

## RANCANG BANGUN ROBOT ARM BERBASIS RASPBERRY PI MENGUNAKAN PRINTER 3D

Nanang Ali Sutisna<sup>1,a</sup>, Narendra Pinandita Suwarno<sup>2,b</sup>, Asep Ropi  
Firmansyah<sup>3,c</sup>

<sup>1,2,3</sup> Program Studi Teknik Mesin, Fakultas Engineering, President University Cikarang, Bekasi,  
Indonesia

<sup>a</sup>[nanang.ali@president.ac.id](mailto:nanang.ali@president.ac.id), <sup>b</sup>pinanditan@gmail.com, <sup>c</sup>asepropipirmansyah@gmail.com

### Abstrak.

Dalam era perkembangan teknologi saat ini, robot merupakan salah satu elemen penting yang digunakan untuk mempermudah manusia dalam menyelesaikan pekerjaannya. Berbagai jenis robot digunakan dengan berbagai kegunaannya untuk membantu melakukan pekerjaan manusia, salah satunya adalah *robot arm*. Dalam projek ini, *Robot Arm* dibuat sebagai sarana edukasi dimana komponennya dibuat dengan printer 3D menggunakan material *Polylactic Acid* (PLA). *Robot arm* ini dirancang dengan menggunakan aplikasi SolidWorks 2020, lalu dengan analisa elemen hingga (FEA) dapat ditemukan bagian mana yang menjadi konsentrasi beban dan dilakukan simulasi untuk mengetahui apakah komponen robot ini aman menahan beban yang diberikan. Selanjutnya dibuat pengontrol *robot arm* dengan menggunakan mikroprosesor Raspberry Pi dan mengembangkan aplikasi perangkat seluler *smartphone* android untuk mengontrol *robot arm* melalui koneksi bluetooth. *Robot arm* ini dirancang memiliki 6 derajat kebebasan (DOF) dan aktuator yang digunakan adalah motor servo, yaitu MG996R dan SG90. Pengembangan program pengontrol *robot arm* dibuat menggunakan bahasa pemrograman Python pada Raspberry Pi, yang kemudian ditransfer ke aplikasi MIT app Inventor yang ada pada *smartphone* android untuk menggerakkan *robot arm*. Kapasitas angkat *robot arm* ini adalah 5 gram dengan ukuran maksimal 30x30x30 mm. Hasil pengujian sistem pada *robot arm* menunjukkan akurasi gerakan dengan kecepatan lambat menghasilkan persentase rata – rata error sumbu X adalah 3,23%, dan sumbu Y 3,58%. Sedangkan pada kecepatan sedang menghasilkan persentase rata – rata error sumbu X adalah 10,05%, dan sumbu Y 7,62%. dan pada kecepatan tinggi menghasilkan persentase rata – rata error sumbu X adalah 11,44%, dan sumbu Y adalah 6,58%.

**Kata kunci:** Rancang Bangun, Robot Arm, Printer 3D, PLA, Raspberry pi

### Abstract.

In the current era of technological development, robots are one of the important elements used to make it easier for humans to complete their work. Various types of robots are used with various uses to help do human work, one of them is the *robot arm*. In this study, the *Robot Arm* was made as an educational media where the components are made with a 3D printer using *Polylactic Acid* (PLA). This *robot arm* was designed using the SolidWorks 2020 application, then with finite element analysis (FEA) it can be found which part is the concentration of the load and a simulation is carried out to determine whether the components of this robot are safe to withstand the given load. Next, a *robot arm* controller was made using a Raspberry Pi microprocessor and developed an android *smartphone* mobile device application to control the *robot arm* via a bluetooth connection. This *robot arm* is designed to have 6 degrees of freedom (DOF) and the actuators used are servo motors, namely MG996R and SG90. The development of the *robot arm* controller program was made using the Python programming language on the Raspberry Pi, which was then transferred to the MIT app Inventor application on the android *smartphone* to drive the arm robot. The lifting capacity of this

*robot arm* is 5 grams with a maximum size of 30x30x30 mm. The results of system testing on the *robot arm* show the accuracy of movement at a slow speed producing an average percentage of X-axis error is 3.23%, and Y-axis is 3.58%. While at medium speed it produces an average percentage of X-axis error is 10, 05%, and the Y-axis 7.62%., and at high speed the percentage of the X-axis error is 11.44%, and the Y-axis is 6.58%.

**Keywords:** *Arm Robot, Forward Kinematics, MATLAB, Robotic Tool Kit, Design.*

## **PENDAHULUAN**

Penggunaan robot di dalam kehidupan manusia semakin meningkat dari waktu ke waktu. Robot sering digunakan untuk menangani berbagai tugas yang tidak dapat ditangani oleh manusia, seperti di bidang nuklir, kimia, perjalanan ke luar angkasa dan tugas – tugas lain yang dilakukan di lingkungan yang berbahaya. Hal ini terjadi karena robot memiliki banyak kelebihan yang tidak dimiliki manusia, diantaranya: menghasilkan output yang sama ketika mengerjakan suatu pekerjaan secara berulang-ulang, tidak mudah lelah, ketelitian dan kecepatan menyelesaikan tugas, dapat diprogram ulang sehingga dapat difungsikan untuk beberapa tugas yang berbeda, dan lebih sedikit melakukan kesalahan dibandingkan manusia.

Berbagai variasi robot yang diciptakan oleh manusia dapat membantu melakukan tugas – tugas yang tidak dapat atau sulit dilakukan oleh manusia, salah satunya adalah *robot arm*. *Robot arm* biasanya digunakan untuk mengambil dan memindahkan objek/barang. Pada umumnya *robot arm* dapat melakukan dua gerakan yaitu gerakan berputar dan gerakan memanjang atau memendek. *Robot arm* memiliki dua sisi yang digunakan dalam melakukan pergerakan tersebut. Salah satu sisi yang disebut poros ditanam pada bidang yang statis dan sisi lain disebut ujung (*end effector*) yang dapat dimuati dengan alat pemegang tertentu sesuai dengan tugas robot. Umumnya alat pemegang yang digunakan adalah *gripper* yang dapat difungsikan sebagai penjepit atau pencengkram objek yang akan diambil. Selain itu pergerakan *robot arm* dapat diatur sesuai dengan kebutuhan yang diinginkan, baik itu sudut putaran *robot arm* atau pun jarak jangkauan *robot arm* [1].

Beberapa penelitian telah dilakukan dalam perancangan dan pengendalian *robot arm*, antara lain menggunakan AVR Atmega 8535 [1] dan Arduino Uno [2,3], baik dikendalikan melalui kabel ataupun nirkabel misalnya melalui bluetooth [3]. Dalam proyek ini, perancangan *robot arm* sederhana dilakukan menggunakan perangkat lunak Solidworks dengan mempertimbangkan proses pembuatannya yaitu dengan printer 3D dan menggunakan material plastik PLA (Polylactic Acid) yang ramah lingkungan, sedangkan pengendaliannya menggunakan mikroprosesor Raspberry Pi. Proyek ini bertujuan membuat prototipe *robot arm* yang ringan dengan sedikit komponen serta material yang ringan, aman dan ramah bagi lingkungan, yang bisa digunakan sebagai sarana edukasi. Dalam makalah ini pula akan dihitung berapa beban maksimal yang bisa diangkat oleh robot saat posisi menjangkau terjauh (*maximum stretch*).

*Robot arm* ini menggunakan konfigurasi *Articulated Robot* dimana robot memiliki gerakan yang alamiah seperti gerakan tangan sehingga akan membuat robot lebih mudah berinteraksi dengan objek atau lingkungannya. *Robot arm* mempunyai 6 motor penggerak pada masing masing persendian yang digerakan dengan motor servo mikro yang ditenagai oleh listrik DC bertegangan 5V dan dikontrol dengan mikroprosesor Raspberry Pi 3 model B+, dan *gripper* sebagai penjepit atau

pencengkram objek/barang. Robot ini bekerja secara otomatis sesuai dengan instruksi yang telah diprogram dan disimpan.

## TINJAUAN PUSTAKA

Secara umum robot adalah sebuah alat mekanik yang dapat melakukan tugas fisik, baik menggunakan pengawasan dan kontrol manusia, ataupun menggunakan program yang telah didefinisikan terlebih dahulu (kecerdasan buatan). Istilah robot berasal dari bahasa Cheko “Robota” yang berarti pekerja atau kuli yang tidak mengenal lelah atau bosan. Robot sejak awal dibuat dengan berbagai tugas untuk dapat menggantikan pekerjaan manusia dalam melakukan pekerjaan yang berbahaya, membosankan dan berulang.

Dalam dunia industri umumnya robot digunakan sebagai perangkat automasi yang biasa digunakan dalam lini produksi yaitu untuk melakukan pekerjaan berulang seperti proses perakitan, pengelasan, pengecatan, handling/transfer material atau barang dan pengepakan, sebagaimana terlihat dalam Gambar 1 di bawah ini.



**Gambar 1. Robot digunakan di lini produksi**

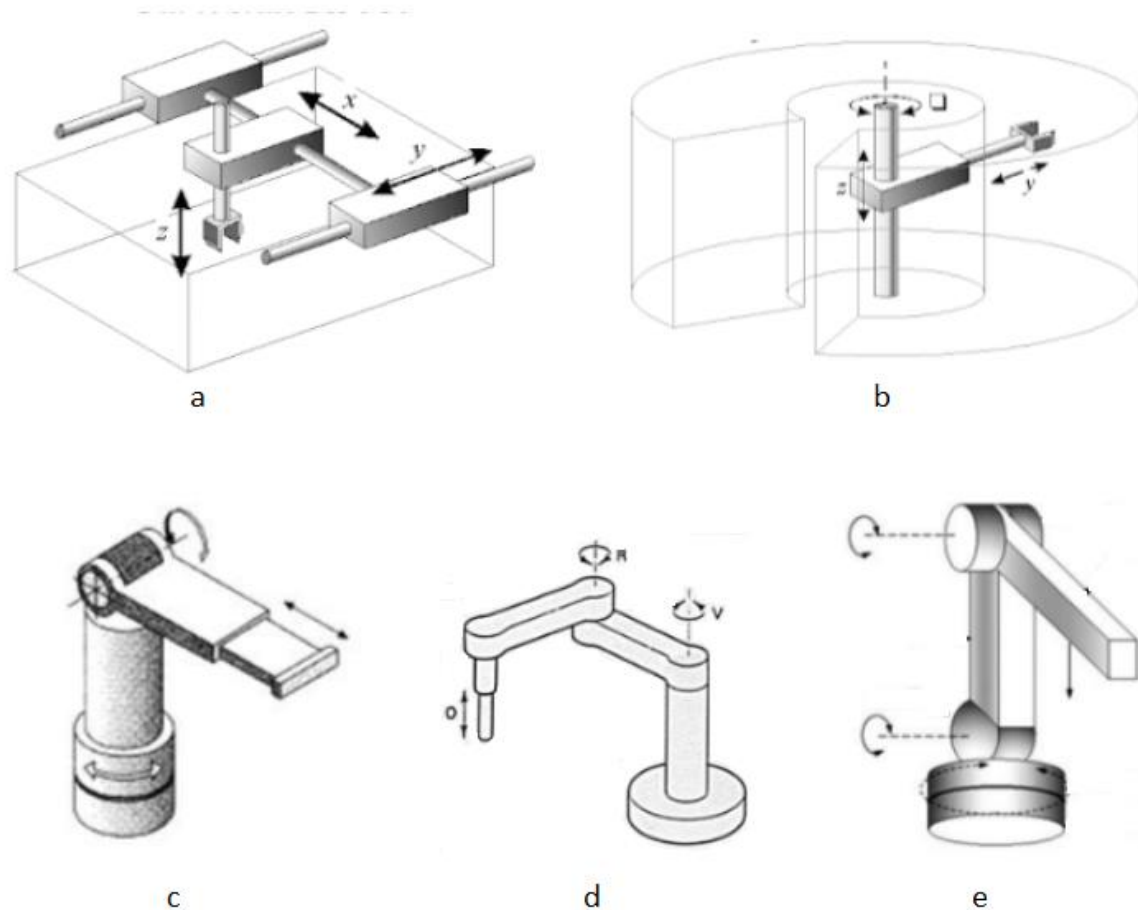
Selain banyak digunakan dalam dunia industri, penggunaan robot juga cukup banyak dalam dunia kedokteran atau robot medis, antara lain robot bedah. Robot ini ada di sebagian besar *telem manipulator*, yang menggunakan aktivator ahli bedah di satu sisi untuk mengontrol *efektor* di sisi lain.

Penggunaan robot lainnya adalah untuk pekerjaan pelayanan sederhana atau pekerjaan berulang yang biasa dilakukan oleh manusia di rumah, restoran, bandara. Seperti halnya robot *telepresence* dalam dunia medis, robot ini juga biasanya mempunyai kemampuan mobilitas, berkomunikasi, dan tugas handling sederhana.

### ***Konfigurasi Robot***

Dalam klasifikasinya ada beberapa jenis konfigurasi robot yang umum dipakai, beberapa diantaranya digerakan secara elektronik, hidrolis dan pneumatic. Setiap jenis konfigurasi mempunyai area dan ruang lingkup kerjanya masing-masing sehingga penting bagi kita melihat satu persatu setiap konfigurasi untuk bisa disesuaikan dengan area atau ruang lingkup dimana robot tersebut akan digunakan.

Secara umum terdapat lima konfigurasi robot yang biasa digunakan di dunia industri yaitu: *Cartesian Robot*, *Cylindrical Robot*, *Spherical Robot*, *Articulated Robot* dan *SCARA ( Selectively Compliant Assembly Robot Arm )* [4]. Berbagai konfigurasi robot tersebut dapat dilihat pada Gambar 2.



**Gambar 2. Konfigurasi robot (a) Cartesian, (b) Cylindrical, (c) Spherical, (d) SCARA, (e) Articulated**

Konfigurasi *articulated* banyak digunakan pada *robot arm*, terdiri dari beberapa bagian yang semuanya tersambung oleh *revolute joint*. Konfigurasi seperti ini biasa digunakan dalam lini proses manufaktur untuk melakukan berbagai macam pekerjaan seperti mengangkat, pengelasan, memindahkan material, dan masih banyak lagi pekerjaan yang dapat dilakukan. Kemampuan diatas diperoleh dengan mengadaptasi pergerakan alamiah dari anatomi lengan manusia. [5]

Konfigurasi *robot arm* dipilih karena konfigurasinya yang sangat sebguna, dibantu dengan *grip* dan pergelangan tangan pada ujung lengan yang dapat berputar ke kanan dan ke kiri, walau jika dibandingkan dengan konfigurasi robot lainnya robot ini mempunyai susunan konfigurasi paling rumit, dikarenakan setiap sendi pada robot ini mempunyai derajat kebebasannya sendiri namun robot ini menawarkan fleksibilitas, jangkauan area dan daya guna yang tinggi diantara konfigurasi robot yang lain.

### **Aktuator**

Aktuator adalah komponen penggerak yang jika dilihat dari prinsip penghasil geraknya dapat dibagi menjadi tiga bagian yaitu penggerak berbasis motor listrik (motor servo, motor stepper, motor DC dan sebagainya), penggerak pneumatik (berbasis kompresi: udara, nitrogen) dan penggerak hidrolis (berbasis benda cair: minyak pelumas) [6].

Pada project ini, motor servo digunakan sebagai penggerak lengan-lengan robot dan ditempatkan pada sendi-sendi. Motor servo adalah motor DC yang memiliki system *close loop* dan mempunyai keluaran sebuah poros dengan torsi besar. Poros ini dapat dikontrol melalui modulasi lebar pulsa (PWM, *Pulse Width Modulation*). Poros dapat diatur untuk bertahan pada posisi tertentu, berputar

searah jarum jam atau sebaliknya. Motor servo terdiri dari servo continuous yang dapat berputar 360° dan servo standar yang dapat berputar 180° [7].

### ***Mikroprosesor***

Mikroprosesor sering disebut CPU (*Central Processing Unit*), merupakan elemen kontrol pada sistem komputer. Sistem Mikroprosesor, yaitu suatu sistem yang didalamnya terdapat mikroprosesor atau komponen I/O, dan memori yang mana sering disebut juga komputer. Sedangkan Mikrokontroler adalah mikrokomputer chip tunggal yang didalamnya mengandung unit mikroprosesor, memori, I/O (*Input/Output*), ADC (*Analog to Digital Converter*), *timer*, *clock* dan lain-lain.

Salah satu contoh mikroprosesor adalah Raspberry Pi, dan contoh mikrokontroler adalah Arduino. Pada proyek ini penulis menggunakan Raspberry Pi, dimana Raspberry pi digunakan untuk menjalankan program, untuk di perintah dan dikontrol, Raspberry pi sendiri bahkan dapat memiliki *operating system*, sehingga dapat berfungsi layaknya komputer seperti yang kita tahu. Mulai dari membuat spreadsheet, pengolah kata, menyimpan file, memutar music, memutar video, *browsing internet*, bahkan memainkan game layaknya di komputer [8].

Raspberry Pi adalah komputer seukuran kartu kredit berbiaya rendah yang dihubungkan ke monitor komputer atau TV, menggunakan keyboard dan mouse standar. Ini adalah perangkat kecil yang mampu memungkinkan orang dari segala usia untuk menjelajahi komputasi dan untuk belajar bagaimana memprogram dalam bahasa seperti Scratch dan Python. Ini mampu melakukan semua yang diharapkan dari komputer desktop, mulai dari menjelajah internet dan memutar video kualitas tinggi, hingga membuat spreadsheet, pengolah kata dan bermain game [10].

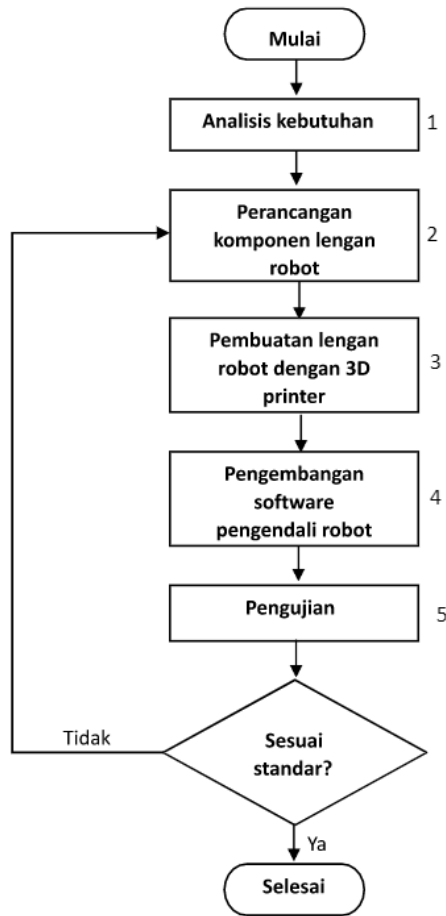
### ***MIT App Inventor***

*MIT App Inventor* adalah aplikasi web sumber terbuka yang awalnya dikembangkan oleh Google, dan saat ini dikelola oleh *Massachusetts Institute of Technology* (MIT).

*MIT App Inventor* membantu penulis membuat aplikasi Android untuk sistem kendali robot. MIT App Inventor menggunakan antarmuka grafis, serupa dengan antarmuka pengguna pada *Scratch* dan *StarLogo TNG*, yang memungkinkan pengguna untuk *men-drag-and-drop* obyek visual untuk menciptakan aplikasi yang bisa dijalankan pada perangkat Android. Dalam menciptakan App Inventor, Google telah melakukan riset yang berhubungan dengan komputasi edukasional dan menyelesaikan lingkungan pengembangan *online* Google [11].

## **METODE**

Metode yang digunakan dalam studi ini akan ditunjukkan dengan diagram alir proses pada Gambar 3 di bawah ini.

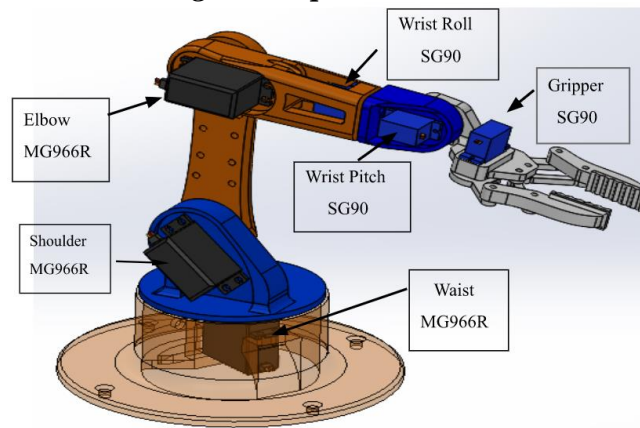


Gambar 3. Diagram alir proses kajian

1. Analisis Kebutuhan

Bagian pertama yang dilakukan saat merencanakan proyek ini adalah dengan menganalisa kebutuhan yang disesuaikan dengan tujuan dari awal perancangan hingga proses pencetakan. Dengan menganalisa bagaimana robot ini akan bekerja, karena ide dari design ini adalah dengan mengadopsi prinsip kerja lengan manusia, dan beberapa kesamaan fungsi seperti memindahkan, menggenggam, mengambil, maka konfigurasi yang paling cocok digunakan untuk robot ini adalah konfigurasi *articulated robot*.

2. Perancangan Komponen Robot Arm



Gambar 4. Rancangan Robot Arm dan penempatan motor servo

Rancangan *robot arm* ini terbuat dari beberapa komponen atau *link* yaitu: base, waist, shoulder, elbow, wrist, dan gripper. Antara *link* satu dengan *link* lainnya dihubungkan oleh persendian yang disebut *joint*. Penempatan posisi Motor Servo pada *robot arm* ditunjukkan pada gambar 4. Analisis kekuatan komponen pada *robot arm* ini dilakukan menggunakan software analisis element hingga (FEA, *Finite Element Analysis*) pada SolidWork 2020, berikut adalah parameter-parameter yang digunakan untuk menjalankan simulasi tersebut, seperti material properties, mesh, bagian yang digunakan sebagai fixture dan pada bagian apa saja force/loads diberikan sesuai besarnya masing-masing.

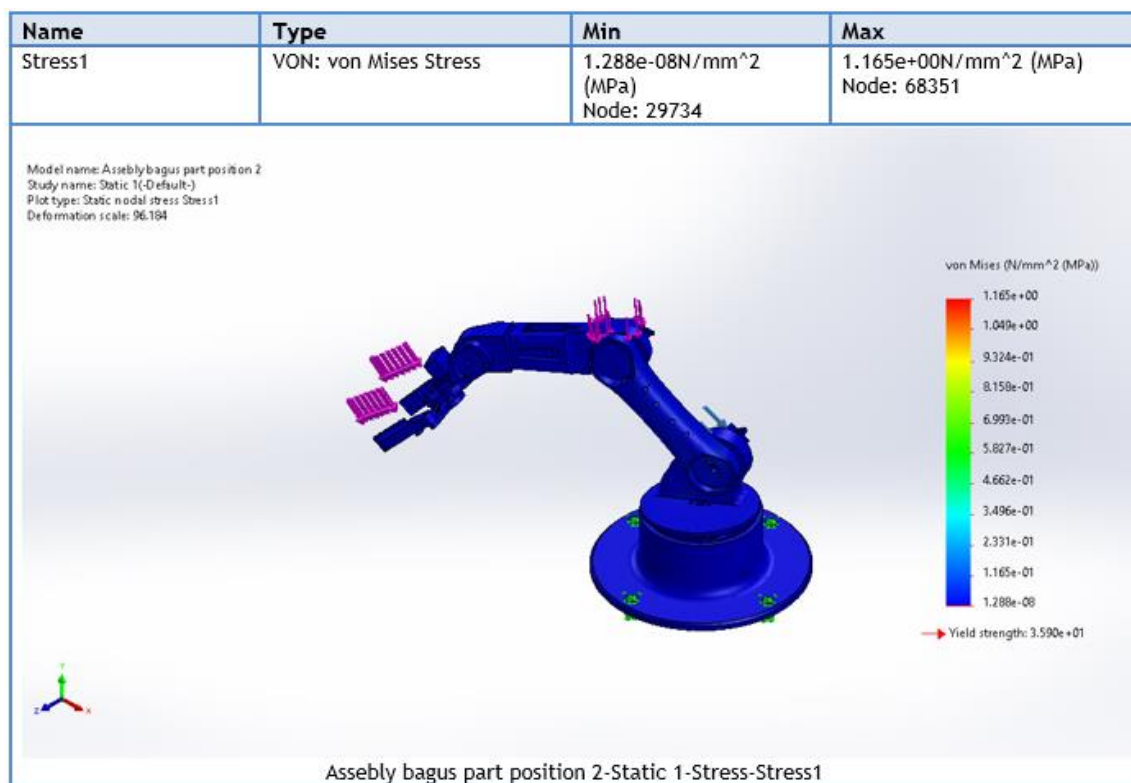
Tabel 2. Material Properties PLA [12]

<i>Mechanical Properties</i>	
Name:	Polylactic Acid
Model type:	Linear Elastic Isotropic
Default failure criterion:	Max von Mises Stress
Yield strength:	2.608e+07 N/m <sup>2</sup>
Tensile strength:	3.293e+07 N/m <sup>2</sup>
Elastic modulus:	2e+09 N/m <sup>2</sup>



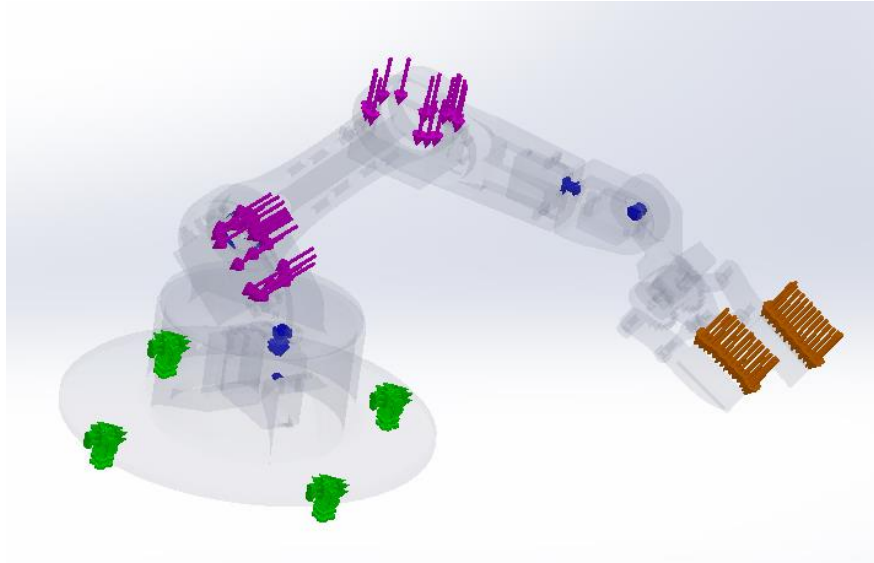
Poisson's ratio:	0.394
Mass density:	1,020 kg/m <sup>3</sup>
Shear modulus:	3.189e+08 N/m <sup>2</sup>

Menurut rencana, robot akan mengangkat beban sebesar 5 gram atau sama dengan 0.05 N maka besaran beban tersebut yang diberikan pada kedua bagian ujung *grip* yang berinteraksi langsung dengan benda tersebut. Berikutnya adalah beban yang berasal dari berat motor pada sendi-sendi antara *Shoulder* dan *Fore Arm* serta antara *Fore Arm* dan *Arm*, masing-masing sebesar 50g atau 0.5 N.



**Gambar 5. Von Mises Stress maksimum 1,165 MPa**

Membandingkan Von Mises stress dari hasil simulasi FEA sebesar 1,165 MPa dengan yield strength PLA sebesar 26 MPa, maka dapat disimpulkan bahwa komponen *robot arm* ini aman. Dengan menggunakan fitur *Design Insight* bisa dilihat bagian mana saja yang secara umum menerima beban paling banyak dari design ini. Gambar 6 dibawah menunjukkan area-area *critical*.



**Gambar 6. identifikasi area kritis dengan fitur Design Insight**

### **3. Pembuatan Robot Arm dengan Printer 3D**

Dalam tahap pembuatan yang termasuk didalamnya adalah pemodelan, *slicing*, dan pencetakan. Untuk proses-proses tersebut membutuhkan software, hardware, dan material yang akan dijelaskan seperti berikut:

#### **Software**

- SOLIDWORKS 2020, Dalam proses pemodelan *robot arm* penulis menggunakan Software SOLIDWORKS 2020, software perancangan yang biasa digunakan untuk menggambar, simulasi machining, rendering dan fitur-fitur simulasi lainnya dalam konsep engineering.
- Ultimaker Cura, adalah aplikasi yang digunakan untuk mengubah model 3D menjadi lapisan atau *layer* (*proses ini disebut slicing*), kemudian mengubah parameter pencetakan kedalam kode G, yang dimana sebelumnya telah diatur posisi pencetakan bagian, kecepatan pencetakan dan durasi pencetakan.

#### **Hardware**

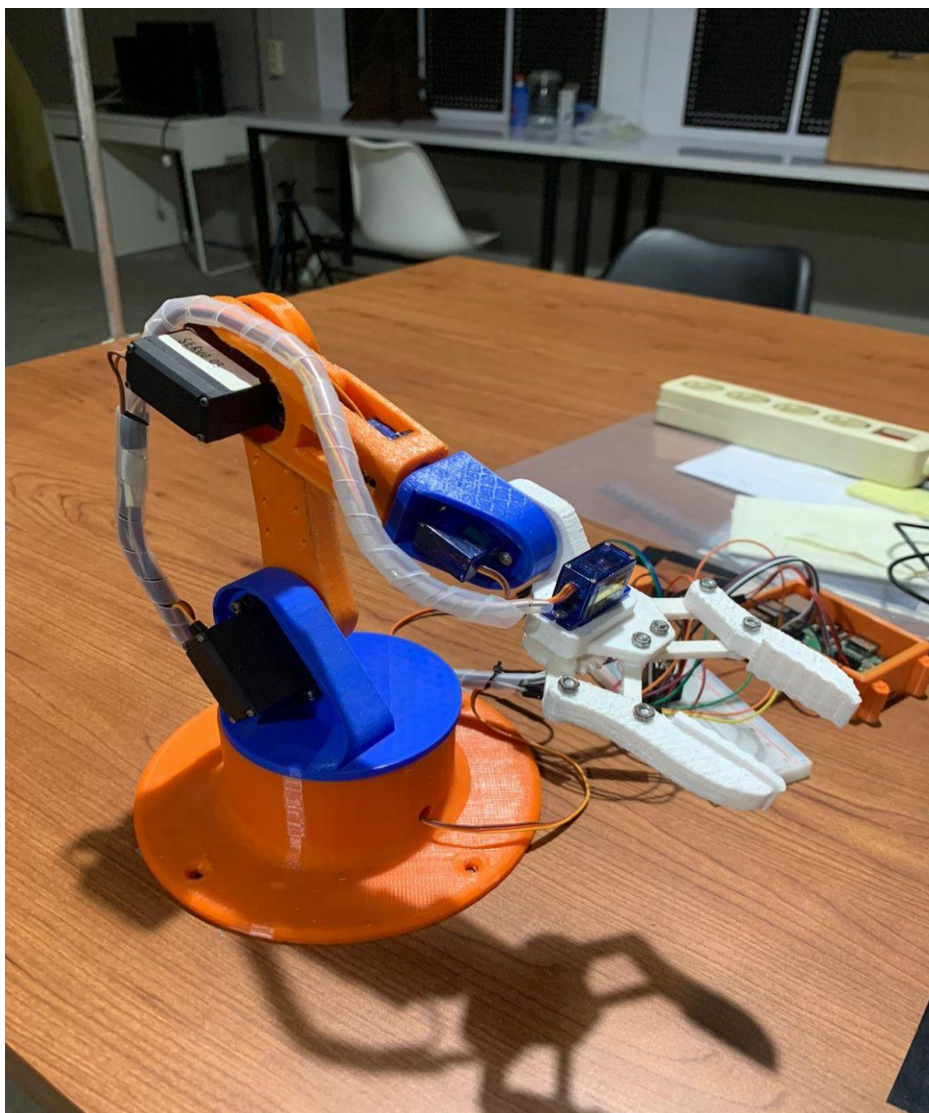
Printer 3D Creality CR-10S yang memiliki 3 sumbu yang dibuat dengan frame aluminium dan alas kaca selebar 300mm x 300mm, diatur dengan control box di samping mesin dan filamen holder terpasang di atasnya.

#### **Material**

Filament PLA (Polylactic Acid) digunakan pada proyek ini karena kemudahannya dan hasil cetak tidak melengkung dan *crack* dikarenakan penyusutannya yang sedikit, selain itu material ini adalah material yang *biodegradable* sehingga lebih mudah terurai di alam dan aman bagi lingkungan.

Komponen-komponen *robot arm* yang telah selesai dicetak kemudian dirakit dan dipasangi motor servo, seperti terlihat pada Gambar 6.

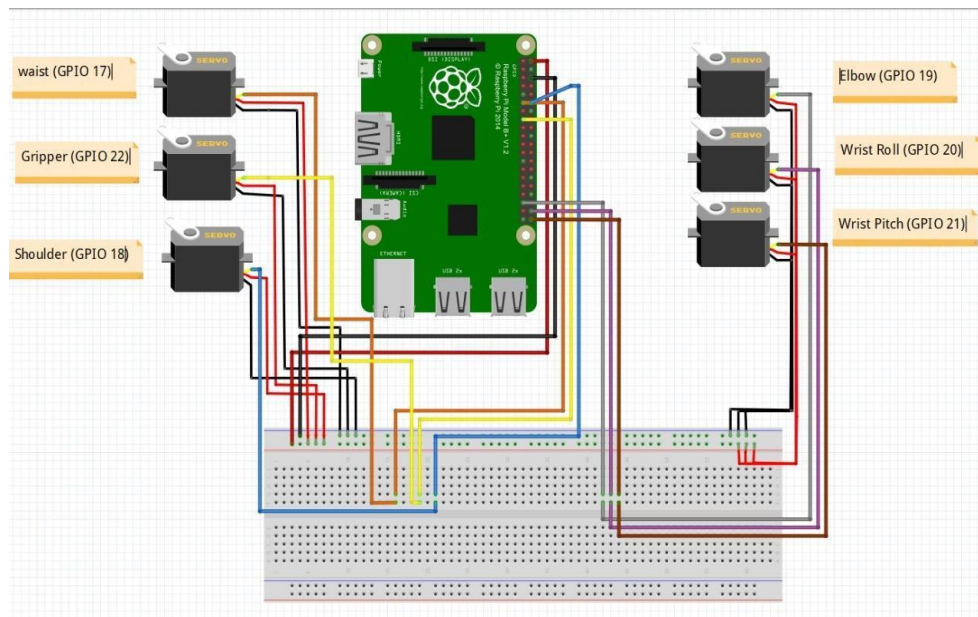




**Gambar 7. Hasil perakitan *robot arm* lengkap dengan motor servo**

### ***Rangkaian Sistem Robot Arm***

Merupakan bagian yang digunakan dalam rangkaian kontroler *robot arm*, seperti menyambungkan Servo, Raspberry Pi, dan catu daya. Untuk detailnya ditunjukkan pada Gambar dibawah 7 ini.



Gambar 8. Rangkaian Raspberry Pi

#### 4. Pengembangan Software Pengendali Robot Arm

Software pengendali *robot arm* dibuat menggunakan bahasa pemrograman Python. Yang dilakukan adalah mendefinisikan semua servo dan juga Bluetooth, termasuk juga memasukan library serial untuk komunikasi serial Bluetooth dan library serial servo yang disertakan dalam pemrograman Python. Dan juga perlu mendefinisikan variabel untuk posisi Servo, dan array untuk mode otomatis serta loop untuk pemrograman instruksi berulang.

Pada *set up* awal membuat perintah awal dengan membuat *library* sebagai dasar perintah untuk menginisiasi perintah program selanjutnya.

```

1  #!/user/bin/python
2
3
4  # import library yang dibutuhkan
5  import serial
6  from signal import pause
7  from time import sleep
8  import pigpio
9

```

Gambar 9. Set up awal

Selanjutnya melakukan koneksi ke bluetooth dengan fungsi status koneksi Bluetooth hingga terkoneksi. Pada kodingan ini jika serial Bluetooth tidak terkoneksi maka status dari Bluetooth akan mengeluarkan tulisan not connected.

```
# Wait until bluetooth device is connected
def bluetooth_status():
    while True:
        try:
            ser = serial.Serial("/dev/rfcomm0", baudrate=9600, timeout=3)
            break
        except:
            status = 'not connected'

bluetooth_status()
# Sleep time- Normal Mode
delay = .001
# Sleep time- Replay Mode
sPeed = [.001]
ser = serial.Serial(
    port='/dev/rfcomm0',
    baudrate=9600)
# Send Greeting and state to arduino device.
ser.write("Hi Boss, I am at your service :)\n".encode())
sleep(4)
ser.write("Manual Mode\n".encode())
# list to store "SAVE" Trigger
save = [0]
```

Gambar 10. Fungsi koneksi Bluetooth

Berikutnya melakukan penyimpanan dari masing-masing posisi servo, dilanjutkan dengan membaca posisi akhir yang ada pada file masing-masing servo. Berikut di bawah ini koding untuk inisiasi servo di pin GPIO dengan format penomoran BCM.

```
73 # servo motors GPIO PINs assignment
74 servo01 = 17 #(Waist)
75 servo02 = 18 # (Shoulder)
76 servo03 = 19 # (Elbow)
77 servo04 = 20 # (Wrist Roll)
78 servo05 = 21 #(Wrist Pitch)
79 servo06 = 22 # (Gripper)
80
```

Gambar 11. Posisi Pin Servo

Koding untuk setting tiap pin GPIO yang terkoneksi sebagai output dapat dilihat pada Gambar 12, sedangkan setting frekuensi pada Bluetooth untuk setiap servo terlihat pada Gambar 13.

```
80
81 # Set tiap pin GPIO yang terkoneksi sebagai output
82 pwm = pigpio.pi()
83 pwm.set_mode(servo01, pigpio.OUTPUT)
84 pwm.set_mode(servo02, pigpio.OUTPUT)
85 pwm.set_mode(servo03, pigpio.OUTPUT)
86 pwm.set_mode(servo04, pigpio.OUTPUT)
87 pwm.set_mode(servo05, pigpio.OUTPUT)
88 pwm.set_mode(servo06, pigpio.OUTPUT)
89
```

Gambar 12. Setting pin GPIO

```
91 # "bluedot.btcomm BluetoothServer" Setup
92 pwm.set_PWM_frequency(servo01, 50)
93 pwm.set_PWM_frequency(servo02, 50)
94 pwm.set_PWM_frequency(servo03, 50)
95 pwm.set_PWM_frequency(servo04, 50)
96 pwm.set_PWM_frequency(servo05, 50)
97 pwm.set_PWM_frequency(servo06, 50)
98
```

Gambar 13. Frekuensi Bluetooth

Kemudian inisiasi posisi awal tiap servo. Dari sini mulai program utama untuk menentukan arah atau gerakan pada saat *robot arm* dijalankan secara manual atau dijalankan secara otomatis melalui *Smartphone*. Ketika menekan tombol save maka script dari servo akan dijalankan, kemudian file dijadikan tipe data string lalu di enter, dan serial koneksi Bluetooth dari masing-masing posisi servo akan diekstrak dengan koneksi serial 600 – 2400.

Servo satu dan servo yang lainnya bergerak dengan menyimpan variabel sementara. Jika posisi servo lebih dari posisi yang sebelumnya, maka nilai servo dikurangi satu. Dan jika posisi servo kurang dari posisi yang sebelumnya, maka nilai servo ditambah satu, dan program ini berjalan seterusnya sampai servo yang ke 6.

Selanjutnya fungsi program untuk *auto run*. Kemudian menyambungkan Bluetooth serial dari raspberry pi dengan android, dan nilai yang dimasukkan pada servo01 ke dalam array, yang kemudian dikirim melalui signal pada Bluetooth.

Kemudian setting speed control yang telah diatur pada aplikasi arm robot dan dijalankan pada Raspberry pi. Dan dengan program yang ada pada scratch MIT app Inventor. Dilakukan berulang-ulang pada setiap servo. Selanjutnya di return, ini adalah akhir dari semua pemilihan speed kontrol, maka dilanjutkan kembali ke script berikutnya.

```
# Speed control
if data[2:4] == 'ss' and len(data) > 3:
    Delay = float(data[4:-3])
    sPeed[0] = (Delay - range_in_min) * \
        (range_out_max - range_out_min) / \
        (range_in_max - range_in_min) + \
        range_out_min
    pwm.set_servo_pulsewidth(servo01, j)
    j -= 1
    sleep(sPeed[0])
```

**Gambar 14. Speed control**

### *Program Aplikasi Smartphone*

Program aplikasi Smartphone dibuat menggunakan MIT app Inventor. Di dalam tahap ini penulis akan menjelaskan program aplikasi smartphone. Dibagian atas antarmuka, ada dua tombol *connect* dan *disconnect* untuk menghubungkan smartphone ke Bluetooth. Di bagian sisi kiri terdapat 7 slider, yang 6 diantaranya untuk menggerakkan servo, dan satu lagi untuk *speed*. Dibawahnya terdapat *saved positions* untuk melihat berapa banyak posisi yang disimpan, dan dibawahnya lagi terdapat 4 tombol *SAVE*, *RUN*, *STOP*, dan *RESET* untuk mode otomatis.

### **5. Pengujian Sistem Robot Arm**

Setelah selesai membuat semua desain program dan aplikasi serta desain perancangan sistem *robot arm*, penulis melakukan proses pengujian sistem. Parameter yang digunakan dalam pengujian sistem ini adalah berasal dari program Raspberry Pi dan aplikasi Smartphone, yang terdiri dari dua jenis pengujian sistem yaitu: pengujian akurasi dan *repearepeatability*.

Pengujian *repeatability* dilakukan dengan menggerakkan *robot arm* dari satu posisi ke posisi yang lain yang diinginkan, dan mengukur perpindahan pada target. Pengujian ini dilakukan sebanyak 5 kali, kemudian hasilnya dibandingkan dengan data posisi target sebelumnya.

Pengujian akurasi gerakan dilakukan dengan kecepatan yang berbeda. Pengujian ini adalah dengan menggerakkan *robot arm* dari satu posisi ke posisi yang lain yang diinginkan, dengan kecepatan bergerak yang pertama di setting dengan kecepatan lambat (20%), kemudian yang kedua di setting dengan kecepatan sedang (50%), dan terakhir di setting dengan kecepatan cepat (100%), lalu

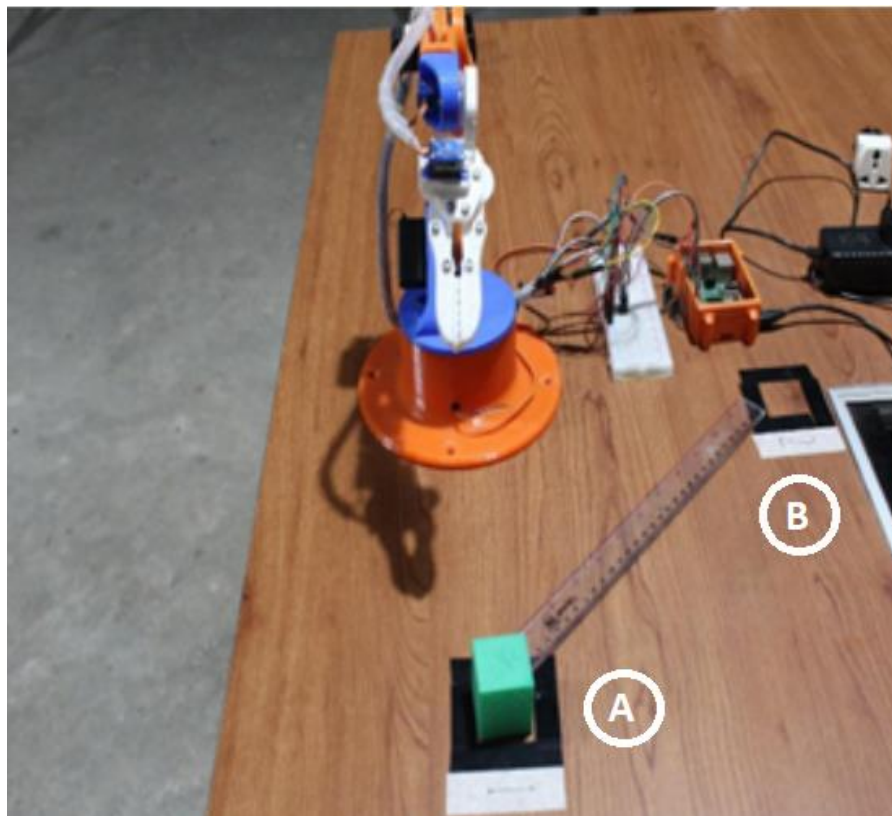


mengukur perpindahan pada target untuk mendapatkan nilai error. Hasil tes akan ditampilkan pada bab berikutnya.

**HASIL DAN PEMBAHASAN**

Komponen-komponen *robot arm* yang dibuat dengan printer 3D dan telah dirakit diuji coba. *Robot arm* ini memiliki 6 derajat kebebasan, tapi hanya ada 5 servo yang mempengaruhi posisi *robot arm*, yaitu waist, shoulder, elbow, dan wrist. Servo terakhir digunakan untuk gripper. Berikut ini merupakan rakitan dan uji tes dari *robot arm*.

Hasil pengujian akurasi dan repeatability dirangkum dalam Tabel 3. Pada gambar di bawah ini jarak dari titik A ke titik B adalah 300 mm.



Gambar 23. Jarak titik A ke titik B

Tabel 3. Hasil tes pergerakan *robot arm* kecepatan 20, 50, dan 100

Kecepatan	Test ke	Posisi awal [mm]		Posisi akhir [mm]		Persentasi error	
		X	Y	X	Y	X	Y
Rendah (20)	1	173	173	170	168	1,73%	2,89%
	2	173	173	167	185	3,46%	6,93%
	3	173	173	176	175	1,75%	1,15%
	4	173	173	176	175	1,75%	1,15%
	5	173	173	160	183	7,51%	5,78%
<b>Rata-rata error</b>						<b>3,28%</b>	<b>3,68%</b>
Sedang (50)	1	173	173	204	149	19,07%	13,87
	2	173	173	175	178	1,15%	2,89%
	3	173	173	198	190	14,45%	9,89%
	4	173	173	178	175	2,89%	1,15%
	5	173	173	151	191	12,71%	10,40%

Rata-rata error						10,06%	7,62%
Tinggi (100)	1	173	173	183	164	5,78%	5,20%
	2	173	173	136	191	21,36%	10,40
	3	173	173	180	163	8,67%	5,78%
	4	173	173	145	186	15,60%	7,51%
	5	173	173	183	166	5,78%	4,04%
Rata-rata error						11,44%	6,58%

Dari data diatas dapat diketahui nilai persentase error perpindahan *robot arm* dari titik A ke titik B pada kecepatan rendah, yang paling rendah dari sumbu X ditunjukkan pada percobaan nomor 1, 3, dan 4 dengan nilai 1,73%, dan persentase error paling tinggi ditunjukkan pada percobaan nomor 5 dengan nilai 7,51%. kemudian nilai persentase error yang paling rendah dari sumbu Y ditunjukkan pada percobaan nomor 3 dan 4 dengan nilai 1,15%, dan persentase error paling tinggi ditunjukkan pada percobaan nomor 2 dengan nilai 6,93%. Sedangkan nilai rata-rata error persentase dari sumbu X adalah 3,23%, dan nilai rata-rata error persentase dari sumbu Y adalah 3,58%.

Pada kecepatan sedang, nilai persentase error perpindahan *robot arm* dari titik A ke titik B yang paling rendah dari sumbu X ditunjukkan pada percobaan nomor 2 dengan nilai 1,15%, dan persentase error paling tinggi ditunjukkan pada percobaan nomor 1 dengan nilai 19,07%. Kemudian nilai persentase error yang paling rendah dari sumbu Y ditunjukkan pada percobaan nomor 4 dengan nilai 1,15%, dan persentase error paling tinggi ditunjukkan pada percobaan nomor 1 dengan nilai 13,87%. Sedangkan nilai rata-rata error persentase dari sumbu X adalah 10,05%, dan nilai rata-rata error persentase dari sumbu Y adalah 7, 62%.

Sedangkan pada kecepatan tinggi, nilai persentase error perpindahan *robot arm* dari titik A ke titik B yang paling rendah dari sumbu X ditunjukkan ditunjukkan pada percobaan nomor 1 dan 5 dengan nilai 5,78%, dan persentase error paling tinggi ditunjukkan pada percobaan nomor 2 dengan nilai 21,38%. Kemudian nilai persentase error yang paling rendah dari sumbu Y ditunjukkan pada percobaan nomor 5 dengan nilai 4,04%, dan persentase error paling tinggi ditunjukkan pada percobaan nomor 2 dengan nilai 10,40%. Sedangkan nilai rata-rata error persentase dari sumbu X adalah 11,44%, dan nilai rata-rata error persentase dari sumbu Y adalah 6,58%.

Penyebab error yang lebih besar pada sumbu X kemungkinan adalah karena adanya kelembaman yang dipengaruhi oleh beban *robot arm* dan jarak yang jauh, sedangkan yang menyebabkan error lebih besar pada sumbu Y adalah dipengaruhi oleh 2 motor servo sehingga nilai error yang dihasilkan akan menjadi dua kali lipat.

Dari serangkaian data percobaan diatas dapat disimpulkan bahwa nilai persentase error perpindahan *robot arm* dari titik A ke titik B yang paling rendah dari sumbu X ditunjukkan pada percobaan dengan kecepatan sedang (50) dengan nilai 1,15%, dan yang paling tinggi ditunjukkan pada percobaan dengan kecepatan tinggi (100) dengan nilai 21,38%. Kemudian nilai persentase error perpindahan yang paling rendah dari sumbu Y ditunjukkan pada percobaan dengan kecepatan sedang (50) dan lambat (20) dengan nilai 1,15%, dan yang paling tinggi ditunjukkan pada percobaan dengan kecepatan sedang (50) dan tinggi (100) dengan nilai 10,40%.

Nilai rata-rata error yang paling rendah dari sumbu X ditunjukkan pada percobaan dengan kecepatan lambat (20) dengan nilai 3,23%, dan yang paling tinggi ditunjukkan pada percobaan dengan kecepatan tinggi (100) dengan nilai 11,44%. Nilai rata – rata error yang paling rendah dari sumbu Y ditunjukkan pada percobaan dengan kecepatan lambat (20) dengan nilai 3,58%, dan yang paling tinggi ditunjukkan pada percobaan dengan kecepatan sedang (50) dengan nilai 7,62%.

Selain itu, pada saat uji coba diketahui bahwa gerakan yang paling lancar dan beraturan dari percobaan gerakan diatas ditunjukkan oleh kecepatan lambat (20) dan sedang (50), sedangkan gerakan yang paling tidak beraturan ditunjukkan oleh kecepatan tinggi (100).

Berdasarkan data dari hasil uji tes, kita dapat mengetahui persentase nilai rata-rata error perpindahan *robot arm* yang paling rendah pada sumbu X ditunjukkan pada percobaan dengan kecepatan lambat



(20) dengan nilai 3,23%, sedangkan persentase nilai rata-rata error perpindahan *robot arm* yang paling tinggi pada sumbu X ditunjukkan pada percobaan dengan kecepatan tinggi (100) dengan nilai 11,44%. Dan persentase nilai rata-rata error perpindahan *robot arm* yang paling rendah pada sumbu Y adalah pada percobaan kecepatan lambat (20) dengan nilai 3.58%, sedangkan yang paling tinggi pada percobaan kecepatan sedang (50) dengan nilai 7,62%.

Pada uji coba ditemukan juga terkadang posisi gerakan servo pada *robot arm* tidak presisi atau melebihi target posisi yang seharusnya, sehingga servo tidak bisa digerakan, cara ini diatasi dengan mereset koneksi yang terhubung dengan aplikasi yang ada pada smartphone, dan melepas USB yang terhubung dengan Raspberry pi, kemudian dikoneksikan kembali.

## KESIMPULAN DAN SARAN

### Kesimpulan

Dari pembahasan pada bagian sebelumnya dapat diambil kesimpulan sebagai berikut:

1. *Robot arm* ini menggunakan konfigurasi *articulated robot* yang mempunyai fleksibilitas yang tinggi sehingga mudah digunakan untuk berbagai macam interaksi sesuai dengan program yang digunakan.
2. Dari hasil analisa Finite Element yang dilakukan dapat diperoleh data bahwa rancangan *robot arm* ini mempunyai nilai Von Mises sebesar 1.54 MPa yang dimana nilai yield strength yang dimiliki material Polylactic Acid adalah 26 MPa, oleh karena itu robot ini dapat dikategorikan tahan terhadap beban yang diaplikasikan.
3. Software pengendali telah berhasil dibuat menggunakan bahasa pemrograman Python dan MIT App Inventor, serta telah diuji coba dapat melaksanakan fungsinya dengan baik.
4. Pengujian akurasi dan repeatability telah dilakukan dengan persentase nilai error rata-rata perpindahan yang paling rendah pada sumbu X dan Y adalah pada percobaan kecepatan lambat (20) dengan nilai 3,23% dan 3,58%, sedangkan error yang paling tinggi pada sumbu X pada percobaan kecepatan cepat atau tinggi (100) dengan nilai 11,44%, dan sumbu Y pada kecepatan sedang (50) dengan nilai 7,62%. Sehingga disarankan untuk menggunakan kecepatan rendah (20).
5. Terkadang posisi gerakan servo tidak presisi, sehingga servo tidak bisa digerakan, dengan mereset dan melepas USB yang terhubung dengan Raspberry pi dan dikoneksikan kembali dapat menjadi solusinya.

### Saran-saran

1. Karena hasil dari analisa beban robot ini masih jauh dibawah batas maksimal yang diijinkan, maka di kemudian hari robot ini bisa menggunakan beban yang lebih berat dari beban yang sekarang digunakan.
2. Untuk penelitian lebih lanjut, disarankan untuk mengembangkan lagi kemampuan *robot arm* yang didukung Raspberry pi dan aplikasi bahasa pemrograman, agar dapat mengendalikan robot melalui internet.
3. Agar pergerakan *robot arm* bisa lebih lancar dan meminimalisir terjadinya kesalahan, dapat ditambahkan sensor dan kontrol aktuator menggunakan close loop.

## DAFTAR PUSTAKA

- [1] R. Arismarjito, "Robot Lengan Otomatis Sebagai Pemisah Barang Berdasarkan Warna Dengan Menggunakan ATMEGA8535," Skripsi, Universitas Sanata Dharma, Yogyakarta, 2011.
- [2] R. A. Pamungkas, "Analisis Penerapan Arm Robot 4 Degree of Freedom (DOF) Sebagai Alat Penyortir Buah Otomatis," Skripsi, Politeknik Negeri Sriwijaya, 2017.

- [3] M. I. Satria, "Development of Robotic Arm Controller Using Arduino Microcontroller and Mobile Device Application," Skripsi, President University, Cikarang, 2021.
- [4] F.G. Ohoirat, et al., "Rancang Bangun Sistem Mekanik Penggiring Dan Penendang Pada Robot Sepak Bola." *Jurnal Teknik Elektro dan Komputer* 8.3, 2019, pp. 171-180.
- [5] Mikell P. Groover., "Automation, Production Systems, and Computer-Integrated Manufacturing, Third Edition". Pearson Education, Inc., Upper Saddle River, NJ, 2008
- [6] A. Wibowo, "Prototipe Robot Manipulator Sendi Lengan (Joint – Arm) Berbasis Arduino Uno Pada Sistem Pemilah Barang," Skripsi, Universitas Nasional, 2020.
- [7] M. Didi, E. D. Marindani and A Elbami. "Rancang Bangun Pengendalian Robot Lengan 4 DOF dengan GUI (Graphical User Interface) Berbasis Arduino Uno." *Jurnal Teknik Elektro Universitas Tanjungpura*, pp. 1-11, 2015.
- [8] A. Sandi, "Pemanfaatan Raspberry Pi Sebagai Server Portable," Skripsi, UIN Alauddin, Makassar, 2016.
- [9] Spesifikasi Raspberry Pi. Tersedia di <https://Raspberrypi.org/help/what-%20is-a-raspberry-pi/>. [Diakses 10 Desember 2021].
- [10] Raspberry Pi. Tersedia di <https://www.raspberrypi.com/products/raspberry-pi-3-model-b-plus/> [Diakses 6 Oktober 2021]
- [11] Y. Efendi. "Rancangan Aplikasi Game Edukasi Berbasis Mobile Menggunakan App Inventor." *Jurnal Intra-Tech*, Vol. 2, No. 1: 40-48, 2018.
- [12] S. R. Subramaniam., et al. "Preliminary investigations of polylactic acid (PLA) properties." AIP conference proceedings. Vol. 2059. No. 1. AIP Publishing LLC, 2019.