

# **KENDALI LENGAN ROBOT BERBASIS ANDROID UNTUK OTOMASI *LIFTING* BARANG**

Tesis S-2

Untuk memenuhi persyaratan memperoleh gelar Magister Teknik  
Program Studi Magister Teknik Elektro



Diajukan oleh  
Eko Sulistyono  
MTE20601700030

**PROGRAM STUDI MAGISTER TEKNIK ELEKTRO  
FAKULTAS TEKNOLOGI INDUSTRI  
UNIVERSITAS ISLAM SULTAN AGUNG  
SEMARANG  
2021**

**TESIS**  
**KENDALI LENGAN ROBOT BERBASIS ANDROID**  
**UNTUK OTOMASI *LIFTING* BARANG**

Yang dipersiapkan dan disusun oleh  
Eko Sulistyono  
MTE20601700030

Telah dipertahankan di depan Dewan Penguji  
Pada tanggal 24 Desember 2021

Susunan Dewan Penguji

Pembimbing Utama



Arief Marwanto, ST., M.Eng., Ph.D.  
NIDN: 0626097501

Ketua Penguji



Dr. Hj. Sri Arttini Dwi P., M.Si  
NIDN: 0620026501

Pembimbing Pendamping



Ir. Suryani Alifah M.T., Ph.D.  
NIDN: 0625036901

Penguji 1



Dr. Ir. H. Muhamad Haddin, MT  
NIDN: 0618066301  
Penguji 2



Dr. Ir. Suhartono, M.Eng.  
NIDN: 0323035903

Tesis ini telah diterima sebagai salah satu persyaratan  
Untuk memperoleh gelar Magister Teknik  
Tanggal 24 Desember 2021

Ketua Program Studi Magister Teknik Elektro



Novi Marlyana, S.T., MT  
NIDN: 0015117601

## MOTTO

**“SESUNGGUHNYA ALLAH TIDAK AKAN MENGUBAH NASIB  
SUATU KAUM SEHINGGA MEREKA MENGUBAH KEADAAN  
YANG ADA PADA DIRI MEREKA SENDIRI”**

**(Q.S AR-RA’D: 11)**

**“BARANG SIAPA KELUAR UNTUK MENCARI ILMU MAKA  
DIA BERADA DI JALAN ALLAH ”**

**(HR.TURMUDZI)**

**“HIDUPLAH SEAKAN KAMU MATI BESOK, BELAJARLAH  
SEAKAN KAMU AKAN HIDUP SELAMANYA”**

**(MAHATMA GANDHI)**



## PERSEMBAHAN



Tesis ini kupersembahkan kepada:

1. Almarhum Ayah dan Ibu terhormat
2. Istri dan Anak-anakku tersayang
3. Almamaterku Universitas Islam Sultan Agung (UNISULLA)  
Semarang

## PERNYATAAN KEASLIAN TESIS

Saya yang bertanda tangan dibawah ini:

Nama : Eko Sulistyono  
NIM : MTE.20601700030  
Program Studi : Magister Teknik Elektro  
Fakultas : Teknologi Industri

Dengan ini saya menyatakan bahwa Tesis yang diajukan kepada Program Studi Magister Teknik Elektro dengan Judul :

### “KENDALI LENGAN ROBOT BERBASIS ANDROID UNTUK OTOMASI *LIFTING* BARANG”

Adalah hasil karya sendiri, judul tersebut belum pernah diajukan untuk memperoleh gelar Magister Teknik pada Program Studi Magister Teknik Elektro Universitas Islam Sultan Agung (UNISSULA) ataupun pada universitas lain serta belum pernah ditulis maupun diterbitkan oleh orang lain kecuali secara tertulis diacu, disitasi dan ditunjuk dalam daftar pustaka. Tesis ini adalah milik saya, segala bentuk kesalahan dan kekeliruan dalam Tesis ini adalah tanggung jawab saya.

Semarang, 24 Desember 2021

Penulis



Eko Sulistyono

MTE.20601700030

## KATA PENGANTAR

Puji syukur kehadirat Tuhan Yang Maha Esa, sehingga penulis dapat menyelesaikan laporan Tugas Akhir (Tesis) dengan judul “**Kendali Lengan Robot Berbasis Android Untuk Otomasi Lifting Barang**” laporan ini merupakan salah satu syarat kelengkapan dalam menyelesaikan Program Studi Magister Teknik Elektro, Fakultas Teknologi Industri Universitas Islam Sultan Agung (UNISSULA) Semarang.

Pada pembuatan laporan ini penulis menyadari benar bahwa masih banyak kekurangan, baik penulisan maupun dalam isinya. Untuk itu penulis sangat mengharapkan adanya kritik dan saran yang sifatnya membangun dari semua pihak demi sempurnanya laporan ini.

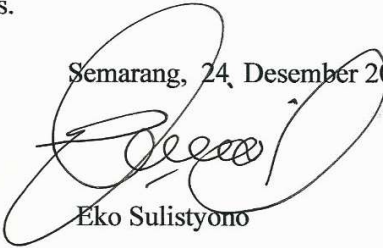
Penulis menyadari bahwa keberhasilan laporan ini tidak terlepas dari bantuan berbagai pihak. Untuk itu penulis mengucapkan banyak terima kasih kepada yang terhormat:

1. Bapak Drs. H. Bedjo santoso, M.T., PH.D. Rektor Universitas Islam Sultan Agung (UNISULLA) Semarang.
2. Ibu Dr. Ir. Hj. Novi Marlyana, S.T., M.T. Dekan Fakultas Teknologi Industri dan Ketua Jurusan Magister Teknik Elektro Universitas Islam Sultan Agung (UNISULLA) Semarang.
3. Bapak Arief Marwanto, S.T.,M.Eng., Ph.D selaku dosen pembimbing I dalam penyelesaian tesis ini.
4. Ibu Ir. Suryani Alifah M.T., Ph.D. selaku dosen pembimbing II dalam penyelesaian tesis ini.
5. Bapak dan Ibu Dosen Magister Teknik Elektro Universitas Islam Sultan Agung (UNISULLA) Semarang
6. Rekan-rekan yang telah banyak membantu dalam menyelesaikan laporan ini.

Sekali lagi penulis mengucapkan banyak terima kasih atas bantuan dan bimbingannya sehingga penulis dapat menyelesaikan laporan tesis ini.

Penulis berharap kiranya laporan tesis ini dapat bermanfaat bagi pembaca, terutama yang melaksanakan penulisan tesis.

Semarang, 24 Desember 2021



Eko Sulistyono



## DAFTAR ISI

HALAMAN JUDUL.....	i
HALAMAN PENGESAHAN.....	ii
MOTTO .....	iii
PERSEMBAHAN.....	iv
HALAMAN PERNYATAAN.....	v
KATA PENGANTAR .....	vi
DAFTAR ISI.....	viii
DAFTAR TABEL.....	x
DAFTAR GAMBAR .....	xi
DAFTAR LAMPIRAN.....	xii
ABSTRAK.....	xiii
ABSTRACT.....	xiv
<b>BAB I PENDAHULUAN.....</b>	<b>1</b>
1.1. Latar Belakang.....	1
1.2. Rumusan Masalah.....	2
1.3. Batasan Masalah .....	3
1.4. Tujuan Penelitian .....	3
1.5. Manfaat Penelitian.....	4
1.6. Keaslian Penelitian .....	4
<b>BAB II TINJAUAN PUSTAKA DAN LANDASAN TEORI.....</b>	<b>5</b>
2.1. Tinjauan Pustaka.....	5
2.2. Landasan Teori .....	7
2.2.1. Lengan Robot .....	7
2.2.2. Servo SG 90.....	8
2.2.3. PWM .....	11
2.2.4. Arduino Mega.....	12
2.2.5. Belt Konveyor .....	13
2.2.6. Motor DC Gearbox.....	13
2.2.7. Sensor Infrared FC-51 .....	14

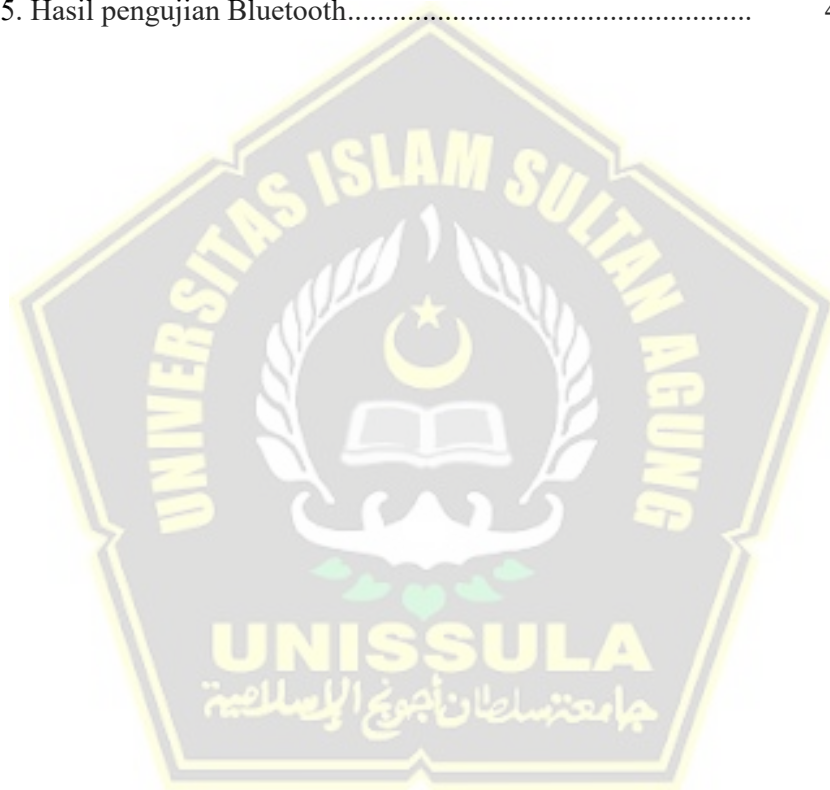


2.2.8. Modul SSC 32 .....	15
2.2.9. Modul Bluetooth HC-05.....	17
2.2.10. Android .....	18
2.2.11. Catu Daya.....	20
<b>BAB III METODE PENELITIAN .....</b>	<b>21</b>
3.1. Model Penelitian.....	21
3.2. Flowchart Program .....	22
3.3. Flowchart Lengan Robot .....	24
3.4. Materi Penelitian .....	26
3.4.1. Hardware .....	26
3.4.2. Software .....	26
3.5. Alat yang Digunakan.....	29
3.6. Langkah-Langkah Penelitian.....	31
3.7. Parameter yang Akan Dianalisis .....	34
3.8. Kesulitan-Kesulitan .....	34
<b>BAB IV HASIL PENELITIAN DAN PEMBAHASAN.....</b>	<b>35</b>
4.1. Hasil Perancangan Perangkat Keras.....	35
4.2. Hasil Pengujian Belt Konveyor .....	36
4.3. Pengujian Modul Infrared FC-51 .....	37
4.4. Hasil Pengujian Motor Servo .....	37
4.5. Hasil Pengujian Modul Bluetooth .....	40
4.6. Hasil Analysis Pengujian Prototype .....	42
<b>BAB V KESIMPULAN</b>	
5.1. Kesimpulan.....	45
5.2. Saran.....	46

## LAMPIRAN

## DAFTAR TABEL

Tabel 2.1. Bagian Modul SSC_32 .....	16
Tabel 3.1. PIN Arduino Mega 2560.....`	31
Tabel 4.1. Bagian <i>prototype</i> beserta fungsinya.....	36
Tabel 4.2. Hasil Pengujian sensor infrared .....	37
Tabel 4.3. hasil pengujian motor servo saat pengambilan benda.....	40
Tabel 4.4. Hasil pengujian motor servo saat <i>lifting</i> benda.....	40
Tabel 4.5. Hasil pengujian Bluetooth.....	42



## DAFTAR GAMBAR

Gambar 2.1. Lengan robot .....	8
Gambar 2.2. Motor servo SG-90.....	9
Gambar 2.3. Modul arduino Mega.....	11
Gambar 2.4. Belt Konveyor .....	12
Gambar 2.5. Motor DC gearbox .....	13
Gambar 2.6. Sensor Infrared FC-51.....	14
Gambar 2.7. Layout PCB Modul SSC-32.....	15
Gambar 2.8. Modul SSC-32.....	16
Gambar 2.9. Modul Bluetooth HC-05.....	17
Gambar 2.10. Power Supply dalam bentuk jadi.....	18
Gambar 3.1. Architectur Sistem Model .....	21
Gambar 3.2. Diagram Flowchart.....	22
Gambar 3.3. Flowchart lengan robot .....	24
Gambar 3.4. Diagram Blok Sistem hardware .....	27
Gambar 3.5. Wiring Diagram Motor Servo Lengan Robot .....	32
Gambar 3.6. Wiring Diagram bluetooth HC_51 .....	33
Gambar 3.7. Wiring Diagram Modul Infrared FC-51 .....	33
Gambar 3.8. Prototype Alat .....	34
Gambar 4.1. Hasil Perancangan Keseluruhan Alat.....	35
Gambar 4.2. Program sudut Motor Servo Standby.....	38
Gambar 4.3. Program Sudut Motor servo .....	39
Gambar 4.4. Tampilan aplikasi pada Smarphone Android.....	41
Gambar 4.5. Grafik Sudut Motor Servo.....	43
Gambar 4.6. Grafik Waktu Gerak Motor Servo.....	44

## DAFTAR LAMPIRAN

LAMPIRAN 1 PROGRAM PADA ARDUINO

LAMPIRAN 2 PROGRAM BASIC 4 ANDROID



## Abstrak

Kebanyakan pemindahan benda di atas konveyor yang mempunyai beberapa ukuran sekaligus masih dilakukan secara manual belum otomatis sehingga masih butuh campur tangan tenaga manusia. Penelitian ini memfokuskan tentang kendali lengan robot berbasis android untuk otomasi *lifting* barang. Model ditentukan sebagai sebuah *lift* barang pada konveyor secara otomatis berdasar ukuran bendanya. Parameter yang ditentukan antara lain adalah benda dengan ukuran kecil, sedang dan besar. Untuk ukuran kecil  $2 \leq x < 3 \text{ cm}^3$ , sedang  $3 \leq x < 4 \text{ cm}^3$  dan besar  $x \geq 4 \text{ cm}^3$ . Prototype ini menggunakan tiga buah lengan robot dengan menggunakan 4 DOF (*Degree of Freedom*) motor servo. Dimana masing-masing lengan robot mengambil benda untuk ukuran kecil, sedang dan besar. Metode yang digunakan dalam penelitian ini yaitu menggunakan metode android dengan menggunakan software arduino IDE untuk pemrograman arduino mega 2560 serta Basic 4 android untuk pemrograman menggunakan perangkat android. Fokus yang dianalisis dalam penelitian ini yaitu ketepatan sudut dan waktu gerak motor servo lengan robot mulai dari persiapan, pengambilan sampai dengan meletakkan benda. Pengukuran sudut serta waktu gerak motor servo dilakukan dengan dua cara yaitu perhitungan secara teori serta pengukuran secara langsung. Hasil pengujian menunjukkan bahwa metode android mampu digunakan untuk mengendalikan lengan robot dalam melakukan *lifting* barang secara otomatis. Hasil pengujian pergerakan sudut motor servo lengan robot didapatkan nilai rata-rata error sudut pergerakan motor servo pada lengan robot 1 adalah 2,5 % lengan robot 2 adalah 1,8 % serta lengan robot 3 adalah 2,2 %. Sedangkan hasil pengujian waktu gerak motor servo terlihat bahwa nilai rata-rata error untuk lengan robot 1 sebesar 4,1 %, lengan robot 2 sebesar 4,8 % serta lengan robot 3 sebesar 3,9 %.

Kata Kunci : Kendali Lengan Robot, *Lifting* Barang, Android

## Abstract

Most of moving objects on conveyors that have several sizes at once is still done manually, not automatically, so it still requires human intervention. This research focuses on the control of an android-based robot arm for the automation of *lifting* goods. The model is defined as a freight *elevator* on a conveyor automatically based on the size of the object. Parameters determined include objects with small, medium and large sizes. For small size  $2 \times 3 \text{ cm}^3$ , medium  $3 \times 4 \text{ cm}^3$  and large  $4 \times 4 \text{ cm}^3$ . This prototype uses three robotic arms using 4 DOF ( *Degree of Freedom* . ) servomotors. Where each robotic arm picks up objects for small, medium and large sizes. The method used in this research is using the android method using Arduino IDE software for programming Arduino Mega 2560 and Basic 4 Android for programming using Android devices. The focus analyzed in this study is the accuracy of the angle and timing of the motion of the robotic arm servo motor from preparation, retrieval to placing objects. The measurement of the angle and time of the servo motor motion is carried out in two ways, namely theoretical calculations and direct measurements. The test results show that the android method can be used to control the robot arm in carrying out *lifting* goods automatically. The results of testing the angular movement of the servo motor of the robotic arm obtained the average error of the angle of movement of the servo motor on the 1st robot arm is 2.5%, the robot arm 2 is 1.8% and the robot arm 3 is 2.2%. While the test results of the servo motor motion time show that the average error value for robot arm 1 is 4.1%, robot arm 2 is 4.8% and robot arm 3 is 3.9%.

Keywords : Robotic Arm Control, Goods *Lifting* , Android

# BAB I

## PENDAHULUAN

### 1.1. Latar Belakang

Pada zaman dahulu, untuk memindahkan atau mengangkat sebuah barang dari tempat satu ke tempat yang lain dibutuhkan tenaga manusia yang cukup banyak, maka untuk sekarang ini hal tersebut sepertinya dirasa kurang efisien. Hal tersebut dilatar belakangi oleh tingkat kualitas produksi, serta efisiensi waktu. Untuk itulah sejalan dengan perkembangan teknologi otomatis yang begitu pesat khususnya dalam dunia industri, maka diciptakan robot – robot otomatis yang dikendalikan oleh teknologi komputer [1].

Disisi lain kemajuan teknologi dalam beberapa dasa warsa ini berkembang sangatlah pesat bagai cendawan yang tumbuh dimusim hujan. Perkembangan teknologi mampu merevolusi dunia sehingga terjadi perubahan yang luar biasa terhadap tatanan kehidupan manusia dalam menjalankan segala aspek kehidupannya. Revolusi Industri 4.0 merupakan fenomena yang mengkolaborasikan teknologi *cyber* dan teknologi otomatisasi. Revolusi Industri 4.0 dikenal juga dengan istilah “*cyber physical system*”. Konsep penerapannya berpusat pada otomatisasi dibantu dengan teknologi informasi dalam proses pengaplikasiannya sehingga keterlibatan tenaga manusia dalam prosesnya dapat berkurang. Dengan demikian, efektivitas dan efisiensi pada suatu lingkungan kerja dengan sendirinya bertambah. Dalam dunia industri, hal ini berdampak signifikan pada kualitas kerja dan biaya produksi. Namun sesungguhnya, tidak hanya industri, seluruh lapisan masyarakat juga bisa mendapatkan manfaat umum dari sistem ini [2].

Pada umumnya teknologi robot merupakan peralatan mekanik yang dikendalikan oleh mikrokontroler yang sudah disematkan program kedalamnya sesuai dengan fungsi yang akan dikehendaki terhadap aplikasi robot tersebut. Disisi lain teknologi yang digunakan pada *smartphone* saat ini berkembang sangatlah pesat dimana di dalamnya sudah disematkan teknologi berbasis komputer dan telah mampu dihubungkan secara nirkabel dengan

berbagai perangkat lainnya atau bisa dikatakan telah mampu berkomunikasi dengan peralatan lain tanpa menggunakan media penghantar berupa kabel. Atas dasar tersebut penulis tertantang untuk mengkolaborasikan fungsi atau pengendalian lengan robot menggunakan perangkat *smartphone* berbasis android sehingga bisa dikendalikan dari jarak jauh.

Banyak penelitian sebelumnya tentang pengambilan benda menggunakan lengan robot, misalkan pada jurnal rancang bangun robot lengan pemindah dan penyeleksi barang berdasarkan warna berbasis arduino uno [3]. Dimana dalam penelitian tersebut penyortiran benda berdasarkan warna benda yang berjalan di atas konveyor. Salah satu kelemahan dari penelitian ini adalah terjadinya salah pembacaan warna ketika cahaya kurang terang. Disamping itu sistem ini sangat sulit diaplikasikan untuk benda yang mempunyai aneka ragam warna. Ada juga jurnal [4] pengendali lengan robot dengan mikrokontroler arduino berbasis *smartphone*. Yang mana dalam penelitian tersebut pengendalian sudah dilakukan dari jarak jauh menggunakan perangkat *smartphone* berbasis android, tetapi benda yang diambil tidak berjalan di atas konveyor tetapi benda diam.

*Prototype* kendali lengan robot dibuat dalam penelitian ini untuk otomasi *lifting* barang berdasarkan ukuran berbasis android. Kelebihan dari penelitian ini adalah pengambilan barang berdasarkan ukuran dilakukan oleh tiga buah lengan robot dengan tugas masing-masing lengan robot mengambil benda dengan ukuran yang berbeda-beda sehingga memudahkan proses penyortiran benda. Selain itu pengoperasiannya bisa dilakukan dari jarak jauh, karena sudah dikendalikan dengan perangkat *smartphone* berbasis android.

## 1.2. Rumusan Masalah

Berdasarkan latar belakang masalah yang diuraikan di atas, maka rumusan masalah dalam penelitian ini adalah sebagai berikut:

1. Bagaimana merancang kendali lengan robot berbasis android untuk otomasi *lifting* barang?



2. Bagaimana optimasi parameter kendali lengan robot berbasis android untuk otomasi *lifting* barang?
3. Bagaimana merealisasikan dan menguji *prototype* kendali lengan robot berbasis android untuk *lifting* barang?

### 1.3. Batasan Masalah

Berikut batasan-batasan yang dilakukan terhadap penelitian ini:

1. Masukan berupa benda yang mempunyai tiga kategori ukuran yaitu jika ukuran benda adalah  $x$  maka  $2 \leq x < 3 \text{ cm}^3$  maka dikategorikan kecil, jika  $3 \leq x < 4 \text{ cm}^3$  ukuran benda dikategorikan sedang dan jika  $x \geq 4 \text{ cm}^3$  ukuran benda dikategorikan besar.
2. Lengan robot dengan kontrol Arduino Mega 2560 dan SSC 32 yang berfungsi untuk mengambil benda yang berjalan di atas konveyor yang digunakan yaitu jenis lengan robot dengan 4 DOF (*Degree of Freedom*).
3. Handphone berbasis android untuk mengendalikan kerja *lifting* benda pada adalah handphone dengan versi android 10 ke atas.
4. Konveyor yang berfungsi untuk menggerakkan benda yang akan diangkat oleh lengan robot adalah konveyor jenis *belt* konveyor.

### 1.4. Tujuan Penelitian

Adapun tujuan yang ingin dicapai adalah :

1. Merancang kendali lengan robot berbasis android untuk otomasi *lifting* barang.
2. Melakukan optimasi parameter kendali lengan robot berbasis android untuk otomasi *lifting* barang.
3. Membuat dan menguji *prototype* lengan robot berbasis android untuk *lifting* barang.

### 1.5. Manfaat Penelitian

Adapun manfaat yang diharapkan dalam penelitian ini adalah :

1. Memberikan kemudahan dalam pengoperasian *lifting* benda berdasar ukuran menggunakan lengan robot.
2. Memberikan ketepatan yang tinggi dalam memisahkan benda berdasarkan ukurannya.

#### 1.6. Keaslian Penelitian

Berdasarkan penelusuran serta tinjauan pustaka yang telah dilakukan terkait dengan penelitian kendali lengan robot untuk otomasi *lifting* barang berdasarkan ukuran pada konveyor berjalan berbasis android, maka didapatkan beberapa penelitian yang berkaitan, yaitu :

1. Penelitian rancang bangun robot lengan pemindah dan penyeleksi barang berdasarkan warna berbasis Arduino uno [3]. Pada penelitian tersebut sensor yang digunakan adalah sensor warna dan sensor cahaya atau LDR sedang pada penelitian kami sensor yang digunakan menggunakan sensor infrared.
2. Penelitian aplikasi pengenalan objek untuk lengan robot pemisah benda berdasarkan bentuk benda[5]. Pada penelitian tersebut benda yang di ambil tidak berjalan di atas konveyor serta tidak dikendalikan oleh android.
3. Pengendali Lengan Robot Dengan Mikrokontroler Arduino Berbasis Smartphone[4]. Pada penelitian tersebut benda yang di ambil tidak berjalan di atas konveyor serta tidak dikendalikan oleh android

Berdasarkan kajian-kajian terhadap penelitian di atas, terdapat beberapa perbedaan dengan penelitian ini, antara lain: tema, implementasi alat, serta penggunaan alat yang telah disebutkan secara spesifik pada kajian-kajian penelitian tersebut.

## **BAB II**

### **TINJAUAN PUSTAKA DAN LANDASAN TEORI**

#### 2.1. Tinjauan Pustaka

Pada tinjauan pustaka ini membahas penelitian sebelumnya terkait tentang pengambilan barang dan melakukan kajian terkait kekurangan serta kelebihan, yang di antaranya adalah :

1. Sy. Syahririni (2018) membuat seleksi benda berwarna menggunakan konveyor dengan lengan robot tujuan dari penelitian ini menyeleksi benda berdasarkan warna menggunakan sensor warna TCS 230 serta mikrokontroler yang digunakan ATmega 32. Pada penelitian ini salah satu kelemahannya yaitu terjadinya kesalahan pembacaan yang di akibatkan pengaruh intensitas cahaya sekitar ruangan, karena kesalahan pendeteksian warna mengakibatkan robot lengan tidak dapat menerima perintah[6].
2. Resa Henggar Prabanegara (2015) membuat penelitian robot lengan pemindah dan penyeleksi barang berdasarkan warna berbasis arduino uno. Tujuan penelitian ini yaitu menyeleksi benda berdasar warna menggunakan sensor warna yang di rancang dengan menggunakan LDR dan RGB led[3]. Pada penelitian ini salah satu kelemahannya yaitu akurasi pembacaan sangat di pengaruhi oleh cahaya.
3. Fajrian Ramadhan (2020) membuat penelitian perancangan penyortiran barang berdasarkan berat dengan sistem pick and place berbasis mikrokontroler. Tujuan dari penelitian ini yaitu menyeleksi benda berdasar berat menggunakan sensor berat Load Cell dan mikrokontroler ATmega 32[7]. Pada penelitian ini salah satu kelemahannya yaitu akan sangat susah untuk menyeleksi barang yang berjalan di atas konveyor dengan jumlah barang lebih dari satu, sehingga keefektifan system secara keseluruhan masih kurang.
4. Erwin Ramadhani (2019) membuat penelitian perancangan robot lengan pemilah obyek sesuai warna berbasis Arduino Mega 2560. Tujuan dari penelitian ini adalah memilah benda berdasarkan warna menggunakan

CMUcam 5 Pixy dan mikrokontroler Arduino Mega 2560[8]. Pada penelitian ini salah satu kelemahannya yaitu terjadinya kesalahan pembacaan dikarenakan besar dan kecilnya benda yang diseleksi.

5. Kartavya Sarware (2020) membuat penelitian lengan robot mengambil dan menaruh secara otomatis benda yang berjalan di atas konveyor. Tujuan dari penelitian ini adalah untuk mempelajari, menganalisis dan membuat Lengan Robot dengan Conveyer Belt yang melakukan operasi pengambilan dan penempatan objek dari satu tempat ke tempat lain[9].
6. Edy Syah Putranta Ginting (2016) membuat penelitian mengangkat dan memindahkan barang dengan lengan robot dengan kendali smartphone android. Tujuan dari penelitian ini yaitu memindahkan barang dengan pengendalian jarak jauh menggunakan perangkat smartphone berbasis android, dimana mikrokontroler yang digunakan adalah ATmega 8535. Salah satu kelemahan dari penelitian ini yaitu belum bisa memilah benda yang diambil[10].
7. Andrea Darmawan (2018) membuat penelitian pembuatan *prototype* alat pemindah barang di pelabuhan berbasis Arduino. Tujuan penelitian ini yaitu mengangkat barang dari atas ke kapal dengan pengendalian jarak jauh[11]. Pada dalam penelitian ini benda tidak berjalan di atas konveyor serta lengan robot yang digunakan menggunakan dua buah motor servo atau 2 DOF(*Degree of Freedom*).
8. Gita Tri Wardana (2014) membuat penelitian robot lengan pemindah barang berdasar ukurannya. Tujuan dari penelitian ini adalah memindahkan benda yang berjalan di atas konveyor berdasarkan warna[12]. Salah satu kelemahan dari penelitian ini yaitu adanya ketidak cocokan antara judul dengan isi penelitian.
9. Harditya Pratama Putra (2013) membuat penelitian Perancangan Simulasi Alat Bantu Pemindah Barang Pada Industri Manufaktur Dengan Robot Lengan. Tujuan penelitian ini yaitu memindahkan benda yang berjalan di atas konveyor berdasarkan warna[13].

10. Irvan Hasan (2015) membuat penelitian aplikasi pengenalan obyek untuk lengan robot berdasar ukurannya. Tujuan penelitian ini adalah memisahkan benda berdasarkan bentuk[5]. Kelemahan penelitian ini yaitu lengan robot hanya satu, sehingga untuk skala benda yang sangat banyak akan sulit dilakukan dalam waktu yang cepat.
11. Purwono Prasetyawan (2018) membuat penelitian Pengendali Lengan Robot Dengan Mikrokontroler Arduino Berbasis *Smartphone*. Tujuan penelitian ini yaitu hanya sekedar mengendalikan gerakan lengan robot menggunakan *smartphone* berbasis android[4]. Fokus penelitian ini lengan robot tidak untuk memindahkan benda.

## 2.2. Landasan Teori

### 2.2.1. Lengan Robot

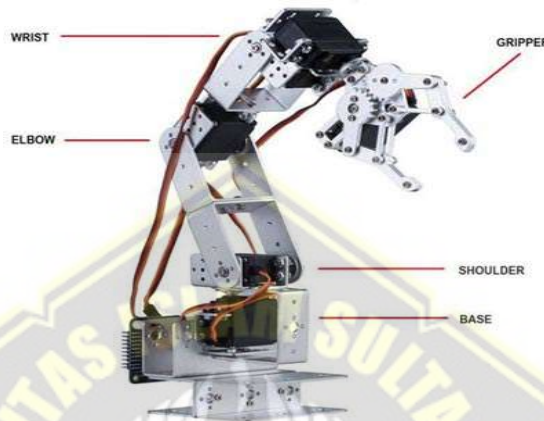
Perkembangan dunia robotika saat ini telah berkembang dengan pesat. Salah satu jenis robot yang banyak digunakan adalah robot lengan (*arm robotic*). Lengan adalah salah satu bagian yang seringkali digunakan pada robot, terutama untuk aplikasi robot *humanoid*. Penguasaan lengan robot dengan teknologi penjepit (*gripper*) dan pengontrolan motor untuk mengetahui posisinya sangat penting dalam membuat robot *humanoid*. Lengan pada robot tersebut digunakan untuk mengambil obyek-obyek di sekitar robot. Namun dalam pengoperasiannya, kebanyakan robot lengan masih dioperasikan menggunakan tombol ataupun tuas, sehingga dibutuhkan operator yang memiliki keahlian khusus untuk mengoperasikannya[14].

Gambar 2.1. adalah lengan robot dengan 5 DOF (*Degree of Freedom*), dimana masing-masing titik pergerakan dilakukan oleh *motor servo* dengan fungsi masing-masing adalah sebagai berikut:

1. *Base servo* : Bagian alas memutar kekanan atau kekiri
2. *Shoulder servo* : Bagian lengan yang bergerak keatas atau kebawah
3. *Elbow servo* : Bagian siku robot fungsinya sama seperti

lengan robot

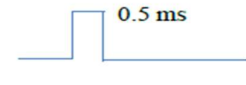

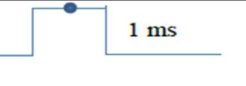

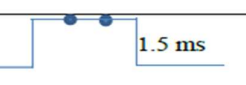



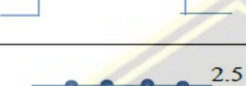

4. *Wrist Servo* : Bagian pergelangan tangan bergerak keatas dan kebawah
5. *Grip Servo* : Bagian *servo* yang berfungsi untuk menjepit benda kerja



Gambar 2.1. Lengan Robot[15]

#### 2.2.2. Motor Servo

Secara umum terdapat 2 jenis motor servo yaitu motor servo standard dan motor *servo continuous*. Servo motor tipe standar hanya mampu berputar 180 derajat. Sedangkan Servo motor *continuous* dapat berputar sebesar 360 derajat. Pengendalian gerakan batang motor servo dapat dilakukan dengan menggunakan metode *Pulse Width Modulation* (PWM). Seperti pada Gambar 2.2, bahwa dengan pulsa 1.5 ms pada periode selebar 20 ms, maka sudut dari sumbu motor akan berada pada posisi tengah. Semakin lebar pulsa OFF maka akan semakin besar gerakan sumbu kearah jarum jam dan semakin kecil pulsa OFF maka akan semakin besar gerakan sumbu ke arah yang berlawanan dengan jarum jam. Untuk menggerakkan motor servo ke kanan atau ke kiri, tergantung dari nilai lebar pulsa yang kita berikan[16].

Input Pulse Width (T=20 ms)	Servo Position
 0.5 ms	 -90°
 1 ms	 -45°
 1.5 ms	 0°
 2 ms	 +45°
 2.5 ms	 +90°

Gambar 2.2. Pensinyalan motor Servo[16]

Setelah didapat nilai lebar pulsa seperti pada Gambar 2.2, kita dapat menghitung lebar pulsa setiap sudut yang kita inginkan dengan rumus persamaan linier. Dengan memasukkan nilai lebar pulsa maksimum untuk sudut +90 dan nilai lebar pulsa minimum untuk sudut -90. Maka dicarilah rumus persamaan lebar pulsa atau  $T_B$ . Dalam perumusan  $T_B$  ini memanfaatkan formulasi garis lurus, hal ini dikarenakan nilai lebar pulsa  $T_B$  linear terhadap nilai sudut B derajat. Untuk penentuan sudut motor servo digunakan persamaan (2.1).

$$\frac{(x-x_2)}{(x_2-x_1)} = \frac{(y-y_1)}{(y_2-y_1)} \dots\dots\dots(2.1)$$

Dimana :

- x = B atau sudut motor servo
- $x_1$  = Sudut Minimum Motor Servo
- $x_2$  = Sudut Maksimum Motor Servo
- $y_1$  = Minimum Pulse
- $y_2$  = Maksimum Pulse
- y = Lebar Pulsa Sudut B
- B = Nilai Sudut (derajat)
- $T_B$  = Lebar Pulsa sudut B (ms)

Apabila penggunaan rumus linear ini disesuaikan dengan permasalahan mencari lebar pulsa untuk mendapat sudut B, maka akan didapat persamaan (2.2) sampai dengan (2.12).

$$\frac{(x-x_1)}{(x_2-x_1)} = \frac{(y-y_1)}{(y_2-y_1)} \dots\dots\dots (2.2)$$

$$\frac{(B-(90))}{(90-(-90))} = \frac{(TB-\text{min.pulse})}{(\text{max.pulse}-\text{min.pulse})} \dots\dots\dots (2.3)$$

$$\frac{(B+90)}{180} = \frac{(TB-0,5)}{(2,5-0,5)} \dots\dots\dots (2.4)$$

$$\frac{(B+90)}{180} = \frac{(TB-0,5)}{2} \dots\dots\dots (2.5)$$

$$2B + 180 = 180TB - 90 \dots\dots\dots (2.6)$$

$$\frac{2B+180+90}{180} = TB \dots\dots\dots (2.7)$$

$$\frac{2B+180+90}{180} = TB \dots\dots\dots (2.8)$$

$$\frac{2B+270}{180} = TB \dots\dots\dots (2.9)$$

$$180 \times TB = 2B + 270 \dots\dots\dots (2.10)$$

$$(180 \times TB) - 270 = 2B \dots\dots\dots (2.11)$$

$$\frac{(180 \times TB - 270)}{2} = B \dots\dots\dots (2.12)$$

Gambar 2.3 memperlihatkan salah satu tipe motor servo standar yang banyak digunakan dalam berbagai penelitian. Dimana motor servo tipe ini biasanya terdiri dari 3 kabel yaitu kabel merah untuk +5V, kabel coklat untuk GND serta kabel oranye untuk memberikan masukan berupa sinyal PWM.





Gambar 2.3. Motor servo SG-90[17]

### 2.2.3. *Pulse With Modulation (PWM)*

PWM (*Pulse Width Modulation*) adalah salah satu teknik modulasi dengan mengubah lebar pulsa (*duty cycle*) dengan nilai amplitudo dan frekuensi yang tetap. Satu siklus pulsa merupakan kondisi *high* kemudian berada di zona transisi ke kondisi *low*. Lebar pulsa PWM berbanding lurus dengan amplitudo sinyal asli yang belum termodulasi. *Duty Cycle* merupakan representasi dari kondisi logika *high* dalam suatu periode sinyal dan dinyatakan dalam bentuk (%) dengan range 0% sampai 100%, sebagai contoh jika sinyal berada dalam kondisi *high* terus menerus artinya memiliki *duty cycle* sebesar 100%. Jika waktu sinyal keadaan *high* sama dengan keadaan *low* maka sinyal mempunyai *duty cycle* sebesar 50%[18].

Aplikasi penggunaan PWM biasanya ditemui untuk pengaturan kecepatan motor dc, pengaturan cerah/redup LED, dan pengendalian sudut pada motor servo. Contoh penggunaan PWM pada pengaturan kecepatan motor dc semakin besar nilai *duty cycle* yang diberikan maka akan berpengaruh terhadap cepatnya putaran motor. Apabila nilai *duty cycle*-nya kecil maka motor akan bergerak lambat.

Untuk membandingkannya terhadap tegangan DC, PWM memiliki 3 mode operasi yaitu :

#### a. *Inverted Mode*

Pada mode *inverted* ini jika nilai sinyal lebih besar dari pada titik pembandingan (*compare level*) maka output akan di set *high* (5v) dan

sebaliknya jika nilai sinyal lebih kecil maka output akan di set *low* (0v) seperti pada gelombang A pada gambar di atas.

b. *Non Inverted Mode*

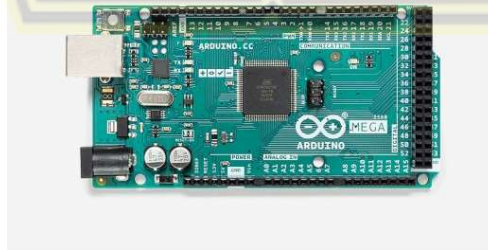
Pada mode *non inverted* ini output akan bernilai high (5v) jika titik pembandingan (*compare level*) lebih besar dari pada nilai sinyal dan sebaliknya jika bernilai *low* (0v) pada saat titik pembandingan lebih kecil dari nilai sinyal.

c. *Toggle Mode*

Pada mode *toggle output* akan beralih dari nilai high (5v) ke nilai low (0v) jika titik pembandingan sesuai dan sebaliknya beralih dari nilai *low* ke *high*.

#### 2.2.4. Arduino Mega

Arduino Mega adalah *board* mikrokontroler berbasis Atmega 1280. Memiliki 54 pin input dan output digital, dimana 14 pin diantara pin tersebut dapat digunakan sebagai output PWM dan 16 pin input analog, 16 MHz osilator kristal, koneksi USB, jack power, ICSP header, dan tombol reset. Untuk mendukung mikrokontroler agar dapat digunakan, cukup hanya menghubungkan Board Arduino Mega ke komputer dengan menggunakan kabel USB atau dengan adaptor AC-DC untuk menjalankannya[19]. Gambar 2.4. memperlihatkan bentuk fisik dari Arduino Mega 2560.



Gambar 2.4. Modul Arduino Mega[20]

### 2.2.5. Belt Konveyor

Konveyor adalah suatu alat yang berfungsi untuk mengangkut atau memindahkan material. Mulai dari material curah hingga material satuan. Menggunakan konveyor bisa memindahkan materi secara mudah dari satu tempat ke tempat lain secara kontinu berapapun jumlahnya. Konveyor system Indonesia memiliki prinsip kerja sederhana. Dimana alat ini akan memindahkan material apa saja yang ada di atas belt. Setibanya umpan di head, maka material akan ditumpahkan lantaran belt bergerak berbalik arah. Belt konveyor sendiri bisa bergerak karena digerakkan oleh *head pulley* atau *drive* yang memakai motor penggerak. *Head pulley* tersebut akan menarik belt konveyor seperti pada Gambar 2.5 memanfaatkan gesekan antara permukaan drum dan belt dengan kapasitas yang bergantung dengan gaya gesekannya[21].



Gambar 2.5. Belt Konveyor[22]

### 2.2.6. Motor DC Gearbox

Motor DC adalah suatu perangkat yang dapat mengubah energi listrik menjadi energi kinetik atau gerakan. Didalam motor DC terdapat dua kumparan yaitu *stator* dan *rotor*. Kumparan medan yang berbentuk kutub merupakan bagian yang tidak berputar (*stator*) dan kumparan jangkarnya merupakan bagian yang berputar (*rotor*)[8]. Motor DC dapat berputar dengan memberi beda tegangan pada masing masing terminal motor DC tersebut [7]. Motor DC yang digunakan adalah motor DC 12V seperti terlihat pada Gambar 2.6.



Gambar 2.6. Motor DC Gearbox[23]

#### 2.2.7. Sensor Infra Red FC 51

Sensor Infrared FC-51 memiliki Infrared *transmitter* dan Infrared *receiver* yang akan mendeteksi keberadaan hambatan di depan modul sensor. Infrared *transmitter* adalah bagian yang memancarkan radiasi infra merah, sehingga biasa disebut IR LED. Meskipun IR LED tampak seperti LED normal pada umumnya, namun radiasi yang dipancarkan oleh IR LED tidak akan terlihat oleh mata manusia. Infrared *receiver* adalah bagian yang mendeteksi radiasi dari infrared *transmitter*. IR transmitter biasanya berbentuk photodiode dan phototransistor.

Photodiode infrared berbeda dari *photodiode* normal karena hanya mendeteksi radiasi infrared saja. Output dari sensor ditentukan berdasarkan intensitas penerimaan yang diterima *photodiode*[24].

Pada modul ini terdapat potensiometer yang dapat digunakan untuk mengatur jangkauan pendeteksian. Modul ini menggunakan komparator IC LM393. Rangkaian komparator pada modul ini menggunakan mode non-inverting di mana ketika tegangan pada pin (+) lebih besar dari pin (-), maka output akan berayun ke arah V+, tetapi jika tegangan pada pin (+) lebih kecil dari pin (-), maka output akan berayun ke arah V-.

Pada Modul rangkaian komparator ini LED *obstacle* yang berada di modul FC-51 ini akan padam ketika output berayun ke arah V+ dan LED *obstacle* yang berada di modul FC-51 ini akan menyala ketika output berayun ke arah V-. Gambar 2.7 adalah salah satu contoh modul sensor infrared tipe HC-51 yang banyak beredar di pasaran dan penggunaannya sering digunakan dengan perangkat arduino[24].

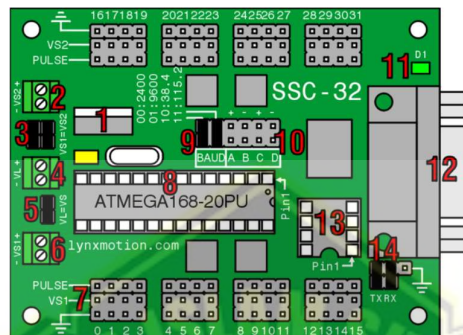


Gambar 2.7. Modul Sensor Infrared FC-51[24]

#### 2.2.8. Modul SSC-32 ATmega168

SSC-32 (*servo controller serial*) adalah *preassembled servo kontroller* kecil dengan beberapa fitur. SSC-32 memiliki resolusi 1 $\mu$ S untuk posisi yang akurat, dan bergerak sangat halus. Kisaran adalah 0.50 mS untuk 2.50 mS untuk jarak sekitar 180°. Kontrol gerak dapat tanggapan langsung, kecepatan dikendalikan, gerak waktunya, atau kombinasi keduanya. Yang unik “*Group Move*” memungkinkan setiap kombinasi servo untuk memulai dan mengakhiri gerakan pada saat yang sama, bahkan jika servo harus pindah jarak yang berbeda. Ini adalah fitur yang sangat bagus untuk menciptakan berjalan kompleks untuk robot berjalan *multi servo*. *Servo* untuk posisi atau gerakan dapat untuk memberikan umpan balik ke komputer. Bahkan ada 12 servo *Hexapod sequencer built in*, memungkinkan kontrol penuh terhadap semua *tripod*, hanya dengan mentransfer beberapa nilai dari *host controller*. Setiap output dapat digunakan sebagai tingkat output TTL. Ada 4 input digital yang statis atau terkunci, sehingga tidak perlu khawatir kehilangan acara

singkat. Mereka juga dapat digunakan sebagai input analog. Ada tiga blok terminal untuk pilihan *powering*. DB9 sebagai input yang terhubung dengan RS-232 untuk dapat digunakan dengan PC {Formatting Citation}. Gambar 2.8 memperlihatkan gambar layout dari modul SSC 32, sedang gambar modul SSC 32 terlihat pada Gambar 2.9.



Gambar 2.8. Layout PCB Modul SSC-32[26]

Tabel 2.1. Bagian Modul SSC-32[26]

No	Keterangan	No	Keterangan
1	Low Dropout Regulator	8	Mikrokontroler ATmega168
2	Power Input V32	9	Jumper Baudrate
3	Jumper VS1 dan VS2	10	Jumper Pullups
4	Tegangan Logika	11	Indikator LED
5	Jumper VL dan VS	12	Port DB9
6	Power Input V31	13	IC EEPROM
7	Perminal Motor Servo	14	Jumper RX-TX



Gambar 2.9. Modul SSC-32 ATMega168[27]

#### 2.2.9. Modul Bluetooth HC-05

Modul Bluetooth HC-05 adalah modul komunikasi nirkabel via bluetooth yang dimana beroperasi pada frekuensi 2.4 GHz dengan pilihan dua mode konektivitas. Mode 1 berperan sebagai *slave* atau receiver data saja, mode 2 berperan sebagai master atau dapat bertindak sebagai *transceiver*. Pengaplikasian komponen ini sangat cocok pada *project* elektronika dengan komunikasi nirkabel atau wireless. Aplikasi yang dimaksud antara lain aplikasi sistem kendali, monitoring, maupun gabungan keduanya. Antarmuka yang dipergunakan untuk mengakses modul ini yaitu serial TXD, RXD, VCC serta GND. Serta terdapat LED (*built in*) sebagai indikator koneksi bluetooth terhadap perangkat lainnya seperti sesama modul, dengan smartphone android, dan sebagainya[28].

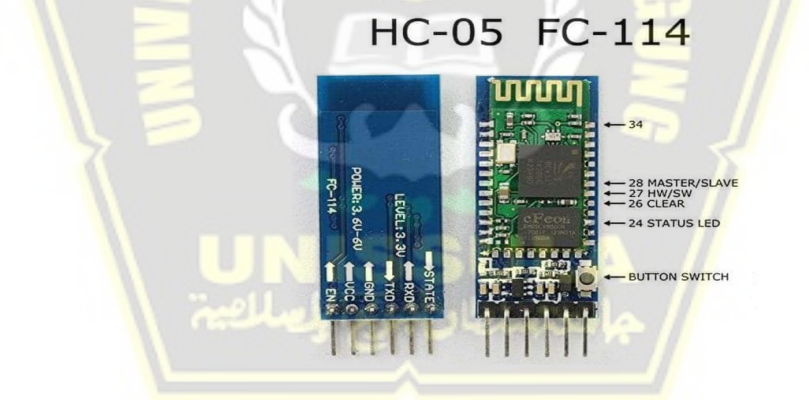
Jangkauan jarak efektif modul ini saat terkoneksi dalam range 10 meter, dan jika melebihi dari range tersebut maka kualitas konektivitas akan semakin kurang maksimal.

Spesifikasi dari module ini antara lain :

1. Frekuensi kerja ISM 2.4 GHz
2. Bluetooth protocol : Bluetooth tipe v2.0+EDR
3. Kecepatan dapat mencapai 1 Mbps pada mode sinkron

4. Kecepatan dapat mencapai 2.1 Mbps / 160 kbps pada mode asinkron maksimum
5. Tegangan kerja pada 3,3 – 6 Volt DC
6. Konsumsi arus kerja yaitu 50 Ma
7. Memiliki modulasi Gaussian Frequency Shift Keying (GFSK)
8. Sensitivitas -84 dBm (0.1% BER)
9. Daya emisi 4 dBm
10. Suhu operasional range -20°C — +75°C
11. Memiliki keamanan dengan enkripsi data dan enkripsi
12. Dimensi modul 15.2×35.7×5.6 mm

Modul yang seperti terlihat pada Gambar 2.10 ini dapat digunakan sebagai mode *slave* (Rx), maupun mode *master* (TX) dan memiliki 2 metode konfigurasi yaitu AT Mode dan *Communication Mode*. Pada AT Mode berfungsi sebagai pengaturan konfigurasi dari HC-05, sedangkan pada *Communication Mode* berfungsi sebagai komunikasi nirkabel dengan perangkat atau piranti lainnya[28].



Gambar 2.10. Modul Bluetooth HC-05[29]

#### 2.2.10. Android

Android merupakan sistem operasi dengan basis *Kernel Linux* yang banyak digunakan pada smartphone dan tablet. Google merancangya demi mendukung kinerja perangkat alat elektronik dengan sentuhan,



ketukan ataupun gesekan. Sedangkan sistem operasi berupa aplikasi dalam perangkat elektronik yang mengatur keseluruhan sumber daya. Seperti manajemen memori, antarmuka tampilan, aplikasi dan lain sebagainya. Pembuat dan pengembang perangkat lunak bebas menggunakan, memodifikasi, sampai mendistribusikan android. Termasuk para pembuat aplikasi yang membuat android mempunyai jutaan aplikasi siap unduh di google play store.

Awalnya android dibuat sebagai pendukung perangkat kamera digital supaya terhubung langsung dengan internet. Karena pangsa pasar tidak terlalu besar, maka selanjutnya android dikembangkan dalam perangkat mobile. Sementara pada saat itu keberadaan Symbian dan windows mobile adalah sebagai pemain utama.

Sejarah perkembangan Android dimulai tahun 2003 saat Andy Rubin, Rich Miner, Chris White dan Nick Sears mendirikan Android, Inc. Sebuah perusahaan besar di Palo Alto, California yang kemudian dipinang oleh Google pada tanggal 17 Agustus 2005[30].

Di bawah naungan Google, android menghilang dan tidak melahirkan apapun sampai pertengahan tahun 2008. Hingga pada 22 Oktober 2008, HTC Dream diluncurkan sebagai ponsel seluler komersial pertama yang berbasis Android.

Dua tahun setelahnya, ponsel pintar seri Nexus One diluncurkan oleh Google dengan bantuan HTC selama proses pembuatan. Hingga akhirnya melahirkan berbagai brand dari OEM yang berbeda seperti Asus, Samsung, LG dan lain sebagainya.

Selanjutnya perangkat Android mulai berkembang pesat dan menenggelamkan para pesaingnya. Bersama dengan perubahan, perkembangan dan perbaikan fitur yang membuat performa android lebih optimal [30].

### 2.2.11. Catu Daya

Arus Listrik yang kita gunakan di rumah, kantor dan pabrik pada umumnya adalah dibangkitkan, dikirim dan didistribusikan ke tempat masing-masing dalam bentuk arus bolak-balik atau arus AC (*Alternating Current*). Hal ini dikarenakan pembangkitan dan pendistribusian arus listrik melalui bentuk arus bolak-balik (AC) merupakan cara yang paling ekonomis dibandingkan dalam bentuk arus searah atau arus DC (*Direct Current*)[31].

Akan tetapi, peralatan elektronika yang kita gunakan sekarang ini sebagian besar membutuhkan arus DC dengan tegangan yang lebih rendah untuk pengoperasiannya. Oleh karena itu, hampir setiap peralatan elektronika memiliki sebuah rangkaian yang berfungsi untuk melakukan konversi arus listrik dari arus AC menjadi arus DC dan juga untuk menyediakan tegangan yang sesuai dengan rangkaian elektroniknya. Rangkaian yang mengubah arus listrik AC menjadi DC ini disebut dengan *DC Power Supply* atau dalam bahasa Indonesia disebut dengan Catu daya DC. *DC Power Supply* atau catu daya juga sering dikenal dengan nama “Adaptor”[31].

Sebuah DC power supply atau adaptor pada dasarnya memiliki 4 bagian utama agar dapat menghasilkan arus DC yang stabil. Keempat bagian utama tersebut diantaranya adalah Transformer, Rectifier, Filter dan Voltage Regulator[31]. Gambar 2.11 memperlihatkan bentuk contoh bentuk fisik dari power supply.

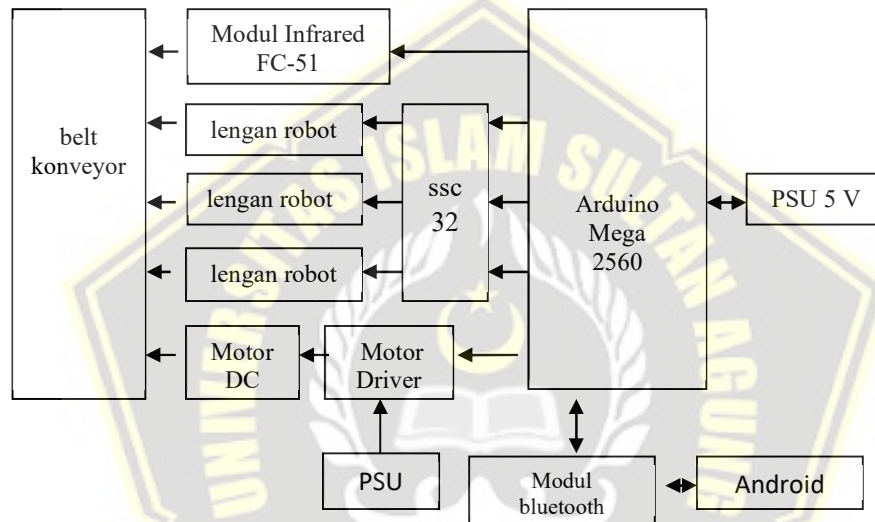


Gambar 2.11. Power supply dalam bentuk jadi[32]

### BAB III METODE PENELITIAN

#### 3.1. Model Penelitian

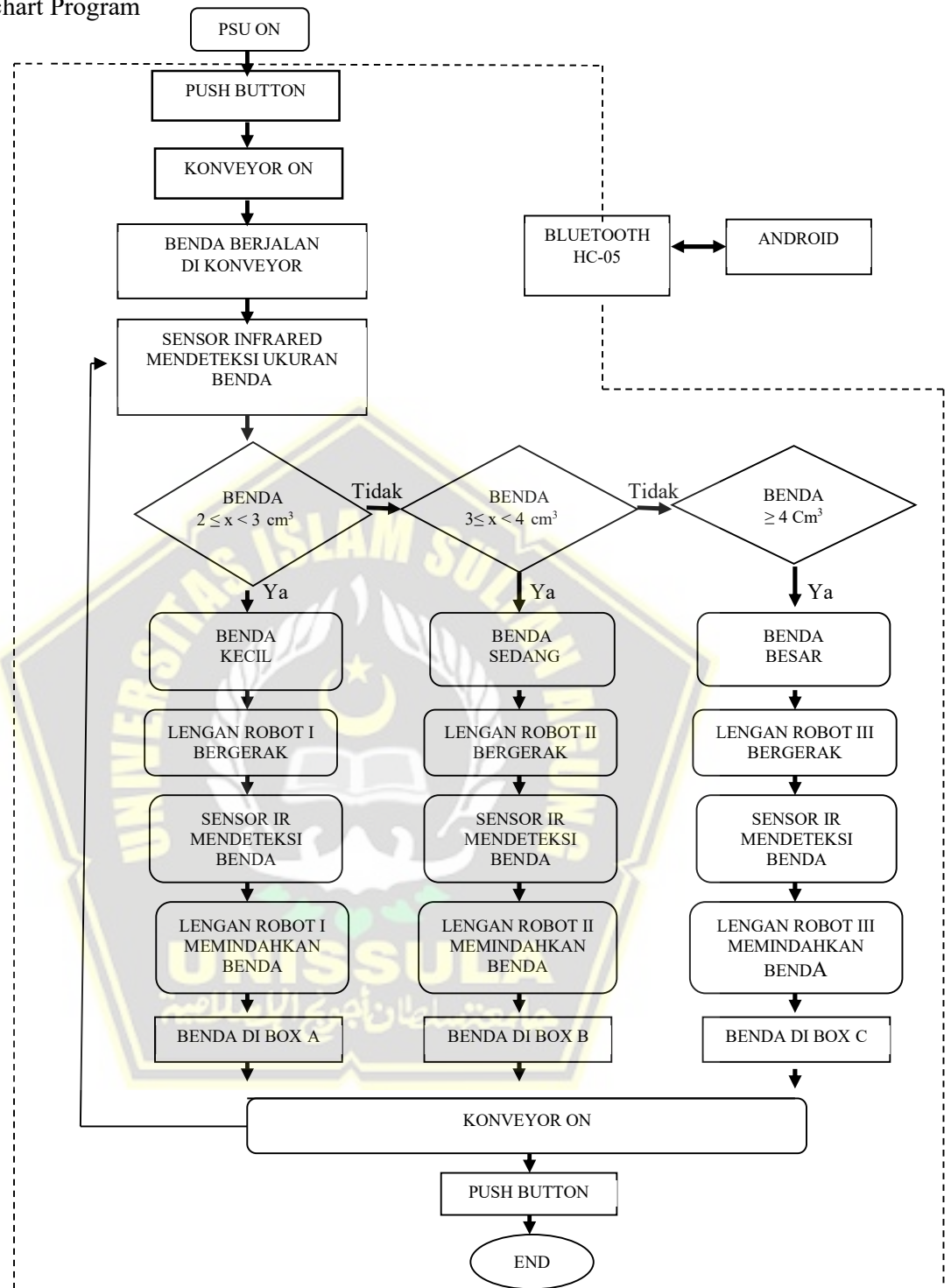
Model pengaturan lengan robot untuk *lifting* barang secara otomatis berdasar ukuran berbasis android akan diimplementasikan menggunakan perangkat smartphone berbasis android dengan skala *prototype*. Dalam pembuatan *prototype*, langkah pertama yang harus dilakukan adalah membuat sistem model, yang ditunjukkan pada Gambar 3.1.



Gambar 3.1. Arsitektur Sistem

Berdasarkan arsitektur sistem pada Gambar 3.1, jantung utama dari rangkaian tersebut adalah mikro kontroler Arduino Mega 2560. Konveyor akan berjalan dengan penggerak sebuah motor DC gearbox yang kecepatannya dikendalikan oleh Stepper Motor DC. Modul infrared FC-51 akan membaca besar kecilnya benda yang berjalan di atas konveyor. Dimana jika ukuran benda adalah  $x$ , maka  $2 \leq x < 3 \text{ cm}^3$  dikategorikan kecil, jika  $3 \leq x < 4 \text{ cm}^3$  ukuran benda dikategorikan sedang dan jika  $x \geq 4 \text{ cm}^3$  ukuran benda dikategorikan besar.

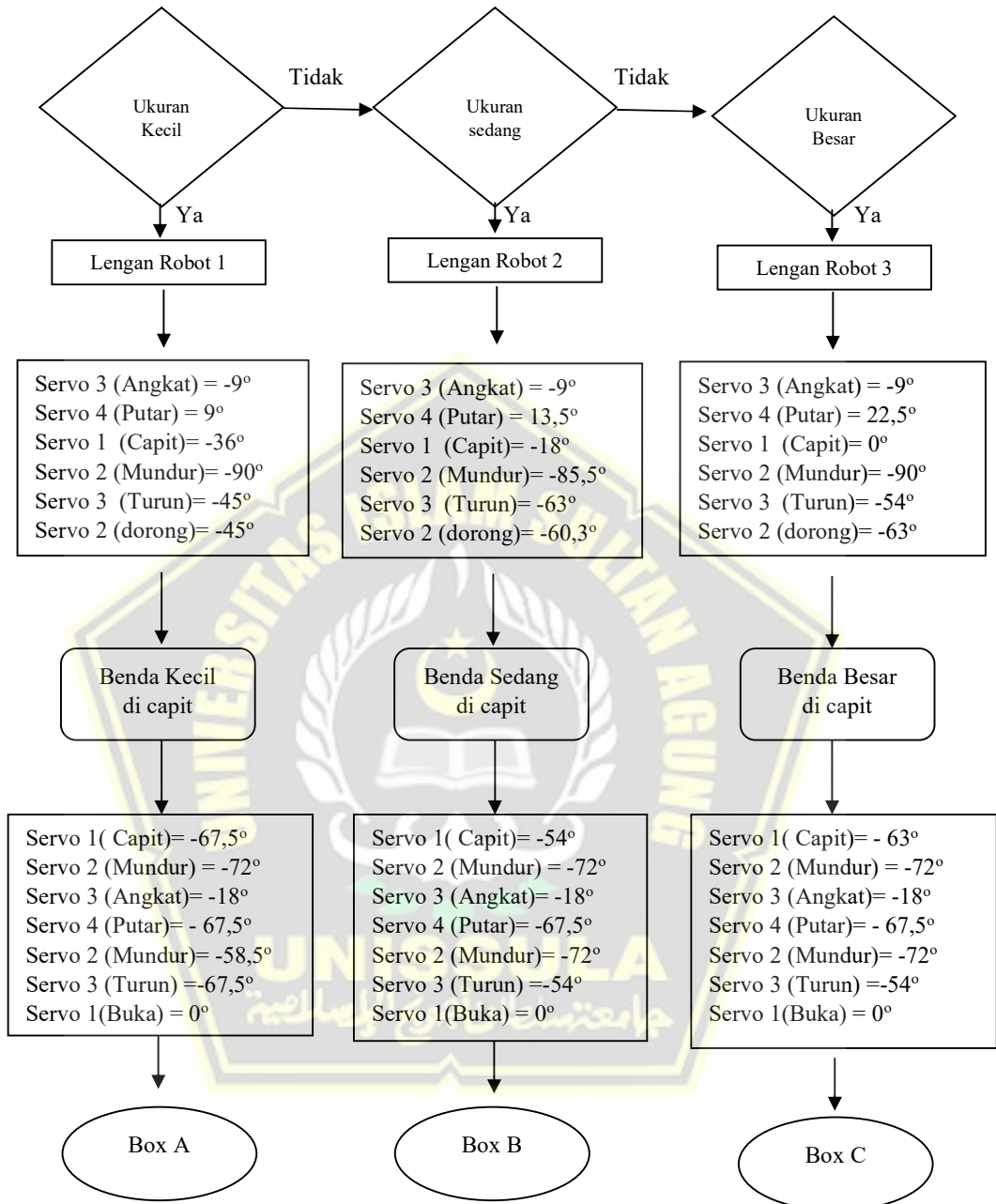
### 3.2. Flowchart Program



Gambar 3.2. Diagram Flow Chart

Untuk mempermudah memahami gambaran sistem model, maka perlu dibuat *flowcart* yang berisi instruksi suatu program. Dimana *flowcart* merupakan mekanisme kerja suatu sistem yang dibuat dalam bentuk simbol - simbol yang masing-masing telah ditentukan makna dan kegunaannya seperti terlihat pada Gambar 3.2 di atas. Ketika power supply dinyalakan, maka sistem belum berjalan atau baru pada mode *standby*. Untuk membuat sistem berjalan ada dua cara yaitu dengan menekan tombol *push button* atau dengan menggunakan perangkat android sehingga bisa dikendalikan dari jarak jauh. Setelah salah satu cara tersebut dilakukan maka motor DC yang ada di konveyor akan bekerja sehingga konveyor akan berjalan. Ketika konveyor berjalan dan meletakkan benda di atasnya benda akan terangkut atau berpindah. Tiga buah sensor infrared yang diletakkan di atas konveyor akan membaca besar kecilnya benda berdasarkan pantulan sinar infrared yang memantul pada benda tersebut. Ketika yang memantul hanya satu infrared, maka benda akan terbaca kecil dan ketika yang memantulkan sinar infrared ada dua maka benda terbaca sedang dan terakhir jika ketiganya memantulkan sinar infrared maka benda akan terbaca besar. Ketika benda yang berjalan sudah teridentifikasi ukurannya maka mikrokontroler akan memerintahkan lengan robot untuk mendekati konveyor dimana ketika benda terbaca kecil, maka lengan robot 1 yang bergerak, ketika benda terbaca sedang maka lengan robot 2 yang bergerak dan ketika benda terbaca besar maka hanya lengan robot 3 yang bergerak mendekati konveyor. Ketika benda yang bergerak sudah mendekati masing-masing lengan robot, maka sensor infrared akan membaca keberadaan benda, kemudian mikrokontroller akan memerintahkan grip untuk mengambil benda dan masing-masing lengan robot akan meletakkan benda di tempatnya sesuai dengan ukurannya. Pada posisi ini konveyor akan tetap berjalan hingga kita mematikan melalui *push button* atau bisa juga melalui perangkat android yang dikendalikan dari jarak jauh.

### 3.3. Flowchart Lengan Robot



Gambar 3.3. Flowchart Pengaturan Sudut Motor Servo Lengan Robot

Untuk membuat lengan robot mampu dengan tepat mengambil benda yang berjalan di atas konveyor, maka perlu dirancang diagram *flowchart* khusus untuk mengatur pergerakan sudut dari masing-masing motor servo pada ketiga lengan robot mulai dari persiapan pengambilan benda sampai dengan meletakkan benda di tempat yang direncanakan seperti terlihat pada Gambar 3.3.

Dari diagram *flowchart* motor servo lengan robot di atas dapat dideskripsikan bahwa secara umum sudut pergerakan ketiga lengan robot mendekati sama, kemudian dari proses gerak pertama sampai gerak terakhir pada saat penempatan benda seluruhnya ada 13 gerakan dimana fungsi motor servo 1 adalah untuk membuka atau menutup capit lengan robot, fungsi motor servo 2 untuk maju atau mundur, fungsi motor servo 3 untuk naik atau turun serta fungsi dari motor servo 4 untuk putar kanan atau kiri.

Jika benda terdeteksi kecil maka lengan robot 1 akan bergerak dengan urutan serta sudut gerak motor servo adalah sebagai berikut: Servo 3 (Angkat) =  $-9^\circ$ , Servo 4 (Putar) =  $9^\circ$ , Servo 1 (Capit) =  $-36^\circ$ , Servo 2 (Mundur) =  $-90^\circ$ , Servo 3 (Turun) =  $-45^\circ$ , Servo 2 (dorong) =  $-45^\circ$  lalu benda kecil di capit dilanjutkan dengan pergerakan motor servo meletakkan benda di tempat yang di tentukan dengan urutan sebagai berikut : Servo 1 (Capit) =  $-67,5^\circ$ , Servo 2 (Mundur) =  $-72^\circ$ , Servo 3 (Angkat) =  $-18^\circ$ , Servo 4 (Putar) =  $-67,5^\circ$ , Servo 2 (Mundur) =  $-58,5^\circ$ , Servo 3 (Turun) =  $-67,5^\circ$ , Servo 1 (Buka) =  $0^\circ$  untuk meletakkan benda ukuran kecil.

Jika Benda terdeteksi sedang maka lengan robot akan bergerak dengan urutan serta sudut gerak motor servo adalah sebagai berikut: Servo 3 (Angkat) =  $-9^\circ$ , Servo 4 (Putar) =  $13,5^\circ$ , Servo 1 (Capit) =  $-18^\circ$ , Servo 2 (Mundur) =  $-85,5^\circ$ , Servo 3 (Turun) =  $-63^\circ$ , Servo 2 (dorong) =  $-60,3^\circ$ , lalu benda ukuran sedang di capit dilanjutkan dengan pergerakan motor servo meletakkan benda di tempat yang di tentukan dengan urutan sebagai berikut : Servo 1 (Capit) =  $-4^\circ$ , Servo 2 (Mundur) =  $-72^\circ$ , Servo 3 (Angkat) =  $-18^\circ$ , Servo 4 (Putar) =  $-$

67,5°, Servo 2 (Mundur) = -72°, Servo 3 (Turun) = -54°, Servo 1 (Buka) = 0° untuk meletakkan benda ukuran sedang.

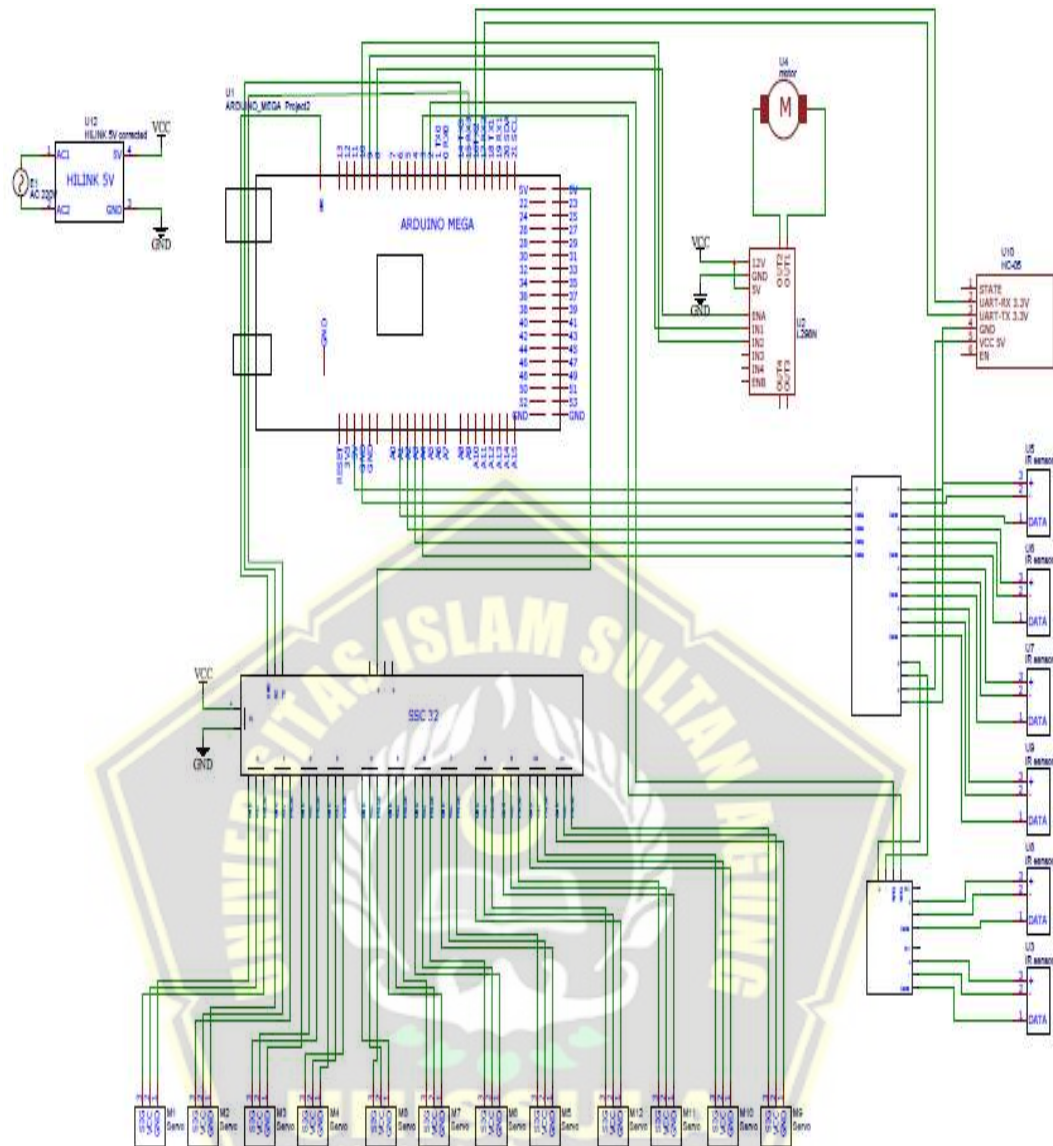
Jika Benda terdeteksi besar maka lengan robot akan bergerak dengan urutan serta sudut gerak motor servo adalah sebagai berikut: Servo 3 (Angkat) = -9°, Servo 4 (Putar) = 22,5°, Servo 1 (Capit) = 0°, Servo 2 (Mundur) = -90°, Servo 3 (Turun) = -54°, Servo 2 (dorong) = -63°, lalu benda ukuran besar di capit dilanjutkan dengan pergerakan motor servo meletakkan benda di tempat yang di tentukan dengan urutan sebagai berikut : Servo 1 (Capit) = - 63°, Servo 2 (Mundur) = -72°, Servo 3 (Angkat) = -18°, Servo 4 (Putar) = - 67,5°, Servo 2 (Mundur) = -72°, Servo 3 (Turun) = -54°, Servo 1 (Buka) = 0° untuk meletakkan benda ukuran besar.

### 3.4. Materi Penelitian

#### 3.4.1. Hardware

Perancangan sistem hardware yang digunakan terdiri dari Arduino Mega 2560 sebagai kontrol utamanya, Modul SSC 32 sebagai kontrol perputaran motor servo, 3 buah lengan robot dengan masing-masing lengan robot menggunakan 4 buah motor servo sehingga jumlah total motor servo yang digunakan ada 12 motor servo, konveyor yang digerakkan motor DC gearbox beserta driver untuk mengendalikan kecepatan putarnya, 6 buah modul infrared FC-51 dimana 3 buah buah untuk mendeteksi besar kecilnya benda yang berjalan di atas konveyor serta 3 buah sebagai sensor titik henti untuk ketiga lengan robot dalam mengambil benda dan Catu Daya SMPS dengan tegangan keluaran 5 Volt. Semua komponen hardware tersebut dirangkai saling berhubungan dan terintegrasi dalam satu sistem seperti digambarkan dalam Gambar 3.4.





Gambar 3.4. Diagram Blok Sistem Hardware

Fungsi dari hardware pada gambar 3.4 adalah sebagai berikut:

1. Modul Arduino Mega 2560 adalah kontrol sistem secara keseluruhan kontroler. Input data dari perangkat modul infrared, lengan robot, dan modul bluetooth HC 05 diproses menggunakan fungsi program (*software*) yang ditanamkan ke dalam chip *mikrokontroler* dari modul Arduino Mega 2560.

2. Modul SSC-32 ATmega 168 merupakan pengontrol motor servo yang ada pada lengan robot. Ada 12 motor servo yang dikendalikan lewat modul SSC-32.
3. Modul infrared FC 51 berfungsi untuk membaca besar kecilnya benda yang berjalan di atas konveyor serta mengendalikan lengan robot dalam mengambil benda yang berjalan di atas konveyor dimana ada enam modul infrared dalam penelitian ini.
4. Modul bluetooth HC 05 berfungsi untuk mengoneksikan sistem dengan perangkat android, sehingga pengoperasian perangkat bisa dikendalikan menggunakan smarphone berbasis android.
5. Motor servo adalah motor DC yang berfungsi untuk menggerakkan lengan robot. Ada empat buah motor servo yang digunakan pada setiap lengan robot. Sehingga jumlah motor servo keseluruhan ada dua belas motor servo.
6. Konveyor adalah sarana berfungsi mengangkut benda yang nantinya akan di pindahkan atau di kelompokkan oleh lengan robot.
7. Motor DC gearbox adalah motor DC yang berfungsi untuk menggerakkan konveyor.
8. Modul driver motor DC L298N adalah modul driver yang berfungsi untuk mengatur kecepatan putar dari motor DC gearbox.
9. Power supply berfungsi sebagai catu daya sistem.

#### 3.4.2. Software

Perancangan software merupakan perancangan program yang dimasukkan ke dalam mikrokontroler Arduino Mega 2560. Pemrograman dimasukkan khusus dengan bahasa pemrograman khusus yang dibuat oleh arduino yaitu Arduino IDE (*Intregated Developmen Environment*). Pemrograman dimulai dari pengaturan kecepatan putar dari konveyor oleh modul driver motor L298N, pembacaan sensor jarak infrared untuk mendeteksi besar kecilnya benda yang berjalan di atas konveyor, penentuan lengan robot yang akan mengambil benda,

pengaturan sudut pada dua belas servo motor pada lengan robot untuk menempatkan posisi pengambilan benda yang berjalan di atas konveyor oleh kontroller servo motor SSC 32, pengaturan sensor infrared FC 51 untuk mendeteksi benda.

Pada saat proses inialisasi, setiap perangkat harus dapat terdeteksi serta datanya dapat terbaca oleh kontroler. Supaya setiap perangkat terdeteksi oleh kontroler, maka yang harus diperhatikan adalah pengaturan program yang terdapat pada bagian inialisasi harus sesuai dengan aturan standar komunikasi yang digunakan oleh masing-masing perangkat kontroler. Jika tidak sesuai dengan aturan standar, maka perangkat tidak bisa dideteksi, yang berakibat kegagalan proses inialisasi.

Untuk mengendalikan dari jarak jauh menggunakan perangkat smartphone yang berbasis android digunakan modul bluetooth HC-05 dimana untuk pemrograman digunakan software Basic4Android. Dimana B4A adalah development tool sederhana yang powerful untuk membangun aplikasi Android. Bahasa Basic4Android mirip dengan bahasa Visual Basic dengan tambahan dukungan untuk objek. Aplikasi Android (APK) yang dicompile oleh Basic4Android adalah aplikasi Android native/asli dan tidak ada extra runtime seperti di Visual Basic yang ketergantungan file `msvbvm60.dll`, yang pasti aplikasi yang dicompile oleh Basic4Android adalah No Dependencies (tidak ketergantungan file oleh lain). IDE Basic4Android hanya fokus pada development Android.

### 3.5. Alat yang Digunakan

Dalam penelitian ini beberapa alat yang digunakan antara lain :

#### 1. Multimeter Digital

Penggunaan multimeter digital dalam penelitian ini mutlak diperlukan guna mengukur tegangan di tiap-tiap titik yang diperlukan, misalkan tegangan yang keluar dari PSU, tegangan yang masuk pada arduino

mega, motor servo, motor DC serta mengukur impedansi antar titik dalam penelitian ini.

#### 2. Tachometer Digital

Tachometer Digital merupakan alat elektronik yang berfungsi untuk mengukur kecepatan putaran, sehingga penggunaan alat ini dalam penelitian adalah untuk mengukur putaran dari belt konveyor.

#### 3. Stopwatch

Stopwatch digunakan untuk mengukur kecepatan putaran dari sabuk konveyor, sehingga bisa di *setting* secara presisi berapa kecepatan putaran belt konveyor yang diinginkan.

#### 4. Obeng Set

Obeng set diperlukan guna merangkai peralatan secara keseluruhan, baik merangkai robot lengan, motor servo, motor DC memasang modul dan lain sebagainya.

#### 5. Laptop

Semua sistem ini untuk dapat bekerja memerlukan pengisian program, sehingga sistem dapat bekerja dengan baik sesuai dengan perencanaan awal. Untuk melakukan pemrograman dibutuhkan sebuah perangkat komputer yang salah satunya adalah laptop dengan spesifikasi minimal window 7.

#### 6. Penggaris Busur

Setiap motor servo dalam lengan robot dalam bekerja membutuhkan pengaturan gerak sudut yang presisi, untuk itu diperlukan penggaris busur untuk menentukan sudutnya sehingga di capai sudut sesuai dengan pengaturan.

### 3.6. Langkah-Langkah Penelitian

Setelah menentukan desain penelitian, bahan-bahan penelitian serta alat yang digunakan sudah siap, maka langkah selanjutnya yaitu merangkai semua bahan yang sudah disiapkan sesuai dengan rangkaian penelitian yang sudah direncanakan. Dimana urutannya adalah sebagai berikut:

## 1. Perakitan belt konveyor

Tahap awal dari penelitian ini yaitu merakit *prototype* belt konveyor dimana bahan bahan yang digunakan diantaranya, motor DC 3 – 12 volt 60 RPM, driver speed motor L298N, catu daya 12 V, sabuk konveyor, gigi gear dengan ratio 11 : 56, roller konveyor serta *bracket* untuk konveyor. Setelah semua terpasang, maka langkah selanjutnya adalah menjalankan konveyor serta mengatur dan mengukur kecepatan putarannya dengan menggunakan tachometer digital sehingga mendapatkan kecepatan putaran yang diinginkan. Setelah konveyor mampu berjalan, langkah selanjutnya yaitu memberi beban beberapa ukuran di atas konveyor yang sedang berjalan untuk menguji apakah konveyor mampu untuk memindah benda dengan baik tanpa adanya kendala yang didapatkan..

## 2. Perancangan Arduino Mega 2560

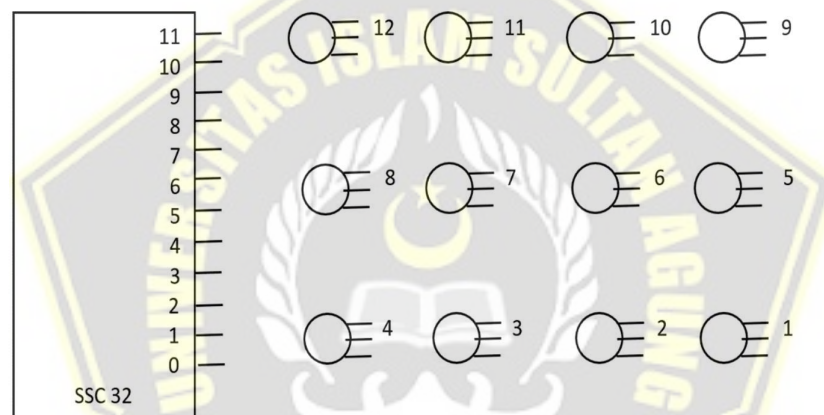
Setelah konveyor bekerja sesuai dengan rencana, maka langkah selanjutnya adalah pemasangan kontrol utama dari penelitian ini yaitu mikrokontroler Arduino Mega 2560 serta merancang wiring diagram yang keluar dan menuju Arduino Mega 2560. Dari hasil perancangan, maka jalur yang ada di Arduino Mega 2560 adalah seperti pada Tabel 3.1.

Tabel 3.1. PIN Arduino Mega 2560

Nomor PIN	Nama PIN	Output
A1-A4	Analog In	Modul Infra red
2-3	PWM	Modul Infra Red
8 – 10	PWM	Motor DC
14 – 15	TX3, RX3	SSC 32
16 – 17	TX2, RX2	Modul Bluetooth

### 3. Perancangan Lengan Robot

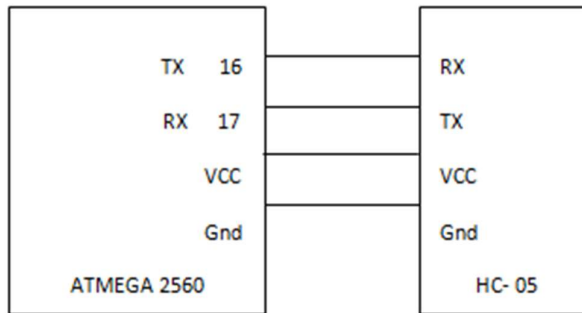
Kelistrikan utama dari lengan robot adalah motor servo SG09, dimana masing-masing lengan robot menggunakan empat buah motor servo SG09. Sehingga jumlah motor servo yang digunakan adalah 12 motor servo. Karena jumlah motor servo yang digunakan ada 12, maka untuk memudahkan dalam pengontrolan, maka digunakan pengontrol motor servo SSC 32 yang mempunyai 32 pin output. Gambar 3.5 memperlihatkan wiring diagram servo lengan robot. Dimana pin output SSC 32 yang berupa sinyal PWM (*Pulse With Modulation*) dihubungkan dengan kaki PWM dari masing-masing motor servo lengan robot yang berjumlah 12 motor servo.



Gambar 3.5 Wiring Diagram Motor Servo Lengan Robot

### 4. Perancangan Modul Bluetooth HC-05

Untuk mengoneksikan rangkaian dengan perangkat smartphone yang berbasis android maka di butuhkan jalur komunikasi menggunakan bluetooth, dimana dalam penelitian ini menggunakan modul bluetooth HC-05 yang di koneksikan dengan Arduino Mega 2560 seperti terlihat pada Gambar 3.6.



Gambar 3.6. Wiring diagram bluetooth HC-05

### 5. Perancangan Modul Infrared FC 51

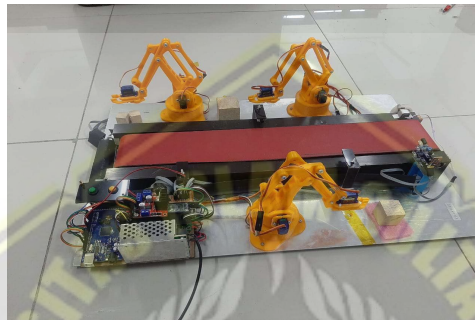
Fungsi utama modul infrared FC 51 dari rangkaian ini adalah untuk membaca besar kecilnya benda yang berjalan di atas konveyor serta mengendalikan *gripper* lengan robot dalam mengambil benda yang berjalan di atas konveyor. Ada 6 buah modul infrared yang digunakan dimana 3 buah berfungsi membaca besar kecilnya benda serta 3 modul berfungsi mengendalikan lengan robot dalam mengambil benda. Wiring diagram modul infrared FC 51 digambarkan pada Gambar 3.7.



Gambar 3.7. Wiring diagram Modul Infrared FC 51

## 6. Pengujian Alat Secara Keseluruhan

Setelah semua komponen terpasang sesuai skema yang telah dirancang, maka langkah selanjutnya adalah menguji alat secara keseluruhan hingga didapatkan hasil yang sesuai dengan harapan yang ditandai dengan kemampuan untuk mengambil benda yang berjalan diatas konveyor berdasarkan ukuran bendanya serta *prototype* mampu untuk dikendalikan dari jarak jauh menggunakan smartphone berbasis android. Gambar 3.8 memperlihatkan *prototype* penelitian.



Gambar 3.8. Prototype alat

## 3.7. Parameter yang di Analisis

Guna mendapatkan hasil penelitian yang lebih terperinci dan sesuai dengan tujuan penelitian, maka perlu kiranya di susun parameter apa saja yang akan dianalisis dalam penelitian ini dimana parameternya diantaranya yaitu :

- a. Parameter pengujian sudut motor servo lengan robot
- b. Parameter pengujian waktu gerak motor servo lengan robot

## 3.8. Kendala

Dalam penelitian ini beberapa kesulitan yang didapat diantaranya adalah:

- a. Penggunaan lengan robot masih banyak kendala yang di antaranya karena bahan yang digunakan dari bahan plastik, sehingga *gear* lengan robot mudah patah, terutama pada bagian capit lengan robot.
- b. Pengujian alat akan mengalami error jika ada cahaya matahari yang masuk ke sensor infrared FC-51, sehingga dalam pengujian sensor tidak boleh terkena cahaya matahari langsung.

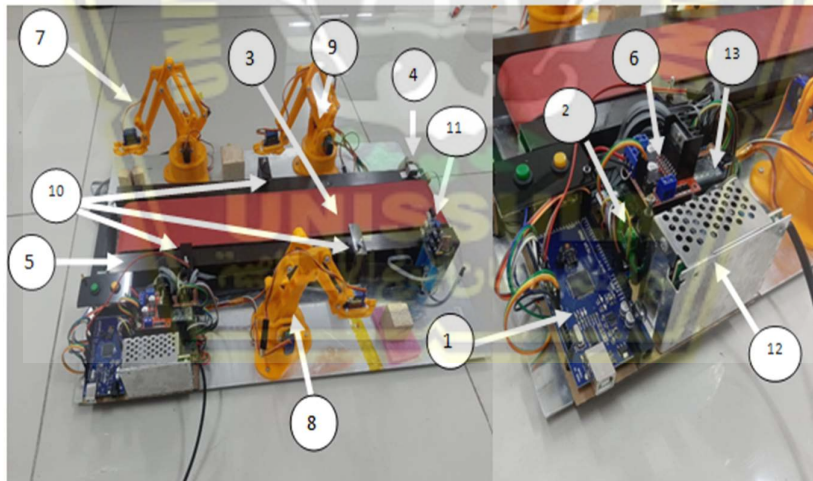


## BAB IV

### HASIL PENELITIAN DAN PEMBAHASAN

#### 4.1. Hasil Perancangan Perangkat Keras

Perangkat keras pada *prototype* kendali lengan robot berbasis android untuk otomasi *lifting* barang berdasarkan ukuran pada konveyor berjalan terdiri dari beberapa bagian, yaitu: belt konveyor, lengan robot, sensor serta catu daya. Dimana dalam belt konveyor terdiri dari motor DC gearbox, rangka, roller konveyor, sabuk konveyor serta motor driver DC. Sedang pada lengan robot terdiri dari 4 DOF dengan menggunakan masing-masing 4 buah motor servo SG90. Sedangkan sensor yang digunakan ada enam modul infrared FC-51. Catu daya sendiri menggunakan catu daya tunggal 5 volt. Bagian terakhir dari *prototype* ini yaitu modul bluetooth HC-05 yang berfungsi untuk menghubungkan *prototype* dengan perangkat smartphone berbasis android. Gambar 4.1 memperlihatkan hasil pembuatan *prototype* beserta bagian-bagian dari *prototype* tersebut. Sedangkan pada Tabel 4.1 memperlihatkan nama-nama komponen dari *prototype* beserta fungsinya.



Gambar 4.1. Hasil perancangan keseluruhan alat

Tabel 4.1. Bagian prototype berserta fungsinya

No	Nama Bagian	Fungsi
1.	Arduino mega 2560	Mikrokontroler 36able36 secara keseluruhan
2.	SC 32 ATmega 168	Pengontrol motor servo pada lengan robot
3	Belt Konveyor	Sabuk pemindah benda
4.	Motor DC gearbox	Penggerak belt konveyor
5.	Rangka konveyor	Tempat belt konveyor
6.	Motor DC driver L298N	Mengatur kecepatan putar belt konveyor
7.	Lengan robot 1	Mengambil benda kategori kecil
8.	Lengan robot 2	Mengambil benda kategori sedang
9.	Lengan robot 3	Mengambil benda kategori besar
10.	Modul Infra red FC-51	Kontrol capit lengan robot dalam mengambil benda
11.	Modul Infra red FC-51	Membaca besar kecilnya benda
12.	Catu Daya 5 Volt	Catu daya 36able3636or36 Arduino Mega 2560 dan motor servo
13.	Modul Bluetooth HC-05	Koneksi perangkat Android

Secara keseluruhan rangkaian elektronika dikontrol oleh Arduino Mega 2560 yang mendapatkan tiga buah masukan, yaitu dari modul infrared FC-51 sensor ukuran benda, modul infrared FC-51 sensor lengan robot serta dari modul bluetooth HC-05 untuk komunikasi dengan perangkat android. Sedangkan outputnya ada dua yaitu pengendalian kecepatan putaran konveyor serta driver servo motor SC-32 untuk mengendalikan putaran motor servo pada lengan robot yang mengambil benda berdasarkan ukurannya dan meletakkan sesuai dengan kotak yang telah di sediakan.

#### 4.2. Hasil Pengujian Belt Konveyor

Pada penelitian ini laju kecepatan konveyor dibuat konstan yaitu dengan memberi tegangan 3,5 volt pada motor DC gearbox. Dari tegangan tersebut setelah diukur menggunakan tachometer didapat kecepatan putaran motor sebesar 39,6 RPM dan kecepatan sumbu konveyor sebesar 7,6 RPM. Sedang kecepatan putaran konveyor ketika di ukur menggunakan stopwatch sebesar 66 cm/menit.

#### 4.3. Pengujian Modul Infrared FC-51

Dalam penelitian ini ada 6 buah modul Infrared FC-51 yang digunakan, dimana 3 buah modul berfungsi membaca besar kecilnya benda yaitu IR 1, IR 2 dan IR3 serta 3 buah modul lainnya berfungsi untuk menentukan titik henti benda berdasarkan ukurannya untuk diambil oleh masing-masing lengan robot yaitu IR 4 untuk benda besar, IR 5 untuk benda sedang dan IR 6 untuk benda kecil. Untuk infrared yang membaca besar kecilnya benda, ketika membaca benda kecil hanya LED IR 1 yang indikatornya menyala, ketika membaca benda sedang hanya IR 1 dan IR 2 yang indikator LEDnya menyala serta ketika membaca benda besar maka indikator LED IR 1, IR 2 dan IR 3 akan menyala.

Ketika ukuran benda sudah terdeteksi berdasar 3 kategori (kecil, sedang dan besar), maka benda akan terus berjalan di atas konveyor. Dimana ketika benda terdeteksi kecil maka benda akan berhenti tepat di depan sensor IR 6, ketika benda terdeteksi sedang maka benda akan berhenti di depan IR 5 dan terakhir ketika benda terdeteksi besar maka benda akan berhenti tepat di depan sensor IR 4. Hal ini terlihat seperti pada Tabel 4.2.

Tabel 4.2. Hasil Pengujian Sensor Infrared

No	Ukuran Benda	Indikator sinyal IR 1	Indikator sinyal IR 2	Indikator sinyal IR 3	Sensor Infrared 4	Sensor Infrared 5	Sensor Infrared 6
1.	kecil	ON	OFF	OFF	-	-	benda berhenti
2.	sedang	ON	ON	OFF	-	Benda berhenti	-
3.	besar	ON	ON	ON	Benda berhenti	-	-

#### 4.4. Hasil Pengujian Motor Servo

Pengujian motor servo pada lengan robot dilakukan bertujuan untuk melihat apakah lengan robot mampu melakukan *lifting* benda di atas konveyor dan menempatkan benda tersebut ketempat yang sudah disediakan sesuai dengan besar kecilnya benda, serta mengukur masing-masing motor servo lengan robot yang meliputi sudut pergerakan serta waktu yang dibutuhkan masing-masing motor servo dalam bergerak. Dimana lengan robot 1 bertugas mengambil benda dengan ukuran kecil, lengan robot 2 bertugas mengambil

benda dengan ukuran sedang serta lengan robot 3 bertugas mengambil benda dengan ukuran besar. Dari masing-masing lengan robot, motor servo 1 merupakan motor servo untuk capit dan lepas, motor servo 2 merupakan motor servo untuk dorong maju dan mundur, motor servo 3 merupakan motor servo untuk angkat dan turun serta motor servo 4 merupakan motor servo untuk putar. Gambar 4.2 merupakan program pengaturan sudut gerak motor servo pada lengan robot pada posisi standby.

```
}  
void standby(){  
  lengan1Standby();  
  lengan2Standby();  
  lengan3Standby();  
}  
void lengan1Standby(){  
  SSC_Kirim("#0P750#1P750 #2P750 #3P750 T1000");  
}  
void lengan2Standby(){  
  SSC_Kirim("#4P750#5P750 #6P750 #7P750 T1000");  
}  
void lengan3Standby(){  
  SSC_Kirim("#8P750#9P750 #10P750 #11P750 T1000");  
}
```

Gambar 4.2 Program sudut motor servo posisi standby

Dari Gambar 4.2 tersebut terlihat bahwa semua motor servo pada masing masing lengan robot *disetting* dengan lebar pulsa yang sama yaitu 750 ms dimana berdasarkan rumus konversi ke sudut besarnya sudut semua motor servo pada saat posisi *standby* adalah  $-67,5^{\circ}$ .

```

}
void lengan1Run()
{
String
dataFx=ssc_format(angkat,"1400","500");
SSC_Kirim(dataFx);delay(600);
dataFx=ssc_format(muter,"1680","1900");
SSC_Kirim(dataFx);delay(2000);
dataFx=ssc_format(capit,"1100","500");
SSC_Kirim(dataFx);
delay(600);
dataFx=ssc_format(dorong,"500","900");
SSC_Kirim(dataFx);
delay(1000);
dataFx=ssc_format(angkat,"1000","500");
SSC_Kirim(dataFx);delay(600);
dataFx=ssc_format(dorong,"1000","900");
SSC_Kirim(dataFx);
delay(1000);
dataFx=ssc_format(capit,"750","500");
SSC_Kirim(dataFx);
delay(600);
dataFx=ssc_format(dorong,"700","600");
SSC_Kirim(dataFx);
delay(700);
dataFx=ssc_format(angkat,"1300","500");
SSC_Kirim(dataFx);delay(600);
dataFx=ssc_format(muter,"750","1900");
SSC_Kirim(dataFx);delay(1950);
dataFx=ssc_format(dorong,"850","900");
SSC_Kirim(dataFx);
delay(1000);
dataFx=ssc_format(angkat,"750","600");
SSC_Kirim(dataFx);delay(600);
dataFx=ssc_format(capit,"1500","500");
SSC_Kirim(dataFx);
delay(600);
lengan1Standby();
}

}
void lengan2Run()
{
String
dataFx=ssc_format(angkat2,"1400","500");
SSC_Kirim(dataFx);delay(600);
dataFx=ssc_format(muter2,"1650","1900");
SSC_Kirim(dataFx);delay(2000);
dataFx=ssc_format(capit2,"1300","500");
SSC_Kirim(dataFx);
delay(600);
dataFx=ssc_format(dorong2,"550","900");
SSC_Kirim(dataFx);
delay(1000);
dataFx=ssc_format(angkat2,"800","500");
SSC_Kirim(dataFx);delay(600);
dataFx=ssc_format(dorong2,"830","900");
SSC_Kirim(dataFx);
delay(1000);
dataFx=ssc_format(capit2,"900","500");
SSC_Kirim(dataFx);
delay(600);
dataFx=ssc_format(dorong2,"700","900");
SSC_Kirim(dataFx);
delay(1000);
dataFx=ssc_format(angkat2,"1300","500");
SSC_Kirim(dataFx);delay(600);
dataFx=ssc_format(muter2,"750","1900");
SSC_Kirim(dataFx);delay(2000);
dataFx=ssc_format(dorong2,"700","900");
SSC_Kirim(dataFx);
delay(1000);
dataFx=ssc_format(angkat2,"900","500");
SSC_Kirim(dataFx);delay(600);
dataFx=ssc_format(capit2,"1500","500");
SSC_Kirim(dataFx);
delay(600);
lengan2Standby();
}

}
void lengan3Run()
{
String
dataFx=ssc_format(angkat3,"1400","500");
SSC_Kirim(dataFx);delay(600);
dataFx=ssc_format(muter3,"1750","1900");
SSC_Kirim(dataFx);delay(2000);
dataFx=ssc_format(capit3,"1500","500");
SSC_Kirim(dataFx);
delay(600);
dataFx=ssc_format(dorong3,"500","900");
SSC_Kirim(dataFx);
delay(1000);
dataFx=ssc_format(angkat3,"900","500");
SSC_Kirim(dataFx);delay(600);
dataFx=ssc_format(dorong3,"800","900");
SSC_Kirim(dataFx);
delay(1000);
dataFx=ssc_format(capit3,"800","500");
SSC_Kirim(dataFx);
delay(600);
dataFx=ssc_format(dorong3,"700","900");
SSC_Kirim(dataFx);
delay(1000);
dataFx=ssc_format(angkat3,"1300","50");
SSC_Kirim(dataFx);delay(600);
dataFx=ssc_format(muter3,"750","1900");
SSC_Kirim(dataFx);delay(2000);
dataFx=ssc_format(dorong3,"700","900");
SSC_Kirim(dataFx);
delay(1000);
dataFx=ssc_format(angkat3,"900","500");
SSC_Kirim(dataFx);delay(600);
dataFx=ssc_format(capit3,"1500","500");
SSC_Kirim(dataFx);
delay(600);
lengan3Standby();
}

```

Gambar 4.3. Program sudut dan waktu putar motor servo lengan robot

Setelah dilakukan pengujian serta perhitungan sudut berdasar teori dan praktek pada masing-masing motor servo, maka didapat perbedaan nilai sudut seperti terlihat pada Tabel 4.3. Dari kedua nilai tersebut didapatkan nilai error berdasar perhitungan nilai teori dikurangi nilai praktek lalu hasilnya dibagi nilai teori dan hasilnya dikalikan 100 %.

Untuk perhitungan nilai teori serta hasil pengukuran praktek waktu yang dibutuhkan motor servo untuk mencapai titik sudut yang diinginkan diperlihatkan pada Tabel 4.4. Dimana pada tabel tersebut juga dihitung nilai selisih antara hasil perhitungan teori dengan hasil pengukuran dalam bentuk nilai error.

Tabel 4.3. Hasil pengujian posisi sudut motor servo

GERAK KE	LENGAN ROBOT 1			LENGAN ROBOT 2			LENGAN ROBOT 3			SERVO	
	Teori	Praktek	Error (%)	Teori	Praktek	Error (%)	Teori	Praktek	Error (%)		
1	-9°	-9,5°	5,5	-9°	-9,5°	5,5	-9°	-9,5°	5,5	3	
2	16,2°	17°	4,9	13,5°	13°	3,7	22,5°	23°	2,2	4	
3	-36°	-37°	2,7	-18°	-19°	5,5	0°	0°	0	1	
4	-90°	-89°	1,1	-85,5°	-85,5°	0	-90°	-91°	1,1	2	
5	-45°	-43°	4,4	-63°	-63°	0	-54°	-56°	3,7	3	
6	-45°	-44°	2,2	-60,3°	-60,5°	0,3	-63°	-65°	3,1	2	
7	-67,5°	-66°	2,2	-54°	-55,5°	2,7	-63°	-65°	3,1	1	
8	-72°	-72,5°	0,7	-72°	-72,5°	0,7	-72°	-73°	1,4	2	
9	-18°	-18,5°	2,7	-18°	-19°	5,5	-18°	-18,5°	2,7	3	
10	-67,5°	-69°	2,2	-67,5°	-68°	0,7	-67,5°	-68°	0,7	4	
11	-58,5°	-70,5°	2,1	-72°	-70°	2,7	-72°	-70°	2,7	2	
12	-67,5°	-66°	2,2	-54°	-52,5°	2,7	-54°	-53°	1,8	3	
13	0	0	0	0	0	0	0	0	0	1	
Rata-rata error			2,5				1,8				2,2

Tabel 4.4. Hasil pengujian waktu putar motor servo

GERAK KE	LENGAN ROBOT 1			LENGAN ROBOT 2			LENGAN ROBOT 3			SERVO	
	Teori (ms)	Praktek (ms)	Error (%)	Teori (ms)	Praktek (ms)	Error (%)	Teori (ms)	Praktek (ms)	Error (%)		
1	500	530	6	500	510	2	500	530	6	3	
2	1900	1960	3,1	1900	1970	3,6	1900	1960	3,1	4	
3	500	510	2	500	530	6	500	530	6	1	
4	900	920	2,2	900	950	5,5	900	920	2,2	2	
5	500	530	6	500	510	2	500	520	4	3	
6	900	910	1,1	900	950	5,5	900	950	5,5	2	
7	500	530	6	500	530	6	500	520	4	1	
8	900	940	4,4	900	950	5,5	900	950	5,5	2	
9	500	520	4	500	530	6	500	510	2	3	
10	1900	1970	3,6	1900	1960	3,6	1900	1970	3,6	4	
11	900	950	5,5	900	940	4,4	900	910	1,1	2	
12	500	520	4	500	530	6	500	520	4	3	
13	500	530	6	500	530	6	500	520	4	1	
Rata-rata error			4,1				4,8				3,9

#### 4.5. Hasil Pengujian Android

Pengujian modul bluetooth bertujuan untuk memastikan bahwa sistem bisa dikendalikan dengan perangkat smartphone berbasis android. Gambar 4.2. memperlihatkan tampilan yang ada di perangkat smartphone dimana ada dua tombol utama yang bisa dikendalikan yaitu tombol START dan STOP serta ada beberapa tampilan status dari perangkat smartphone yaitu status koneksi

perangkat smartphone dengan konveyor dimana status yang ditampilkan *connected* atau *non connected*, ukuran benda yang sedang berjalan di atas konveyor apakah kecil, sedang atau besar serta status konveyor apakah berjalan atau tidak berjalan.



Gambar 4.4. Tampilan aplikasi pada smartphone android

Dari pengujian yang dilakukan didapatkan hasil bahwa semua berjalan sesuai perencanaan tanpa menghadapi kendala sedikitpun sehingga konveyor dan lengan robot bisa dikendalikan dari jarak jauh menggunakan koneksi bluetooth seperti terlihat pada Tabel 4.5. Dalam pengujian ini jarak terjauh yang bisa dilakukan adalah 15 meter dari obyek penelitian.

Tabel 4.5. hasil ujicoba bluetooth

No	Status	Konveyor	Lengan Robot 1	Lengan Robot 2	Lengan Robot 3
1	Non connected	Berhenti	Tidak bergerak	Tidak bergerak	Tidak bergerak
2	Connected	Berhenti	Tidak bergerak	Tidak bergerak	Tidak bergerak
3	STOP	Berhenti	Tidak bergerak	Tidak bergerak	Tidak bergerak
4	START	Berjalan	Tidak bergerak	Tidak bergerak	Tidak bergerak
5	Kecil	Berjalan	Bergerak	Tidak bergerak	Tidak bergerak
6	Sedang	Berjalan	Tidak bergerak	Bergerak	Tidak bergerak
7	Besar	Berjalan	Tidak bergerak	Tidak bergerak	Bergerak

#### 4.6. Hasil dan Analisis Pengujian Prototype

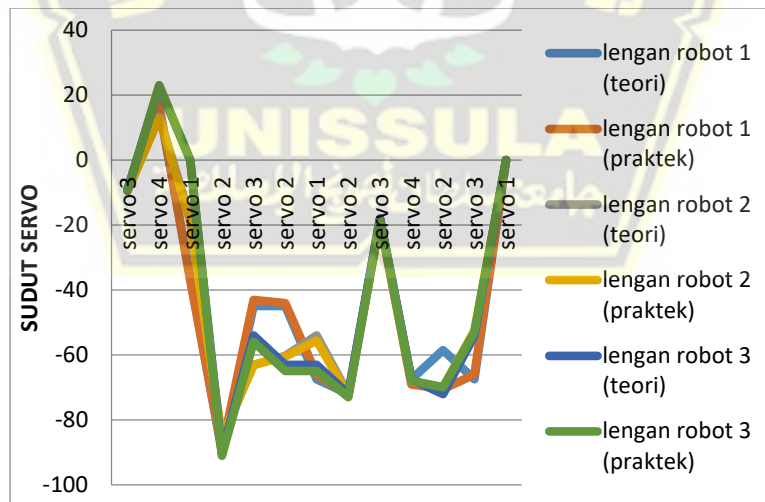
Dari hasil proses pengujian peralatan secara keseluruhan, baik pengujian dan pengukuran perangkat keras (*hardware*) maupun perangkat lunak (*software*), dapat disimpulkan bahwa alat yang dirancang sudah sesuai dengan spesifikasi yang diinginkan. Konveyor telah mampu berjalan dengan konstan yaitu mempunyai kecepatan 66 cm/menit. Sedangkan sensor infrared 1, 2 dan 3 telah mampu mengidentifikasi besar kecilnya benda yang berjalan di atas konveyor dengan sangat baik serta sensor infrared 4, 5 dan 6 telah mampu mendeteksi benda yang berjalan di atas konveyor untuk ditempatkan pada titik pengambilan lengan robot sesuai dengan ukuran benda dimana lengan robot 1 mengambil benda dengan ukuran kecil, lengan robot 2 mengambil benda dengan ukuran sedang serta lengan robot 3 mengambil benda dengan ukuran besar.

Gambar 4.5 memperlihatkan grafik hasil teori dengan praktek besar kecilnya sudut motor servo untuk masing masing pergerakan. Dari ketiga lengan robot dengan masing-masing lengan robot mempunyai 4 motor servo dalam setiap *lifting* benda. Dimana dari tiap lengan robot melakukan 13



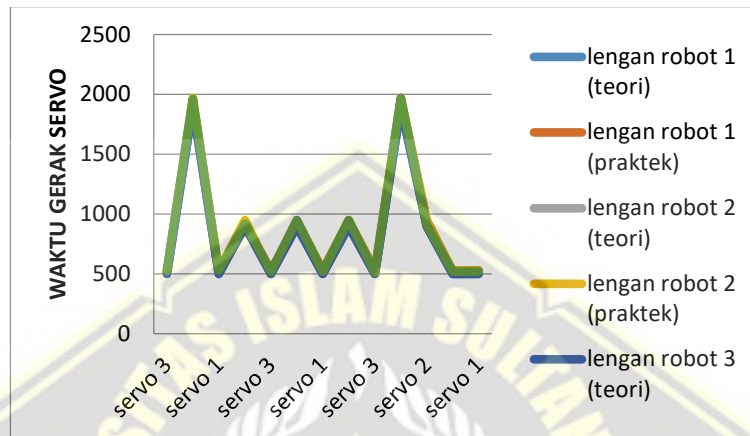
gerakan untuk menyelesaikan setiap *lifting* benda dengan urutan servo 3 (angkat), servo 4 (putar), servo 1 (buka capit), servo 2 (tarik), servo 3 (turun), servo 2 (dorong), servo 1 (tutup capit), servo 2 (dorong), servo 3 (angkat), servo 4 (putar), servo 2 (dorong), servo 3 (turun) dan terakhir servo 1 (buka capit). Dari 13 gerakan lengan robot terlihat hanya 1 sudut pergerakan lengan robot yang bernilai positif yaitu pergerakan ke 2 pada motor servo 4 (putar) dengan posisi sudut motor servo 4 lengan robot 1 adalah  $16,2^\circ$  untuk teori serta , motor servo 4 untuk lengan robot 2 adalah  $13,5^\circ$  untuk teori serta motor servo 4 untuk lengan robot 3 adalah sebesar  $22,5^\circ$ . Dimana kalau di dihitung dari sudut awal sebesar  $-67,5^\circ$  maka sudut putar dari servo 4 untuk lengan robot 1 adalah sebesar  $83,7^\circ$ , lengan robot 2 adalah sebesar  $81^\circ$  serta lengan robot 3 adalah sebesar  $90^\circ$ .

Dari hasil pengujian pergerakan sudut motor servo lengan robot didapatkan nilai rata-rata error sudut pergerakan motor servo pada lengan robot 1 adalah 2,5 % lengan robot 2 adalah 1,8 % serta lengan robot 3 adalah 2,2 %. Sedangkan hasil pengujian waktu gerak motor servo terlihat bahwa nilai rata-rata error untuk lengan robot 1 sebesar 4,1 %, lengan robot 2 sebesar 4,8 % serta lengan robot 3 sebesar 3,9 %.



Gambar 4.5. Grafik sudut motor servo

Pada Gambar 4.6 memperlihatkan hasil perhitungan teori dengan hasil pengukuran praktek besarnya waktu yang dibutuhkan oleh masing-masing pergerakan motor servo dalam berputar. Dari grafik tersebut terlihat bahwa antara perhitungan teori dengan pengukuran praktek selisihnya sangat kecil dan bisa dikatakan hampir mendekati sama, sehingga bisa disimpulkan hasil pengukuran sudut antara teori dengan praktek adalah sama dan sesuai.



Grafik 4.6. Grafik waktu gerak motor servo

## BAB V

### KESIMPULAN

Berdasarkan hasil pengujian tiap blok pada prototype dan pengujian sistem secara keseluruhan yang telah dilakukan, dapat disimpulkan beberapa hal sebagai berikut:

1. Bahwa dalam pengujian *prototype* didapat nilai error yang kecil antara nilai perhitungan teori dengan hasil pengukuran sudut dan waktu pergerakan motor untuk masing-masing lengan robot. Dimana dari hasil pengujian pergerakan sudut motor servo lengan robot didapatkan nilai rata-rata error sudut pergerakan motor servo pada lengan robot 1 adalah 2,5 % lengan robot 2 adalah 1,8 % serta lengan robot 3 adalah 2,2 %. Sedangkan hasil pengujian waktu gerak motor servo terlihat bahwa nilai rata-rata error untuk lengan robot 1 sebesar 4,1 %, lengan robot 2 sebesar 4,8 % serta lengan robot 3 sebesar 3,9 %.
2. Sistem kerja lengan robot untuk otomasi *lifting* barang berdasarkan ukuran pada konveyor berjalan berbasis android adalah konveyor berjalan dengan penggerak motor DC gearbox yang kecepatannya diatur oleh driver motor DC L298N. Ketika benda yang berjalan di atas konveyor melewati tiga buah sensor infrared FC-51, sensor akan mendeteksi ukuran benda apakah kecil, sedang atau besar. Ketika ukuran benda terdeteksi kecil, maka lengan robot 1 akan mengambil benda kecil, ketika benda terdeteksi sedang, maka lengan robot 2 akan mengambil benda sedang dan ketika benda terdeteksi besar maka lengan robot 3 akan mengambil benda besar.
3. Hasil pengujian lengan robot untuk *lifting* barang berdasar ukuran pada konveyor berjalan berbasis android didapat bahwa gerak konveyor sangat stabil. Dimana dengan pemberian tegangan 3.5 volt konveyor mampu berjalan dengan kecepatan 66 cm/menit. Disamping itu semua sensor infrared FC-51 telah mampu bekerja dengan baik yang ditandai dengan kemampuan untuk mendeteksi besar kecilnya benda serta mampu mendeteksi benda yang berjalan di atas konveyor untuk diambil lengan

robot sesuai dengan ukurannya. Hasil pengujian lengan robot didapat bahwa lengan robot mampu melakukan *lifting* benda dengan baik berdasar ukuran bendanya. Untuk pengujian *prototype* alat menggunakan perangkat smartphone berbasis android didapat bahwa *prototype* mampu dikendalikan dengan baik lewat bluetooth dengan jarak terjauh efektif adalah 15 meter dari *prototype*.

### 2.1. Saran

Perancangan dan pembuatan alat masih terdapat kelemahan dan kekurangan sistem. Beberapa hal yang dapat dijadikan saran pada alat ini adalah:

1. Sensor yang mendeteksi ukuran benda bisa menggunakan jenis sensor yang lain yang tidak terpengaruh sumber cahaya luar.
2. Aplikasi android perlu dikembangkan lagi sehingga dalam pengambilan benda yang berjalan di atas konveyor posisi konveyor bisa tetap berjalan, serta ketiga lengan robot mampu bekerja secara bersamaan.
3. Untuk menambah tingkat akurasi kerja lengan robot maka perlu ditambahkan logika fuzzy.

## DAFTAR PUSTAKA

- [1] Amelia, P. 2015. *Penerapan Radio Frekuensi Pada Robot Amphibi* <http://eprints.polsri.ac.id/2052/2/2>. Diakses tanggal 4 Mei 2021.
- [2] Rizkinaswara, L. 2020. *Revolusi Industri 4.0* <https://aptika.kominfo.go.id/2020/01/revolusi-industri-4-0/>. Diakses tanggal 3 Mei 2021.
- [3] R. H. Prabanegara, M. F. Noor, and E. Kunia. 2015. *Rancang Bangun Robot Lengan Pemindah dan Penyeleksi Barang Berdasarkan Warna Berbasis Arduino Uno*. Energy, vol. 5, no. 2, pp. 31–40, 2015.
- [4] P. Prasetyawan, Y. Ferdianto, S. Ahdan, and F. Trisnawati. 2018. *Pengendali Lengan Robot Dengan Mikrokontroler Arduino Berbasis Smartphone*. J. Tek. Elektro ITP, vol. 7, no. 2, pp. 104–109, 2018, doi: 10.21063/jte.2018.3133715.
- [5] Hasan, I. 2015. *Aplikasi pengenalan objek untuk lengan robot pemisah benda berdasarkan bentuk benda*. Yogyakarta : Universitas Sanata Dharma
- [6] S. Syahririni and H. Kurniawan. 2018. *Seleksi Benda Berwarna dengan Conveyor Menggunakan Robot Lengan*. Sidoarjo : Universitas Muhammadiyah
- [7] F. Ramadhan and T. Ta'ali. 2020. *Perancangan Penyortiran Barang Berdasarkan Berat dengan Sistem Pick And Place Berbasis Mikrokontroler*. JTEV (Jurnal Tek. Elektro dan Vokasional),
- [8] E. Ramadhani. 2019. *Perancangan Robot Lengan Pemilah Obyek Sesuai Warna Berbasis Arduino Mega 2560*. Journal Maestro, vol. 2, no. 2, pp. 432–439, 2019.
- [9] K. Sarware, A. Namdeo, A. Dwivedi, A. Shukla, and P. Soni. 2020. *Automated Pick And Throw Robotic Arm From Conveyer Belt*. International Research Journal of Engineering and Tecnology
- [10] E. S. P. Ginting. 2016. *Perancangan Robot Pemindah Barang Berbasis Mikrokontroler Atmega8535 Dengan Kendali Smartphone Android*. Sumatera Utara. Universitas Sumatera Utara.
- [11] A. Darmawan, N. S. Salahuddin, and M. Karjadi. 2018. *Prototipe Alat Pemindah Barang Di Pelabuhan Berbasis Arduino*. Jurnal Ilmiah Teknologi dan Rekayasa, vol. 23, no. 2, pp. 103–111.
- [12] G. T. Wardana, D. E. Setiawan, A. Rahman, and N. Prasetia. 2014. *Robot Lengan Pemindah Barang Berdasarkan Ukurannya Berbasis Mikrokontroler*. Palembang. Amik MDP Palembang.
- [13] H. P. Putra. 2013. *Dengan Robot Lengan Perancangan Simulasi Alat Bantu Pemindah Barang Pada Industri Manufaktur*. Surakarta. Universitas Muhamadiyah Surakarta.
- [14] Asgi Maldy. 2018. *Memahami Lengan Robot (Arm Robot) / Modular Production System (mps)*. <http://elektronikacafe.blogspot.com/2018/10/> Diakses 2 Mei 2021.
- [15] Ardutech. 2021. *Kontrol Robot Arm dengan Arduino*. <https://www.ardutech.com/kontrol-robot-arm-dengan-arduino/>. Diakses 2 Mei 2021.

- [16] A. D. Dwipanegara and A. R. Zulviana. 2020. *Pembuatan Prototype Alat Bantu Simulasi Attitude and Heading Reference System (Ahrs) Pada Avionic Laboratory PT Dirgantara*. Bandung. Universitas Nurtanio Bandung.
- [17] Edukasi Elektronika. 2020. *Motor Servo SG90 - Edukasi Elektronika | Electronics Engineering Solution and Education*. <https://www.edukasielektronika.com/2020/12/motor-servo-sg90.html> Diakses 2 Juli 2021.
- [18] Firman. *Tentang PWM (Pulse Width Modulation)*. <https://kl301.ilearning.me/2015/05/19/tentang-pwm-pulse-width-modulation/>. Diakses 16 Juli 2021.
- [19] D. Caysar, G. D. Nusantoro, and E. Yudaningsy. 2015. *Pengaturan Pergerakan Robot Lengan Smart Arm Robotic Ax-12a Melalui Pendekatan Geometry Based Kinematic Menggunakan Arduino*. Malang. Universitas Brawijaya.
- [20] Arduino. 2021. *Arduino Mega 2560 Rev3*. <https://store.arduino.cc/usa/mega-2560-r3> Diakses 3 Juni 2021.
- [21] Admin. 2018. *Mengenal Conveyor Automation Beserta Bagian Penting nya*. <https://www.dnm.co.id/conveyor/> Diakses 2 Mei 2021.
- [22] Ctec. 2021. *Mengenal Conveyor Belt* <https://ctec.co.id/mengenal-conveyor-belt/>. Diakses 2 Mei 2021.
- [23] Robotic Learning Center. 2021. *Motor Gearbox Roda*. <http://nextsys.web.id/edukasi/produk/lainnya/motor-dan-gearbox>. Diakses 6 Juli 2021.
- [24] Farid Urrohman. 2020. *Sensor Infrared (IR) Proximity FC-51* <https://www.edukasielektronika.com/2020/09/sensor-infrared-ir-proximity-fc-51.html> Diakses 3 Mei 2021.
- [24] Farid Urrohman. 2020. *Sensor Infrared (IR) Proximity FC-51* <https://www.edukasielektronika.com/2020/09/sensor-infrared-ir-proximity-fc-51.html> Diakses 4 Mei 2021
- [26] I. Chayati. 2007. *Pengenalan Multipied Robot 2 DOF*. Yogyakarta. Universitas Negeri Yogyakarta.
- [27] Sergio Juan. 2012. *SSC 32*. <https://www.scribd.com/document/76917165/ssc-32>. Diakses 2 Juli 2021.
- [28] Agus Faudin. 2018. *Tutorial Arduino mengakses module Bluetooth HC-05*. <https://www.nyebarilmu.com/tutorial-arduino-module-bluetooth-hc-05/> Diakses 3 Mei 2021.
- [29] Msyefudin. 2018. *Membahas Bluetooth HC-05 Arduino dan Programnya*. <https://digitalapik.blogspot.com/2020/03/membahas-bluetooth-hc-05-arduino-dan.html>. Diakses 3 Mei 2021.
- [30] Andy Nugroho. 2020. *Sejarah Android dan Perkembangannya Dari Masa ke Masa - Qwords*. <https://qwords.com/blog/sejarah-android/>. Diakses 3 mei 2021.
- [31] Dickson Kho. 2020. *Prinsip Kerja DC Power Supply (Catu Daya / Adaptor)*. <https://teknikelektronika.com/prinsip-kerja-dc-power-supply-adaptor/> Diakses 3 Mei 2021.



## LAMPIRAN 1 : PROGRAM PADA ARDUINO

```
#include <Servo.h>
```

```
//===== Lengan =====/
```

```
#define capit "0"
```

```
#define dorong "1"
```

```
#define angkat "2"
```

```
#define muter "3"
```

```
#define capit2 "4"
```

```
#define dorong2 "5"
```

```
#define angkat2 "6"
```

```
#define muter2 "7"
```

```
#define capit3 "8"
```

```
#define dorong3 "9"
```

```
#define angkat3 "10"
```

```
#define muter3 "11"
```

```
//===== End Lengan =====//
```

```
//===== Sensor ===//
```

```
#define pinSensor1 A1
```

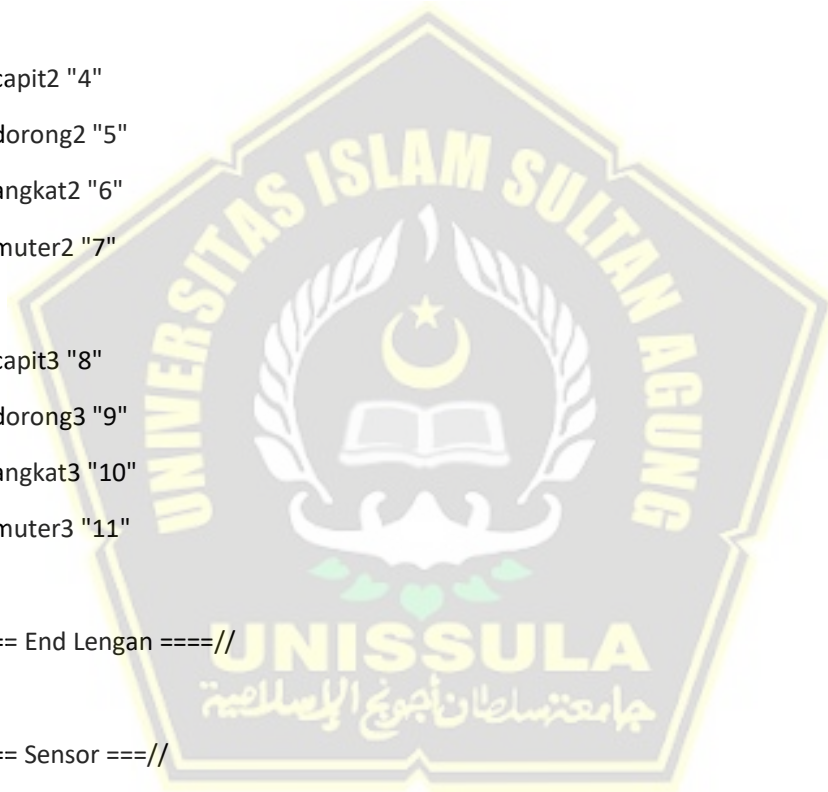
```
#define pinSensor2 A2
```

```
#define pinSensor3 A3
```

```
#define pinSensor4 A4
```

```
#define pinSensor5 2
```

```
#define pinSensor6 3
```





```

#define sensor1 digitalRead(pinSensor1)
#define sensor2 digitalRead(pinSensor2)
#define sensor3 digitalRead(pinSensor3)
#define sensor4 digitalRead(pinSensor4)
#define sensor5 digitalRead(pinSensor5)
#define sensor6 digitalRead(pinSensor6)

//===== End Sensor =====/

//===== Konveyor =====//
#define pinPwm 8
#define pinDirA 9
#define pinDirB 10
#define spdKonveyor 200

#define konveyorJalan analogWrite(pinPwm,spdKonveyor);
#define konveyorStop analogWrite(pinPwm,0);

//===== End Konveyor =====//

//===== Ultrasonic =====//

#define trigPin_ULTRASONIC 7
#define echoPin_ULTRASONIC 6

//===== Tombol ON/OFF =====//

#define pinButtonON A15
#define pinButtonOFF A14

#define button_ON digitalRead(A15)

```

```

#define button_OFF digitalRead(A14)

//=====//

boolean sistemON=true;

int sizeObject=0;

unsigned long previousMillis = 0;
const long interval = 200;

void initSensor(){
  pinMode(pinSensor1,INPUT_PULLUP);
  pinMode(pinSensor2,INPUT_PULLUP);
  pinMode(pinSensor3,INPUT_PULLUP);
  pinMode(pinSensor4,INPUT_PULLUP);
  pinMode(pinSensor5,INPUT_PULLUP);
  pinMode(pinSensor6,INPUT_PULLUP);
}

void initButton()
{
  pinMode(pinButtonON,INPUT_PULLUP);
  pinMode(pinButtonOFF,INPUT_PULLUP);
}

void initKonveyor(){
  pinMode(pinPwm,OUTPUT);
  pinMode(pinDirA,OUTPUT);
  pinMode(pinDirB,OUTPUT);
}

```

```
digitalWrite(pinDirB,1);  
digitalWrite(pinDirA,0);  
}
```

```
void initSerial(){  
  Serial.begin(9600); // Serial Monitor  
  Serial3.begin(115200); // Serial SSC  
  Serial2.begin(9600); // Serial Bluetooth  
}
```

```
void serialEvent2(){
```

```
  char data=Serial2.read();  
  if(data=='A'){  
    konveyorJalan;  
    sistemON=true;  
  }if(data=='B'){  
    konveyorStop;  
    sistemON=false;  
  }  
}
```

```
}
```

```
void setup()  
{  
  initSensor();  
  initButton();  
}
```



```

initKonveyor();
initSerial();

standby();
delay(1100);
sistemON=false;

}

void loop()
{

if(button_ON==0){
  sistemON=true;
  konveyorJalan;
}
if(button_OFF==0){
  sistemON=false;
}
Serial.print(button_ON);Serial.print(" ");Serial.print(button_OFF);
Serial.println(sistemON);
unsigned long currentMillis = millis();

if (currentMillis - previousMillis >= interval) {
  // save the last time you blinked the LED
  previousMillis = currentMillis;

  androidKirim("
~"+String(sensor1)+String(sensor2)+String(sensor3)+String(sensor4)+String(sizeObject)+
String(sistemON)+"#");
}
}

```

```

}

if(sistemON==true){
    if((sensor1==0) && (sizeObject==3)){
        androidKirim("
~"+String(sensor1)+String(sensor2)+String(sensor3)+String(sensor4)+String(sizeObject)+
"#");
        delay(1800);
        konveyorStop;
        lengan1Run();
        konveyorJalan;
        sizeObject=0;
    }

    if((sensor2==0) && (sizeObject==1)){
        androidKirim("
~"+String(sensor1)+String(sensor2)+String(sensor3)+String(sensor4)+String(sizeObject)+
"#");
        delay(1300);
        konveyorStop;
        lengan2Run();
        konveyorJalan;
        sizeObject=0;
    }

    if((sensor3==0) && (sizeObject==2)){
        androidKirim("
~"+String(sensor1)+String(sensor2)+String(sensor3)+String(sensor4)+String(sizeObject)+
"#");
        delay(1300);
    }
}

```



```

    konveyorStop;
    lengan3Run();
    konveyorJalan;
    sizeObject=0;
}

if(sensor4==0){
    delay(2000);
    konveyorStop;
    //delay(200);
    readObj();
    //delay(200);

    androidKirim("
~"+String(sensor1)+String(sensor2)+String(sensor3)+String(sensor4)+String(sizeObject)+
"#");
    // androidKirim(" ~"+sizeObject+"#");
    konveyorJalan;
    delay(5000);
    //lengan3Run();
}
}

if(sistemON==false){
    konveyorStop;
    standby();
}
}

void standby(){

```

```
lengan1Standby();  
lengan2Standby();  
lengan3Standby();  
}
```

```
void lengan1Standby(){  
    SSC_Kirim("#0P750 #1P750 #2P750 #3P750 T1000");  
}
```

```
void lengan2Standby(){  
    SSC_Kirim("#4P750 #5P750 #6P750 #7P750 T1000");  
}
```

```
void lengan3Standby(){  
    SSC_Kirim("#8P750 #9P750 #10P750 #11P750 T1000");  
}
```

```
void lengan1Run()  
{  
    String dataFx=ssc_format(angkat,"1400","500");  
    SSC_Kirim(dataFx);delay(600);  
  
    dataFx=ssc_format(muter,"1750","1900");  
    SSC_Kirim(dataFx);delay(2000);  
  
    dataFx=ssc_format(capit,"1500","500");//buka  
    SSC_Kirim(dataFx);  
    delay(600);  
    //  
    dataFx=ssc_format(dorong,"500","900");  
    SSC_Kirim(dataFx);  
    delay(1000);  
}
```

```
dataFx=ssc_format(angkat,"900","500");//turunke  
SSC_Kirim(dataFx);delay(600);
```

```
dataFx=ssc_format(dorong,"800","900");  
SSC_Kirim(dataFx);  
delay(1000);
```

```
dataFx=ssc_format(capit,"800","500");//capiten  
SSC_Kirim(dataFx);  
delay(600);
```

```
dataFx=ssc_format(dorong,"700","900");  
SSC_Kirim(dataFx);  
delay(1000);
```

```
dataFx=ssc_format(angkat,"1300","500");//turunke  
SSC_Kirim(dataFx);delay(600);
```

```
dataFx=ssc_format(muter,"750","1900");//puter balik  
SSC_Kirim(dataFx);delay(2000);
```

```
dataFx=ssc_format(dorong,"700","900");  
SSC_Kirim(dataFx);  
delay(1000);
```

```
dataFx=ssc_format(angkat,"900","500");//turunke  
SSC_Kirim(dataFx);delay(600);
```



```

dataFx=ssc_format(capit,"1500","500");//lepas
SSC_Kirim(dataFx);
delay(600);

lengan1Standby();

}

void lengan2Run()
{
String dataFx=ssc_format(angkat2,"1400","500");
SSC_Kirim(dataFx);delay(600);

dataFx=ssc_format(muter2,"1680","1900");
SSC_Kirim(dataFx);delay(2000);

dataFx=ssc_format(capit2,"1100","500");//buka
SSC_Kirim(dataFx);
delay(600);
//
dataFx=ssc_format(dorong2,"500","900");
SSC_Kirim(dataFx);
delay(1000);

dataFx=ssc_format(angkat2,"1000","500");//turunke
SSC_Kirim(dataFx);delay(600);

dataFx=ssc_format(dorong2,"1000","900");
SSC_Kirim(dataFx);

```

```
delay(1000);
```

```
dataFx=ssc_format(capit2,"750","500");//capiten
```

```
SSC_Kirim(dataFx);
```

```
delay(600);
```

```
dataFx=ssc_format(dorong2,"700","600");
```

```
SSC_Kirim(dataFx);
```

```
delay(700);
```

```
dataFx=ssc_format(angkat2,"1300","500");//turunke
```

```
SSC_Kirim(dataFx);delay(600);
```

```
dataFx=ssc_format(muter2,"750","1900");//puter balik
```

```
SSC_Kirim(dataFx);delay(1950);
```

```
dataFx=ssc_format(dorong2,"850","900");
```

```
SSC_Kirim(dataFx);
```

```
delay(1000);
```

```
dataFx=ssc_format(angkat2,"750","600");//turunke
```

```
SSC_Kirim(dataFx);delay(600);
```

```
dataFx=ssc_format(capit2,"1500","500");//lepas
```

```
SSC_Kirim(dataFx);
```

```
delay(600);
```

```
lengan2Standby();
```

```
}
```

```

void lengan3Run()
{
    String dataFx=ssc_format(angkat3,"1400","500");
    SSC_Kirim(dataFx);delay(600);

    dataFx=ssc_format(muter3,"1650","1900");
    SSC_Kirim(dataFx);delay(2000);

    dataFx=ssc_format(capit3,"1300","500");//buka
    SSC_Kirim(dataFx);
    delay(600);
    //
    dataFx=ssc_format(dorong3,"550","900");
    SSC_Kirim(dataFx);
    delay(1000);

    dataFx=ssc_format(angkat3,"800","500");//turunke
    SSC_Kirim(dataFx);delay(600);

    dataFx=ssc_format(dorong3,"830","900");
    SSC_Kirim(dataFx);
    delay(1000);

    dataFx=ssc_format(capit3,"900","500");//capiten
    SSC_Kirim(dataFx);
    delay(600);

    dataFx=ssc_format(dorong3,"700","900");

```

```
SSC_Kirim(dataFx);
```

```
delay(1000);
```

```
dataFx=ssc_format(angkat3,"1300","500");//turunke
```

```
SSC_Kirim(dataFx);delay(600);
```

```
dataFx=ssc_format(muter3,"750","1900");//puter balik
```

```
SSC_Kirim(dataFx);delay(2000);
```

```
dataFx=ssc_format(dorong3,"700","900");
```

```
SSC_Kirim(dataFx);
```

```
delay(1000);
```

```
dataFx=ssc_format(angkat3,"900","500");//turunke
```

```
SSC_Kirim(dataFx);delay(600);
```

```
dataFx=ssc_format(capit3,"1500","500");//lepas
```

```
SSC_Kirim(dataFx);
```

```
delay(600);
```

```
lengan3Standby();
```

```
}
```

```
void lengan1Run_BalokSedeng()
```

```
{
```

```
String dataFx=ssc_format(angkat,"1400","500");
```

```
SSC_Kirim(dataFx);delay(600);
```

```
dataFx=ssc_format(muter,"1720","1900");
```

```
SSC_Kirim(dataFx);delay(2000);
```

```
dataFx=ssc_format(capit,"1700","500");//buka
```

```
SSC_Kirim(dataFx);
```

```
delay(600);
```

```
//
```

```
dataFx=ssc_format(dorong,"500","900");
```

```
SSC_Kirim(dataFx);
```

```
delay(1000);
```

```
dataFx=ssc_format(angkat,"1000","500");//turunke
```

```
SSC_Kirim(dataFx);delay(600);
```

```
dataFx=ssc_format(dorong,"800","900");
```

```
SSC_Kirim(dataFx);
```

```
delay(1000);
```

```
dataFx=ssc_format(capit,"800","500");//capiten
```

```
SSC_Kirim(dataFx);
```

```
delay(600);
```

```
dataFx=ssc_format(angkat,"1300","500");//turunke
```

```
SSC_Kirim(dataFx);delay(600);
```

```
dataFx=ssc_format(muter,"750","1900");//puter balik
```

```
SSC_Kirim(dataFx);delay(2000);
```

```
dataFx=ssc_format(dorong,"700","900");
```



```

SSC_Kirim(dataFx);

delay(1000);

dataFx=ssc_format(angkat,"900","500");//turunke
SSC_Kirim(dataFx);delay(600);

dataFx=ssc_format(capit,"1500","500");//lepas
SSC_Kirim(dataFx);
delay(600);

lengan1Standby();

```

```

}

```

```

String ssc_format(String port, String Speda, String waktu){
    String Ffix="#" + port + " P" + Speda;
    if(waktu!=""){
        Ffix=Ffix+ " T" + waktu;
    }
    return Ffix;
}

```

```

void SSC_Kirim(String data_ssc){
    Serial3.println(data_ssc);
}

```

```

void androidKirim(String datas)
{
    Serial2.print(datas);
}

```



```
}
```

```
float baca_ULTRASONIC()
```

```
{
```

```
int co;
```

```
long duration, distance;
```

```
float jarak_UL=0;
```

```
for(co=0;co<10;co++)
```

```
{
```

```
digitalWrite(trigPin_ULTRASONIC, LOW); // Added this line
```

```
delayMicroseconds(2); // Added this line
```

```
digitalWrite(trigPin_ULTRASONIC, HIGH);
```

```
delayMicroseconds(10); // Added this line
```

```
digitalWrite(trigPin_ULTRASONIC, LOW);
```

```
duration = pulseIn(echoPin_ULTRASONIC, HIGH);
```

```
// distance = distance+ (duration/2) / 29.1;
```

```
distance = distance+(duration*0.034/2);
```

```
delay(10);
```

```
}
```

```
distance=distance/co;
```

```
//if(distance>200){distance=200;}
```

```
jarak_UL=distance;
```

```
// jarak_UL=duration*0.034/2;
```

```
return jarak_UL;
```

```
}
```

```
void readObj()
```

```
{
```

```

if(sensor4==0 && sensor5==0 && sensor6==0){
    sizeObject=3;
    goto keluar;
}
if(sensor4==0 && sensor5==0)
{
    sizeObject=2;
    goto keluar;
}
if(sensor4==0){
    sizeObject=1;
    goto keluar;
}
keluar;;
}
void search(){
    float minVal,maxVal,avrVal;
    int co;
    float avr;
    boolean ft=true;
    for(co=0;co<10;co++){
        float datas=baca_ULTRASONIC();
        avr=avr+datas;
        if(ft==true){minVal=datas;}else{
            if(datas<minVal){
                minVal=datas;
            }
        }
    }
    if(datas>maxVal){

```





```

    maxVal=datas;
}
delay(1);
}
avrVal=avr/(co);

if(avrVal>=4.5 && avrVal<=5.9){
    sizeObject=3;
}
else if(avrVal>5.9 && avrVal<=7){
    sizeObject=2;
}else{
    sizeObject=1;
}
Serial.print(avrVal);Serial.print(" ");
Serial.print(minVal);Serial.print(" ");
Serial.print(maxVal);Serial.print(" ");
Serial.print(sizeObject);Serial.print(" ");
Serial.println();

/* if((avrVal>=3.6) && (avrVal<=4.5))
{
    sizeObject="Kecil";
}
if(*/
}

```



## LAMPIRAN 2 :PROGRAM BASIC 4 ANDROID

```
#Region Project Attributes
    #ApplicationLabel: Conveyor
    #VersionCode: 1
    #VersionName:
    'SupportedOrientations possible values: unspecified, landscape or
portrait.
    #SupportedOrientations: unspecified
    #CanInstallToExternalStorage: True
#End Region
```

```
#Region Activity Attributes
    #FullScreen: True
    #IncludeTitle: True
    #BridgeLogger:true
#End Region
```

### **Sub Process Globals**

'These global variables will be declared once when the application starts.

'These variables can be accessed from all modules.

```
Dim AStreams As AsyncStreams
Dim Serial1 As Serial
Dim connected As Boolean
```

End Sub

### **Sub Globals**

'These global variables will be redeclared each time the activity is created.

'These variables can only be accessed from this module.

```
Dim Twriter As TextWriter
```

```
Private S_Conv As SeekBar
Private L_Conv As Label
Private LK As Label
Private P_BL As Panel
Private D_Object As EditText
Private L_Start As Label
Private B_START As Button
Private B_STOP As Button
```

```

Private indInfrared1 As Panel
Private indInfrared3 As Panel
Private indInfrared4 As Panel
Private indInfrared2 As Panel
Private tempText As EditText
End Sub

```

```

Sub Activity_Create(FirstTime As Boolean)
'Do not forget to load the layout file created with the visual designer.

```

```

For example:
Activity.LoadLayout("Layout")
Serial1.Initialize("Serial1")

```

```

End Sub

```

```

Sub Activity_Resume

```

```

If Serial1.IsEnabled = False Then
Msgbox("Silahkan Hidupkan Bluetooth", "")
Else
Serial1.Listen
End If

```

```

End Sub

```

```

Sub Activity_Pause (UserClosed As Boolean)

```

```

If UserClosed Then
AStreams.Close
End If

```

```

End Sub

```

```

Sub Serial1_connected(success As Boolean)

```

```

If success Then
ToastMessageShow("Koneksi Berhasil", False)
Twriter.Initialize(Serial1.OutputStream)

```

```

AStreams.Initialize(Serial1.InputStream,Serial1.OutputStream,"AStreams")

```

```

connected = True
P_BL.Color=Colors.ARGB(100,0,255,0)
LK.Text="Connected"

```

```

Else
connected = False

```

```

        MsgBox>LastException.Message, "koneksi gagal")
        P_BL.Color=Colors.ARGB(100,255,0,0)
        LK.Text="Disconnected"
    End If
End Sub

Sub AStreams_NewData (Buffer() As Byte)
    Dim rxChar As String
    ' rxChar=rxChar&BytesToString(Buffer, 0, Buffer.Length, "UTF8")

    'Log(rxChar.SubString2(rxChar.IndexOf("~")+1,rxChar.IndexOf("#")))
    ' LoadRX(BytesToString(Buffer, 0, Buffer.Length, "UTF8"))

    LogMessage("",BytesToString(Buffer, 0, Buffer.Length, "UTF8"))
    tempText.Text=""

    ' If EditText1.Text.Contains("*") Then
    ' EditText1.Text=""
    ' End If
    ' Log(BytesToString(Buffer, 0, Buffer.Length, "UTF8"))
    ' Dim dataRx As String
    ' dataRx=BytesToString(Buffer, 0, Buffer.Length, "UTF8")
    ' D_Object.Text=dataRx

End Sub

Sub LogMessage(From As String, Msg As String)
    Dim inf1,inf2,inf3,inf4,data_serial,unk,sisON As String
    tempText.Text= tempText.Text & From & Msg
    tempText.SelectionStart=tempText.Text.Length
    Log(tempText.Text)
    If tempText.Text.Contains("~") And tempText.Text.Contains("#") Then

        unk=tempText.text.SubString2(tempText.Text.IndexOf("~")+1,tempText.Text.IndexOf("#"))& "#"
        Log(unk)
        inf1=unk.SubString2(0,1)
        inf2=unk.SubString2(1,2)
        inf3=unk.SubString2(2,3)
        inf4=unk.SubString2(3,4)
        data_serial=unk.SubString2(4,5)
        sisON=unk.SubString2(5,6)
        If inf1="0" Then
            indInfrared1.Color=Colors.Green
        Else if inf1="1" Then
            indInfrared1.Color=Colors.White
        End If
    End If
End Sub

```

```

End If
If inf2="0" Then
    indInfrared2.Color=Colors.Green
Else if inf2="1" Then
    indInfrared2.Color=Colors.White
End If

If inf3="0" Then
    indInfrared3.Color=Colors.Green
Else if inf3="1" Then
    indInfrared3.Color=Colors.White
End If
If inf4="0" Then
    indInfrared4.Color=Colors.Green
Else if inf4="1" Then
    indInfrared4.Color=Colors.White
End If
If data_serial = "0" Then
    D_Object.Text="Tidak Terdeteksi"
End If
If data_serial = "1" Then
    D_Object.Text="Kecil"
End If
If data_serial = "2" Then
    D_Object.Text="Sedang"
End If
If data_serial = "3" Then
    D_Object.Text="Besar"
End If
If sisON = 0 Then
    L_Start.Text="Berhenti"
    L_Start.Color=Colors.ARGB(100,255,0,0)
End If
If sisON = 1 Then
    L_Start.Text="Berjalan"
    L_Start.Color=Colors.ARGB(100,0,255,0)
End If
End If

End If

'
data_int=data_serial

End Sub

```

**Sub B\_START\_Click**

```
' L_Start.Text="Berjalan"
' L_Start.Color=Colors.ARGB(100,0,255,0)
  If connected Then
    Twriter.WriteLine("A")
    Twriter.Flush
  Else
    MsgBox("Bluetooth belum terkoneksi", "Gagal")
  End If
End Sub
```

**Sub B\_STOP\_Click**

```
' L_Start.Text="Berhenti"
' L_Start.Color=Colors.ARGB(100,255,0,0)

  If connected Then
    Twriter.WriteLine("B")
    Twriter.Flush
  Else
    MsgBox("Bluetooth belum terkoneksi", "Gagal")
  End If
End Sub
```

**Sub P\_BL\_Click**

```
  If connected=False Then
    Dim PairedDevices As Map
    PairedDevices = Serial1.GetPairedDevices
    Dim List1 As List

    List1.Initialize
    For i = 0 To PairedDevices.Size - 1
      List1.Add(PairedDevices.GetKeyAt(i))
    Next
    Dim res As Int
    res = InputList(List1, "Pilih Bluetooth", -1)
    If res <> DialogResult.CANCEL Then
      Serial1.connect(PairedDevices.Get(List1.Get(res)))
    End If
  Else if connected Then
    Serial1.Disconnect
    connected=False
    AStreams.Close
    P_BL.Color=Colors.ARGB(100,255,0,0)
```

LK.Text="Disconnected"

End If

End Sub

