

BAB I

PENDAHULUAN

A. Latar Belakang

Pada era industri 4.0 saat ini, robot adalah suatu hal yang tidak asing lagi bagi kehidupan manusia, dimana pada jaman sekarang, robot sudah berada pada setiap sektor kehidupan manusia agar pekerjaan manusia lebih mudah, baik pada sektor ekonomi, pendidikan, sosial, ekonomi, bahkan pada sektor industri, dimana peran manusia seperti pelabelan, pengelasan dan kegiatan lain, dapat dilakukan oleh robot.

Saat ini hampir semua orang mengenal yang namanya robot. Banyak robot-robot yang telah dibuat oleh para ahli menirukan bentuk anatomi makhluk hidup. Salah satu yang sangat disukai adalah robot berkaki. Robot merupakan perangkat otomatis yang mampu bergerak sendiri untuk menyelesaikan sebuah pekerjaan. Banyak sekali jenis-jenis robot yang ada. Namun berdasarkan alat geraknya robot ini diklasifikasikan menjadi dua jenis yaitu robot berkaki dan robot beroda. Robot berkaki adalah robot yang bermanuver dengan kaki-kaki buatan, baik dengan robot berkaki dua yang sering disebut dengan robot humanoid, robot berkaki tiga disebut dengan robot tripod, robot berkaki empat yang disebut dengan robot quadruped, robot berkaki enam yang disebut dengan robot hexapod, dan robot berkaki banyak yang lainnya. Sedangkan, robot beroda

adalah robot yang bermanuver dengan menggunakan roda, baik itu dengan dua roda ataupun lebih dari dua roda. (W. Ansori Dkk,2014).

Dan disini saya membahas robot berkaki empat yang disebut dengan robot quadruped. Penggunaan robot berkaki ini sangat meningkat didunia robotika karena dengan keterbatasan sistem gerak pada robot beroda. Pergerakan robot quadruped berkaki empat ini terdiri dari dua belas motor servo yang digunakan dan harus dikontrol sudutnya untuk mencapai posisi yang diinginkan. Untuk kontrol robot menggunakan bluetooth smartphone yang terkoneksi dengan modul HC-05, sehingga setiap pergerakan robot akan dikontrol melalui smartphone..

B. Rumusan Masalah

1. Bagaimana merancang robot berkaki empat 12 dof ?
2. Bagaimana menerapkan smartphone control dengan bluetooth ke dalam robot berkaki empat 12 dof?

C. Tujuan Penelitian

Pengembangan sistem keberagaman robotik yang sudah ada dengan menambahkan variasi dalam pembuatannya,dengan hal tersebut diharapkan semakin banyak juga variasi robot yang dapat digunakan dalam mempermudah setiap kegiatan manusia bahkan dapat menggantikan beberapa kegiatan manusia yang dapat digantikan oleh robot.

Dengan dibuatnya robot quadruped menggunakan 12 dof yang dapat dikontrol menggunakan smartphone diharapkan dapat menjadi alternatif atau

varian baru dalam pembuatan robot yang telah digunakan sebelumnya, dengan tingkat keseimbangan dan sudut gerak yang lebih variatif.

D. Manfaat Penelitian

1. Sebagai sarana implementasi keberagaman sistem robot dengan berbagai variasi yang telah ada.
2. Meningkatkan bentuk kreativitas pada dunia robotik
3. Sebagai pijakan dan referensi pada penelitian-penelitian selanjutnya yang berhubungan dengan robotik

E. Ruang Lingkup

Batasan masalah dari proyek ini adalah penggunaan mikrokontroler arduino nano sebagai pemroses data, HC-05 bluetooth module untuk menghubungkan smartphone dengan robot, SG90 mini servo sebagai penggerak kaki-kaki robot, dan Lithium battery 2 buah sebagai pemasok daya robot, kemudian robot akan dikendalikan melalui aplikasi pada smartphone

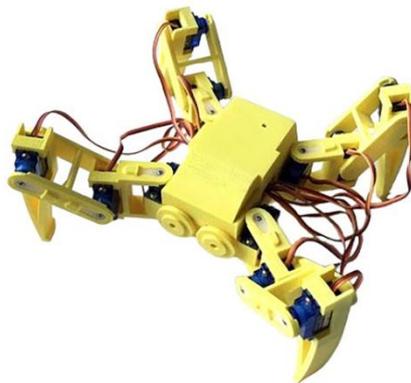
BAB II

TINJAUAN PUSTAKA

A. Kajian Teori

1. Robot

Robot berasal dari kata “robota” yang dalam bahasa Ceko memiliki arti bekerja. Robot adalah sebuah mesin perangkat dari hasil rakitan tangan manusia yang bekerja tanpa mengenal lelah.



Gambar 2.1 Robot Empat Kaki.

Sumber : Robot Quadruped (www.tokopedia.com)

. Banyak yang beranggapan bahwa robot adalah suatu alat yang menyerupai manusia, Tapi pada struktur tubuhnya terbuat dari logam. Beberapa pakar robot berupaya memberikan beberapa definisi mengenai robot, sebagai berikut:

1. Robot adalah sebuah alat manipulator yang dapat dilakukan program ulang untuk memindahkan alat, material, atau hal tertentu dengan berbagai gerakan yang akan diprogram untuk berbagai macam tugas.

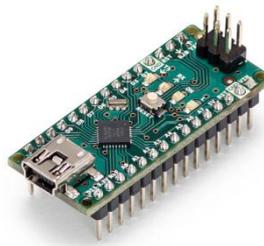
2. Robot adalah sebuah peralatan mesin yang dilengkapi dengan perangkat memori dan juga perangkat yang mampu berputar dan menggantikan pekerjaan manusia dengan gerakan otomatisasi.

2. Arduino IDE

IDE merupakan singkatan dari Integrated Development Environment, atau secara bahasa mudah merupakan suatu lingkungan terintegrasi yang digunakan untuk melakukan suatu pengembangan. Arduino IDE menggunakan bahasa pemrograman tersendiri yang hampir menyerupai bahasa C. Bahasa pemrograman Arduino sudah dilakukan suatu perubahan untuk memudahkan bagi para pemula dalam melakukan suatu pemrograman dari bahasa aslinya. Sebelum didistribusikan dan dijual ke pasaran, IC mikrokontroler Arduino sudah ditanamkan suatu program bernama *bootloader* yang memiliki fungsi sebagai penengah antara mikrokontroler dengan *compiler* arduino. Arduino IDE dibuat dari bahasa pemrograman JAVA. Arduino IDE juga dilengkapi dengan perpustakaan bahasa C/C++ yang biasa disebut dengan Wiring, fungsinya membuat suatu operasi masukan dan keluaran dengan mudah. Arduino IDE menunjukkan board yang terkonfigurasi beserta dengan ISSN/e-ISSN: 1907-4093/2087-9814 35 COM port yang digunakan.

3. Mikrokontroller Arduino Nano

Mikrokontroller Arduino Nano ATmega328 adalah sebuah papan pengembangan (development board), mikrokontroller yang berbasis pada chip ATmega328 dengan bentuk kecil. Sementara secara fungsi tidak berbeda dengan arduino uno.



Gambar 2.2 Arduino Nano.
Sumber : Arduino nano (store.arduino.cc)

Untuk perbedaan utama hanya terletak pada ketiadaan jack power DC dan pada penggunaan *connector* Mini-B USB. Pada chip ATmega328 memiliki arsitektur Harvard, yaitu dapat memisahkan memori untuk kode program dan memori untuk data. Sehingga dapat memaksimalkan kinerja. Instruksi dalam memori program dilakukan dalam satu alur tunggal, dimana pada saat satu instruksi dilakukan dengan instruksi berikutnya yang telah diambil dari memori program. Konsep inilah akan memungkinkan suatu instruksi-instruksi yang dapat dieksekusi dalam satu siklus clock. Pemrograman board Arduino Nano dilakukan dengan menggunakan software Arduino. Pada chip ATmega328 yang terdapat pada Arduino Nano telah dimasukkan program awal yang disebut dengan *bootloader*. Bootloader ini bertugas memudahkan pengguna dalam melakukan

suatu pemrograman yang lebih sederhana dengan menggunakan arduino *software* tanpa harus menggunakan tambahan *hardware* yang lain. Agar dapat digunakan, pengguna hanya cukup menghubungkan arduino dengan kabel USB ke PC. *Development board* Arduino Nano dapat diberi tenaga dengan daya yang diperoleh dari koneksi kabel Mini-B USB ataupun melalui power dari luar. Pada Arduino Software ini terdapat monitor serial yang memudahkan data untuk dikirim atau keluar dari arduino.

4. Motor Servo

Motor Servo adalah sebuah motor dengan arus DC yang memiliki sistem umpan balik tertutup dimana posisi rotor akan diinformasikan kembali kepada suatu rangkaian kontrol yang berada dalam motor servo.



Gambar 2.3 Motor Servo.

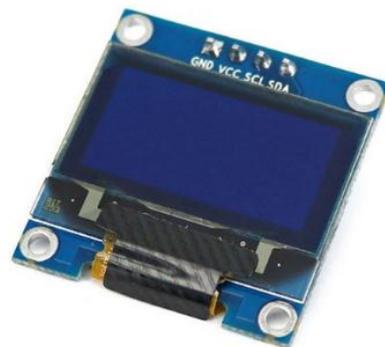
Sumber : Motor servo (www.edukasioelektronika.com)

Motor Servo adalah sebuah motor dengan arus DC yang memiliki siklus sistem umpan balik tertutup dimana posisinya akan diinformasikan kembali kepada rangkaian kontrol yang ada dalam servo. Motor terdiri dari motor arus DC, serangkaian gir, potensiometer, dan rangkaian kendali. Pemilihan motor

servo sangatlah berpengaruh besar pada pergerakan pada kaki robot dalam menentukan cara kerja pada saat melakukan pergerakan. Motor servo harus mempunyai daya dan torsi yang besar untuk mengatasi total berat dari robot. Sedangkan sudut motor servo diatur dengan berdasarkan berapa sudut yang akan dimasukkan ke setiap kaki robot, maka lebar pulsa baru akan didapatkan. Robot quadruped memakai 12 motor servo SG90.

5. OLED 0.96 inch

LCD OLED 0.96 memiliki jenis layar kecil yang menggunakan teknologi *Organic Light-Emitting Diode*, dengan ukuran 0.96 inci. OLED merupakan jenis teknologi layar di mana setiap *pixel* terdiri dari bahan organik yang bisa menghasilkan cahaya sendiri ketika diberikan tegangan pada komponennya.



Gambar 2.4 Oled 0.96 inch.

Sumber : Oled 0.96 inch (www.Randomnerdtutorial.com)

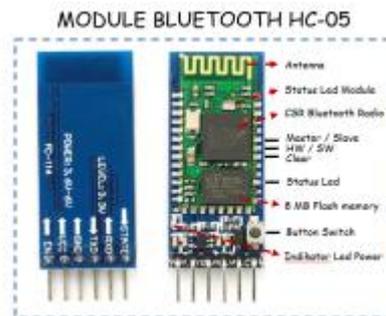
Beberapa informasi tambahan yang mungkin relevan:

1. 0.96 Inchi: Ukuran ini mengindikasikan diagonal pada layar, yang dalam kasus berikut adalah 0.96 inci. Layar OLED biasanya digunakan dalam proyek elektronika kecil atau perangkat yang dapat disesuaikan dengan penggunaan layar ini.
2. Warna dan Resolusi: OLED mempunyai keunggulan dalam kualitas warna dan juga kontras, karena dalam piksel menghasilkan cahaya sendiri. Tapi, resolusi layar (jumlah piksel) dapat bervariasi. Biasanya, layar OLED 0.96 inci memiliki resolusi yang cukup untuk menampilkan informasi dasar.
3. Aplikasi: Layar OLED 0.96 inci biasa digunakan dalam proyek elektronika, mikrokontroler, atau *prototipe*. Mereka dapat digunakan menampilkan teks, grafik, atau animasi tergantung pada kemampuan dan kompleksitas layar.
4. Antarmuka: Layar ini dilengkapi dengan antarmuka yang berbeda, seperti I2C, yang berfungsi untuk memudahkan pemakaian dengan mikrokontroler atau papan pengembangan lain.

Layar OLED 0.96 inci dapat menjadi pilihan yang baik untuk proyek kecil atau proyek DIY yang memerlukan tampilan kecil, dan hemat daya dengan kualitas tampilan yang baik.

6. HC05 bluetooth module

Bluetooth adalah peralatan industri untuk jaringan kawasan pribadi *Personal Area Networks* atau jaringan tanpa kabel.



Gambar 2.5 Modul Bluetooth HC-05.

Sumber : Modul HC-05 (www.anakteknik.co.id)

Bluetooth dapat menghubungkan dan dapat dipakai untuk melakukan pertukaran informasi atau data di antara peralatan seperti PDA, laptop, HP, dan lainnya. *Bluetooth* beroperasi dalam pita frekuensi 2,4 Ghz yang mampu menyediakan layanan komunikasi data dan suara secara real antara host *bluetooth* dengan jarak terbatas. Salah satu contoh modul *bluetooth* yang paling banyak digunakan adalah tipe HC-05. Modul Bluetooth HC-05 adalah sebuah modul Bluetooth SPP (Serial Port Protocol) yang mudah digunakan untuk komunikasi *serial wireless* yang merubah port serial ke *bluetooth*. HC-05 menggunakan modulasi *bluetooth* V2.0 + EDR (Enhanced Data Rate) 3 Mbps dengan memanfaatkan gelombang radio frekuensi 2,4 GHz. Modul ini dapat digunakan sebagai *slave* maupun master. HC-05 memiliki 2 mode konfigurasi, yaitu AT *mode* dan *Communication mode*. AT mode memiliki fungsi untuk melakukan

pengaturan dari HC-05. Sedangkan *Communication mode* memiliki fungsi melakukan komunikasi *bluetooth* dengan perangkat lain. Dalam penggunaannya, HC-05 dapat beroperasi tanpa menggunakan driver yang khusus. Untuk berkomunikasi antar *bluetooth*, minimal harus memenuhi dua kondisi berikut : 1. Komunikasi harus antara master dan slave. 2. Password peralatan harus benar (saat melakukan penghubungan). Jarak sinyal dari HC-05 adalah 30 meter, dengan kondisi tanpa penghalang. Sedangkan jangkauan jarak efektif modul ini saat terhubung dalam jarak 10 meter, dan jika melebihi dari jarak tersebut maka kualitas konektivitas akan semakin berkurang.

B. Kajian Hasil Penelitian Terdahulu

sebagai referensi untuk penelitian ini maka akan dicantumkan beberapa hasil penelitian terdahulu yang telah dilakukan.

1. Penelitian terdahulu yang telah dilakukan oleh Triyadi Zafarudding, Dkk. (Pada tahun 2018) tentang “Pengembangan Prototype Robot Quadruped Dengan Monitoring Control Android”. Pada penelitian ini mengacu pada pembuatan robot berkaki empat,yang memiliki desain kaki-kaki lebih adaptif,dan dengan adanya tambahan slot kamera untuk melakukan monitoring menurut point of view dari pengguna,robot ini juga menggunakan bluetooth sebagai koneksi antara robot dengan smartphone
2. Penelitian terdahulu yang dilakukan oleh Nanda Syafitri,Dkk. (Pada tahun 2018) tentang “Rancang Bangun Robot Quadruped Penghindar Halangan Berbasis Arduino”. Pada penelitian ini mengacu pada pembuatan robot empat

kaki yang dapat menghindari objek di depannya agar tidak terbentur, dengan tambahan sensor ultrasonik sebagai deteksi benda di depan robot.

BAB III

METODE PENELITIAN

A. Waktu Dan Tempat Penelitian.

Waktu dan tempat pelaksanaan penelitian akan dilakukan di Kabupaten Barru. Dengan menggunakan peralatan yang sudah disediakan dan memanfaatkan referensi yang ada. Penelitian ini dilakukan ± 2 (Dua) bulan dimulai pada awal bulan Juli 2024 sampai dengan akhir bulan Agustus 2024.

Tabel 3. 1 Jadwal Pelaksanaan Penelitian.

Uraian Kegiatan	Tahun 2024		
	Juli	Agustus	
Studi Literatur	■		
Perancangan Alat	■		
Pengadaan Alat Dan Komponen		■	
Perakitan Dan Realisasi Alat			
Penentuan Parameter Pengujian			■
Pengujian Untuk Kerja			■
Analisis Hasil Pengujian			■

B. Jenis Penelitian

Penelitian yang dilakukan dengan menggunakan metode eksperimen. Tujuan dari metode ini adalah untuk menguji rangkain alat apakah dapat berfungsi sesuai dengan yang diharapkan, dengan berpatokan pada referensi-referensi yang telah dikumpulkan.

C. Alat Dan Bahan

Berikut alat dan komponen yang digunakan pada perancangan ini adalah sebagai berikut :

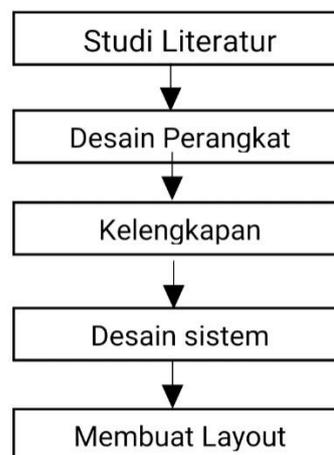
Tabel 3.2 Alat dan bahan yang dibutuhkan.

No	Uraian	Qty
1	Arduino Nano	1 buah
2	SG 90 Servo Motor	12 buah
3	HC05 Bluetooth Module	1 buah
4	Li-Ion Battery 3.7 KV	2 buah
5	OLED 0.96 inch	1 buah
6	DC Power Switch	1 buah

D. Rancangan Penelitian

1. Perancangan Perangkat Keras

Pada perancangan perangkat keras, dilakukan beberapa langkah agar dalam perancangan alat dapat berjalan efektif

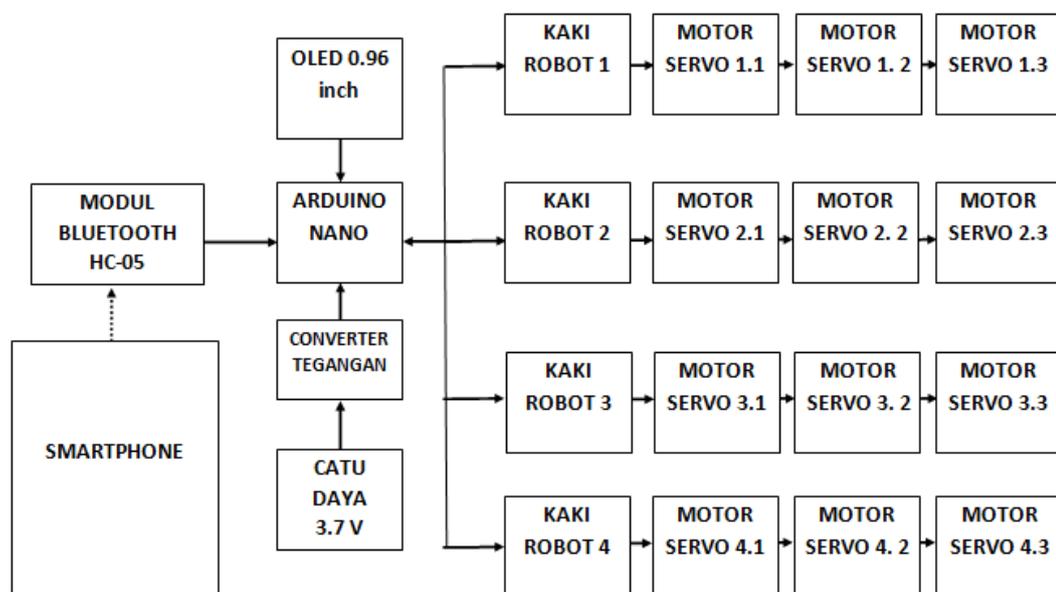


Gambar 3.1 Blok diagram perancangan perangkat keras

Untuk perancangan perangkat keras diawali dengan studi literatur dengan mengumpulkan teori-teori yang berhubungan dengan perancangan alat ini.

Tahap berikutnya yaitu melakukan desain perangkat keras sesuai dengan teori-teori yang didapatkan pada studi literature. Desain sistem pada perancangan alat ini menggunakan Arduino Nano sebagai Mikrokontroler dan penggunaan motor servo SG 90 untuk menggerakkan kaki robot serta penggunaan modul bluetooth HC-05 sebagai penghubung antara smartphone dan robot. dilanjutkan dengan pembuatan desain sistem serta skematik rangkaian.

2. Perancangan Blok Diagram



Gambar 3.2 Blok diagram secara keseluruhan.

E. Teknik Pengumpulan Data

Adapun teknik pengumpulan data yang digunakan dalam menyelesaikan rancangan alat ini yaitu :

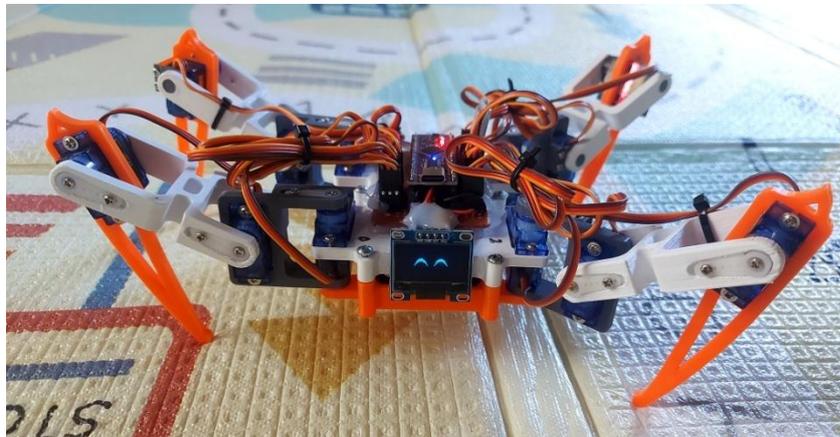
- 1) Studi Literatur, dalam metode ini penulis mengumpulkan sumber-sumber yang mendukung dalam perancangan alat ini, baik berupa artikel, buku-buku, jurnal, serta informasi-informasi yang berada di internet yang relevan sesuai dengan permasalahan perancangan alat.
- 2) Metode Konsultasi, pada metode ini penulis melakukan konsultasi pada pembimbing mengenai perihal perancangan dan penyusunan laporan. Dengan demikian adanya panduan akan memudahkan penulis untuk lebih baik dalam penyusunan perancangan alat. Pada metode ini penulis juga dapat melakukan konsultasi pada dosen-dosen selain dari dosen pembimbing.
- 3) Metode Pengujian, pada metode ini rangkaian alat akan diuji terlebih dahulu dengan menggunakan program simulasi rangkaian elektronika dan melakukan pengujian secara langsung di lapangan.

1. Arduino Nano, jenis mikrokontroler yang digunakan pada penelitian kali ini yaitu menggunakan arduino nano, yang berfungsi sebagai Mikrokontroler pada rancang bangun robot empat kaki 12 dof dengan kendali *bluetooth handphone*.
2. Motor Servo SG90, berfungsi sebagai sendi penggerak pada robot empat kaki yang dirancang, pada setiap pergerakan kaki robot, motor servo ini yang memiliki peran sangat penting dalam kesatuan robot.
3. Modul Bluetooth HC-05, untuk koneksi antara robot empat kaki dengan handphone, modul bluetooth yang digunakan memiliki spesifikasi dapat beroperasi pada jarak 10 meter, *input* tegangan 3.3 sampai dengan 5 volt dan memiliki total 6 pin.
4. Handphone, penggunaan handphone pada rancang bangun robot ini berfungsi sebagai pusat kontrol dari robot, dengan sebelumnya pada *handphone* terpasang aplikasi *Arduino Bluetooth Controller*
5. Arduino Bluetooth Controller, merupakan aplikasi gratis pada *play store* yang memiliki fungsi mengontrol robot pada menu dalam aplikasi, penggunaannya cukup buka aplikasi kemudian konesikan *handphone* ke *bluetooth* robot dan pilih menu terminal.
6. Dua buah baterai 3.7 Volt, berfungsi sebagai sumber daya pada robot, dimana kapasitas baterai yang digunakan adalah 3600 mAh.
7. Kerangka Robot, berfungsi sebagai bagian utama pada robot, karena pada kerangka robot inilah semua komponen pada robot empat kaki 12 dof

terpasang, kerangka robot ini memiliki material plastik yang cukup kokoh, sehingga pada tiap bagiannya mampu menopang pergerakan robot.

8. OLED 0,96 inchi, penggunaan OLED pada penelitian ini bertujuan untuk menampilkan kondisi mimik wajah pada robot ini, karena pada rancang bangun robot empat kaki 12 dof dibuat seperti robot serangga empat kaki yang memiliki wajah.

Berikut tampilan robot secara menyeluruh pasca dilakukan pemasangan tiap komponen menjadi robot berkaki empat.

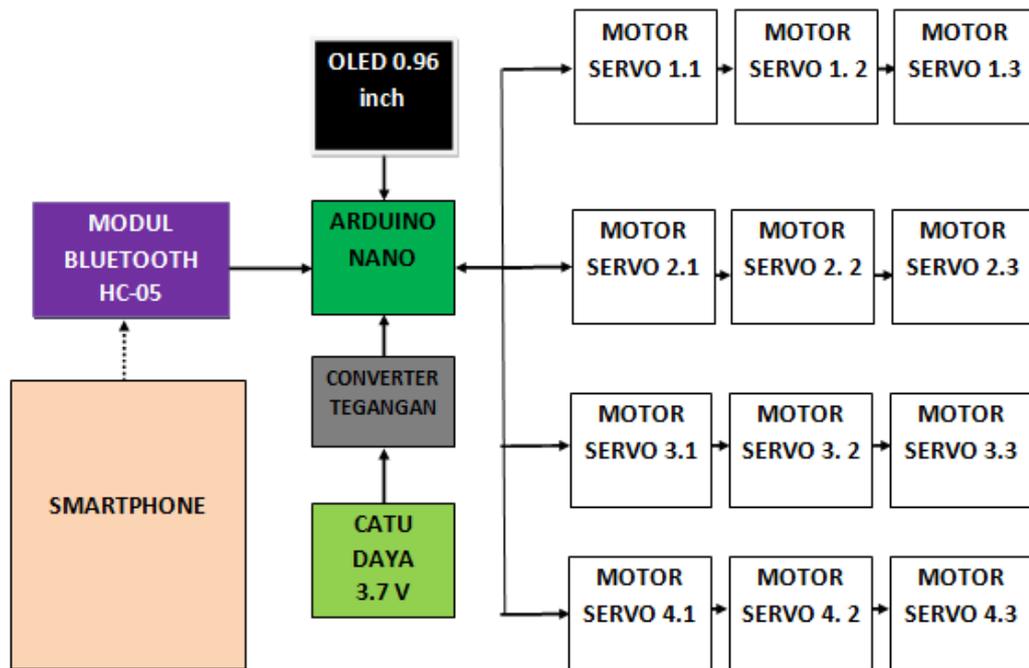


Gambar 4.2 Hasil Perakitan Robot.

B. Perancangan Sistem *Software*

Untuk perancangan perangkat lunak atau *Software* digunakan aplikasi Arduino IDE. Untuk setiap baris program yang telah dibuat akan di *Upload* masuk ke mikrokontroler arduino nano. Perancangan ini bertujuan untuk menggerakkan robot ke beberapa arah yang telah ditentukan sebelumnya dengan menggunakan aplikasi *arduino bluetooth controller* pada handphone, pada aplikasi ini nantinya akan di *input* kan beberapa perintah yang nantinya akan diproses dan

dilakukan oleh robot. Prinsip kerja sistem ini dapat dilihat pada blok diagram dibawah ini.



Gambar 4. 3 Blok diagram Perancangan *Software*

Pada gambar 4.1 kita dapat melihat rancangan pada robot empat kaki menggunakan 12 motor servo yang akan digunakan sebagai penggerak utama setiap pergerakan robot, ada beberapa pergerakan robot yang akan diujikan pada penelitian ini, yaitu robot maju, robot mundur, robot putar ke kanan 180 derajat, robot putar ke kiri 180 derajat, setiap pergerakan akan menggunakan perintah berupa huruf f untuk maju, b untuk mundur, r dilakukan dua kali perintah untuk putar robot ke kanan 180 derajat, dan juga perintah l dilakukan dua kali perintah untuk memutar robot ke kiri 180 derajat. Berikut tiap sketch pada perintah yang akan dimasukkan ke dalam arduino nano :

1. Sketch program untuk mimik wajah robot

Adapun sketch program pada OLED sebagai berikut :

```

void membuka() {
  display.clearDisplay();
  display.fillCircle (50, 15, 12, WHITE); //
MATA KANAN YANG BESAR
  display.fillCircle (82, 20, 7, WHITE); //
MATA KANAN YANG KECIL
  display.display();
}

void menutup() {
  display.clearDisplay();
  display.drawFastHLine(40, 15, 20, WHITE);
  display.drawFastHLine(72, 20, 15, WHITE);
  display.display();
}

void mengelilingi() {
  display.clearDisplay();
  display.fillCircle (42, 10, 20, WHITE);
//MATA KIRI YANG BESAR
  display.fillCircle (82, 10, 15, WHITE);
//MATA KIRI YANG KECIL
  display.fillRect (0, 0, 128, 15, BLACK);
//ALIS ATAS
  display.fillRect (0, 40, 128, 15, BLACK);
//ALIS BAGIAN BAWAH
  display.display();
}

void sedih() {
  display.clearDisplay();
  display.fillCircle (42, 10, 17, WHITE);
//MATA KIRI YANG BESAR
  display.fillCircle (82, 10, 17, WHITE);
//MATA KIRI YANG BESAR
  display.fillTriangle (0, 0, 0, 35, 78, 0,
BLACK); //ALIS ATAS
  display.fillTriangle (50, 0, 128, 35, 128,
0, BLACK); //ALIS ATAS
  display.display();
}

void senang() {
  display.clearDisplay();
  display.fillCircle (42, 25, 15, WHITE);
//MATA KIRI YANG BESAR
  display.fillCircle (82, 25, 15, WHITE);
//MATA KANAN KECIL
  display.fillCircle (42, 33, 20, BLACK);
//MATA KIRI YANG BESAR

```

```

        display.fillCircle (82, 33, 20, BLACK);
//MATA KANAN KECIL
        display.display();
    }

    void marah() {
        display.clearDisplay();
        display.fillCircle (42, 10, 18, WHITE);
//MATA KIRI YANG BESAR
        display.fillCircle (82, 10, 12, WHITE);
//MATA KANAN KECIL
        display.fillTriangle (0, 0, 54, 26, 118, 0,
BLACK); //ALIS ATAS
        display.display();
    }
    void marah1() {
        display.clearDisplay();
        display.fillCircle (42, 10, 18, WHITE);
//MATA KIRI YANG BESAR
        display.fillCircle (82, 10, 12, WHITE);
//MATA KANAN KECIL
        display.fillTriangle (0, 0, 65, 15, 120, 0,
BLACK); //ALIS ATAS
        display.display();
    }
}

```

2. Sketch program untuk robot bergerak maju

Adapun sketch program untuk robot bergerak maju sebagai berikut :

```

void step_forward(unsigned int step){
    move_speed = leg_move_speed;
    while (step-- > 0){
        if (site_now[2][1] == y_start){
            //leg 2&1 move
            set_site(2, x_default + x_offset, y_start, z_up);
            wait_all_reach();
            set_site(2, x_default + x_offset, y_start + 2 *
y_step, z_up);
            wait_all_reach();
            set_site(2, x_default + x_offset, y_start + 2 *
y_step, z_default);
            wait_all_reach();

            move_speed = body_move_speed;

            set_site(0, x_default + x_offset, y_start,
z_default);
            set_site(1, x_default + x_offset, y_start + 2 *
y_step, z_default);
            set_site(2, x_default - x_offset, y_start +
y_step, z_default);

```

```

        set_site(3, x_default - x_offset, y_start +
y_step, z_default);
        wait_all_reach();

        move_speed = leg_move_speed;

        set_site(1, x_default + x_offset, y_start + 2 *
y_step, z_up);
        wait_all_reach();
        set_site(1, x_default + x_offset, y_start, z_up);
        wait_all_reach();
        set_site(1, x_default + x_offset, y_start,
z_default);
        wait_all_reach();
    }
    else{
        //leg 0&3 move
        set_site(0, x_default + x_offset, y_start, z_up);
        wait_all_reach();
        set_site(0, x_default + x_offset, y_start + 2 *
y_step, z_up);
        wait_all_reach();
        set_site(0, x_default + x_offset, y_start + 2 *
y_step, z_default);
        wait_all_reach();

        move_speed = body_move_speed;

        set_site(0, x_default - x_offset, y_start +
y_step, z_default);
        set_site(1, x_default - x_offset, y_start +
y_step, z_default);
        set_site(2, x_default + x_offset, y_start,
z_default);
        set_site(3, x_default + x_offset, y_start + 2 *
y_step, z_default);
        wait_all_reach();

        move_speed = leg_move_speed;

        set_site(3, x_default + x_offset, y_start + 2 *
y_step, z_up);
        wait_all_reach();
        set_site(3, x_default + x_offset, y_start, z_up);
        wait_all_reach();
        set_site(3, x_default + x_offset, y_start,
z_default);
        wait_all_reach();
    }
}
}

```

3. Sketch program untuk robot mundur

Adapun sketch program untuk robot bergerak mundur sebagai berikut :

```

void step_back(unsigned int step){
  move_speed = leg_move_speed;
  while (step-- > 0){
    if (site_now[3][1] == y_start){
      //leg 3&0 move
      set_site(3, x_default + x_offset, y_start, z_up);
      wait_all_reach();
      set_site(3, x_default + x_offset, y_start + 2 *
y_step, z_up);
      wait_all_reach();
      set_site(3, x_default + x_offset, y_start + 2 *
y_step, z_default);
      wait_all_reach();

      move_speed = body_move_speed;

      set_site(0, x_default + x_offset, y_start + 2 *
y_step, z_default);
      set_site(1, x_default + x_offset, y_start,
z_default);
      set_site(2, x_default - x_offset, y_start +
y_step, z_default);
      set_site(3, x_default - x_offset, y_start +
y_step, z_default);
      wait_all_reach();

      move_speed = leg_move_speed;

      set_site(0, x_default + x_offset, y_start + 2 *
y_step, z_up);
      wait_all_reach();
      set_site(0, x_default + x_offset, y_start, z_up);
      wait_all_reach();
      set_site(0, x_default + x_offset, y_start,
z_default);
      wait_all_reach();
    }
    else{
      //leg 1&2 move
      set_site(1, x_default + x_offset, y_start, z_up);
      wait_all_reach();
      set_site(1, x_default + x_offset, y_start + 2 *
y_step, z_up);
      wait_all_reach();
      set_site(1, x_default + x_offset, y_start + 2 *
y_step, z_default);
      wait_all_reach();

      move_speed = body_move_speed;

      set_site(0, x_default - x_offset, y_start +
y_step, z_default);
      set_site(1, x_default - x_offset, y_start +
y_step, z_default);

```

```

        set_site(2, x_default + x_offset, y_start + 2 *
y_step, z_default);
        set_site(3, x_default + x_offset, y_start,
z_default);
        wait_all_reach();

        move_speed = leg_move_speed;

        set_site(2, x_default + x_offset, y_start + 2 *
y_step, z_up);
        wait_all_reach();
        set_site(2, x_default + x_offset, y_start, z_up);
        wait_all_reach();
        set_site(2, x_default + x_offset, y_start,
z_default);
        wait_all_reach();
    }
}
}

```

4. Sketch program untuk robot putar kiri

Adapun sketch program untuk robot putar ke kiri 90 derajat sebagai berikut :

```

void turn_left(unsigned int step){
    move_speed = spot_turn_speed;
    while (step-- > 0){
        if (site_now[3][1] == y_start){
            //leg 3&1 move
            set_site(3, x_default + x_offset, y_start, z_up);
            wait_all_reach();

            set_site(0, turn_x1 - x_offset, turn_y1,
z_default);
            set_site(1, turn_x0 - x_offset, turn_y0,
z_default);
            set_site(2, turn_x1 + x_offset, turn_y1,
z_default);
            set_site(3, turn_x0 + x_offset, turn_y0, z_up);
            wait_all_reach();

            set_site(3, turn_x0 + x_offset, turn_y0,
z_default);
            wait_all_reach();

            set_site(0, turn_x1 + x_offset, turn_y1,
z_default);
            set_site(1, turn_x0 + x_offset, turn_y0,
z_default);
            set_site(2, turn_x1 - x_offset, turn_y1,
z_default);
            set_site(3, turn_x0 - x_offset, turn_y0,
z_default);

```

```

        wait_all_reach();

        set_site(1, turn_x0 + x_offset, turn_y0, z_up);
        wait_all_reach();

        set_site(0, x_default + x_offset, y_start,
z_default);
        set_site(1, x_default + x_offset, y_start, z_up);
        set_site(2, x_default - x_offset, y_start +
y_step, z_default);
        set_site(3, x_default - x_offset, y_start +
y_step, z_default);
        wait_all_reach();

        set_site(1, x_default + x_offset, y_start,
z_default);
        wait_all_reach();
    }
    else {
        //leg 0&2 move
        set_site(0, x_default + x_offset, y_start, z_up);
        wait_all_reach();

        set_site(0, turn_x0 + x_offset, turn_y0, z_up);
        set_site(1, turn_x1 + x_offset, turn_y1,
z_default);
        set_site(2, turn_x0 - x_offset, turn_y0,
z_default);
        set_site(3, turn_x1 - x_offset, turn_y1,
z_default);
        wait_all_reach();

        set_site(0, turn_x0 + x_offset, turn_y0,
z_default);
        wait_all_reach();

        set_site(0, turn_x0 - x_offset, turn_y0,
z_default);
        set_site(1, turn_x1 - x_offset, turn_y1,
z_default);
        set_site(2, turn_x0 + x_offset, turn_y0,
z_default);
        set_site(3, turn_x1 + x_offset, turn_y1,
z_default);
        wait_all_reach();

        set_site(2, turn_x0 + x_offset, turn_y0, z_up);
        wait_all_reach();

        set_site(0, x_default - x_offset, y_start +
y_step, z_default);
        set_site(1, x_default - x_offset, y_start +
y_step, z_default);
        set_site(2, x_default + x_offset, y_start, z_up);
        set_site(3, x_default + x_offset, y_start,
z_default);

```

```

        wait_all_reach();

        set_site(2,  x_default  +  x_offset,  y_start,
z_default);
        wait_all_reach();
    }
}

```

5. Sketch program untuk robot putar ke kanan

Adapun sketch program untuk robot putar ke kanan 90 derajat sebagai berikut

```

/*
- spot turn to right
- blocking function
- parameter step steps wanted to turn
-----*/
-----*/
void turn_right(unsigned int step){
    move_speed = spot_turn_speed;
    while (step-- > 0){
        if (site_now[2][1] == y_start){
            //leg 2&0 move
            set_site(2, x_default + x_offset, y_start, z_up);
            wait_all_reach();

            set_site(0, turn_x0 - x_offset, turn_y0, z_default);
            set_site(1, turn_x1 - x_offset, turn_y1, z_default);
            set_site(2, turn_x0 + x_offset, turn_y0, z_up);
            set_site(3, turn_x1 + x_offset, turn_y1, z_default);
            wait_all_reach();

            set_site(2, turn_x0 + x_offset, turn_y0, z_default);
            wait_all_reach();

            set_site(0, turn_x0 + x_offset, turn_y0, z_default);
            set_site(1, turn_x1 + x_offset, turn_y1, z_default);
            set_site(2, turn_x0 - x_offset, turn_y0, z_default);
            set_site(3, turn_x1 - x_offset, turn_y1, z_default);
            wait_all_reach();

            set_site(0, turn_x0 + x_offset, turn_y0, z_up);
            wait_all_reach();

            set_site(0, x_default + x_offset, y_start, z_up);
            set_site(1, x_default + x_offset, y_start, z_default);
            set_site(2,  x_default  -  x_offset,  y_start  +  y_step,
z_default);
            set_site(3,  x_default  -  x_offset,  y_start  +  y_step,
z_default);
            wait_all_reach();

            set_site(0, x_default + x_offset, y_start, z_default);

```

```

    wait_all_reach();
}
else{
    //leg 1&3 move
    set_site(1, x_default + x_offset, y_start, z_up);
    wait_all_reach();

    set_site(0, turn_x1 + x_offset, turn_y1, z_default);
    set_site(1, turn_x0 + x_offset, turn_y0, z_up);
    set_site(2, turn_x1 - x_offset, turn_y1, z_default);
    set_site(3, turn_x0 - x_offset, turn_y0, z_default);
    wait_all_reach();

    set_site(1, turn_x0 + x_offset, turn_y0, z_default);
    wait_all_reach();

    set_site(0, turn_x1 - x_offset, turn_y1, z_default);
    set_site(1, turn_x0 - x_offset, turn_y0, z_default);
    set_site(2, turn_x1 + x_offset, turn_y1, z_default);
    set_site(3, turn_x0 + x_offset, turn_y0, z_default);
    wait_all_reach();

    set_site(3, turn_x0 + x_offset, turn_y0, z_up);
    wait_all_reach();

    set_site(0, x_default - x_offset, y_start + y_step,
z_default);
    set_site(1, x_default - x_offset, y_start + y_step,
z_default);
    set_site(2, x_default + x_offset, y_start, z_default);
    set_site(3, x_default + x_offset, y_start, z_up);
    wait_all_reach();

    set_site(3, x_default + x_offset, y_start, z_default);
    wait_all_reach();
}
}
}
}

```

6. Sketch program untuk bluetooth

Sketch program untuk *bluetooth* dapat dilihat sebagai berikut :

```

#include <Servo.h> //to define and control servos
#include <FlexiTimer2.h> //to set a timer to manage
all servos
#include <Wire.h>
#include <Adafruit_SSD1306.h>
#include <Adafruit_GFX.h>

#define OLED_ADDR 0x3C
Adafruit_SSD1306 display(-1);

/* Servos -----

```

```

-----*/
//define 12 servos for 4 legs
Servo servo[4][3];
//define servos' ports
const int servo_pin[4][3] = { {11,10,12}, {7, 8, 9},
{14,13,15}, {17,16,2} };
//=====          L0,K0,P0,    K1,L1,P1,
L2,K2,P2,    L3,K3,P3

/* Size of the robot -----
-----*/
const float length_a = 55;
const float length_b = 77.5;
const float length_c = 27.5;
const float length_side = 71;
const float z_absolute = -28;

/* Constants for movement -----
-----*/
const float z_default = -50, z_up = -30, z_boot =
z_absolute;
const float x_default = 62, x_offset = 0;
const float y_start = 0, y_step = 40;
const float y_default = x_default;

/* variables for movement -----
-----*/
volatile float site_now[4][3];          //real-time
coordinates of the end of each leg
volatile float site_expect[4][3];      //expected
coordinates of the end of each leg
float temp_speed[4][3]; //each axis' speed, needs to
be recalculated before each movement
float move_speed;          //movement speed
float speed_multiple = 1; //movement speed multiple

const float spot_turn_speed = 4;
const float leg_move_speed = 8;
const float body_move_speed = 3;
const float stand_seat_speed = 1;
volatile int rest_counter;          //+1/0.02s, for
automatic rest

//functions' parameter
const float KEEP = 255;
const float pi = 3.1415926; //define PI for
calculation

/* Constants for turn -----
-----*/
//temp length
const float temp_a = sqrt(pow(2 * x_default +
length_side, 2) + pow(y_step, 2));
const float temp_b = 2 * (y_start + y_step) +
length_side;
const float temp_c = sqrt(pow(2 * x_default +

```

```

length_side, 2) + pow(2 * y_start + y_step +
length_side, 2));
const float temp_alpha = acos((pow(temp_a, 2) +
pow(temp_b, 2) - pow(temp_c, 2)) / 2 / temp_a /
temp_b);
//site for turn
const float turn_x1 = (temp_a - length_side) / 2;
const float turn_y1 = y_start + y_step / 2;
const float turn_x0 = turn_x1 - temp_b *
cos(temp_alpha);
const float turn_y0 = temp_b * sin(temp_alpha) -
turn_y1 - length_side;
/* -----
-----*/
char dataBTH = NULL;

void setup() {
  Serial.begin(9600);
  display.begin(SSD1306_SWITCHCAPVCC, OLED_ADDR);
  display.clearDisplay();
  display.display();
  delay(1000);

  //initialize default parameter
  set_site(0, x_default - x_offset, y_start + y_step,
z_boot);
  set_site(1, x_default - x_offset, y_start + y_step,
z_boot);
  set_site(2, x_default + x_offset, y_start, z_boot);
  set_site(3, x_default + x_offset, y_start, z_boot);
  for (int i = 0; i < 4; i++){
    for (int j = 0; j < 3; j++){
      site_now[i][j] = site_expect[i][j];
    }
  }
  //start servo service
  FlexiTimer2::set(20, servo_service);
  FlexiTimer2::start();

  //initialize servos
  servo_attach();
  stand();
  happy(); delay (random (500, 1000)); //bahagia
  cierra(); delay (150); //terpejam
  enfado(); delay (random (1000, 3000)); //marah
  cierra(); delay (150);
  entorna(); delay (random (100, 1000)); //melirik
  cierra(); delay (150);
  enfado1(); delay (random (100, 1000));
  cierra(); delay (150);
  triste(); delay (random (100, 1000)); //sedih
  cierra(); delay (150);
  abre(); delay (random (100, 1000)); //melek
  cierra(); delay (150);
  happy(); delay (random (100, 1000));
}

```

```

void servo_attach(void){
  for (int i = 0; i < 4; i++){
    for (int j = 0; j < 3; j++){
      servo[i][j].attach(servo_pin[i][j]);
      delay(100);
    }
  }
}

void servo_detach(void){
  for (int i = 0; i < 4; i++){
    for (int j = 0; j < 3; j++){
      servo[i][j].detach();
      delay(100);
    }
  }
}

// loop function
/*=====
=====*/
void loop() {
  if(Serial.available(>0) {
    char dataBTH = Serial.read();
    if(dataBTH=='f') { step_forward(20);}
    if(dataBTH=='b') { step_back(20); }
    if(dataBTH=='l') { turn_left(6); }
    if(dataBTH=='r') { turn_right(6); }
    if(dataBTH=='d') { body_dance(10);}
    if(dataBTH=='s') { sit();}
    dataBTH=NULL;
  }
}

void servo_service(void){
  sei();
  static float alpha, beta, gamma;
  for (int i = 0; i < 4; i++){
    for (int j = 0; j < 3; j++){
      if (abs(site_now[i][j] - site_expect[i][j]) >=
abs(temp_speed[i][j]))
        site_now[i][j] += temp_speed[i][j];
      else site_now[i][j] = site_expect[i][j];
    }

    cartesian_to_polar(alpha, beta, gamma,
site_now[i][0], site_now[i][1], site_now[i][2]);
    polar_to_servo(i, alpha, beta, gamma);
  }

  rest_counter++;
}

void set_site(int leg, float x, float y, float z){

```

```

float length_x = 0, length_y = 0, length_z = 0;
if (x != KEEP) length_x = x - site_now[leg][0];
if (y != KEEP) length_y = y - site_now[leg][1];
if (z != KEEP) length_z = z - site_now[leg][2];

float length = sqrt(pow(length_x, 2) +
pow(length_y, 2) + pow(length_z, 2));

temp_speed[leg][0] = length_x / length * move_speed
* speed_multiple;
temp_speed[leg][1] = length_y / length * move_speed
* speed_multiple;
temp_speed[leg][2] = length_z / length * move_speed
* speed_multiple;

if (x != KEEP) site_expect[leg][0] = x;
if (y != KEEP) site_expect[leg][1] = y;
if (z != KEEP) site_expect[leg][2] = z;
}

void wait_reach(int leg){
while (1)
if (site_now[leg][0] == site_expect[leg][0])
if (site_now[leg][1] == site_expect[leg][1])
if (site_now[leg][2] == site_expect[leg][2])
break;
}

void wait_all_reach(){ for (int i = 0; i < 4; i++)
wait_reach(i); }

void cartesian_to_polar(volatile float &alpha,
volatile float &beta, volatile float &gamma,
volatile float x, volatile
float y, volatile float z) {
//calculate w-z degree
float v, w;
w = (x >= 0 ? 1 : -1) * (sqrt(pow(x, 2) + pow(y,
2)));
v = w - length_c;
alpha = atan2(z, v) + acos((pow(length_a, 2) -
pow(length_b, 2) + pow(v, 2) +
pow(z, 2)) / 2 / length_a / sqrt(pow(v, 2)
+ pow(z, 2)));
beta = acos((pow(length_a, 2) + pow(length_b, 2) -
pow(v, 2) -
pow(z, 2)) / 2 / length_a / length_b);
//calculate x-y-z degree
gamma = (w >= 0) ? atan2(y, x) : atan2(-y, -x);
//trans degree pi->180
alpha = alpha / pi * 180;
beta = beta / pi * 180;
gamma = gamma / pi * 180;
}

void polar_to_servo(int leg, float alpha, float beta,

```

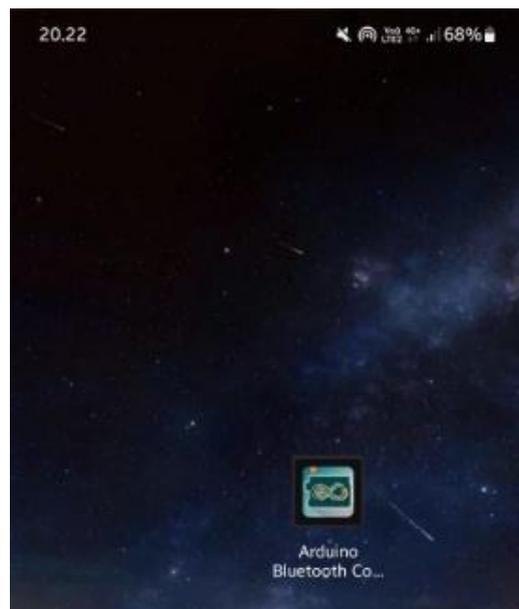
```

float gamma){
  if (leg == 0){ alpha = 90-alpha; beta = beta; gamma
  += 90; }
  else if (leg==1){ alpha += 90; beta = 180-beta;
  gamma = 90-gamma; }
  else if (leg==2){ alpha += 90; beta = 180-beta;
  gamma = 90-gamma; }
  else if (leg==3){ alpha = 90-alpha; beta = beta;
  gamma += 90; }

  servo[leg][0].write(alpha);
  servo[leg][1].write(beta);
  servo[leg][2].write(gamma);
}

```

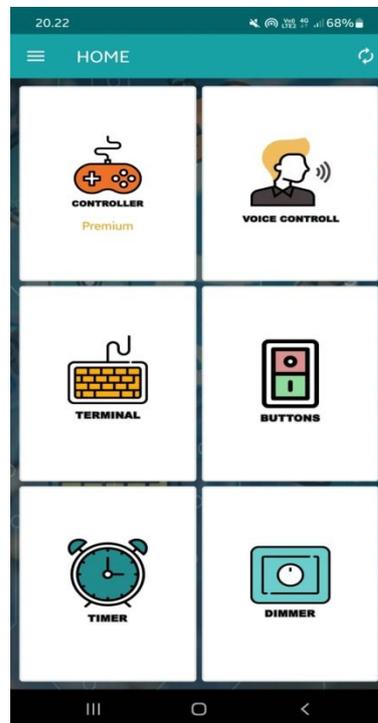
Setiap sketch yang ditulis, diupload ke dalam mikrokontroler untuk kemudian diproses setiap sketch yang diinstruksikan ke dalamnya, dalam melakukan kendali pada robot, digunakan juga aplikasi *Arduino Bluetooth Controller* sebagai alat kendali pada robot, yang menginstruksikan berbagai perintah berupa huruf, berikut tampilan dari aplikasi yang digunakan :



Gambar 4.4 Tampilan Aplikasi Kendali Robot

Untuk mengoperasikan aplikasi *arduino bluetooth controller* dimulai dengan menghubungkan *bluetooth handphone* dengan *bluetooth robot*, jika

bluetooth sudah terhubung, akan muncul tampilan beranda seperti gambar berikut:



Gambar 4.5 Tampilan Dashboard Aplikasi

Pada tampilan *dashboard* aplikasi, terdapat beberapa menu yang muncul pada tampilan awal aplikasi, namun dalam melakukan kontrol robot, opsi yang digunakan adalah menu terminal.



Gambar 4.6 Tampilan Menu Terminal

Pada tampilan menu terminal, tersedia slot untuk melakukan *command* terhadap robot, dalam slot tersebut bisa diinputkan beberapa perintah yang sudah ditentukan sebelumnya dalam mengendalikan robot, beberapa perintah tersebut adalah perintah *f* untuk robot bergerak maju ke depan, perintah *b* untuk robot bergerak mundur, perintah *r* untuk robot berputar ke kanan 90 derajat, , dan yang terakhir adalah perintah *l* , sama dengan perintah *r*, perintah *l* ini menginstruksikan robot untuk berputar ke kiri 90 derajat, agar dapat berputar 180 derajat, perintal *l* harus di *inputkan* 2 kali.

C. Pengujian Pergerakan Robot

Ada 6 (enam) tahap pengujian pada penelitian rancang bangun robot empat kaki 12 dof dengan kendali *bluetooth handphone*, yang pertama adalah pengujian pergerakan robot maju, pada pengujian ini robot dipantau saat bergerak maju, dari urutan kaki robot yang melangkah dan waktu yang dibutuhkan untuk bergerak maju ke depan, pengujian yang kedua adalah pengujian robot bergerak mundur, pemantauan yang dilakukan sama dengan pergerakan robot sebelumnya, mulai dari pergerakan kaki robot dan waktu yang dibutuhkan robot untuk bergerak mundur, yang ketiga adalah pergerakan robot menghadap ke kanan, yang ke empat adalah pergerakan robot menghadap ke kiri, pergerakan robot berbalik 180 derajat, dan yang terakhir adalah pengujian robot bergerak maju dengan tiga variasi kecepatan, berikut rincian dari pengujian robot empat kaki 12 dof dengan kendali *bluetooth handphone* :

1. Pengujian pergerakan robot maju

Pada pengujian robot maju, yang dilakukan adalah dengan memberikan perintah kepada robot melalui aplikasi di *handphone* dengan perintah f, dengan perintah tersebut robot akan bergerak maju sesuai dengan program yang telah di *input* kan ke dalam mikrokontroler.

Tabel 4.1 Data Hasil Pengujian Robot Bergerak Maju.

Tampilan Perintah Pada Aplikasi	Urutan Pergerakan Kaki Robot	Jarak Yang Ditempuh robot	Waktu Yang Dibutuhkan Untuk Bergerak
	Kaki kanan depan,kaki kiri belakang ,kaki kiri depan,kaki kanan belakang berulang sepuluh kali	0.04 meter	a. 18.92
			Detik
			b. 18.84
			Detik
			c. 18.90
			Detik

Dari data yang ada pada tabel 4.1 didapatkan hasil mengenai pergerakan robot maju yaitu pada saat robot diberikan perintah f , robot akan bergerak maju diawali kaki kanan depan,kemudian kaki kiri belakang,kaki kiri depan,kaki kanan belakang,dan berulang sebanyak sepuluh kali sejauh 0.04 meter,pengujian dilakukan sebanyak tiga kali,dan tiap pergerakan membutuhkan waktu yang berbeda-beda,pada pengujian pertama didapatkan waktu 18.92 detik,pada pengujian kedua didapatkan waktu 18.84 detik,dan pada percobaan terakhir didapatkan waktu 18.90 detik.

Dari data tersebut dapat disimpulkan bahwa pada pergerakan robot maju,dengan jarak yang sama,diperlukan waktu bergerak yang berbeda-beda.

2. Pengujian robot bergerak mundur

Pada pengujian robot mundur,yang dilakukan adalah dengan memberikan perintah kepada robot melalui aplikasi di *handphone* dengan perintah b ,dengan perintah tersebut robot akan bergerak mundur sesuai dengan program yang telah di *input* kan ke dalam mikrokontroler.

Tabel 4.2 Data Hasil Pengujian Robot Bergerak Mundur.

Tampilan Perintah Pada Aplikasi	Urutan Pergerakan Kaki Robot	Jarak Yang Ditempuh robot	Waktu Yang Dibutuhkan Untuk Bergerak
	Kaki kanan belakang ,kaki kiri depan ,kaki kiri belakang ,kaki kanan depan berulang sepuluh kali	0.04 meter	a. 18.71 Detik b. 18.79 Detik c. 18.86 Detik

Dari data yang ada pada tabel 4.2 didapatkan hasil mengenai pergerakan robot maju yaitu pada saat robot diberikan perintah b , robot akan bergerak mundur diawali kaki kanan belakang,kemudian kaki kiri depan,kaki kiri belakang,kaki kanan depan,dan berulang sebanyak sepuluh kali sejauh 0.04 meter ,pengujian dilakukan sebanyak tiga kali,dan tiap pergerakan membutuhkan waktu yang berbeda-beda,pada pengujian pertama didapatkan waktu 18.71 detik,pada pengujian kedua didapatkan waktu 18.79 detik,dan pada percobaan terakhir didapatkan waktu 18.86 detik.

Dari data tersebut dapat disimpulkan bahwa pada pergerakan robot mundur,dengan jarak yang sama,diperlukan waktu bergerak yang berbeda-beda.

3. Pengujian robot menghadap kanan

Pada pengujian robot menghadap kanan,yang dilakukan adalah dengan memberikan perintah kepada robot melalui aplikasi di *handphone* dengan perintah r ,dengan perintah tersebut robot akan bergerak 90 derajat ke kanan sesuai dengan program yang telah di *input* kan ke dalam mikrokontroler

Tabel 4.3 Pengujian robot menghadap kanan.

Tampilan Perintah Pada Aplikasi	Urutan Pergerakan Kaki Robot	Waktu Yang Dibutuhkan Untuk Bergerak
	<p>Kaki kanan belakang,kaki kiri belakang,kaki kiri depan,kaki kanan belakang berulang tiga kali</p>	<p>a. 06.61 Detik b. 06.58 Detik c. 06.67 Detik</p>

Dari data yang ada pada tabel 4.3 didapatkan hasil mengenai pergerakan robot menghadap kanan, yaitu pada saat robot diberikan perintah r , robot akan bergerak 90 derajat ke arah kanan diawali kaki kanan belakang,kemudian kaki kiri belakang,kaki kiri depan,kaki kanan belakang,dan berulang sebanyak tiga kali,pengujian dilakukan sebanyak tiga kali,tiap pergerakan membutuhkan waktu yang berbeda-beda,pada pengujian pertama didapatkan waktu 06.61 detik,pada pengujian kedua didapatkan waktu 06.58 detik,dan pada percobaan terakhir didapatkan waktu 06.67 detik.

Dari data tersebut dapat disimpulkan bahwa pada pergerakan robot menghadap kanan,dengan sudut pergerakan yang sama,diperlukan waktu bergerak yang berbeda-beda.

4. Pengujian robot menghadap kiri

Pada pengujian robot menghadap kiri, yang dilakukan adalah dengan memberikan perintah kepada robot melalui aplikasi di *handphone* dengan perintah `l`, dengan perintah tersebut robot akan bergerak 90 derajat ke kiri sesuai dengan program yang telah di *input* ke dalam mikrokontroler.

Tabel 4.4 Pengujian robot menghadap kiri.

Tampilan Perintah Pada Aplikasi	Urutan Pergerakan Kaki Robot	Waktu Yang Dibutuhkan Untuk Bergerak
	<p>Kaki kanan depan, kaki kiri depan, kaki kiri belakang, kaki kanan belakang berulang tiga kali</p>	<p>a. 06.68 Detik b. 06.61 Detik c. 06.65 Detik</p>

Dari data yang ada pada tabel 4.4 didapatkan hasil mengenai pergerakan robot menghadap kiri, yaitu pada saat robot diberikan perintah `l`, robot akan bergerak 90 derajat ke arah kiri diawali kaki kanan depan, kemudian kaki kiri depan, kaki kiri belakang, kaki kanan depan, dan berulang sebanyak tiga kali, pengujian dilakukan sebanyak tiga kali, tiap pergerakan membutuhkan waktu yang berbeda-beda, pada pengujian pertama didapatkan waktu 06.68 detik, pada pengujian kedua didapatkan waktu 06.61 detik, dan pada percobaan terakhir didapatkan waktu 06.65 detik.

Dari data tersebut dapat disimpulkan bahwa pada pergerakan robot menghadap kiri, dengan sudut pergerakan yang sama, diperlukan waktu bergerak yang berbeda-beda.

5. Pengujian robot berbalik arah 180 derajat

Pada pengujian robot berbalik arah 180 derajat, yang dilakukan adalah dengan memberikan perintah kepada robot melalui aplikasi di *handphone* dengan perintah r sebanyak dua kali, dengan perintah tersebut robot akan bergerak 180 derajat ke kanan sesuai dengan program yang telah di *input* ke dalam mikrokontroler.

Tabel 4.5 Pengujian robot berbalik arah 180 derajat.

Tampilan Perintah Pada Aplikasi	Urutan Pergerakan Kaki Robot	Waktu Yang Dibutuhkan Untuk Bergerak
	<p>Kaki kanan belakang, kaki kiri belakang, kaki kiri depan, kaki kanan belakang berulang enam kali</p>	<p>a. 12.67 Detik b. 12.77 Detik c. 12.59 Detik</p>

Dari data yang ada pada tabel 4.5 didapatkan hasil mengenai pergerakan robot berputar 180 derajat ke kanan, yaitu pada saat robot diberikan perintah r dua kali, robot akan bergerak 180 derajat ke arah kanan, diawali kaki kanan belakang, kemudian kaki kiri belakang, kaki kiri depan, kaki kanan belakang, dan berulang sebanyak tiga kali, pengujian dilakukan sebanyak tiga kali, tiap pergerakan membutuhkan waktu yang berbeda-beda, pada pengujian pertama didapatkan waktu 12.67 detik, pada pengujian kedua didapatkan waktu 12.77 detik, dan pada percobaan terakhir didapatkan waktu 12.59 detik.

Dari data tersebut dapat disimpulkan bahwa pada pergerakan robot berbalik arah 180 derajat, dengan pergerakan yang sama, diperlukan waktu bergerak yang berbeda-beda.

6. Pengujian robot bergerak maju dengan 3 (tiga) variasi kecepatan

Pada pengujian robot bergerak maju dengan 3 (tiga) variasi kecepatan, dilakukan sedikit perubahan pada sketch program arduino.

```
float move_speed;           //movement speed
float speed_multiple = 1; //movement speed multiple
```

Bagian sketch yang dirubah ada pada *float speed_multiple = 1*, pada bagian tersebut akan dirubah pada nilai 1, untuk 3 variasi kecepatan robot nilai tersebut akan dirubah menjadi 2 dan 3, kemudian akan diambil data kecepatan pada perubahan nilai tersebut.

a. Perubahan kecepatan menjadi bernilai 2 (dua)

Yang dilakukan pada sketch program adalah mengganti nilai kecepatan menjadi 2

```
float move_speed;           //movement speed
float speed_multiple = 2; //movement speed multiple
```

Dari data perubahan tersebut, jika dimasukkan ke dalam tabel akan didapatkan data sebagai berikut.

Tabel 4.6 Tabel Pergerakan Robot Maju Dengan Kecepatan 2

Jarak	Waktu Tempuh	Kecepatan Pergerakan
0.04meter	a. 16.82 Detik	a. 0.00237 m/detik
	b. 16.80 Detik	b. 0.00238 m/detik
	c. 16.89 Detik	c. 0.00236 m/detik

Dari data tabel 4.6 didapatkan hasil, dengan kecepatan robot yang dinaikkan menjadi 2 (dua), pada tiga percobaan yang dilakukan dengan jarak tempuh yang sama, pada percobaan pertama pada jarak 0,04 meter, waktu tempuh 16.82 detik, kecepatan pergerakan robot adalah 0.00237 m/detik, pada percobaan kedua waktu tempuh 16.80 detik, kecepatan 0.00238 m/detik, dan percobaan ketiga waktu tempuh 16.89 detik, kecepatan 0.00236 m/detik, pada data kecepatan diambil dari rumus kecepatan dan didapatkan data yang tertera sesuai dengan tabel 4.6.

b. Perubahan kecepatan menjadi 3 (tiga)

Yang dilakukan pada sketch program adalah mengganti nilai kecepatan menjadi 3.

```
float move_speed;           //movement speed
float speed_multiple = 3; //movement speed multiple
```

Dari data perubahan tersebut, jika dimasukkan ke dalam tabel akan didapatkan data sebagai berikut :

Tabel 4.7 Tabel Pergerakan Robot Maju Dengan Kecepatan 3

Jarak	Waktu Tempuh	Kecepatan Pergerakan
0.04 meter	a. 14.79 Detik	a. 0.00270 m/detik
	b. 14.92 Detik	b. 0.00268 m/detik
	c. 14.83 Detik	c. 0.00269 m/detik

Dari data tabel 4.7 didapatkan hasil, dengan kecepatan robot yang dinaikkan menjadi 3 (tiga), pada tiga kali percobaan yang dilakukan, percobaan pertama dengan jarak 0.04 meter, waktu tempuh 14.79 detik, kecepatan 0.00270 m/detik, pada percobaan kedua waktu tempuh 14.92 detik, kecepatan 0.00268 m/detik, dan percobaan ketiga waktu tempuh 14.83 detik, kecepatan 0.00269 m/detik, pada data kecepatan diambil dari rumus kecepatan sesuai dengan yang tertera di tabel 4.7.

c. Perbandingan Kecepatan Bergerak Maju 1(satu) ,2 (dua) ,dan 3 (tiga)

Pada perbandingan kecepatan pergerakan robot maju, dimasukkan ketiga data kecepatan dari tiap percobaan pergerakan, berikut tabel yang menyajikan data , mengenai perbandingan variasi kecepatan robot bergerak maju.

Tabel 4.8 Tabel Perbandingan Kecepatan Robot Maju

Input Kecepatan Pada Sketch	Jarak	Waktu Tempuh	Kecepatan Pergerakan
1	0.04 m	18.92 Detik	0.00211 m/detik
		18.84 Detik	0.00212 m/detik
		18.90 Detik	0.00211 m/detik
2	0.04 m	16.82 Detik	0.00237 m/detik
		16.80 Detik	0.00238 m/detik
		16.89 Detik	0.00236 m/detik
3	0.04 m	14.79 Detik	0.00270 m/detik
		14.92 Detik	0.00268 m/detik
		14.83 Detik	0.00269 m/detik

Dari data tabel 4.8 ,pada percobaan 3 (tiga) variasi kecepatan robot yang telah dilakukan,didapatkan hasil semakin tinggi nilai yang dimasukkan pada sketch program,pergerakan robot akan semakin cepat,waktu tempuh yang dilakukan robot akan semakin singkat,dan kecepatan bergerak robot akan meningkat,pada data tersebut dapat diambil kecepatan rata-rata pada setiap percobaan variasi kecepatan,dari kecepatan 1 (satu) rata-rata kecepatan pergerakan robot adalah 0.00211 m/detik, dari kecepatan 2 (dua) rata-rata kecepatan pergerakan robot adalah 0.00237 m/detik,dan dari kecepatan 3 (tiga) rata-rata kecepatan pergerakan robot adalah 0.00269 m/detik.

BAB V

KESIMPULAN DAN SARAN

A. Kesimpulan

Setelah dilakukan berbagai macam pengujian pada robot empat kaki 12 dof, terdapat beberapa kesimpulan yang dapat diambil, diantaranya yaitu :

1. Pergerakan robot sesuai dengan perintah yang dimasukkan sebelumnya ke dalam program arduino nano, namun dari enam percobaan yang dilakukan, waktu yang tercatat berbeda pada tiap percobaanya.
2. Pergerakan robot yang dapat dilakukan adalah bergerak maju, bergerak mundur, gerakan menghadap kanan, dan gerakan menghadap kiri, gerakan robot berbalik arah 180 derajat, dan 3 variasi kecepatan pergerakan robot, semua gerakan dapat dilakukan dengan baik.

B. Saran

1. Disarankan menggunakan material kerangka robot dan tipe motor servo yang lebih kokoh agar pergerakan robot lebih stabil.
2. Pada pengembangan selanjutnya dapat difokuskan pada variasi gerakan robot.
3. Pengoperasian robot sebaiknya dilakukan pada permukaan yang datar agar pergerakan robot lebih maksimal.

DAFTAR PUSTAKA

- W. Ansori, S. Hardhinata, and T. A. Zuraiyah, “Sistem Kontrol Robot Beroda Berbasis Mikrokontroler Atmega128 Menggunakan Speech Recognition Dengan Komunikasi Bluetooth Sebagai Transfer Data,” *J. Online Mhs. Bid. Ilmu Komputer/Informatika*, vol. 2, no. 2, 2014.
- S. H. Tirtamiharja and others, “Perancangan automatic tower crane berbasis mikrokontroler AT89S51,” *SKRIPSI-2012*, 2016.
- A. K. Muhammad, “Aplikasi Accelerometer Pada Penstabil Monopod Menggunakan Motor Servo,” Politeknik Negeri SRIWIJAYA, 2016.
- N. S. Ahlina, “Sistem Kendali Motor Servo Sebagai Penggerak Kamera Pada Robot Boat Pengintai Menggunakan Xbee Series 1 Berbasis Arduino,” Politeknik Negeri Sriwijaya, 2015.
- A. Darmawan and H. Adrianto, *ARDUINO belajar cepat dan pemrograman*. Bandung:Informatika, 2016.
- Rouse, M., 2016. *Robot*. [Online].
- Setiawan, L. & Utomo, D., 2011. *Inverse Kinematic dengan Real Time Operating System pada robot Quadruped*, Salatiga: Universitas Kristen Satya Wacana.
- Wicaksono, H. et al., 2008. *Perancangan Sistem Navigasi Otonom pada Behavior Based Hexapod Robot*, Surabaya: Petra.
- Santoso, H. (2018). *Monster Arduino 3*. ElangSakti. Wicaksana, M. F. (2017). *Aplikasi Arduino Dan Sensor*. Bi Obsess 2.
- H. D. Siswaja, “Prinsip Kerja dan Klasifikasi Robot,” *Media Inform.*, vol. 7, no. 3, pp. 148–157, 2008.
- P. Prasetyawan, Y. Ferdianto, S. Adan, and F. Trisnawati, “Pengendali Lengan Robot Dengan Mikrokontroler Arduino Berbasis Smartphone,” vol. 7, no. 2, pp. 104–109, 2018.
- R. Syam, “Kinematika dan Dinamika Robot Lengan,” in *Seri Buku Ajar Robotika*, 2015, p. 76
- H. Chandra, “Perancangan Lengan Robot Dengan Micro Servo Berbasis Arduino Uno,” no. February. p. 5, 2017.

- A. Triwiyatno, "Konsep Umum Sistem Kontrol," *Buku Ajar Siste Kontrol Analog*. pp. 1–11, 2011.
- M. Didi, E. D. Marindani, and A. Elbani, "Rancang Bangun Pengendalian Robot Lengan 4 DOF dengan GUI (Graphical User Interface) Berbasis Arduino Uno I-2." pp. 1–11, 2006.
- Hasyflatsy, "Potentiometer Servo Controller Glove," 2018. [Online]. Available: <https://www.thingiverse.com/thing:2782111>. [Accessed: 07-Aug-2019].