth *transceiver* diganti menggunakan sensor HC-05. Sensor HC-05 digunakan sebagai *receiver*. *Transmitter* menggunakan Bluetooth dari HP, melalui aplikasi “Arduino Bluetooth Controller”. Wujud Bluetooth *transceiver* dapat dilihat pada Gambar 3.14. Kontroler melalui HP dapat dilihat pada Gambar 3.15 dan 3.16. Pengujian Bluetooth *transceiver* dapat dilihat pada Tabel 3.7.



Gambar 3.14 Pengujian Bluetooth *Transceiver* HC-05



Gambar 3.15 Tampilan Kontroler dari HP untuk Menyalakan dan Mematikan LED



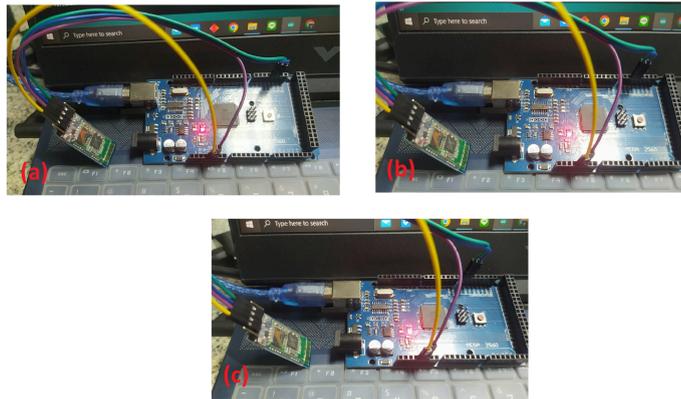
Gambar 3.16 Tampilan Kontroler dari HP untuk Kontrol Robot

Tabel 3.7 Hasil Pengujian Bluetooth *Transceiver* HC-05

Perlakuan	Hasil Deteksi
Panah atas	'F'
Panah kanan	'R'
Panah bawah	'B'
Panah kiri	'L'
Segitiga	'T'
Bulat	'C'
Silang	'X'
Kotak	'S'
Start	'A'
Pause	'P'

Sensor HC-05 digunakan dengan bantuan SoftwareSerial dikarenakan pin RX TX (pin digital 0 dan 1) Arduino Uno digunakan untuk komunikasi dengan komputer. Dengan bantuan SoftwareSerial, pin digital 10 dan 11 bisa digunakan sebagai RX TX, sehingga tidak mengganggu komunikasi dengan komputer pada saat *testing* dan *debugging*. Namun masalah timbul saat menggunakan pin RX TX Arduino Mega. Arduino Mega memiliki pin RX TX tambahan sebanyak 3 pasang. Sensor HC-05 tidak bisa menggunakan pin RX TX tambahan tersebut. Lampu LED pin digital 13 tidak menyala saat dikendalikan melalui HP, seperti pada Gambar 3.17. Hal ini kemungkinan disebabkan cacat produksi pada Arduino Mega atau sensor HC-05, dikarenakan tidak ada kesalahan dalam pengkabelan dan pemrograman. Oleh karena itu, sensor HC-05 dihubungkan dengan pin digital

10 dan 11 menggunakan *library* SoftwareSerial. Risiko menggunakan SoftwareSerial adalah adanya *delay* serta *baudrate* terbatas dari 9600 hingga 38400 (jangkauan optimal). *Delay* dan *baudrate* tersebut dapat mengurangi efektivitas komunikasi nirkabel.



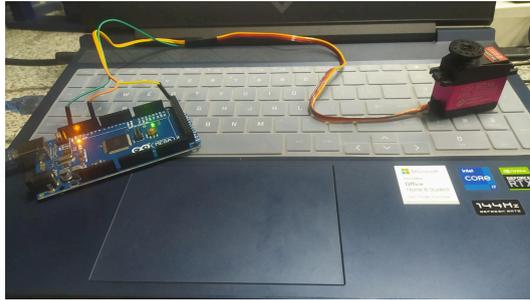
Gambar 3.17 (a) RX1 TX1 Tidak Merespon Input Kontroler, (b) RX2 TX2 Tidak Merespon Input Kontroler, (c) RX3 TX3 Tidak Merespon Input Kontroler

3.4 Pengujian Aktuator

Semua aktuator yang digunakan perlu diprogram terlebih dahulu agar dapat berfungsi sebagaimana mestinya. Pemrograman dan pengujian aktuator dijabarkan sebagai berikut.

3.4.1 Pengujian Servo

Servo digunakan untuk pergerakan dengan akurasi tinggi, seperti pada lengan robot pemilah dan pengumpan sampah. Tipe servo yang digunakan adalah DS3225 yang memiliki torsi besar. Wujud servo dapat dilihat pada Gambar 3.18. Servo dapat diatur untuk bergerak ke posisi tertentu, dengan rentang dari 0° hingga 180° . Servo diprogram agar bergerak secara perlahan dengan tujuan meminimalisir energi momentum yang dapat merusak persendian lengan robot. Pergerakan servo diatur seperti pergerakan motor *stepper*, yakni bergerak per derajat dalam waktu tertentu. Pengujian servo dapat dilihat pada Tabel 3.8.



Gambar 3.18 Pengujian Servo

Tabel 3.8 Hasil Pengujian Servo

Servo	Status
Servo 1	Berputar sesuai derajat tertentu
Servo 2	Berputar sesuai derajat tertentu
Servo 3	Berputar sesuai derajat tertentu
Servo 4	Berputar sesuai derajat tertentu
Servo 5	Berputar sesuai derajat tertentu

3.4.2 Pengujian Motor DC

Motor DC digunakan untuk pergerakan robot pemilah dan pengumpulan sampah. Tipe motor DC yang digunakan adalah motor DC 12 V yang mampu berputar dengan kecepatan hingga 12 ribu rpm. Wujud motor DC dapat dilihat pada Gambar 3.19. Setelah diuji, motor ini kurang cocok untuk aplikasi motor penggerak roda. Motor ini memiliki kecepatan tinggi, tetapi torsi yang dihasilkan kecil. Motor diuji dengan diberi beban jepitan jari di ujung *shaft* motor. Tidak lama kemudian, motor langsung memutus suplai listrik.



Gambar 3.19 Pengujian Motor DC *High Speed*

Tipe motor DC diganti dengan jenis motor DC 12 V yang dilengkapi dengan *gearbox* bawaan. Wujud motor DC dapat dilihat pada Gambar 3.20. Dengan adanya *gearbox*, torsi motor dapat ditingkatkan. Konsekuensi adanya *gearbox* adalah kecepatan motor yang berkurang menjadi 130 rpm.



Gambar 3.20 Pengujian Motor DC dengan *Gearbox*

Agar dapat dikendalikan, motor disambungkan dengan *motor driver*. Tipe *motor driver* yang digunakan adalah L298N yang umum digunakan untuk Arduino. *Motor driver* digunakan untuk mengendalikan kecepatan dan arah gerak motor. Kecepatan motor dapat diatur dengan *duty cycle* PWM. Arah gerak motor dapat diatur dari sinyal *HIGH and LOW* pin digital untuk mengubah polaritas motor. Pengujian motor dapat dilihat pada Gambar 3.21 dan Tabel 3.9.



Gambar 3.21 Pengujian Motor DC Dihubungkan dengan *Motor Driver*

Tabel 3.9 Hasil Pengujian Motor DC

Motor DC	Perlakuan Polaritas +-	Perlakuan Polaritas -+
Motor 1	Berputar	Berputar dengan arah berlawanan
Motor 2	Berputar	Berputar dengan arah berlawanan
Motor 3	Berputar	Berputar dengan arah berlawanan
Motor 4	Berputar	Berputar dengan arah berlawanan
Motor 5	Berputar	Berputar dengan arah berlawanan
Motor 6	Berputar	Berputar dengan arah berlawanan

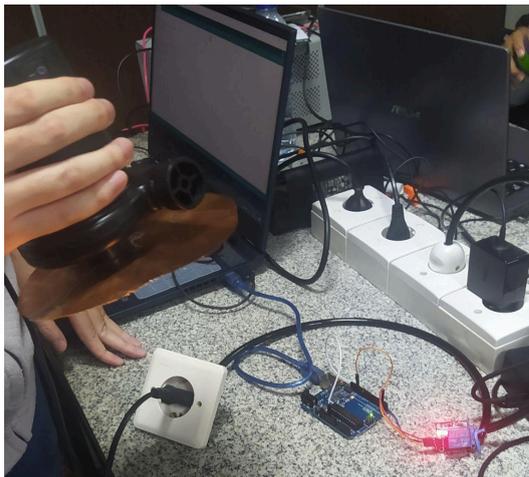
3.4.3 Pengujian Motor Pompa

Motor pompa digunakan untuk menyedot sampah. Motor pompa berasal dari alat pemompa balon. Motor menggunakan listrik AC rumahan (220 V, 50 Hz). Wujud motor pompa dapat dilihat pada Gambar 3.22.



Gambar 3.22 Pengujian Motor Pompa untuk Menyedot Sampah Besi

Agar dapat dikendalikan, motor disambungkan dengan *relay*. Tipe *relay* yang digunakan adalah *relay* SPDT (*Single Pole Double Throw*) SRD-05VDC-SL-C yang umum digunakan untuk Arduino. *Relay* bertindak sebagai saklar yang dikendalikan oleh listrik. Tidak mungkin untuk mengendalikan listrik AC rumahan menggunakan pin digital Arduino secara langsung. Dibutuhkan *relay* sebagai perantara. *Relay* diberi sinyal *HIGH or LOW* dari Arduino untuk mengendalikan saklar yang terhubung ke listrik AC rumahan. Pengujian motor dapat dilihat pada Gambar 3.23 dan Tabel 3.10.



Gambar 3.23 Pengujian Motor Pompa untuk Menyedot Sampah Besi Dikendalikan dengan *Relay*

Tabel 3.10 Hasil Pengujian Pompa Motor

Jenis Sampah	Status
Logam feromagnetik	Tersedot
Logam non-feromagnetik	Tersedot
Daun	Tersedot
Plastik	Tersedot
Kertas	Tersedot

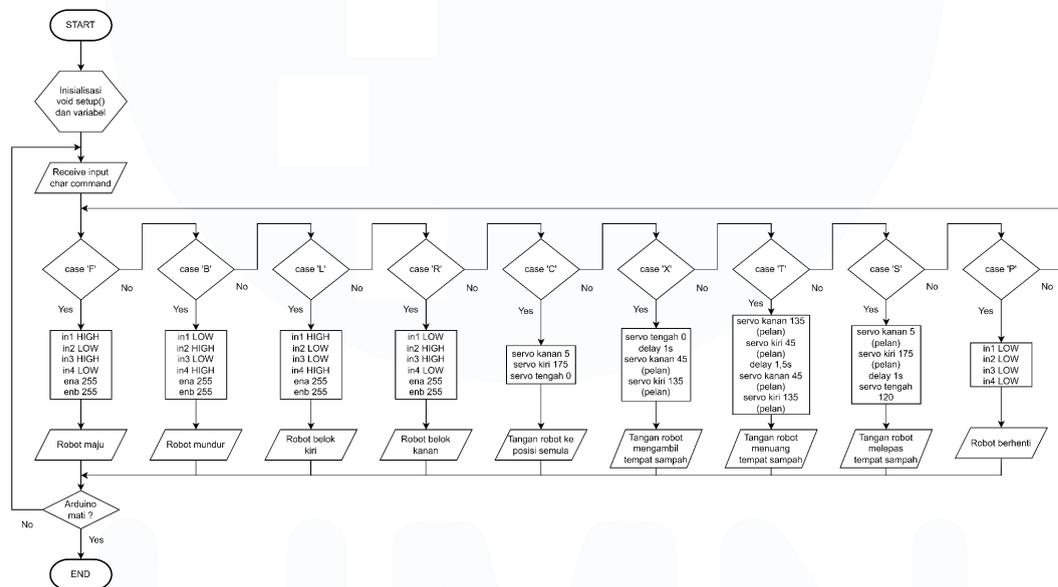
3.5 Integrasi Sistem Robot Pengumpan

Sensor dan aktuator yang sudah diprogram dan diuji dapat diintegrasikan menjadi suatu sistem robot. Program integrasi sistem dibuat berdasarkan rancangan program yang telah ditentukan sebelumnya. Program dimulai dengan mendefinisikan variabel karakter-karakter yang digunakan. Lalu berikutnya dilanjut dengan memasukkan *library* SoftwareSerial dan Servo. Selanjutnya, variabel servo, sudut penyesuaian, dan variabel pin motor *driver* dideklarasikan.

Pada bagian *void setup()*, *baud rate* diatur. Pin servo juga dideklarasikan pada bagian ini. Servo diatur pada posisi lengan normal pada saat *setup*. Selanjutnya, pin motor *driver* dideklarasikan menjadi pin output.

Pada bagian *void loop()*, program akan mendeteksi apakah ada protokol komunikasi yang terhubung. Bila ada, maka karakter yang dikirim dari kontroler akan dibaca. Fungsi robot diprogram dalam sebuah *switch case*. Bila karakter 'F' (FORWARD) terbaca, maka pin motor *driver* diatur supaya motor dapat bergerak dengan arah maju. Bila karakter 'B' (BACKWARD) terbaca, maka pin motor *driver* diatur supaya motor dapat bergerak dengan arah mundur. Bila karakter 'L' (LEFT) terbaca, maka pin motor *driver* diatur supaya motor dapat bergerak dengan arah belok ke kiri. Bila karakter 'R' (RIGHT) terbaca, maka pin motor *driver* diatur supaya motor dapat bergerak dengan arah belok ke kanan. Bila karakter 'C' (CIRCLE) terbaca, maka sudut servo diatur agar posisi lengan dalam kondisi normal. Bila karakter 'X' (CROSS) terbaca, maka sudut servo diatur agar

lengan bisa menjepit tempat sampah, lalu mengangkatnya. Bila karakter 'T' (TRIANGLE) terbaca, maka sudut servo diatur supaya lengan dapat menuang isi tempat sampah, lalu kembali ke posisi mengangkat tempat sampah. Bila karakter 'S' (SQUARE) terbaca, maka sudut servo diatur supaya lengan dapat menurunkan tempat sampah, lalu merenggangkan jepitan. Bila karakter 'P' (PAUSE) terbaca, maka pin motor *driver* diatur supaya motor berhenti bergerak. Karena program dijalankan secara sekuensial, maka saat sedang melakukan suatu fungsi, fungsi lainnya tidak bisa dijalankan secara bersamaan. *Flowchart* dari program ini dapat dilihat pada Gambar 3.24.



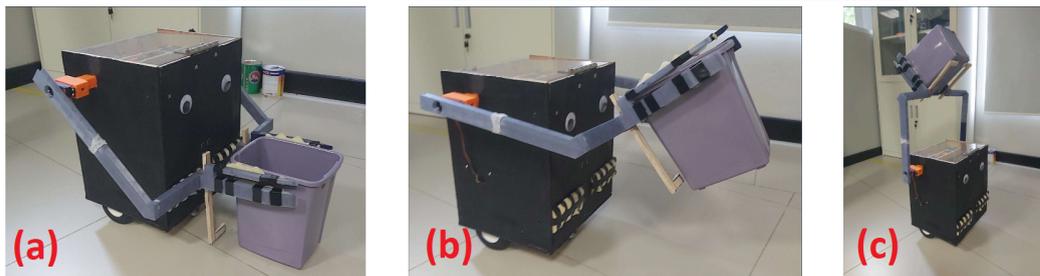
Gambar 3.24 *Flowchart* Program Hasil Integrasi Sistem

3.6 Keberhasilan Integrasi Sistem dan Pengujian Robot Pengumpan

Berdasarkan semua proses pengujian sensor dan aktuator serta integrasi, dapat disimpulkan bahwa semua sensor dan aktuator berjalan sesuai dengan fungsinya, sehingga dapat diintegrasikan menjadi sebuah sistem untuk mengoperasikan robot. Walaupun sempat menemui kegagalan dalam pengujian Bluetooth *transceiver* dan motor DC, masalah tersebut dapat diatasi. Bluetooth *transceiver* dapat diatasi dengan penggantian komponen serta penggunaan pin

digital lain yang tersedia pada Arduino Mega. Pada motor DC, masalah dapat diatasi dengan penggantian komponen.

Robot pengumpan sampah berhasil dibuat, serta berhasil dioperasikan secara nirkabel untuk melakukan fungsi-fungsi yang telah ditentukan. Hasil robot pengumpan sampah dapat dilihat pada Gambar 3.25.



Gambar 3.25 (a) Robot Deimos pada Keadaan Posisi Normal, (b) Robot Deimos pada Keadaan Mengangkat Tempat Sampah, (c) Robot Deimos pada Keadaan Membuang Isi Tempat Sampah

UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA