

RANCANG BANGUN ROBOT LOGISTIK UNTUK PENGIRIMAN MINUMAN

Mada Jimmy Fonda Arifianto

Mekatronika, Politeknik Manufaktur Astra, Jl. Gaya Motor Raya 8, Sunter II, Jakarta 14330, Indonesia

E-mail : mada.jimmy@polman.astra.ac.id

Abstrak - Penelitian ini mencakup perancangan dan pembuatan robot yang secara khusus digunakan untuk mengirim gelas berisi minuman dan harus mampu melewati tangga tanpa menumpahkan minuman. Robot juga tidak boleh rentan terhadap kondisi lingkungan di ruang publik seperti kebisingan suara dan fluktuasi cahaya. Setelah melewati tahap perancangan, pembuatan dan ujicoba, hasil rancang bangun robot logistik ini terdiri dari tiga kerangka yaitu depan, tengah dan belakang. Kerangka depan dan belakang dapat bergerak translasi naik dan turun relatif terhadap kerangka tengah untuk mengatur sokongan robot saat naik dan turun tangga. Sambungan antar kerangka ini menggunakan rak dan pinion.

Bagian rancangan elektronika robot terdiri dari mikrokontroler Arduino sebagai pengendali utama robot. Komponen lain adalah penggerak motor DC, motor DC dengan transmisi roda gigi planet untuk roda, motor DC dengan transmisi roda cacing untuk sambungan kerangka. Sensor-sensor diterapkan untuk mendeteksi dinding, lantai dan posisi kerangka pada saat di tengah, atas atau bawah.

Sedangkan pemrograman robot menggunakan permodelan „mesin keadaan-hingga“ (*finite-state machine*). Model ini biasa diterapkan pada sistem yang berjalan secara terus menerus untuk menjamin perangkat tetap berjalan secara otomatis.

Kata Kunci: rancang bangun robot, mesin keadaan (*state machine*), pemrograman mikrokontroler

I. PENDAHULUAN

Penelitian ini dilatarbelakangi oleh kegiatan kompetisi robot yang diadakan oleh sebuah perusahaan di Belanda yang diselenggarakan di kota Eindhoven pada bulan Juni 2015 dan diikuti oleh tim yang terdiri dari beberapa orang dari lembaga pendidikan. Secara garis besar, robot peserta harus membawa tiga gelas minuman (beban total sekitar 1,5 kg) sambil berjalan, melewati rintangan berupa menaiki tiga anak tangga dan menuruni tiga anak tangga, selanjutnya berjalan menuju lokasi tertentu dan diakhiri dengan membunyikan bel. Robot harus beroperasi secara otomatis tanpa interaksi dengan manusia kecuali saat pengaktifan pertama kali. Dimensi robot dibatasi 60cm x 60cm x 160 cm (panjang x lebar x tinggi), sedangkan besaran lain tidak dibatasi. Pada saat berjalan, minuman yang dibawa robot tidak boleh tumpah lebih dari 20% dari total volume minuman.

Berdasarkan hal tersebut, masalah yang dapat dirumuskan adalah bagaimana membuat robot dengan ketentuan di atas. Aspek teknik yang digunakan untuk landasan perancangan adalah sebagai berikut:

1. Syarat geometri: maksimal lebar 0,6m x panjang 0,6m x tinggi 1,6m
2. Mampu naik dan turun tangga, dengan kata lain titik berat robot tidak bergerak di luar bidang tumpu pada anak tangga.
3. Mampu bergerak otomatis saat diaktifkan dan mengenali kondisi lingkungan yaitu dinding anak tangga dan lantai.
4. Karena berada di ruang publik, maka robot tidak boleh terganggu oleh suara, cahaya matahari, lampu penerang atau lampu kilat

5. Syarat keselamatan: robot tidak boleh melukai orang atau dirinya sendiri (dalam hal ini cairan yang tumpah bisa merusak rangkaian elektronika)

Aspek-aspek tersebut selanjutnya menjadi batasan masalah dalam mengembangkan desain robot.

II. METODE PENELITIAN

Secara keseluruhan, penelitian ini terdiri dari pengumpulan data, proses pemecahan masalah, desain dan simulasi, pembuatan robot, uji coba dan evaluasi.

1. Pengumpulan data
Dalam tahap ini dilakukan observasi lapangan dan studi kasus dari peraturan kompetisi yang tersedia.
2. Pemecahan masalah
Metode yang digunakan untuk pemecahan masalah adalah metode 635 dan analisa morfologi. Tahap ini menghasilkan satu sketsa robot dan sketsa langkah gerakan saat naik dan turun tangga.
3. Desain dan simulasi
Permodelan CAD dilakukan dengan menggunakan perangkat lunak Solidworks berdasarkan sketsa pada tahap sebelumnya. Model rakitan dibuat untuk menghitung berat robot dan menganalisa gerakan masing-masing kerangka. Berat robot selanjutnya digunakan untuk menghitung kapasitas motor yang akan digunakan.
4. Pembuatan robot
Model pada Solidworks selanjutnya diubah menjadi gambar teknik yang akan digunakan untuk landasan dalam merakit robot.

Perakitan komponen elektronika dilakukan setelah kerangka robot selesai dirakit.

5. Uji coba
Setelah dirakit, robot dicoba untuk dioperasikan sesuai dengan aspek teknik yang diharapkan.
6. Evaluasi
Jika ada penyimpangan, maka sebagian langkah penelitian diulang dan modifikasi dilakukan sampai mendekati hasil yang diinginkan.

III. HASIL DAN PERANCANGAN

III. 1. Solusi Pemecahan Masalah

Metode pemecahan masalah yang digunakan oleh tim adalah metode 635 dan analisa morfologi. Metode 635 dilakukan oleh 6 anggota tim yang masing-masing menuliskan/menggambarkan 3 ide di selembar kertas. Selanjutnya kertas tersebut diputar di antara anggota tim. (pertukaran ide melalui tulisan/gambar, *brainwriting / minddrawing*). Setelah itu masing-masing anggota menuliskan lagi 3 ide di kertas baru dan diputar lagi.

Proses pertukaran ide ini dilakukan sebanyak 5 kali, sehingga dalam waktu yang singkat akan dihasilkan banyak ide. Beberapa ide besar dipilih untuk dianalisa lebih lanjut yaitu menggunakan analisa morfologi.

Tabel 1. Analisa morfologi dengan tiga pilihan solusi

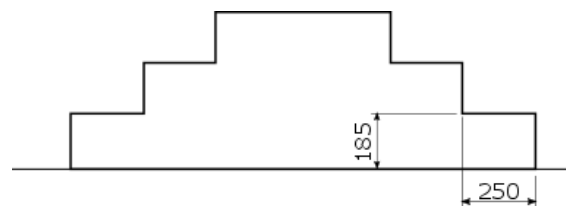
Sub Fungsi	Pilihan 1	Pilihan 2	Pilihan 3
Mekanisme naik dan turun tangga	Roda tank dan mekanisme penyeimbang pada beban	Robot berkaki dan mekanisme penyeimbang	Rak & pinion tiga kerangka badan tanpa penyeimbang
Mekanisme pemukulan lonceng	Batang vertikal tanpa aktuator	Mekanisme gantung	Mekanisme kipas
Penggerak	Motor langkah	Motor Servo	Motor DC
Panel operasi	Tombol	Jarak jauh	Saklar
Pengendali	Arduino	Raspberry Pi	Intel Galileo
Sensor jarak	Foto-resistor	Ultrasonik	IR proximity
Sensor posisi	Ultrasonik	IR proximity	Limit Switch & Magnetik
Catu daya	Lithium polymer	Alkaline	NiMH

Analisa ini menggunakan tabel yang terdiri dari sub-fungsi pada bagian baris dan pilihan solusi pemecahan masalah pada bagian kolom. Dengan kombinasi sistematis, solusi dan sub-fungsi dapat digabung untuk menyelesaikan masalah keseluruhan sistem.

Berdasarkan tahap pemecahan masalah menggunakan metode 635 maka diperoleh beberapa alternatif solusi seperti yang tercantum pada Tabel 1. Analisa morfologi dilakukan dengan cara menilai konsep yang terbaik dari masing-masing solusi pada tiap sub-fungsi. Pilihan solusi terbaik pada masing-masing sub-fungsi ditunjukkan pada Tabel 1 dengan tanda garis bawah.

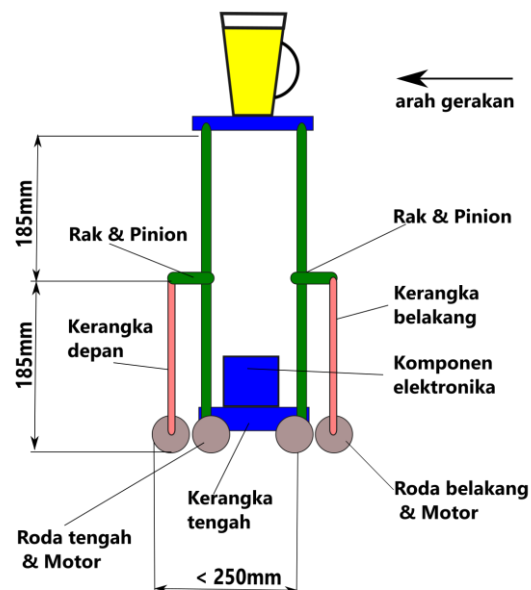
III. 2. Perancangan Mekanik

Robot terdiri dari 3 kerangka utama yaitu kerangka depan, tengah dan belakang. Antara kerangka depan - tengah dan tengah - belakang dihubungkan dengan sambungan translasi dan penggerak rak pinion, agar bisa bergerak translasi relatif terhadap kerangka tengah dengan jarak jangkauan 2 x 18,5 cm (menyesuaikan ketinggian anak tangga) yang bisa dilihat di Gambar 1. Struktur tiga rangka ini menjamin minuman tetap pada posisi horizontal.



Gambar 1. Ukuran anak tangga di arena

Komponen utama untuk membentuk badan robot adalah batang profil aluminium yang dapat dipotong sesuai kebutuhan.



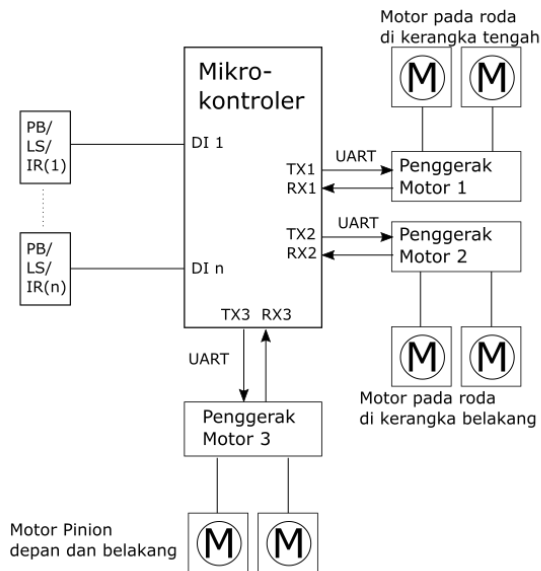
Gambar 2. Sketsa desain robot

Gambar 2 menunjukkan sketsa robot dan bagian-bagiannya. Perlu diperhatikan bahwa jarak roda paling depan dengan titik tengah roda belakang pada kerangka tengah harus kurang dari 250mm atau lebar anak tangga supaya tidak jatuh. Komponen elektronika ditempatkan di

bawah dan tertutup agar terhindar dari kemungkinan terkena tumpahan minuman.

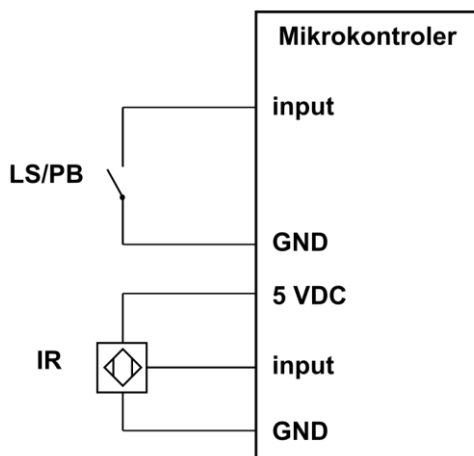
III.3. Perancangan Rangkaian Elektronik

Mikrokontroler Arduino Mega adalah pengendali utama robot. Pemilihan mikrokontroler ini berdasarkan jumlah masukan/keluaran yang diperlukan, harga dan ketersediaan dukungan modul-modul pemrograman yang siap pakai. Hubungan mikrokontroler dengan piranti lain dapat dilihat di Gambar 3.



Gambar 3. Diagram blok sistem pengendali robot

Pada Gambar 4 dijelaskan koneksi antara mikrokontroler dengan piranti masukan berupa tombol, saklar pembatas, atau sensor foto infra merah.



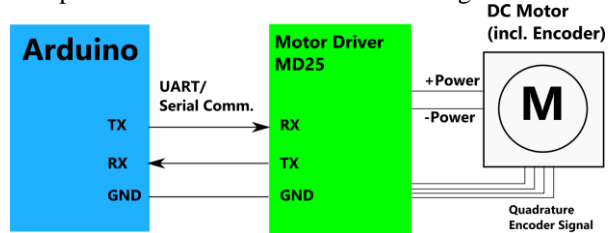
Gambar 4. Koneksi piranti masukan ke mikrokontroler.

Koneksi tombol atau saklar dihubungkan ke masukan digital dari mikrokontroler dan terminal 0 volt (*ground*). Mikrokontroler Arduino sudah memiliki hambatan internal tarik ke atas (*internal pull-*

up resistor) sehingga dengan mengaktifkan tombol maka akan menghasilkan logika nol (0) atau *active low*.

Sensor foto (IR) memiliki tiga kabel yang masing-masing dihubungkan ke positif 5 volt, masukan digital dan 0 volt (*ground*). Aktifnya sensor ini akan menghasilkan logika 1 pada mikrokontroler atau *active high*. Sensor infra merah digunakan karena sesuai dengan kondisi lingkungan yang tidak terganggu dengan cahaya luar.

Perangkat keluaran berupa motor DC dengan penggerak motor (*motor driver*) lewat komunikasi serial UART. Penggerak motor yang digunakan yaitu dua buah MD25 dengan arus maksimal 2,8 A untuk roda di kerangka tengah-belakang, dan sebuah Sabertooth dengan arus maksimal 30A digunakan untuk penggerak motor pada rak/pinion. MD25 dan Sabertooth keduanya dapat mengendalikan kecepatan dan arah dua buah motor sekaligus.

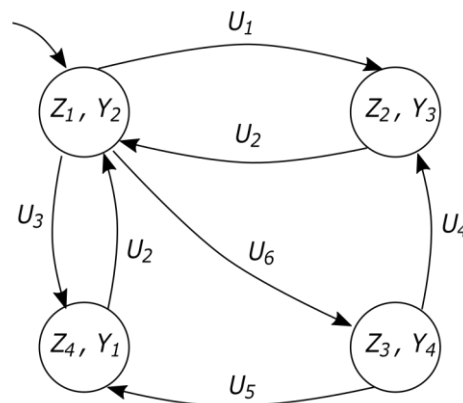


Gambar 5. Koneksi penggerak motor

Gambar 5 menjelaskan koneksi penggerak motor MD25 dengan mikrokontroler melalui komunikasi serial. Motor DC memiliki pulsa umpan balik dari enkoder putar yang digunakan oleh MD25 untuk kendali kecepatan.

III. 4. Permodelan Matematika

Karena elemen kerja pada robot ini bekerja secara berurutan maka model yang sesuai untuk keperluan ini adalah mesin keadaan-hingga (*finite-state machine*).



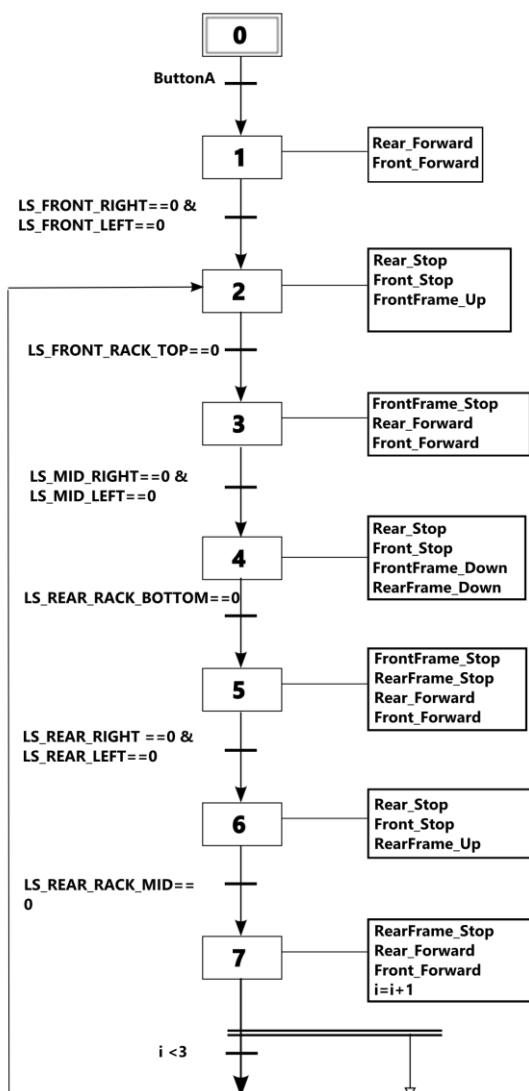
Gambar 6. Contoh mesin keadaan

Pada mesin keadaan-hingga, suatu keadaan (*state*) akan berubah ke keadaan lain pada saat fungsi input tertentu aktif. Sebuah contoh mesin keadaan dapat dilihat pada Gambar 6. Pada contoh ini notasi Z menyatakan keadaan, Y menyatakan keluaran dan U adalah masukan atau fungsi transisi. Kondisi inialisasi berada pada Z1 yang ditandai

dengan satu anak panah dari luar. ^[2] Jika U_1 aktif pada saat keadaan Z_1 , maka keadaan akan berubah menjadi keadaan Z_2 dan Y_3 akan aktif. Keadaan Z_2 menunggu sampai ada transisi U_2 aktif dan kembali ke keadaan Z_1 .

Pada robot ini, himpunan hingga-keadaan (*finite-state*) merupakan himpunan elemen kerja robot seperti jalan maju, kerangka depan naik, kerangka belakang naik dan seterusnya. Sedangkan fungsi input adalah suatu sensor atau kumpulan sensor yang aktif pada keadaan tertentu.

Sequential Function Chart for Climbing Up



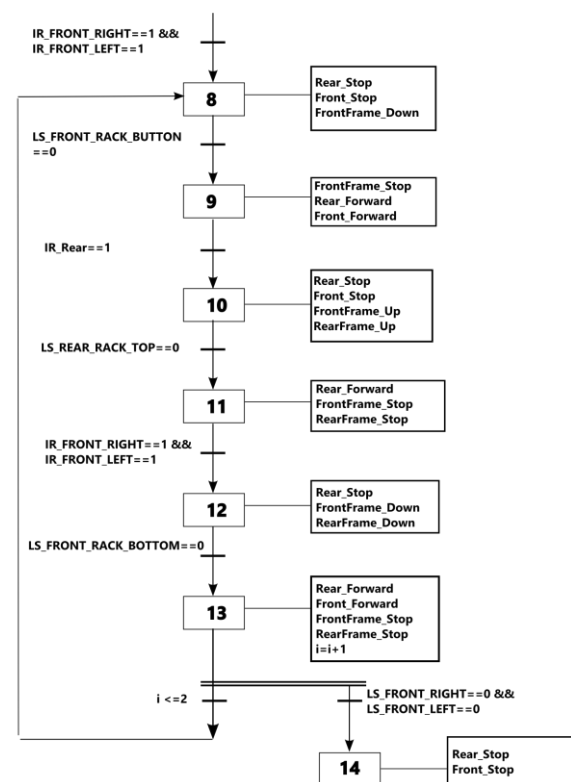
Gambar 7. Diagram alir keadaan saat naik tangga

Gambar 7 menjelaskan mesin keadaan (*state machine*) yang sudah dikonversi ke SFC (*Sequential Function Chart*), pada saat robot berjalan di lantai dan proses menaiki tangga. Kita lihat di sini terdapat keadaan ke-0 sampai ke-7. Keadaan ke-0 adalah kondisi awal pada saat robot siap menerima perintah untuk mulai beroperasi. Saat kondisi transisi aktif

(ditandai dengan penekanan tombol), maka keadaan akan berubah dari 0 ke 1. Keadaan 1 adalah aktifnya motor depan dan motor belakang atau bisa dikatakan robot sedang maju. Keadaan 1 berubah ke keadaan 2 saat transisi 1-2 aktif yaitu sensor depan kanan dan kiri. Pada saat keadaan 2, motor depan dan belakang berhenti dan secara bersamaan kerangka depan mulai naik. Demikian seterusnya sampai keadaan ke-7. Perlu diingat bahwa anak tangga ada tiga, sehingga keadaan 1 sampai 7 perlu diulang sebanyak tiga kali, karena prosesnya sama.

Ketika robot bergerak turun tangga lalu memukul lonceng dapat dijelaskan pada Gambar 8. Keadaan 8 dicapai setelah proses menaiki tangga 3 kali terlampaui. Dengan konsep yang sama, keadaan 8 berpindah ke keadaan 9 ketika transisi 8-9 aktif. Setelah mencapai keadaan 13, keadaan diulang ke-8 sebanyak tiga kali dan selanjutnya bergerak ke titik tujuan dengan menyentuh lonceng.

Sequential Function Chart for Climbing Down



Gambar 8. Diagram alir keadaan saat turun tangga

Tabel 2 memuat daftar sub program yang sering dipanggil di dalam program utama saat keadaan tertentu tercapai.

Tabel 2. Daftar sub program

Referensi program	Deskripsi
Front_Forward	Motor di kerangka tengah maju
Rear_Forward	Motor di kerangka belakang maju

Front_Stop	Motor di kerangka tengah berhenti
Rear_Stop	Motor di kerangka belakang berhenti
Front_Frame_Down	Kerangka depan bergerak turun
Front_Frame_Up	Kerangka depan bergerak naik
Front_Frame_Stop	Kerangka depan berhenti
Rear_Frame_Down	Kerangka belakang bergerak turun
Rear_Frame_Up	Kerangka belakang bergerak naik
Rear_Frame_Stop	Kerangka belakang berhenti

Tabel 3. Daftar piranti masukan

Referensi program	Deteksi sensor	Letak sensor/fungsi
LS_FRONT_RIGHT	Dinding	kerangka depan kanan
LS_FRONT_LEFT	Dinding	kerangka depan kiri
LS_MID_RIGHT	Dinding	kerangka tengah kanan
LS_MID_LEFT	Dinding	kerangka tengah kiri
LS_REAR_RIGHT	Dinding	kerangka belakang kanan
LS_REAR_LEFT	Dinding	kerangka belakang kiri
LS_FRONT_RACK_TOP	Posisi rak	rak/pinion depan atas
LS_FRONT_RACK_MID	Posisi rak	rak/pinion depan tengah
LS_FRONT_RACK_BOTTOM	Posisi rak	rak/pinion depan bawah
LS_REAR_RACK_TOP	Posisi rak	rak/pinion belakang atas
LS_REAR_RACK_MID	Posisi rak	rak/pinion belakang tengah
LS_REAR_RACK_BOTTOM	Posisi rak	rak/pinion belakang bawah
IR_FRONT_RIGHT	Lantai	kerangka depan kanan
IR_FRONT_LEFT	Lantai	kerangka depan kiri
IR_REAR	Lantai	kerangka belakang (tengah)
ButtonA	-	Tombol mulai
ButtonB, C,D,E	-	Kendali manual rak depan dan belakang untuk naik atau turun.

Daftar sensor dan tombol dapat dilihat di Tabel 3. Secara umum ada tiga kegunaan sensor yaitu untuk mendeteksi dinding (bidang vertikal), lantai (bidang horizontal) dan posisi kerangka untuk membatasi gerak. Sedangkan tombol-tombol digunakan untuk kendali manual pada panel kontrol.

III. 5. Pemrograman Mikrokontroler

Struktur program Arduino terdiri dari sub program **setup()** dan **loop()**. Sub program **setup()** hanya dieksekusi sekali pada saat awal program, sedangkan **loop()** dieksekusi setelahnya secara berulang-ulang. Sub program atau sub fungsi lain dapat kita tambahkan sesuai keperluan dan bisa dipanggil di dalam **setup()** atau **loop()**.

Mesin keadaan (*state machine*) dapat diaplikasikan ke dalam program mikrokontroler dengan menggunakan struktur pengendalian **switch case break** seperti terlihat pada Gambar 9.

```

contoh_mesin_keadaan $
int keadaan=0; // Deklarasi variabel
int sensor1;

void loop()
{
  switch (keadaan) // evaluasi nilai variabel keadaan
  {
    case 0: // saat keadaan ke-0
      if (sensor1==0) // jika transisi 0 ke 1 aktif (sensor1)
      { // maka
        sub_program_1(); // sub program 1, sub fungsi 1
        sub_fungsi_1(); // dan yang lainnya akan dikerjakan
        .
        . // pada saat yang sama nilai keadaan
        keadaan=1; // diubah ke keadaan berikutnya
      }
      break;

    case 1: // saat keadaan ke-1 berlaku
      . // struktur yang sama
      .
      break;
    default: // kondisi inisialisasi
      keadaan=0; // berada di keadaan ke-0
      break;
  }
}

```

Gambar 9. Implementasi struktur mesin keadaan pada program Arduino

Pertama kita tentukan variabel keadaan dengan tipe integer dan kita beri nilai inisialisasi 0. Pada sub program **loop()**, variabel keadaan diuji nilainya menggunakan **switch()**. Jika nilainya 0 maka program akan diulang-ulang pada **case 0** sampai kondisi pada transisi (input: sensor1) aktif. Pada saat ini instruksi di dalam **case..break** ini dieksekusi dan nilai variable keadaan diubah ke keadaan berikutnya.

Saat keadaan bernilai 1 maka program akan berulang pada **case 1** dan menunggu transisi yang sesuai. Demikian seterusnya sampai semua keadaan masuk dalam struktur **select().. case**.

Dengan menggunakan model ini, pemrograman dapat dilakukan dengan terstruktur untuk menghindari terlewatnya suatu keadaan atau fungsi yang harus dikerjakan oleh robot.

III. 6. Uji Coba

Uji coba dilakukan secara bertahap yaitu pertama pada fungsi kerja roda depan dan tengah. Kecepatan dan arah dievaluasi untuk menghasilkan gerakan yang optimal.

Selanjutnya fungsi gerakan vertikal kerangka depan dan belakang diperiksa untuk mendapatkan nilai parameter yang tepat supaya depan dan belakang memiliki kecepatan yang sama. Parameter kecepatan berbeda pada saat kerangka naik dan saat turun. Uji coba berikutnya adalah pemeriksaan semua sinyal masukan dari sensor agar terbaca oleh mikrokontroler.



Gambar 10. Foto robot saat uji coba menuruni tangga

Setelah masing-masing sub sistem berjalan dengan normal, uji coba program utama dilakukan tanpa dijalankan di tangga. Hal ini untuk menghindari kerusakan akibat robot jatuh jika ada keadaan yang belum tercakup di dalam program.

Tahap berikutnya adalah uji coba robot di arena yang sebenarnya, yaitu dari awal, jalan, naik dan turun tangga. Gambar 10 menunjukkan proses uji coba di tangga. Uji coba keseluruhan sistem menunjukkan bahwa logika program berjalan sesuai harapan. Pada saat robot naik maupun turun, terdapat kemiringan robot yang dikarenakan perbedaan kecepatan antara kerangka depan dan belakang. Pengaturan parameter

kecepatan dapat mengurangi kemiringan ini sehingga minuman tidak tumpah saat dibawa.

IV. KESIMPULAN DAN SARAN

Rancang bangun robot logistik dapat direalisasikan dengan solusi permasalahan menggunakan metode 635 dan analisis morfologi. Mekanisme robot menggunakan struktur tiga kerangka tanpa memerlukan penyeimbang beban karena titik berat berada di antara bidang tumpu. Pemrograman mikrokontroler dapat dilakukan dengan terstruktur berkat bantuan permodelan mesin keadaan.

Uji coba menunjukkan bahwa robot dapat bertugas mulai dari titik awal, berjalan lurus, menaiki tangga, menuruni tangga dan berjalan lurus sampai posisi yang diinginkan. Walaupun ada sedikit kemiringan saat naik atau turun tangga, tetapi minuman tidak tumpah.

Rancang bangun robot ini secara khusus dilakukan untuk memenuhi kriteria kompetisi robot. Namun dari hasil yang dicapai, penulis memberikan saran bahwa pengembangan robot ini dapat dilakukan lebih lanjut untuk dimanfaatkan oleh masyarakat umum. Contohnya, intelegensi robot dapat dikembangkan agar bisa naik tangga dengan ukuran dan jumlah anak tangga yang bervariasi. Dalam hal konstruksi elektro-mekanik pada bagian rak dan pinion, perlu dikembangkan menggunakan kendali posisi dan kecepatan untuk menjamin tidak terjadinya kemiringan badan robot.

V. DAFTAR PUSTAKA

- [1] Bayle, Julien (2013). *C programming for Arduino*. Packt Publishing Ltd., UK
- [2] Litz, Lothar (2005). *Grundlagen der Automatisierungstechnik*. Oldenbourg Wissenschaftsverlag GmbH, München.
- [3] Lunze, Jan. (2012). *Ereignisdiskrete Systeme*. Oldenbourg Wissenschaftsverlag GmbH, München.