**Kendali Robot Berdasarkan Perintah Inisialisasi Awal Pengguna**

**Adit Ilham Nugroho1, Mochammad Hannats Hanafi I. 2, Rizal Maulana3**

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya

Email: 1aditilham152@gmail.com, 2hanas.hanafi@ub.ac.id, 3rizal\_lana@ub.ac.id

**Abstrak**

Selama 30 tahun terakhir, penelitian robotika dan perkembangan robotika terus berkembang robot autonomous yang dulu robot industri tradisional dan animal like atau humanoid robot. Salah satu dari perkembangan tersebut adalah teknologi robot industri atau disebut juga industrial robot. Ada beberapa alasan mengapa robot yang diartikulasikan banyak dikembangkan antara lain robot ini masuk kedalam robot pabrik yang dimana industri saat ini terus berkembang dan juga memiliki ruang gerak yang bebas 360°. Robot industri dapat meningkatkan produktivitas dan efiensi pada sebuah pabrik untuk menjalankan lini produksi. Penulis membuat sebuah protype robot yang dapat bergerak sesuai dengan inisialisasi awal pengguna, robot ini memudahkan pengguna dalam menjalankan robot jika ingin merubah gerakan robot, dibandingkan dengan robot autonomous yang sudah ada. Robot ini tergolong robot manual yang menekan biaya pembelian robot. Pada penelitian ini hanya menggunakan software Arduino dengan fungsi EEPROM untuk menyimpan gerakan yang telah diberikan. Software Arduino merupakan perangkat lunak open-source untuk memudahkan penulisan kode. Alat bantu hardware Arduino tipe Mega 2560, Arduino Mega 2560 merupakan perangkat keras pengembangan mikrokontroller berbasis Arduino dengan menggunakana chip ATmega2560 untuk membuat prototipe penelitian. Penelitian ini menggunakan total 5 joint dengan menggunakan penggerak Servo, Servo Motor yang digunakan adalah Servo SG90. Pengujian gerakan servo 0⁰-180⁰ memiliki nilai rata-rata error sebesar 1.85 % dan pengujian gerakan servo 0⁰-360⁰ memiliki nilai rata-rata error sebesar 2.32 %. Lalu pada analisis perintah 1 gerakan nilai rata-rata error hanya sebesar 2.9 % dan pada analisis perintah keseluruhan nilai rata-rata error hanya sebesar 1.5 %.

**Kata kunci**: robot, servo, EEPROM, Arduino Mega

**Abstract**

*Over the past 30 years, robotics research and robotics developments have continued to evolve the autonomous robots that used to be traditional industrial robots and animal like or humanoid robots. One of these developments is industrial robot technology or also called industrial robot. There are several reasons why many articulated robots are developed, among others, this robot into the robot factory where the current industry continues to grow and also has a free space 360 °. Industrial robots can increase productivity and efficiency in a factory to run the production line. The author created a protype robot that can move according to the initial initialisation of the user, this robot allows users to run the robot if you want to change the robot movement, compared with existing autonomous robot. This robot is classified as a manual robot that reduces the cost of purchasing a robot. In this study only use Arduino software with EEPROM function to save the movement that has been given. Arduino software is an open-source software for easy code writing. Arduino hardware device type Mega 2560, Arduino Mega 2560 is a hardware development of Arduino-based microcontroller using ATmega2560 chip to create research prototype. This study uses a total of 5 joints using Servo drives, Servo motors used are Servo SG90. Testing servo movement 0⁰-180⁰ has an average error rate of 1.85% and servo movement 0⁰-360⁰ has an average error rate of 2.32%. Then in order analysis 1 movement the average value of error is only 2.9% and in the overall command analysis the average error rate is only 1.5%.*

**Keywords**: *robot, servo, EEPROM, Arduino Mega*

# PENDAHULUAN

Perkembangan dunia robotika terjadi sangat cepat pada era ini. Ketertarikan manusia yang tinggi terhadap perkembangan robot berujung pada penelitian yang semakin bermunculan.

Selama 30 tahun terakhir, penelitian robotika dan perkembangan robotika terus berkembang robot *autonomous* yang dulu robot industri tradisional dan animal like atau *humanoid* robot. Salah satu dari perkembangan tersebut adalah teknologi robot industri atau disebut juga *industrial* robot. Berdasarkan penjelasan dari *website* Administrasi Keselamatan Kerja dan Kesehatan Kerja Departemen Tenaga Kerja Amerika Serikat yang diakses pada tanggal 23 September 2017 mengatakan bahwa robot industri berarti alat mekanis multifungsi yang dapat diprogram yang dirancang untuk memindahkan material, komponen, peralatan, atau perangkat khusus melalui gerakan terprogram variabel untuk melakukan berbagai tugas. Sistem robot industri tidak hanya mencakup robot industri tetapi juga perangkat dan / atau sensor yang dibutuhkan robot untuk melakukan tugasnya serta melakukan sekuensing atau memantau antarmuka komunikasi.

Robot umumnya digunakan untuk melakukan tugas yang tidak aman, berbahaya, sangat berulang, dan tidak menyenangkan. Mereka memiliki banyak fungsi yang berbeda seperti penanganan material, perakitan, pengelasan busur, pengelasan ketahanan, pemuatan alat mesin dan fungsi pembongkaran, pengecatan, penyemprotan, dll. (Department of Labor, 2017)

*Articulated arm robot* berdasarkan penjelasan yang telah dibuat oleh Carlos M. Gonzales, seorang *technology*. Articulated Robots atau *Robots Arms* dilengkapi sendi putar yang dapat berkisar dari struktur dua sendi sederhana hingga struktur rumit dengan 10 atau lebih sambungan. Lengan terhubung ke alas yang memiliki sambungan memutar. Sendi putar menghubungkan kaitan di lengan; Setiap sendi adalah sumbu yang berbeda dan memberikan tingkat kebebasan tambahan. Lengan robot industri memiliki empat atau enam sumbu. Robot semacam itu terutama digunakan untuk operasi perakitan, mesin *die-casting*, *fettling*, pengelasan gas dan busur, dan penerapan cat. (Gonzales, 2016)

Ada beberapa alasan mengapa robot yang diartikulasikan banyak dikembangkan antara lain robot ini masuk kedalam robot pabrik yang dimana industri saat ini terus berkembang dan juga memiliki ruang gerak yang bebas 360°. Robot industri dapat meningkatkan produktivitas dan efiensi pada sebuah pabrik untuk menjalankan lini produksi.

Inisialisasi adalah proses mencari dan menggunakan nilai yang ditentukan untuk data variabel yang digunakan oleh program komputer. Misalnya, sistem operasi atau program aplikasi diinstal dengan nilai default atau nilai yang ditentukan pengguna yang menentukan aspek tertentu dari bagaimana sistem atau program berfungsi. Biasanya, nilai-nilai ini disimpan dalam file inisialisasi. Ketika sistem operasi atau program aplikasi pertama kali dimasukkan ke dalam memori, sebagian program melakukan inisialisasi - yaitu, tampilannya dalam file inisialisasi, menemukan nilai pasti untuk menggantikan nilai variabel, dan bertindak sesuai. Disini menggunakan inisialisasi awal pengguna, jadi pengguna memberikan gerakan terhadap robot dan gerakan tersebut menjadi inisialisasi untuk menjadi gerakan robot. (Thing, 2016)

Penelitian yang sudah dilakukan oleh Adiyatma Ghazian ST, Ir. Nurussa’adah, MT., Mochammad Rif’an ST., MT. menjelaskan terkait gerakan robot secara koordinat menggunakan metode invers kinematik. Invers kinematik berfokus pada penentuan nilai-nilai variabel joint adar didapat posisi yang diinginkan pada orientasi *end-effector*. Didapat kan hasil pada posisi 0⁰ membutuhkan waktu sebesar 2,1ms dan pada posisi 90⁰ membutuhkan waktu sebesar 0,9ms. Penelitian berikutnya oleh Dr. Bindu A Thomas, Stafford Michahial, Shreeraksha.P, Vijayashri B Nagvi, Suresh M. menjelaskan tentang *autonomous* robot lengan dibidang industri. Kondisi robot tersebut akan menunggu dengan waktu yang telah ditentukan ketika ada hambatan pada saat menggerakan objek. Dan akan kembali bergerak ketika hambatan telah dibersihkan, namun jika masih terjadi hambatan maka *buzzer* akan dihidupkan untuk memanggil operator atau personil industri untuk mengatasi masalah.

Penulis membuat sebuah *protype* robot yang dapat bergerak sesuai dengan inisialisasi awal pengguna, robot ini memudahkan pengguna dalam menjalankan robot jika ingin merubah gerakan robot, dibandingkan dengan robot *autonomous* yang sudah ada. Robot ini tergolong robot manual yang menekan biaya pembelian robot.

Pada penelitian ini hanya menggunakan *software* Arduino. *Software* Arduino merupakan perangkat lunak *open-source* untuk memudahkan penulisan kode. Alat bantu hardware Arduino tipe Mega 2560, Arduino Mega 2560 merupakan perangkat keras pengembangan mikrokontroller berbasis Arduino dengan menggunakana chip ATmega2560 untuk membuat prototipe penelitian. Penelitian ini menggunakan total 5 joint dengan menggunakan penggerak Servo, Servo Motor yang digunakan adalah Servo SG90. Servo Motor merupakan sebuah motor dengan sistem umpan balik tertutup di mana posisi dari motor akan diinfirmasikan kembali ke rangkaian kontrol yang ada di dalam motor servo. Dengan total ada 5 joint dan diinisialisasi awal dari pengguna, sehingga pengguna bisa memberikan gerakan yang diinginkan.

# PERANCANGAN SISTEM

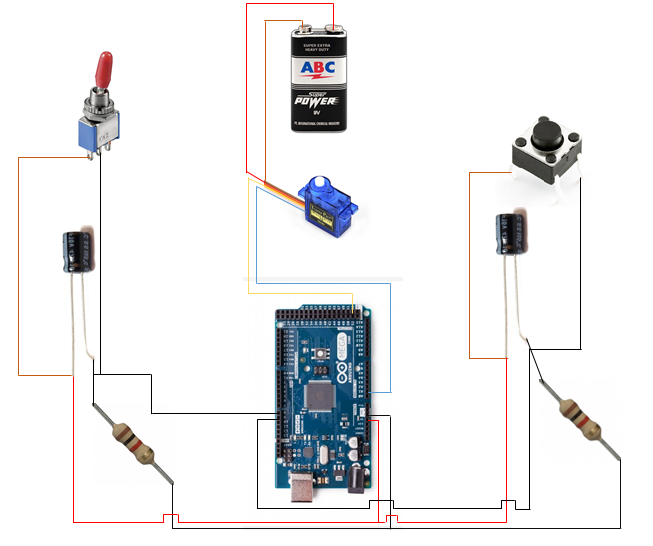
Pada Perancangan Sistem disini menjelaskan tahap yang akan dilakukan dalam pembuatan sistem ini

* 1. **Perancangan Sistem Perangkat Keras**

Perancangan sistem Perangkat Keras ini terbagi menjadi 3 subbab yang akan menjelaskan tahapan dalam permbuatan perangkat keras sistem.

* + 1. **Perancangan Rangkaian Perangkat Keras**

Perancangan sistem perangkat keras terdapat 7 komponen utama yang harus ada dalam robot, yaitu terdapat servo motor, arduino mega, *push button*, *toggle switch*,rangkaian *debounce*,baterai dan power bank sebagai sumber tegangan.



Gambar 3 Skematik Keseluruhan Sistem

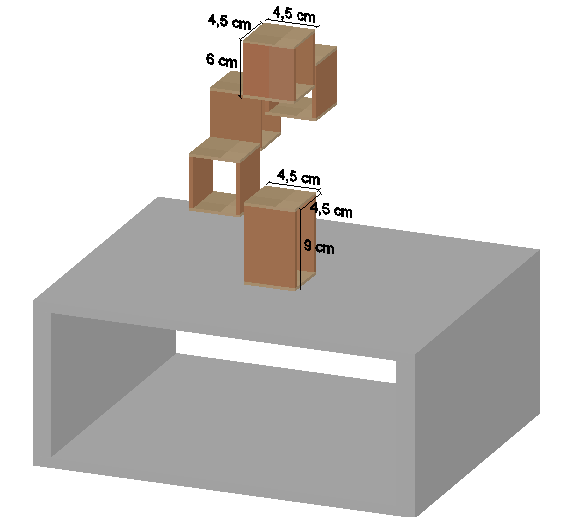
Dari skematik pada Gambar 3 diatas, berikut merupakan pin-pin dari keseluruhan sistem

Tabel 1 Pin Skematik Keseluruhan Sistem

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Arduino Mega | Servo | *Push Button* | *Toggle Switch* |
| 1 | Pin A0 | *Feedback* Servo 1 |  |  |
| 2 | Pin A1 | *Feedback* Servo 2 |  |  |
| 3 | Pin A2 | *Feedback Servo 2* |  |  |
| 4 | Pin A3 | *Feedback* Servo 4 |  |  |
| 5 | Pin A4 | *Feedback* Servo 5 |  |  |
| 6 | Pin 4 |  | *Push Button* 1 |  |
| 7 | Pin 5 |  | *Push Button* 2 |  |
| 8 | Pin 6 |  | *Push Button* 3 |  |
| 9 | Pin 7 |  | *Push Button* 4 |  |
| 10 | Pin 8 |  | *Push*  *Button* 5 |  |
| 11 | Pin 9 |  |  | Merekam |
| 12 | Pin 10 |  |  | Bergerak sekali |
| 13 | Pin 11 |  |  | Bergerak kontinu |
| 14 | Pin 53 | Pin Servo 1 |  |  |
| 15 | Pin 51 | Pin Servo 2 |  |  |
| 16 | Pin 49 | Pin Servo 3 |  |  |
| 17 | Pin 47 | Pin Servo 4 |  |  |
| 18 | Pin 45 | Pin Servo 5 |  |  |
| 19 | Pin 5V |  | Pin VCC | Pin VCC |
| 20 | Pin GND |  | Pin GND | Pin GND |

* + 1. **Perancangan Robot Lengan**

Perancangan robot lengan ini berdasarkan perhitungan panjang servo yang akan diletakan disetiap lengan. Lengan robot ini dibuat dari karton duplex yang masing-masing ukurannya adalah 6 x 4.5 cm sebanyak 8 buah, 4.5 x 4.5 cm sebanyak 10 buah, dan 9 x 4.5 cm sebanyak 2 buah. Yang nantinya lengan pertama akan lebih panjang daripada lengan lainnya, tujuannya adalah untuk menumpu servo dengan lengan lainnya dan dapat membentuk pola setengah bola. Disetiap sisi lengan akan dipasangi dengan *push button*, push button ini nantinya bertujuan untuk menandakan servo mana yang akan memulai sistem perekaman pergerakan. Gambar 4 merupakan desain dari lengan robot

. 

Gambar 4 Desain Lengan Robot

* + 1. **Perancangan Pergerakan Lengan Robot**

Percangan pergerakan lengan robot dan posisi lengan ini diposisikan agar lengan tetap stabil. Nantinya lengan kedua akan diletakan disebelah kiri lengan pertama, lengan ketiga disebelah kiri lengan kedua yang berarti berada dibelakang lengan pertama, lengan keempat berada disebelah kiri lengan ketiga yang berarti membelakangi lengan kedua, lengan kelima berada disebelah kiri lengan keempat yang berarti berada diatas lengan pertama. Posisi lengan nantinya akan membentuk memutar.

Pergerakan lengan robot sendiri adalah 2 servo bergerak dari 0⁰ hingga 180⁰ dan 3 servo bergerak dari 0⁰ hingga 180⁰. Penggunaan 2 servo servo tersebut terletak pada penampang menuju lengan 1 dan lengan 2 menuju lengan 2, penempatan ini agar tidak ada kabel yang menghambat dari gerak servo tersebut. Dan penggunaan 3 servo 0⁰ hingga 360⁰ terletak pada sisa lengan, agar bagian atas robot dapa bergerak bebas. Sehingga membentuk gerakan setelah bola.

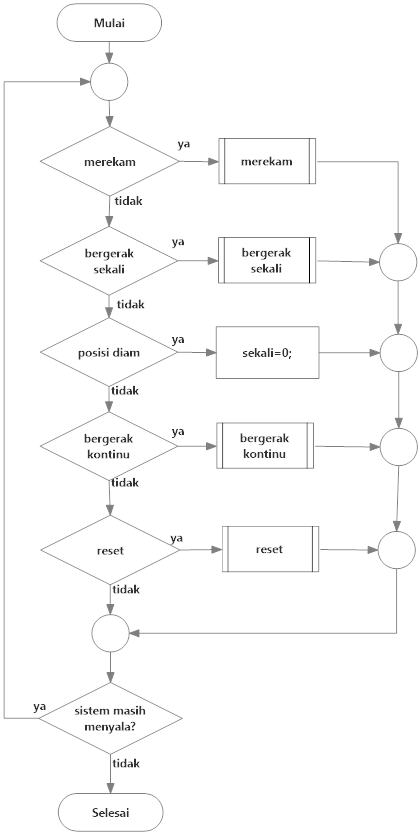
Langkah-langkah dalam membaca inisialisasi yang pertama adalah dengan menyalakan atau memposisikan *toggle switch* merekam dalam kondisi menyala atau *on*, lalu disetiap lengan robot terdapat push button yang berfungsi untuk memberikan tanda bahwa servo yang lebih dulu ditekan *push button*-nya akan memulai merekam, berikutnya pengguna akan menggerakan lengan tersebut dengan kurun waktu 5 detik dan terakhir akan tersimpan pada slot EEPROM yang telah dibagi menjadi 5 *slot* alamat, dengan masing-masing 50 alamat, lalu *toggle switch* dimatikan untuk memberhentikan perintah perekaman.

Untuk langkah-langkah menjalankan perintah adalah yang pertama *toggle switch* bergerak dalam kondisi menyala atau *on*, lalu program akan melakukan pembacaan pada setiap EEPROM yang telah disimpan sebelumnya, robot akan bergerak satu persatu dengan diawali dari lengan satu dan lanjut ke lengan dua dan seterusnya. Perbedaan bergerak sekali dan bergerak berkali-kali adalah ketika bergerak sekali maka alat akan bergerak dari lengan pertama hingga lengan kelima lalu berhenti, sedangkan bergerak berkali-kali maka robot akan terus bergerak dari lengan pertama hingga lengan kelima dan langsung melakukan gerakan pada lengan pertama hingga lengan kelima, bergerak berkali-kali akan berhenti ketika *toggle switch* bergerak berkali-kali dalam keadaan mati atau *off*.

Selanjutnya adalah langkah-langkah penyimpanan pada EEPROM, pertama setelah melakukan pembacaan nilai ADC (*Analog to Digital Converter*) nilai tersebut akan dilakukkan *mapping* untuk mengalokasikannya pada EEPROM, satu lengan akan diberi alamat sebesar 50, sehingga membutuhkan 250 alamat pada EEPROM. Pembacaan tersebut akan dilakukan 50 alamat / 5 detik.

* 1. **Perancangan Sistem Perangkat Lunak**

Perancangan sistem perangkat lunak ini membahas tentang bagaimana cara agar sistem ini mampu bergerak seimbang dengan sistem perangkat keras yang akan menjalankan instruksi yang dibutuhkan sistem untuk bergerak sebagaimana fungsi yang telah ditentukan. Perangkat lunak yang digunakan dalam perancangan sistem yaitu Arduino. Perancangan perangkat lunak dijelaskan untuk mengetahui alur kerja sistem melalui diagram alir yang mana diagram tersebut berisi tentang cara perekaman inisialisasi kemudian bergerak sekali sesuai inisialisasi dan bergerak secara kontinu sesuai dengan inisialisasi. pada Gambar 5 merupakan diagram alir sistem.



Gambar 5 Diagram Alir Sistem

* + 1. **Perancangan Fungsi Merekam**

Sistem memiliki beberapa fungsi, diantaranya yaitu fungsi, diantaranya yaitu fungsi merekam, bergerak sekali, dan bergerak kontinu.

Tahapan pertama adalah membaca nilai dari inisialisasi pada *push button*, setelah itu memasuki tahap permisalan kondisi ketika. Ketika a atau *push button* pertama dipencet maka dia akan langsung membaca nilai dari a tersebut yang nantinya nilai tersebut akan digunakan di permisalan kondisi ketika. Setelah itu memasuki permisalan kondisi jika, jika a bernilai 0 maka otomatis akan berhenti, tetapi ketika a bernilai 1 maka akan memasuki fungsi selanjutnya yaitu memanggila method *recoding*. Setelah selesai menjalankan method *recording* selanjutnya akan kembali kekondisi ketika. Begitu pula kepada *push button* 2, 3, 4, dan 5

* + 1. **Perancangan Fungsi Bergerak Sekali**

Fungsi selanjutnya merupakan fungsi untuk menjalakan perintah yang sudah diinisialisasikan dan sudah disimpan kedalam EEPROM. Fungsi ini akan berjalan ketika nilai x atau inisialisasi fungsi bergerak sekali bernilai 1.

Fungsi ini menginisialisasikan bahwa bergerak sekali adalah y. ketika y bernilai tidak sama dengan 0 maka akan menuju fungsi pembacaan nilai baru dari y, setelah itu menuju fungsi berikutnya yaitu permisalan kondisi jika, jika sekali bernilai 0 maka akan menuju fungsi proses yaitu menyimpan nilai baru dari sekali, maka sekali akan bernilai 1 dan menuju fungsi pemanggilan, yaitu memanggil method *play*. Sudah dijelaskan penulis sebelumnya jika nilai dari sekali akan kembali bernilai 0 jika kondisi semua *toggle switch* bernilai 0. Dapat dilihat pada kode programbergerak sekali bahwa servo akan bergerak dimulai pada servo pertama lanjut hingga servo kelima. Hal ini dilalukan karena kode program tidak dapat berjalan secara bersamaan.

* + 1. **Perancangan Fungsi Bergerak Berkali-kali**

Fungsi berikutnya adalah fungsi bergerak berkali-kali atau gerak kontinu. Sama sepeti bergerak sekali, fungsi ini akan menjalankan perintah inisialisasi dari pengguna ketika selesai merekam.

Perbedaan dengan fungsi bergerak sekali adalah dibagian pembentukan inisialisasi baru. Di fungsi berikut hanya akan terus bergerak jika nilai dari *toggle switch* bernilai 1. Dan otomatis berhenti ketika toggle switch bernilai 0. Sama halnya dengan bergerak berkali-kali, kode program bergerak berkali-kali juga hanya dapat menggerakan servo satu persatu dimulai dari servo pertama hingga seterusnya.

* + 1. **Perancangan Fungsi *Recording***

Fungsi ini berguna untuk menyimpan nilai yang telah diberikan oleh awal pengguna yang akan disimpan di EEPROM. Fungsi ini berbentuk method baru yang bias dipanggil.

Fungsi ini ketika terpanggil akan membaca pinMode yaitu analogPin. Lalu masuk ke fungsi permisalan kondisi ketika. Ketika nilai buttonPin bernilai sesuai maka akan menuju fungsi pembacaan kembali analogPin lalu akan melakukan pencetakan nilai a. Setelah itu akan masuk ke permisalan kondisi jika, jika a kurang dari kalibrasi minimal makan nilai a menjadi nilai kalibrasi minimal, begitu pula dengan kalibrasi maksimal. Setelah itu menuju fungsi *mapping* dan penyimpanan pada EEPROM. Fungsi ini akan terus berulang hingga nilai alamat sudah 50. Setelah itu akan menjalankan fungsi permisalan kondisi jika, jika nilai alamat tidak sama dengan 25 maka akan menyimpan nilai EEPROM berupa alamat terakhir dan nilai perubahan.

* + 1. **Perancangan Fungsi *Play***

Fungsi play merupakan method yang dapat dipanggil untuk menjalankan perintah yang sudah disimpan pada EEPROM.

Fungsi ini hampir sama seperti fungsi *recording* fungsi ini berjalan hanya untuk membaca apa yang sudah ditulis atau direkam dan disimpan di EEPROM. Pembacaan tersebut peralamat dan sesuai dengan hasil *recording*.

# IMPLEMENTASI SISTEM

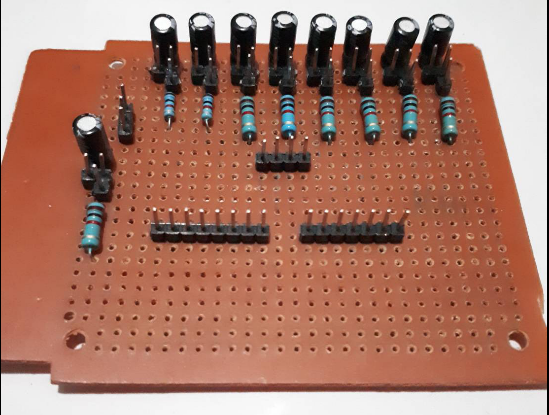
Pada Implementasi Sistem ini dijelaskan tahapan-tahapan dalam pembuatan sistem sesuai dengan perancangan sistem.

* 1. **Implementasi Sistem Perangkat Keras**

Implementasi Sistem Perangkat Keras terbagi dari beberapa subbab yang akan menjelaskan terkait pembuatan sistem sesuai dengan perancangan sistem.

* + 1. **Implementasi Rangkaian Perangkat Keras**

Sebelum memulai perakitan, mekanik membuat rangkaian RC atau rangkaian *debounce* yang berfungsi untuk membuat nilai dari *push button* dan *toggle switch* stabil atau tidak berubah-ubah nilai variabelnya. Dapat dilihat pada Gambar 6 adalah pemasangan rangkaian debounce

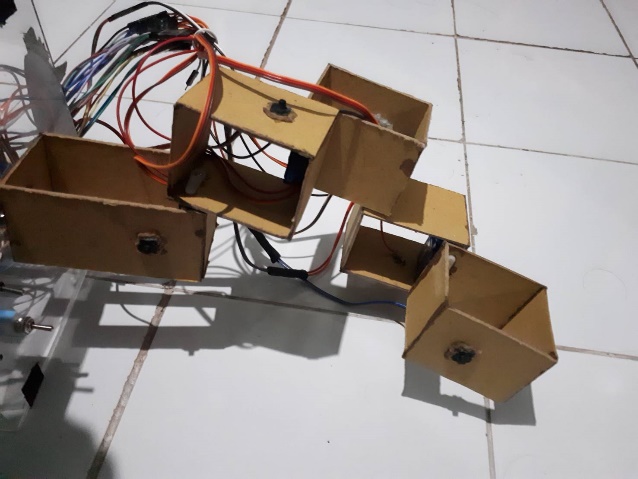


Gambar 6 Rangkaian RC/Rangakaian Debounce

Selanjutnya melakukan modifikasi pada Servo SG90. Modifikasi ini dilakukan agar mendapatkan *feedback* dari servo itu sendiri. *Feedback* ini nantinya menjadi media untuk mengirimkan nilai untuk menggerakkan servo. Gambar 7 berikut ini merupakan pembongkaran Servo SG90 untuk mencari *feedback*. Pencarian *feedback* sendiri menggunakan multimeter.

* + 1. **Implementasi Robot Lengan**

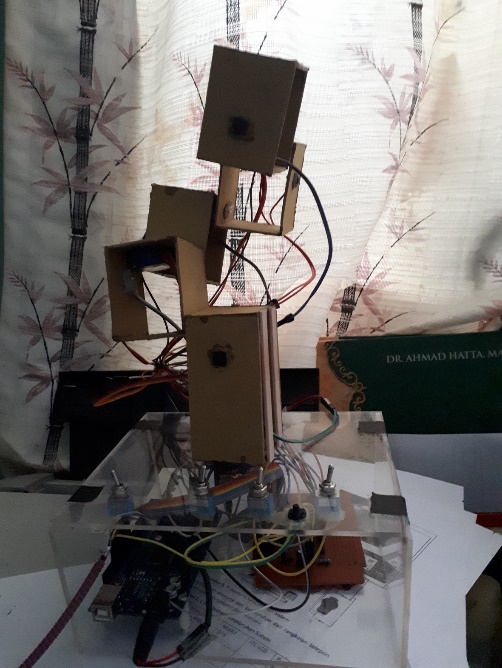
Setelah melakukan modifikasi pada servo, setelah itu melakukan implementasi lengan robot menggunakan karton duplex, lengan berbentuk balok terbuka, dengan perhitungan 6 x 4.5 cm dengan 4.5 x 4.5 cm sebanyak 4 buah dan 9 x 4.5 cm dengan 4.5 x 4.5 cm sebanyak 1 buah. Seperti Gambar 7 berikut ini merupakan bentuk balok terbuka dan sudah dipasang servo serta *push button*.



Gambar 7 Bentuk Lengan Balok Terbuka

* + 1. **Implementasi Pergerakan Lengan Robot**

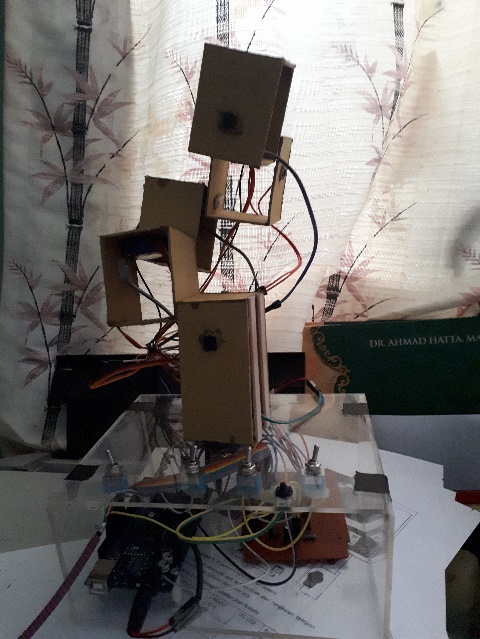
Posisi lengan robot yang telah dirancang maka diimplementasikan dan akan membentuk seperti pada Gambar 8. Pada gambar dapat dilihat bahwa posisi lengan robot berbentuk memutar. Dan pergerakan robot akan membentuk setengah bola dengan kondisi tertentu.



Gambar 8 Posisi Lengan Robot

Setelah membuat posisi lengan, dilanjutkan dengan pembuatan penampang menggunakan akrilik dengan ukuran 18 x 15 cm dan sisi penampang 7 x 15 cm. Selanjutnya penampang dipasang dengan 4 buah *toggle switch* dan sudah terpasang dengan lengan.

Gambar 9 dibawah ini merupakan hasil dari total perancangan sistem perangkat keras yang sudah dijelaskan diatas.



Gambar 9 Total Implementasi Perangkat Keras

# PENGUJIAN

Berdasarkan pengujian ketepatan pembacaan servo pada servo yang bergerak 0⁰-180⁰ yang merupakan hasil dari perbandingan nilai ADC yang dihasilkan oleh servo dan oleh multimeter. Didapatkan hasil dari pembacaan nilai ADC servo serta nilai analog multimeter ketika keadaan servo berada pada posisi sudut istimewa. Posisi sudut tersebut didapatkan dengan cara memprogram servo agar berada pada sudut tersebut setelah itu melakukan analisa hasil. Dapat diketahui nilai rata-rata error adalah 1.85 %.

Dan hasil dari pembacaan nilai ADC servo 0⁰-360⁰ serta nilai ADC multimeter saat servo berada diposisi sudut istimewa. Posisi sudut tersebut didapatkan dengan cara memprogram servo agar berada pada sudut tersebut setelah itu dilakukan analisa. Servo 0⁰-360⁰ tidak dapat menentukan titik 0⁰, jika diprogram untuk berada pada 0⁰ maka servo akan terus berputar tanpa henti. Dapat dilihat bahwa nilai rata-rata error adalah 2.32 %.

Selanjutnya pengujian pembacaan perintah 1 gerakan dengan posisi awal semua servo 90⁰ lalu digerakan dengan beberapa sudut istimewa. Posisi servo awal ditentukan dengan cara mengisialisasi pada program, setelah itu dilakukan penempatan posisi busur dan dilakukan fungsi merekam pada servo sesuai dengan derajat pada busur. Setelah perubahan sudut direkam, lengan akan dikembalikan pada posisi 90⁰, selanjutnya menjalankan program bergerak sekali dan dibandingkan posisi lengan yang sudah direkam sebelumnya dengan nilai aktual. Dapat diketahui nilai rata-rata error adalah 2.9 %.

Terakhir pengujian pembacaan perintah keseluruhan dengan posisi awal semua servo 90⁰ lalu disetiap percobaan digerakan dengan beberapa sudut istimewa. Posisi servo awal ditentukan dengan cara mengisialisasi pada program, setelah itu dilakukan penempatan posisi busur dan dilakukan fungsi merekam pada setiap servo sesuai dengan derajat pada busur. Setelah perubahan sudut direkam, lengan akan dikembalikan pada posisi 90⁰, selanjutnya menjalankan program bergerak sekali dan dibandingkan posisi lengan yang sudah direkam sebelumnya dengan nilai aktual. Dapat diketahui nilai rata-rata error adalah 1.5 %.

# KESIMPULAN

Dari keseluruhan tahap yang telah dilakukan, mulai dari tahap perancangan, tahap implementasi, hasil pengujian, dan hasil analisis, berikut adalah kesimpulan yang dapat diambil:

* + 1. Untuk merancang gerak robot berdasarkan perintah inisialisasi awal pengguna dibutuhkan perangkat keras berupa servo motor, arduino mega2560, *push button*, *toggle switch*, rangkaian *debounce* serta perangkat laptop dan perangkat lunak yaitu ardunio.
    2. Implementasi kendali robot berdasarkan perintah inisialisasi awal pengguna dapat dilakukan setelah melakukan rancangan sistem, implementasi yang dilakukan pertama adalah dengan membuat rangkaian *debounce* dilanjutkan dengan mencari *feedback* yang ada di servo, lalu pembuatan lengan robot, sebelumnya peneliti menggunakan bahan arkilik dalam pembuatan lengan robot, lalu diganti menggunakan karton duplex. Dilanjutkan dengan implementasi penampang dan pemasangan kabel.
    3. Anasilis hasil gerak servo pada kendali robot berdasarkan perintah inisialisasi awal pengguna ini dapat dihasilkan bahwa nilai ADC yang dihasilkan hampir serupa dengan nilai ADC yang diukur menggunakan multimeter. Dapat dilihat nilai rata-rata error hanya sebesar 1.85 % dan 2.32 %. Lalu pada analisis perintah 1 gerakan nilai rata-rata error hanya sebesar 2.9 % dan pada analisis perintah keseluruhan nilai rata-rata error hanya sebesar 1.5 %.

# DAFTAR PUSTAKA

Aditya Hidayat, A. N. J. a. R. E. S., 2016. Autonomous Quadruped Robot Locomotion Control Using Inverse Kinematics and Sine Pattern Methods. *The 2016 International Conference on Control, Electronics, Renewable Energy and Communications (ICCEREC).*

Biorobotics Laboratory, E., 2015. *BIOROBOTICS LABORATORY BIOROB.* [Online]   
Available at: http://biorob.epfl.ch/cheetah  
[Accessed 3 March 2017].

Biswas Palok, S. A. S., 2016. Design and Development of a 3 axes Pneumatic Robotic Arm. *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering,* V(9).

Department of Labor, U. S., 2017. *Occupational Safety and Health Administration.* [Online]   
Available at: https://www.osha.gov/dts/osta/otm/otm\_iv/otm\_iv\_4.html  
[Accessed 23 September 2017].

Dr. Bindu A Thomas, S. M. S. V. B. N. S. M., 2013. Industry Based Automatic Robotic Arm. *International Journal of Engineering and Innovative Technology (IJEIT),* II(11).

Gonzales, C. M., 2016. What's the differences between industrial robots?. *machine design.*

Jinrong Zhang, C. W. a. J. Z., 2014. The Kinematics Analysis and Configuration Optimize of Quadruped Robot. *The Open Automation and Control Systems Journal.*

K. Weinmeister1, P. E. H. W. a. A.-J. I., 2015. Cheetah-cub-S: Steering of a Quadruped Robot using Trunk Motion.

KOLLMORGEN, 2016. *Motion Control Solution.* [Online]   
Available at: https://www.kollmorgen.com/en-us/products/motors/servo/servo-motors/  
[Accessed September 2017].

Liu, C.-Y. C. a. Y.-C., 2016. Towards a Walking, Turning, and Jumping Quadruped Robot with Compliant Mechanisms. *2016 IEEE International Conference on Advanced Intelligent Mechatronics (AIM).*

Massimo Banzi, D. C. T. I. G. M. a. D. M., 2017. *Arduino.* [Online]   
Available at: arduino.cc

Md. Anisur Rahman, A. H. K. D. T. A. d. M. M. S., 2013. Design, Analysis and Implementation of a Robotic Arm- The Animator. *American Journal of Engineering Research (AJER),* II(10).

Narwane, S. O. a. V., 2014. Design,Analysis and Fabrication of Quadruped Robot with Four bar Chain Leg Mechanism. *IJISET - International Journal of Innovative Science, Engineering & Technology,* I(6).

Park, B. T., 2014. *Immersa Lab.* [Online]   
Available at: http://www.immersa-lab.com/  
[Accessed December 2017].

Putra, D. A. E., 2009. *DSP & Embedded Electronics.* [Online]   
Available at: agfi.staff.ugm.ac.id  
[Accessed 2017].

Ryosuke Kawasaki, R. S. E. K. A. M. a. M. S., 2016. Development of A Flexible Coupled Spine Mechanism For A Small Quadruped Robot. *International Conference on Robotics and Biomimetics.*

Satoshi Nishikawa1, K. S. a. Y. K., 2016. Musculoskeletal Quadruped Robot with Torque-Angle Relationship Control System. *2016 IEEE International Conference on Robotics and Automation (ICRA).*

Thing, L., 2016. *TechTarget WhatIs?.* [Online]   
Available at: http://whatis.techtarget.com/definition/initialization  
[Accessed December 2017].

Thing, L., 2017. *TechTarget Whatis.com?.* [Online]   
Available at: http://whatis.techtarget.com/definition/EEPROM-electrically-erasable-programmable-read-only-memory  
[Accessed December 2017].