

RANCANG BANGUN ROBOT HUMANOID SENI TARI DAN SISTEM PEMROGRAMAN TERINTEGRASI

PROYEK AKHIR

Laporan akhir ini dibuat dan diajukan untuk memenuhi salah satu syarat kelulusan Sarjana Terapan Diploma IV Politeknik Manufaktur Negeri Bangka Belitung



Disusun Oleh :

Mardinata Indra Kristianto NIM : 1052216
Danny Pranata NIM : 1052235

**POLITEKNIK MANUFAKTUR NEGERI
BANGKA BELITUNG
TAHUN 2025**

LEMBAR PENGESAHAN

JUDUL PROYEK AKHIR

RANCANG BANGUN ROBOT HUMANOID SENI TARI DAN SISTEM PEMROGRAMAN TERINTEGRASI

Oleh:

Mardinata Indra Kristianto/1052216
Danny Pranata/1052235

Laporan akhir ini telah disetujui dan disahkan sebagai salah satu syarat kelulusan
Program Sarjana Terapan Politeknik Manufaktur Negeri Bangka Belitung

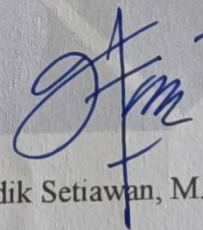
Menyetujui,

Pembimbing 1



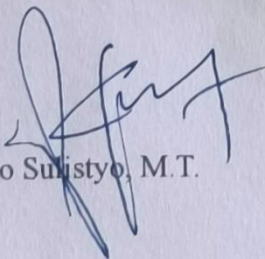
Indra Dwisaputra, M.T.

Pembimbing 2



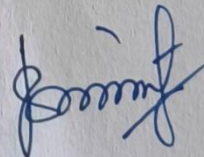
I Made Andik Setiawan, M.Eng., Ph.D,

Penguji 1



Eko Sulistyono, M.T.

Penguji 2



Yuke Mareta Ariesta Sandra, S.P., M.Si.

PERNYATAAN BUKAN PLAGIAT

Yang bertanda tangan di bawah ini:

Nama Mahasiswa 1 : Mardinata Indra Kristianto NIM: 1052216

Nama Mahasiswa 2 : Danny Pranata NIM: 1052235

Dengan Judul : RANCANG BANGUN ROBOT HUMANOID SENI
TARI DAN SISTEM PEMROGRAMAN
TERINTEGRASI

Menyatakan bahwa laporan akhir ini adalah hasil kerja kami sendiri dan bukan merupakan plagiat. Pernyataan ini kami buat dengan sebenarnya bila ternyata dikemudian hari ternyata melanggar pernyataan ini, kami bersedia menerima sanksi yang berlaku.

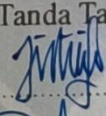
Sungailiat, 24 Juli 2025

Nama Mahasiswa

Tanda Tangan

1. Mardinata Indra Kristianto

2. Danny Pranata


.....

.....

ABSTRAK

Perkembangan teknologi yang begitu pesat, khususnya di bidang robotika telah memberikan sejuta peluang penerapan hingga ke berbagai sektor. Salah satunya pada pelestarian seni dan budaya lewat Kontes Robot Seni Tari Indonesia (KRSTI). Penelitian ini bertujuan untuk merancang, membangun, dan memprogram robot humanoid seni tari sesuai ketentuan yang berlaku pada Kontes Robot Seni Tari Indonesia (KRSTI) 2024. Robot humanoid ini memiliki 25 derajat kebebasan, dilengkapi modul Audio Bluetooth Receiver untuk mendeteksi irama musik yang dimainkan. Perangkat lunak berbasis Java Swing dibuat untuk kontrol servo robot dalam pemrograman bentuk gerak tari robot humanoid. Proses pelaksanaan meliputi perancangan struktur mekanik, sistem pengendalian servo, serta komunikasi antar robot. Hasil pengujian diperoleh robot mampu menampilkan gerak tari dengan durasi yang selaras dengan musik, berkomunikasi antar unit robot untuk melakukan gerak tari secara sinkron, serta perangkat lunak yang dikembangkan dapat membantu pengguna dalam proses rancang dan susun gerak tari dengan lebih presisi. Berdasarkan hasil tersebut dapat disimpulkan bahwa sistem ini dapat digunakan pada ajang kompetisi dan berpotensi sebagai media pembelajaran pelestarian budaya tradisional.

Kata kunci: *Java Swing, komunikasi robot, kontrol servo, KRSTI, robot humanoid*

ABSTRACT

The rapid development of technology, especially in the field of robotics, has provided a million opportunities for application in various sectors. One of them is in the preservation of arts and culture through the Kontes Robot Seni Tari Indonesia (KRSTI). This research aims to design, build, and program a humanoid dance robot according to the provisions applicable to the Kontes Robot Seni Tari Indonesia 2024 (KRSTI). This humanoid robot has 25 degrees of freedom, equipped with a Bluetooth Audio Receiver module to detect the rhythm of the music being played. Java Swing-based software was created for robot servo control in programming the humanoid robot's dance movements. The implementation process includes designing the mechanical structure, servo control system, and communication between robots. The test results obtained the robot is able to display dance movements with a duration that is in sync with the music, communicate between robot units to perform dance movements synchronously, and the developed software can assist users in the process of designing and composing dance movements with more precision. Based on these results, it can be concluded that this system can be used in competitions and has the potential to be a learning medium for preserving traditional culture.

Keywords: humanoid robot, java swing, KRSTI, robot communication, servo control

KATA PENGANTAR

Puji dan syukur penulis panjatkan atas karunia dan rahmat Tuhan Yang Maha Esa atas segala rezeki, nikmat, rahmat, dan hidayah sehingga Penulis dapat menyelesaikan laporan Proyek Akhir yang berjudul “Rancang Bangun Robot Humanoid Seni Tari dan Sistem Pemrograman Terintegrasi” ini dengan baik. Adapun tujuan dari laporan Proyek Akhir ini disusun adalah sebagai salah satu persyaratan dan kewajiban Mahasiswa untuk menyelesaikan kurikulum program Pendidikan Diploma IV (DIV) di Politeknik Manufaktur Negeri Bangka Belitung.

Selama pelaksanaan penelitian Proyek Akhir ini, penulis mengakui bahwa telah mendapatkan bantuan dan dukungan dari berbagai pihak dalam membuat alat maupun dalam menyelesaikan laporan Proyek Akhir ini dengan baik. Untuk itu Penulis mengucapkan terima kasih kepada:

1. Orang tua serta keluarga penulis yang selalu memberikan do’a, nasihat, motivasi, dukungan, dan karena keberadaan merekalah Penulis tetap semangat untuk menyelesaikan penelitian ini.
2. Bapak I Made Andik Setiawan, M.Eng., Ph.D. selaku Direktur Politeknik Manufaktur Negeri Bangka Belitung sekaligus dosen pembimbing 2 yang telah meluangkan waktu, tenaga serta pikiran dalam membimbing penulis pada penulisan laporan proyek akhir ini dan telah banyak pula memberikan saran-saran terhadap permasalahan yang penulis hadapi selama proses penyusunan proyek akhir.
3. Bapak Indra Dwisaputra, M.T. selaku pembimbing 1 yang telah meluangkan waktu, tenaga serta pikiran dalam membimbing penulis pada penulisan laporan proyek akhir ini dan telah banyak pula memberikan saran-saran terhadap permasalahan yang penulis hadapi selama proses penyusunan proyek akhir.
4. Bapak Catur Pebriandani, S.Tr.T. selaku pendamping setia penulis yang selalu memberikan dukungan, arahan, saran, serta semangat dalam menghadapi tantangan dan menjadikannya pengalaman yang sangat bermakna.

5. Bapak Zanu Saputra, M.Tr.T. selaku Ketua Jurusan Teknik Rekayasa Elektro dan Industri Pertanian Polman Babel.
6. Bapak Aan Febriansyah, M.T. selaku Ketua Program Studi Diploma IV Teknik Elektronika Polman Babel.
7. Seluruh staff pengajar dan karyawan di Polman Babel.
8. Teman-teman mahasiswa Polman Babel yang telah banyak membantu memberikan dukungan dan saran selama pembuatan laporan proyek akhir.
9. Pihak-pihak lain yang telah memberi bantuan baik secara langsung maupun tidak langsung yang tidak dapat disebutkan satu persatu.

Penulis memahami bahwa laporan proyek akhir ini masih jauh dari kata sempurna, baik dari segi isi maupun perencanaan karena keterbatasan hambatan yang dialami oleh penulis. Oleh karena itu, penulis sangat mengharapkan masukan dan saran yang berharga dari para pembaca sekalian agar dapat dijadikan bahan pemikiran penulis untuk lebih mengembangkan laporan proyek akhir ini. Mudah-mudahan laporan proyek akhir ini dapat bermanfaat dalam menambah pemahaman dan wawasan.

Sungailiat, 24 Juli 2025
Hormat kami,

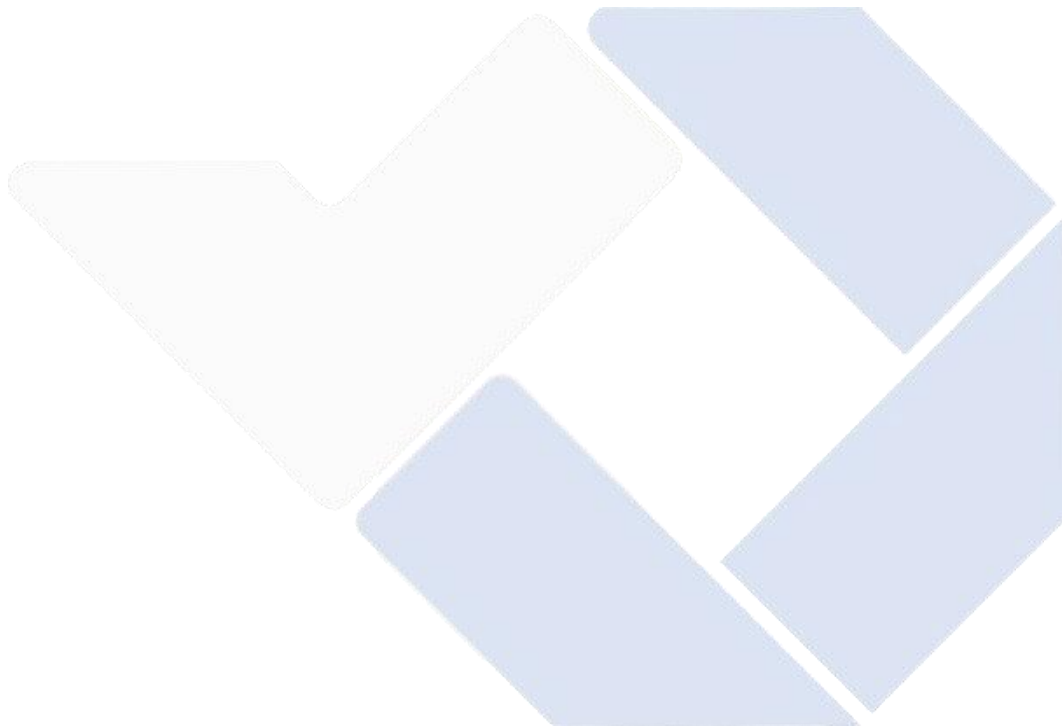
Penulis

DAFTAR ISI

	Halaman
LEMBAR PENGESAHAN	ii
PERNYATAAN BUKAN PLAGIAT	iii
ABSTRAK	iv
ABSTRACT	v
KATA PENGANTAR.....	vi
DAFTAR ISI	viii
DAFTAR TABEL.....	xi
DAFTAR GAMBAR	xii
DAFTAR LAMPIRAN	xiv
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	4
1.3 Tujuan Proyek Akhir	4
BAB II DASAR TEORI.....	5
2.1 Robot Humanoid	5
2.2 Kontes Robot Seni Tari Indonesia (KRSTI)	6
2.3 ESP32.....	7
2.4 Motor Servo Sonic SN30MG-PRO.....	8
2.5 Motor Servo MG996R	10
2.6 Servo MG90S.....	11
2.7 HC-12.....	12
2.8 UGREEN CM523	13

2.9	PCA9685	14
2.10	<i>Buck Converter</i> XL4016	15
2.11	<i>Java Swing</i>	16
2.12	Komunikasi Serial (USART)	17
2.13	Baterai <i>Lithium Polymer</i>	18
BAB III METODE PELAKSANAAN		19
3.1.	Desain 3D <i>Hardware</i>	19
3.2.	Perancangan dan Implementasi Rangkaian Elektrik.....	27
3.3.	Desain <i>Software</i>	30
3.4.	Desain Komunikasi Robot	32
3.5.	Pembuatan Rangkaian Elektrik dan PCB.....	33
3.6.	Pembuatan <i>Software</i>	35
3.7.	Proses Pembuatan Gerakan Tari.....	35
BAB IV PEMBAHASAN.....		39
4.1	Deskripsi Alat.....	39
4.2	Perakitan <i>Hardware</i> Robot.....	40
4.2.1	Instalasi Servo	41
4.2.2	Pemasangan ESP32.....	41
4.2.3	Pemasangan dan Perakitan Rangkaian Komponen Elektrik	42
4.3	Pengujian HC-12.....	42
4.4	Pengujian Sinyal ADC Terhadap Kondisi Suara Musik	45
4.5	Pengujian Data Serial Sudut Servo	47
4.6	Pengujian Komunikasi <i>Software</i> dengan Servo	51
4.7	Pengujian Keseluruhan.....	66
BAB V PENUTUP.....		75

5.1	Kesimpulan	75
5.2	Saran.....	76
DAFTAR PUSTAKA.....		77
LAMPIRAN.....		81



DAFTAR TABEL

Tabel	Halaman
Tabel 2.1 Perbandingan Arduino, NodeMCU, dan ESP32 [12]	8
Tabel 2.2 Spesifikasi Servo Sonic SN30MG-PRO[15]	9
Tabel 2.3 Spesifikasi Servo MG996R	11
Tabel 2.4 Spesifikasi Servo MG90S[20]	12
Tabel 2.5 Spesifikasi UGREEN CM523 [25].....	14
Tabel 2.6 Spesifikasi PCA9685 [27]	15
Tabel 3.1 Tabel dimensi robot	27
Tabel 3.2 Tabel hasil pengukuran arus total semua servo	29
Tabel 4.1 Tabel selisih waktu kirim dan terima HC-12	44
Tabel 4.2 Tabel data sinyal ADC	46
Tabel 4.3 Tabel Pengukuran Servo	49
Tabel 4.4 Tabel Pengujian Data Serial Sudut Servo	50
Tabel 4.5 Tabel Kondisi Awal Servo	52
Tabel 4.6 Tabel Pergerakan Servo Terhadap <i>Software</i>	59
Tabel 4.7 Tabel Pengujian Keseluruhan Servo Terhadap <i>Software</i>	65
Tabel 4.8 Tabel Data Deteksi Musik dari ESP1_ <i>Master</i>	67
Tabel 4.9 Tabel Data yang diterima dari ESP1_ <i>Slave</i>	67
Tabel 4.10 Tabel ESP2_ <i>Slave</i> saat menerima perintah dari ESP1_ <i>Slave</i>	68
Tabel 4.11 Pengujian ke-1 sinkronisasi gerakan kedua robot.....	69
Tabel 4.12 Rata-rata persentase error pengujian	70
Tabel 4.13 Sinkronisasi gerakan robot	70

DAFTAR GAMBAR

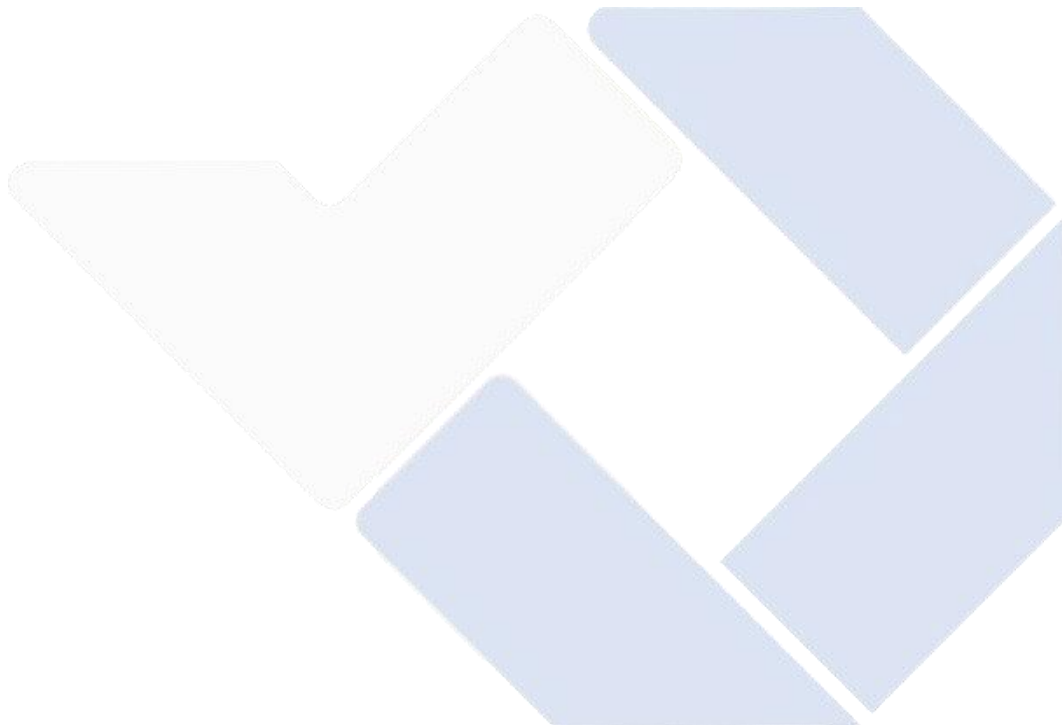
Gambar	Halaman
Gambar 1.1 Kontes Robot Seni Tari Indonesia	2
Gambar 2.1 Robot Humanoid.....	6
Gambar 2.2 Gambaran Pertandingan Kontes Robot Seni Tari Indonesia	7
Gambar 2.3 Mikrokontroler ESP32.....	8
Gambar 2.4 Motor Servo Sonic SN30MG-PRO	10
Gambar 2.5 Motor Servo MG996R.....	11
Gambar 2.6 Motor Servo MG90S	12
Gambar 2.7 HC-12	13
Gambar 2.8 <i>Audio Bluetooth dongle</i> UGREEN CM523	14
Gambar 2.9 PC9685	15
Gambar 2.10 <i>Buck Converter</i> XL4016.....	16
Gambar 2.11 Antarmuka <i>Java Swing</i> (Eclipse)	17
Gambar 2.12 Baterai Li-Po.....	18
Gambar 3.1 (a) Desain telapak kaki kanan (b) Desain telapak kaki kiri	19
Gambar 3.2 (a) Desain pergelangan kaki kanan (b) Desain pergelangan kaki kiri	20
Gambar 3.3 (a) Desain betis kanan (b) Desain betis kiri.....	20
Gambar 3.4 (a) Desain paha kanan (b) Desain paha kiri.....	20
Gambar 3.5 (a) Desain <i>Right Hip Pitch</i> (b) Desain <i>Left Hip Pitch</i>	21
Gambar 3.6 (a) Desain <i>Right Hip Roll</i> (b) Desain <i>Left Hip Roll</i>	21
Gambar 3.7 Desain Pinggang	22
Gambar 3.8 Desain Main <i>Body</i> Robot.....	22
Gambar 3.9 Desain Penutup <i>Body</i> Robot	22
Gambar 3.10 (a) Desain <i>Right Shoulder Pitch</i> (b) Desain <i>Left Shoulder Pitch</i> ..	23
Gambar 3.11 (a) Desain <i>Right Shoulder Roll</i> (b) Desain <i>Left Shoulder Roll</i>	23
Gambar 3.12 (a) Desain <i>Right Upper Arm</i> (b) Desain <i>Left Upper Arm</i>	24
Gambar 3.13 (a) Desain <i>Right Arm Roll</i> (b) Desain <i>Left Arm Roll</i>	24

Gambar 3.14 (a) Desain <i>Right Lower Arm</i> (b) Desain <i>Left Lower Arm</i>	24
Gambar 3.15 (a) Desain <i>Right Hand</i> (b) Desain <i>Left Hand</i>	25
Gambar 3.16 Desain <i>Head Pitch</i>	25
Gambar 3.17 Desain <i>Head Bracket</i>	26
Gambar 3.18 Desain Keseluruhan Robot Humanoid	26
Gambar 3.19 Diagram derajat kebebasan robot humanoid	27
Gambar 3.20 Rangkaian Elektrik Robot <i>Master</i>	28
Gambar 3.21 Rangkaian Elektrik Robot <i>Slave</i>	28
Gambar 3.22 Hasil pengukuran arus maksimum servo	29
Gambar 3.23 <i>Flowchart</i> Desain <i>Software</i>	30
Gambar 3.24 Komunikasi Antar Robot	33
Gambar 3.25 Schematic Kontroler ESP32(1) Robot <i>Master</i> dan <i>Slave</i>	33
Gambar 3.26 Schematic Kontroler ESP32(2) Robot <i>Master</i> dan <i>Slave</i>	34
Gambar 3.27 Hasil cetak PCB	34
Gambar 3.28 Tampilan <i>Software</i> Kontrol Servo	35
Gambar 3.29 Bagian Kontrol Sudut Servo pada <i>Software</i>	36
Gambar 3.30 Bagian Data Gerakan pada <i>Software</i>	37
Gambar 3.31 Pose Robot dengan <i>software</i> kontrol servo	38
Gambar 4.1 Flowchart sistem keseluruhan	40
Gambar 4.2 Instalasi Servo	41
Gambar 4.3 Pemasangan ESP32	42
Gambar 4.4 Pemasangan Rangkaian Elektrik	42
Gambar 4.5 (a). Rangkaian Pengujian <i>Master</i> Robot (b). Rangkaian Pengujian <i>Slave</i> Robot	43
Gambar 4.6 Grafik selisih waktu kirim dan terima HC-12	43
Gambar 4.7 Dokumentasi Pengujian HC-12	45
Gambar 4.8 (a) <i>Audio transmitter</i> terpasang pada laptop (b) <i>Audio receiver</i> terpasang pada robot humanoid	45
Gambar 4.9 Grafik sinyal ADC yang diterima <i>receiver</i>	46

DAFTAR LAMPIRAN

Lampiran 1 : Daftar Riwayat Hidup

Lampiran 2 : Tabel Pengujian Sinkronisasi Robot



BAB 1

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi pada kehidupan manusia saat ini begitu pesat, terutama pada elektronika. Kemajuan teknologi saat ini ditandai dengan diciptakannya berbagai peralatan yang dapat dikendalikan secara manual maupun otomatis. Perkembangan ini juga turut mendorong kemajuan dalam cabang ilmu elektronika, khususnya di bidang robotika. Perkembangan di bidang robotika ini juga dipengaruhi oleh meningkatnya kebutuhan yang banyak menggunakan teknologi robotika seperti bidang industri, sipil, medis, olahraga, manajemen bencana, hingga bidang kesenian. Tidak sedikit juga riset yang dilakukan pada bidang robotika, seperti robot humanoid atau robot yang menyerupai manusia [1].

Robot humanoid dikembangkan untuk memiliki karakteristik selayaknya manusia dari bentuk fisik hingga fungsionalnya. Hal ini tentunya menjadi keunggulan dan menjadikannya teknologi yang fleksibel untuk diaplikasikan diberbagai lingkup pekerjaan manusia seperti *Humanoid Robotics Project* (HRP) oleh *The Ministry of Economy, Trade and Industry* (METI) dari Jepang yang telah mengaplikasikan robot humanoid untuk melakukan pekerjaan manusia dalam *maintenance* mesin industri, menjadi pelayan manajemen di gedung dan rumah, melakukan eksperimen jarak jauh atau di tempat yang berbahaya [2].

Semakin banyak orang di Indonesia yang mengembangkan atau meneliti robot humanoid. Hal ini tidak lepas karena adanya kompetisi yang diselenggarakan di tingkat regional, nasional, hingga internasional. Kementerian Riset, Teknologi, dan Pendidikan Tinggi Republik Indonesia biasanya menyelenggarakan perlombaan yaitu Kontes Robot Indonesia (KRI) yang diadakan setahun sekali. KRI memiliki banyak divisi, salah satunya adalah Kontes Robot Seni Tari Indonesia (KRSTI) [3]. KRSTI sendiri adalah kompetisi robot yang menggabungkan elemen seni dan budaya Indonesia, terutama seni tari yang telah terkenal. Ajang ini difokuskan pada perancangan, pembuatan, dan pemrograman robot humanoid agar robot dapat melakukan beberapa gerakan tarian ketika alunan musik dimainkan dan

berhenti jika alunan musik di-stop yang dimana musik dihubungkan dari juri melalui *Audio Bluetooth Transmitter* dan dihubungkan ke robot melalui *Audio Bluetooth Receiver*. Dalam pertandingan KRSTI, robot harus melakukan tarian yang dimulai dari zona *Start*, zona A, zona B, zona C, dan zona *Finish* dimana robot harus melewati setiap zona untuk mendapatkan poin [4].



Gambar 1.1 Kontes Robot Seni Tari Indonesia

Perancangan robot humanoid yang dilaksanakan pada penelitian sebelumnya, salah satu aspek yang cukup signifikan adalah desain mekanik yang sampai saat ini tidak memenuhi persyaratan dan aturan yang terbaru dari KRSTI pada tahun 2024. Robot humanoid yang digunakan saat itu hanya memiliki 16 DoF (*Degree of Freedom*). Namun, pada pedoman KRSTI terbaru harus memiliki minimal 24 DoF agar robot dapat melakukan berbagai gerakan tari lebih luwes dan lebih kompleks [4]. Pengembangan sebelumnya hanya berfokus pada proses pengenalan suara musik dengan membaca sinyal ADC (*Analog to Digital Converter*) dari output signal analog suara tetapi struktur mekaniknya tetap sama [5]. Hal ini menjadi masalah karena robot harus dapat melakukan berbagai gerakan yang sesuai dengan tema dan irama lagu dalam waktu terbatas di arena sepanjang 3 meter. Ketidakmampuan robot untuk bergerak dengan tepat dan fleksibel karena keterbatasan jumlah DoF pasti akan berdampak pada penilaian, performa dan

keselarasan gerakan dengan musik yang merupakan komponen penting dalam KRSTI.

Permasalahan berikutnya ada pada proses pemrograman robot, yang dimana kompleksitas dalam seni tari dan terdapat banyak aktuator seperti servo sangat menyita tenaga dan waktu. Cara pemrograman yang biasa dilakukan adalah dengan mengetik kode pada software Arduino IDE (*Integrated Development Environment*), lalu diunggah ke mikrokontroler yaitu ESP32. Namun, ketika terdapat kesalahan dalam program, pengguna harus memperbaiki kode lagi, lalu mengunggah ulang. Jika masih ada kesalahannya, proses ini harus diulang berulang kali hingga program berjalan benar-benar seperti yang diinginkan. Masalahnya, ESP32 memerlukan waktu yang relatif lama untuk mengelola proses upload, jadi pengulangan seperti ini menjadi sangat tidak efektif dan mengganggu produktivitas. Ini menjadi masalah tersendiri, terutama ketika robot harus langsung diuji untuk menampilkan gerakan yang tepat dan sinkron dengan musik. Kondisi tersebut tentu menimbulkan kelelahan dan kejenuhan dalam proses *debugging* yang berulang, dan menjadi penghambat dalam pengembangan gerakan robot yang seharusnya bisa lebih cepat dan fleksibel dilakukan.

Atas dasar permasalahan di atas, perlu dilakukan penelitian dalam pengembangan robot humanoid seni tari. Penelitian ini akan merancang ulang desain mekanik robot dengan struktur yang lebih presisi dan fleksibel, lengkap dengan 25 derajat kebebasan (DOF) untuk memenuhi syarat terbaru KRSTI serta mendukung kemampuan robot dalam melakukan variasi gerakan tari yang lebih kompleks dan harmonis di sepanjang arena kompetisi. Selain itu, untuk mengatasi hambatan dalam proses pemrograman yang selama ini masih bergantung pada metode konvensional dengan penggunaan Arduino IDE yang bernilai waktu proses *upload* yang berulang kali saat terjadi *error*, dikembangkan sebuah perangkat lunak berbasis Java Swing yang bertindak sebagai kontrol gerakan servo. *Software* ini dimanfaatkan untuk mempercepat penyusunan, *editing*, dan pengujian gerakan tari langsung, sehingga berhasil menghemat waktu dan menekan potensi kesalahan saat pengembangan program, terutama pada perangkat ESP32 yang memiliki waktu unggah cukup lama.

1.2 Perumusan Masalah

Rumusan masalah yang didapat dari proyek akhir "Rancang Bangun Robot Humanoid Seni Tari dan Sistem Pemrograman Terintegrasi" sebagai berikut:

1. Bagaimana cara merancang dan membangun robot humanoid seni tari dengan desain yang presisi sesuai dengan ketentuan KRSTI?
2. Bagaimana mengembangkan perangkat lunak untuk mengontrol servo yang dapat mempermudah proses pemrograman gerakan robot secara efisien tanpa perlu melakukan proses *upload* yang berulang?
3. Bagaimana merancang sistem komunikasi antar robot yang lebih optimal agar kedua robot dapat bergerak dengan sinkron?

1.3 Tujuan Proyek Akhir

Penelitian pada proyek akhir ini bertujuan sebagai berikut:

1. Merancang dan membangun robot humanoid seni tari dengan desain mekanik yang presisi dan memiliki 25 *Degree of Freedom* (DoF) sesuai dengan ketentuan terbaru KRSTI.
2. Mengembangkan *software* berbasis *Java Swing* untuk mengontrol servo secara *real-time* guna mempermudah proses pemrograman gerakan robot.
3. Meningkatkan komunikasi antar robot agar kedua robot dapat bergerak secara sinkron dalam melakukan gerakan tari.

BAB II DASAR TEORI

2.1 Robot Humanoid

Robot merupakan sebuah alat mekanik yang dapat melakukan tugas fisik, baik menggunakan pengawasan dan kontrol manusia, ataupun menggunakan program yang telah didefinisikan terlebih dahulu (Gonzales, 1987). Kata "robot" berasal dari bahasa Cheko yaitu "robota", yang artinya pekerja atau buruh yang bekerja keras dan tidak terlalu lelah. Umumnya, robot dibutuhkan saat kita akan melakukan pekerjaan yang berat, berbahaya, atau pekerjaan yang harus diulang-ulang dan kotor. Robot juga banyak digunakan dalam dunia industri, pekerjaan yang ideal jika dijalankan oleh robot adalah proses produksi selayaknya manusia. Penyebutan umum dalam dunia industri ini adalah arm robots. Robot juga digunakan dalam hal pekerjaan lain seperti mengendapkan limbah beracun, eksplorasi bawah air dan ruang angkasa, pertambangan, pekerjaan "cari dan tolong" (*search and rescue*), dan dalam menemukan tambang. Beberapa tahun yang lalu, robot juga mulai digunakan dalam dunia konsumen khususnya alat pembantu rumah tangga, seperti penyedot debu dan pemotongan rumput [6].

Robot humanoid adalah robot yang memiliki bentuk tubuh yang menyerupai manusia, dasarnya mereka memiliki kepala, dua lengan dan dua kaki. Beberapa robot humanoid dirancang hanya menyerupai bagian tertentu dari tubuh manusia, namun ada juga yang memiliki wajah lengkap dengan mata dan mulut. Pesatnya kemajuan teknologi menghadirkan tantangan baru bagi pengembangan robot humanoid, yaitu menciptakan robot yang mampu mempelajari keterampilan baru melalui pengamatan dan peniruan terhadap aktivitas manusia. Proses ini dimulai dengan pendekatan yang berfokus pada pembelajaran tindakan manusia melalui penemuan serta pengalaman pribadi. Model yang dihasilkan kemudian digunakan untuk menyimpulkan tujuan dari setiap tindakan manusia, sehingga memungkinkan robot melakukan peniruan secara efektif dalam proses pembelajaran robotik, guna mendukung kolaborasi antara manusia dan robot. Terlepas dari kenyataan bahwa robot humanoid memiliki aktuator yang sangat berbeda dengan manusia, beberapa

faktor ini memungkinkan robot menjadi lebih baik dalam menginterpretasikan tindakan manusia yang semakin kompleks yang kemudian dapat ditiru. Hal ini tidak diragukan lagi memungkinkan robot untuk mengerjakan tugas-tugas yang dianggap berbahaya atau tidak mungkin dilakukan oleh manusia, sehingga menumbuhkan kolaborasi antara kecerdasan manusia dan kekuatan robot [7].



Gambar 2.1 Robot Humanoid

2.2 Kontes Robot Seni Tari Indonesia (KRSTI)

Kontes Robot Indonesia (KRI) ialah kegiatan kompetisi tahunan bagi mahasiswa yang gemar berkecimpung di bidang perancangan dan pembuatan robotika yang diselenggarakan oleh Pusat Prestasi Nasional, Kementerian Pendidikan dan Kebudayaan Republik Indonesia. Kompetisi tahunan ini pertama kali diselenggarakan pada tahun 2003 oleh Direktorat Jendral Pendidikan Tinggi, Departemen Pendidikan dan Kebudayaan di masa itu. Kontes Robot Indonesia atau KRI memiliki beberapa divisi, yang salah satunya memiliki nilai seni yaitu Kontes Robot Seni Tari Indonesia (KRSTI) [8].

Kontes Robot Seni Tari Indonesia (KRSTI) merupakan salah satu kontes untuk merancang, membangun, dan memprogram robot dengan menggabungkan tarian dan aspek budaya Indonesia. Dari nama divisi itu sendiri, Kontes Robot Seni Tari Indonesia, sudah terlihat bahwa robot yang dilombakan adalah robot yang nantinya akan menampilkan tarian-tarian daerah yang ada di Indonesia. “Robot Penari Oleg Tamulilingan gaya Peliangan” yang berasal dari Bali terpilih menjadi tema KRSTI tahun 2024 lalu [9].

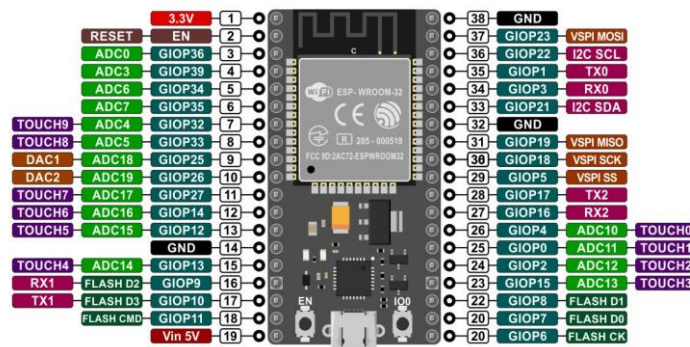
Robot dalam Kontes Robot Seni Tari Indonesia (KRSTI) harus dapat melakukan beberapa gerakan tari ketika musik dinyalakan dan berhenti ketika musik dimatikan. Robot harus menjaga keseimbangan sampai zona lapangan selesai. Terdapat papan lapangan yang digunakan sebagai dasar perlombaan yang terdiri dari zona-zona seperti Zona A, Zona B, dan Zona C. Zona A memiliki dua robot di zona awal. Selain Zona A dan Zona C, Zona B memiliki ketinggian 4mm, oleh karena itu, ada Zona Larangan di Zona B yang tidak dapat dilewati dengan diameter 300mm dan tinggi 10mm.



Gambar 2.2 Gambaran Pertandingan Kontes Robot Seni Tari Indonesia

2.3 ESP32

Diperkenalkan oleh *Espressif System*, ESP32 adalah mikrokontroler versi terbaru dari mikrokontroler terdahulu yaitu ESP8266. Modul Wi-Fi 802.11 b/g/n on-chip pada mikrokontroler ini membuatnya cukup membantu untuk merancang sistem aplikasi *Internet of Things*. Selain itu, ESP32 juga sudah dilengkapi dengan Bluetooth 4.2 dan berbagai peripheral serta GPIO (*General Purpose Input Output*) yang bisa digunakan untuk rangkaian pengganti pada Arduino. LCD (*Liquid Crystal Display*), lampu, dan bahkan gerakan motor DC (*Direct Current*), semuanya dapat dilakukan dengan menggunakan pin-pin ini sebagai *input* atau *output* [10].



Gambar 2.3 Mikrokontroler ESP32

ESP32 menggunakan set instruksi Xtensa LX16 dan dilengkapi dengan prosesor *dual-core* yang dapat beroperasi hingga 240 MHz. Performa yang mencapai 600 DMIPS, prosesor ini mampu menangani tugas-tugas berat secara efisien. Selain itu, meskipun perangkat IoT umumnya membutuhkan sistem manajemen daya tambahan, ESP32 telah dilengkapi dengan koprosesor ultra-low power (ULP) yang mendukung konsumsi daya sangat rendah dan memungkinkan penggunaan *deep sleep mode* [11].

Perbedaan ESP32 dengan mikrokontroler lain dipaparkan pada tabel 2.1.

Tabel 2.1 Perbandingan Arduino, NodeMCU, dan ESP32 [12]

Parameter	Arduino Uno	Node MCU (ESP8266)	ESP32
Tegangan	5 Volt	3.3 Volt	3.3 Volt
CPU	ATmega328 – 16MHz	Xtensa single core L106 – 60MHz	Xtensa dual core LX6 -160MHz
Arsitektur	8 bit	32 bit	32 bit
Flash Memory	32 kB	16 MB	16 MB
SRAM	2kB	160kB	512kB
GPIO PIN (ADC/DAC)	14(6/-)	17(1/-)	36(18/2)
Bluetooth	Tidak ada	Tidak ada	Ada
Wi-Fi	Tidak ada	Ada	Ada
SPI/I2C/UART	1/1/1	2/1/2	4/2/2

2.4 Motor Servo Sonic SN30MG-PRO

Motor servo merupakan jenis motor DC yang dilengkapi dengan *gearbox* serta sistem pengendali berbasis sinyal pulsa. Sistem kendali ini bekerja dengan mengirimkan sinyal pulsa untuk mengatur posisi sudut putaran motor, umumnya

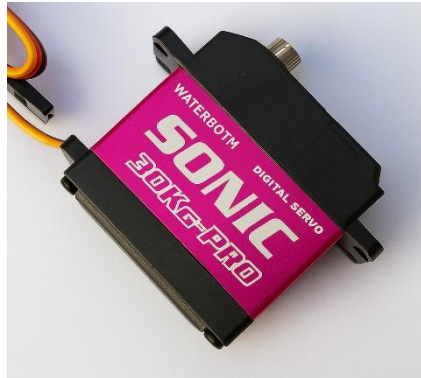
dalam rentang 0° hingga 180° . Motor servo memiliki torsi yang relatif kuat karena terdiri dari kombinasi motor DC, gearbox, variabel resistor, dan rangkaian kontrol. Motor ini memiliki tiga kabel utama, yaitu kabel daya (power), kabel *ground* (GND), dan kabel sinyal (data). Pengendalian posisi motor servo dilakukan dengan menggunakan sinyal PWM (Pulse Width Modulation), di mana motor akan bergerak menuju posisi tertentu dan berhenti secara otomatis setelah sinyal diterima. [13].

Selain itu, motor servo merupakan motor dengan sistem umpan balik tertutup (closed-loop system), di mana posisi poros motor secara terus-menerus diinformasikan kembali ke rangkaian kontrol internalnya. Komponen utama dalam motor servo meliputi motor DC, rangkaian roda gigi, potensiometer, dan sirkuit kendali. Potensiometer berperan penting dalam menentukan batas maksimum sudut putaran servo, sedangkan posisi sudut poros motor dikendalikan berdasarkan lebar pulsa (*pulse width*) yang dikirim melalui kabel sinyal. Sistem umpan balik ini memastikan motor bergerak dan berhenti secara presisi sesuai perintah sinyal yang diberikan [14].

Motor servo Sonic SN30MG Pro dipilih untuk digunakan pada bagian kaki, pinggang, dan pundak robot humanoid karena kemampuannya menahan beban dan menggerakkan lengan dengan stabilitas tinggi. Servo ini memiliki karakteristik torsi besar serta kecepatan respons yang memadai sehingga dapat mengangkat dan menahan beban masing-masing tangan robot selama gerakan berlangsung.

Tabel 2.2 Spesifikasi Servo Sonic SN30MG-PRO[15]

Spesifikasi	Nilai
Torsi maksimum (6,0 V)	30 kg·cm
Kecepatan operasi ($0-60^\circ$)	$\pm 0,10$ detik
Tegangan operasi	4,8–6,0 V
Tipe gear	Metal Gear
Material housing	Aluminium / logam campuran
Dimensi ($P \times L \times T$)	$\pm 40 \times 20 \times 38$ mm
Berat	± 55 g



Gambar 2.4 Motor Servo Sonic SN30MG-PRO

2.5 Motor Servo MG996R

Servo MG996R diproduksi oleh TowerPro dan telah banyak digunakan dalam aplikasi robotika dan biomedis karena torsi serta keakuratannya yang tinggi [16]. Sebuah penelitian terbaru yang dipublikasikan dalam International Journal of Robotics and Control Systems menggunakan MG996R dan berhasil meningkatkan ketepatan gerakan servo melalui pengendalian tertutup non-invasif yang disesuaikan menggunakan PID yang dioptimalkan PSO tercapai model presisi RMSE sekitar $0,85^\circ$ pada gerakan step 10° [17].

Servo ini sangat cocok diterapkan pada bagian tangan robot humanoid karena kemampuannya memberikan daya tahan dan kekuatan untuk mengangkat beban lengan. Torsi tinggi dan bodi gear metal memungkinkan kontrol posisi yang presisi meskipun dituntut gerakan dinamis. Keandalan MG996R dalam aplikasi biomedical menegaskan kestabilan performanya pada beban statis maupun dinamis. Dibandingkan mikroservo, MG996R memberikan dampak signifikan dalam menghadirkan postur tangan robot yang kokoh dan mampu melakukan tugas angkat atau dorong dengan keandalan tinggi.

Tabel 2.3 Spesifikasi Servo MG996R

Spesifikasi	Nilai
Torsi <i>stall</i>	9,4 kg·cm
Torsi <i>stall</i>	11 kg·cm
Kecepatan (0–60°)	0,17 s
Kecepatan (0–60°)	0,14 s
Arus operasi	500 mA–900 mA
Arus <i>stall</i>	2,5 A
Tegangan kerja	4,8–7,2 V
Gear	Metal



Gambar 2.5 Motor Servo MG996R

2.6 Servo MG90S

Servo MG90S juga diproduksi oleh TowerPro, merupakan mikroservo berukuran kompak dengan berat hanya sekitar 13,4g. Sebuah penelitian berjudul “*Design and implementation of Arduino based robotic arm*” menggunakan MG90S untuk menggerakkan sendi pergelangan pada lengan robot 6-DOF, menunjukkan bahwa servo ini memiliki performa baik dalam pengendalian presisi ringan [18]. Servo ini juga menggunakan *gear* metal dan *bearing*, menyediakan torsi hingga 2,2 kgcm pada 6V dan kecepatan 0,1s per 60°, membuatnya unggul dari versi plastik SG90 [19].

Robot humanoid ini menggunakan MG90S sehingga ideal untuk bagian pergelangan tangan dan kepala karena kecepatan dan ukuran kecilnya memungkinkan gerakan ekspresif dan responsif tanpa menambah beban signifikan. *Gear metal* meningkatkan daya tahan mekanis saat melakukan orientasi dinamis,

sementara ketepatan sudutnya mendukung ekspresi wajah dan kontrol sudut kepala yang halus. Kombinasi ini menjadikan MG90S pilihan tepat untuk aplikasi gerakan ringan yang menuntut kecermatan dan kecepatan.

Tabel 2.4 Spesifikasi Servo MG90S[20]

Spesifikasi	Nilai
Berat	13,4 g
Dimensi	$\pm 22 \times 11,8 \times 31$ mm
Torsi <i>stall</i>	1,8 kg·cm
Torsi <i>stall</i>	2,2 kg·cm
Kecepatan	0,10 s/60°
Tegangan kerja	4,8–6,0 V
<i>Dead band width</i>	~5 μ s
<i>Gear</i>	Metal dengan bearing



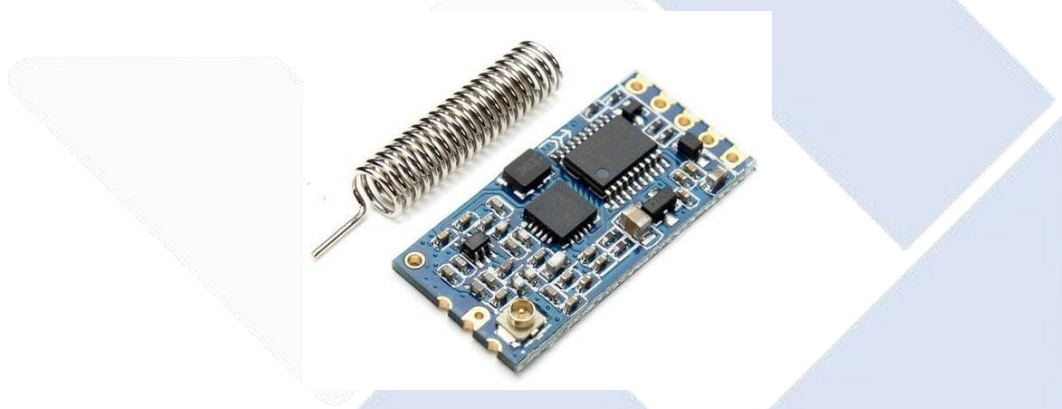
Gambar 2.6 Motor Servo MG90S

2.7 HC-12

Modul HC-12 merupakan modul komunikasi serial nirkabel generasi terbaru yang mendukung transmisi data *multichannel*. Modul ini beroperasi pada frekuensi nirkabel antara 433,4 hingga 473,0 MHz, dengan jarak antar *channel* sebesar 400 KHz, sehingga tersedia hingga 100 *channel* yang dapat dikonfigurasi. Daya transmisi maksimalnya mencapai 100mW (setara 20dBm), sedangkan sensitivitas penerimaannya mencapai -117dBm pada kecepatan baud di udara sebesar 5.000

bps. Jangkauan modul ini dapat mencapai hingga 1.000 meter apabila digunakan dalam kondisi tanpa hambatan atau di ruang terbuka. [21].

Modul ini sudah dilengkapi dengan mikrokontroler (MCU) internal, sehingga pengguna tidak perlu melakukan pemrograman terpisah. Semua mode transmisinya bersifat transparan, yang artinya hanya menangani proses pengiriman dan penerimaan data melalui port serial, menjadikannya sangat mudah digunakan. Modul ini mendukung berbagai mode transmisi transparan melalui port serial, yang dapat dipilih sesuai kebutuhan dengan menggunakan perintah AT (*Attention*). Tiga mode utama yaitu FU1, FU2, dan FU3 ini memiliki konsumsi arus rata-rata dalam kondisi siaga masing-masing sebesar 80 μ A, 3,6mA, dan 16mA. Sementara itu, arus kerja maksimum saat modul mengirimkan data dapat mencapai 100 mA. [22].



Gambar 2.7 HC-12

2.8 UGREEN CM523

Adaptor Bluetooth nirkabel UGREEN 3,5 mm mendukung dua mode, yaitu TX (*transmitter*) dan RX (*receiver*), sehingga dapat digunakan sebagai *transmitter* maupun *receiver* sesuai kebutuhan. Perangkat ini memungkinkan perangkat non-Bluetooth seperti TV, komputer, mobil, atau speaker untuk memperoleh konektivitas *Bluetooth* [23].

Menggunakan *chip Bluetooth 5.1*, *transmitter* dan *receiver Bluetooth* dari UGREEN menawarkan kompatibilitas tinggi dengan berbagai perangkat nirkabel serta meminimalkan latensi transmisi sinyal, sehingga memastikan koneksi yang stabil dan andal. Nikmati pengalaman audio yang jernih dan mulus tanpa gangguan

atau putusnya koneksi. Jarak transmisinya mencapai hingga 33 kaki (sekitar 10 meter) [24].

Berikut adalah spesifikasi teknis dari UGREEN CM523

Tabel 2.5 Spesifikasi UGREEN CM523 [25]

Spesifikasi	Detail
Model	UGREEN CM523
Fungsi	Transmitter (TX) & Receiver (RX) audio Bluetooth
Versi Bluetooth	5.1
Port Audio	Jack 3.5 mm AUX
Jarak Transmisi	Hingga 10 meter
Daya Tahan Baterai	Hingga 8 jam penggunaan
Port Pengisian Daya	USB Type-C



Gambar 2.8 Audio Bluetooth dongle UGREEN CM523

2.9 PCA9685

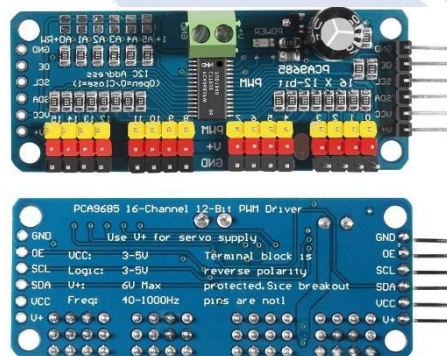
Driver motor servo PCA9685 merupakan IC yang mampu menghasilkan hingga 16 sinyal PWM melalui antarmuka komunikasi TWI/I2C. Modul ini menyediakan 16 output PWM dengan *duty-cycle* yang dapat diatur secara independen untuk tiap kanal, meskipun semuanya bekerja pada frekuensi yang sama. Modul PCA9685 hanya memerlukan dua pin I2C/TWI, sehingga sangat efisien dalam penggunaan pin mikrokontroler seperti ESP32. Jika 16 kanal PWM masih kurang, modul ini dapat dikembangkan hingga 62 unit, menghasilkan total 992 kanal PWM. Penggunaan pin pada ESP32 dapat diminimalkan melalui pemanfaatan modul tambahan. Modul PCA9685 tidak hanya digunakan untuk

mengendalikan motor servo, tetapi juga dioptimalkan dalam aplikasi pencahayaan, seperti pada LCD dan LED RGBA (Merah, Hijau, Biru, dan Kuning). [14].

Driver motor servo tipe servo *shield* PCA9685 memungkinkan pengendalian beberapa motor servo secara paralel. Setiap motor servo umumnya membutuhkan arus antara 200 mA hingga 900 mA untuk beroperasi. Dalam pengujian, tegangan *input* yang digunakan biasanya berada pada nilai tengah dari rentang tegangan yang direkomendasikan dalam *datasheet* PCA9685, yaitu antara 2,5 hingga 5,5 volt DC [26].

Tabel 2.6 Spesifikasi PCA9685 [27]

Parameter	Nilai
Jumlah Kanal PWM	16 <i>channel</i> independen
Resolusi PWM	12-bit (4096 langkah)
Frekuensi PWM	Dapat disesuaikan antara 24 Hz hingga 1526 Hz
Antarmuka Komunikasi	I2C (alamat default 0x40, dapat diubah hingga 62 alamat unik)
Tegangan Operasi (VCC)	2,3 V hingga 5,5 V
Tegangan <i>Input</i> Servo	Hingga 6 V (melalui konektor eksternal V+)
Arsitektur <i>Clock</i>	<i>Internal oscillator</i> 25 MHz
Proteksi	Tersedia fitur <i>overcurrent protection</i>
Kompatibilitas	Kompatibel dengan berbagai mikrokontroler seperti Arduino, ESP32, dsb



Gambar 2.9 PC9685

2.10 Buck Converter XL4016

XL4016 merupakan konverter DC-DC tipe buck (*step-down*) dengan frekuensi tetap 180 KHz, yang mampu memberikan daya hingga 8A secara efisien,

dengan tingkat riak rendah serta performa regulasi beban dan saluran yang sangat baik. Regulator ini dirancang agar mudah digunakan, hanya memerlukan sedikit komponen eksternal, dan telah dilengkapi dengan kompensasi frekuensi internal serta osilator bawaan. Rangkaian pengendali PWM (*Pulse Width Modulation*) memungkinkan penyesuaian rasio siklus kerja secara linear dari 0 hingga 100%. Modul ini juga dilengkapi fitur perlindungan arus lebih. Saat terjadi hubung singkat, frekuensi operasi akan otomatis diturunkan dari 180 KHz menjadi 48 KHz. Kompensasi internal dirancang untuk mengurangi kebutuhan akan komponen eksternal tambahan [28].



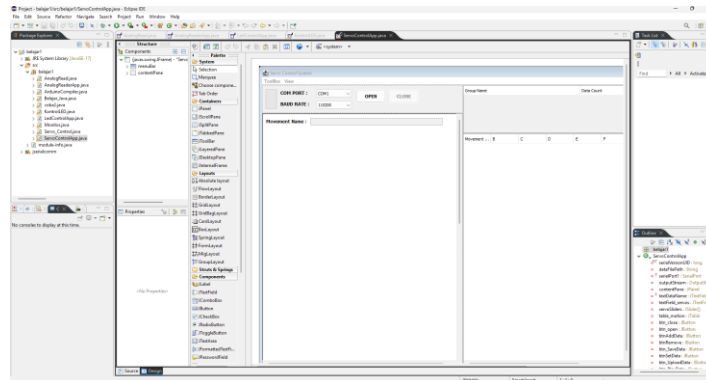
Gambar 2.10 *Buck Converter XL4016*

2.11 *Java Swing*

Java Swing merupakan *library* dalam bahasa pemrograman *Java* yang digunakan untuk membangun *Graphical User Interface* (GUI). Salah satu keunggulan *Java Swing* adalah kemampuannya untuk menciptakan antarmuka yang bersifat *cross-platform* atau independen terhadap sistem operasi. GUI yang telah dikembangkan dapat beroperasi di berbagai sistem operasi berbasis *Java* dengan tampilan visual yang cenderung konsisten. Bahkan, *Java Swing* memungkinkan pengembang menciptakan antarmuka yang menyerupai tampilan *Windows XP*, *Mac OS*, atau *Linux*, tanpa bergantung pada sistem operasi yang digunakan [29].

Swing merupakan bagian dari *Java Foundation Classes* (JFC), yang juga mencakup berbagai fitur grafis lanjutan. *Package Swing* menyediakan beragam kelas yang berguna untuk mengembangkan aplikasi GUI, dan semuanya terdapat

dalam package `javax.swing`. Komponen-komponen dalam *Swing* sepenuhnya ditulis dalam Java, sehingga memiliki tampilan "*look and feel*" yang seragam di berbagai *platform*. Selain itu, *Swing* menawarkan elemen GUI tambahan yang menarik, seperti pemilih warna (*color chooser*) dan kotak dialog pilihan (*option pane*) [28].



Gambar 2.11 Antarmuka *Java Swing* (Eclipse)

2.12 Komunikasi Serial (USART)

USART (*Universal Synchronous Asynchronous Receiver*) merupakan sistem komunikasi dengan fleksibilitas tinggi yang dapat digunakan untuk mentransfer data, baik antar mikrokontroler maupun antara mikrokontroler dengan modul eksternal, termasuk komputer yang mendukung fitur USART. Modul ini mendukung dua mode transmisi data, yaitu secara sinkron (*synchronous*) maupun asinkron (*asynchronous*), sehingga perangkat yang memiliki USART secara otomatis juga kompatibel dengan UART [29].

Sementara itu, UART berfungsi sebagai pengubah data dari format paralel menjadi serial dan sebaliknya. Umumnya, UART merupakan sirkuit terintegrasi yang digunakan dalam komunikasi serial, baik di komputer maupun perangkat periferan melalui port serial. UART sendiri adalah salah satu protokol komunikasi yang paling umum digunakan dalam pertukaran data serial antar perangkat, misalnya komunikasi antar mikrokontroler [30].

2.13 Baterai *Lithium Polymer*

Baterai *Lithium Polymer* (Li-Po) adalah salah satu varian baterai lithium yang menggunakan elektrolit polimer dalam bentuk padat atau semi-padat, berbeda dari Li-Ion yang menggunakan elektrolit cair [31]. Keunggulan utama baterai Li-Po mencakup fleksibilitas desain baterai dapat dibuat dalam berbagai bentuk dan ukuran sesuai kebutuhan perangkat portable, bobot yang ringan sehingga ideal untuk aplikasi yang mengutamakan mobilitas tinggi, tingkat *self-discharge* rendah, memungkinkan daya tetap tersedia setelah periode penyimpanan, dan kemampuan pengisian cepat dan *discharge rate* tinggi [32].



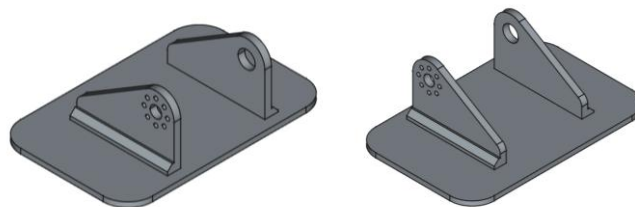
Gambar 2.12 Baterai Li-Po

BAB III METODE PELAKSANAAN

Untuk bab ini menjelaskan tentang desain robot, perancangan sistem, *software* dan implementasi secara keseluruhan, yang dimulai dari desain 3D *hardware*, desain rangkaian elektrik, desain *software*, desain komunikasi pada robot, pembuatan *software*, implementasi desain mekanik robot, dan implementasi sistem elektrik. *Hardware* yang digunakan adalah dua buah robot humanoid yaitu robot humanoid *master*, dan robot humanoid *slave* yang dirancang dengan 25 derajat kebebasan dengan menggunakan kontroler ESP32. Terdapat dua kontroler ESP32 yang digunakan pada masing-masing robot humanoid, ESP32 yang pertama yaitu *subcontroller* digunakan sebagai kontroler komunikasi robot yang terhubung ke HC-12, sedangkan ESP32 yang kedua yaitu *main controller* digunakan sebagai kontroler aktuator servo yang terhubung ke PCA9685 untuk menggerakkan 25 buah servo. Secara keseluruhan, data pergerakan robot atau pergerakan servo dibuat dan diatur pada *software*, kemudian data dimasukkan ke dalam program dan diupload ke kontroler ESP32 yang pertama sebagai kontroler servo sehingga dapat menggerakkan servo sesuai dengan data yang ada pada *software*.

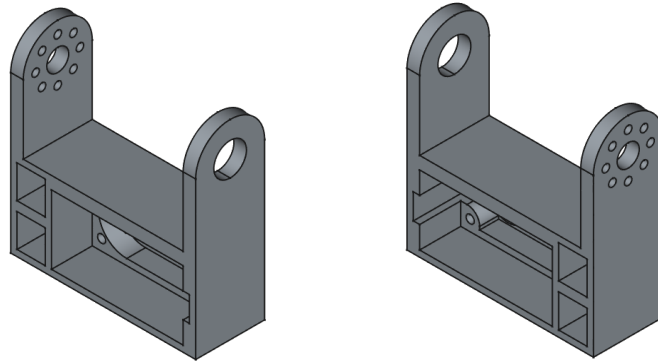
3.1. Desain 3D *Hardware*

Desain 3D robot humanoid pada proyek akhir ini menggunakan aplikasi FreeCAD yang akan dicetak berbahan dasar filament PLA (Polylactic Acid). Desain alat dapat dilihat pada gambar dibawah ini.



Gambar 3.1 (a) Desain telapak kaki kanan (b) Desain telapak kaki kiri

Pada gambar 3.2. merupakan gambar telapak kaki pada robot yang diprint menggunakan 3D *printing*.



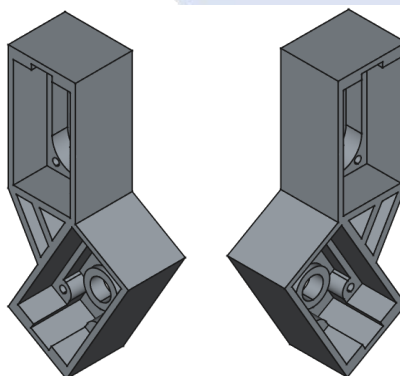
Gambar 3.2 (a) Desain pergelangan kaki kanan (b) Desain pergelangan kaki kiri

Gambar di atas merupakan gambar pergelangan kaki pada robot yang diprint menggunakan *3D printing*.



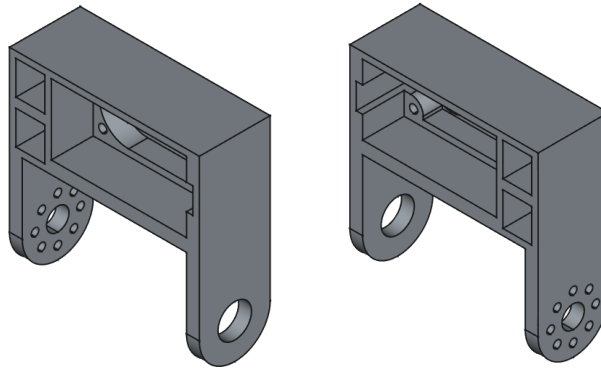
Gambar 3.3 (a) Desain betis kanan (b) Desain betis kiri

Gambar di atas merupakan gambar betis pada kaki robot yang diprint menggunakan *3D printing*.



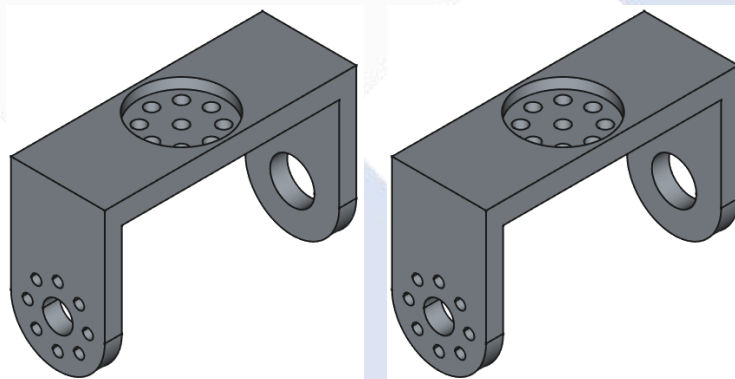
Gambar 3.4 (a) Desain paha kanan (b) Desain paha kiri

Gambar diatas merupakan gambar paha pada kaki robot yang diprint menggunakan 3D *printing*.



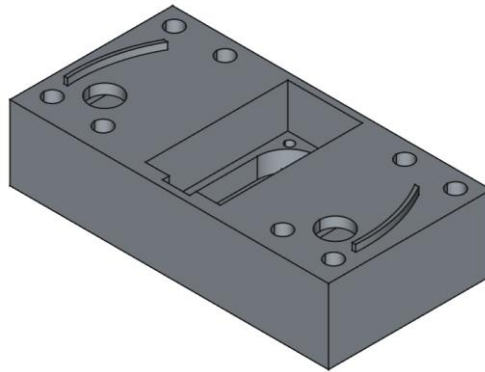
Gambar 3.5 (a) Desain *Right Hip Pitch* (b) Desain *Left Hip Pitch*

Gambar di atas merupakan gambar *Right* dan *Left Hip Pitch* pada kaki robot yang diprint menggunakan 3D *printing*. Bagian ini membuat robot dapat menggerakkan kaki maju-mundur.



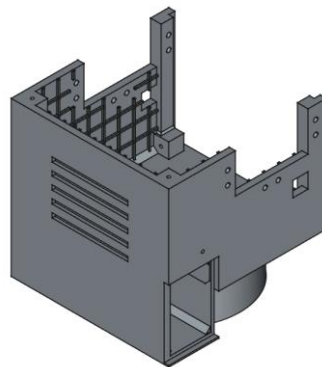
Gambar 3.6 (a) Desain *Right Hip Roll* (b) Desain *Left Hip Roll*

Gambar di atas merupakan gambar *Right* dan *Left Hip Roll* pada kaki robot yang diprint menggunakan 3D *printing*. Bagian ini membuat robot dapat menggerakkan kaki kiri-kanan.



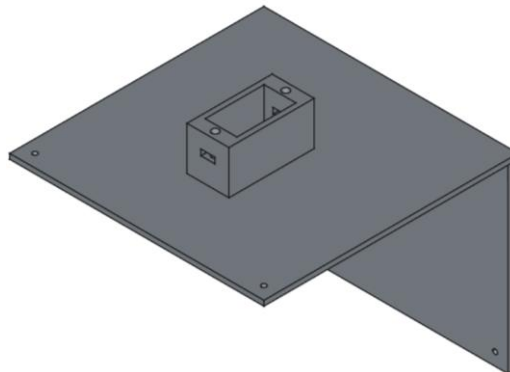
Gambar 3.7 Desain Pinggang

Gambar di atas merupakan gambar pinggang robot yang diprint menggunakan 3D *printing*. Bagian ini membuat robot dapat menggerakkan kaki dan juga *body* pada sumbu *yaw*.



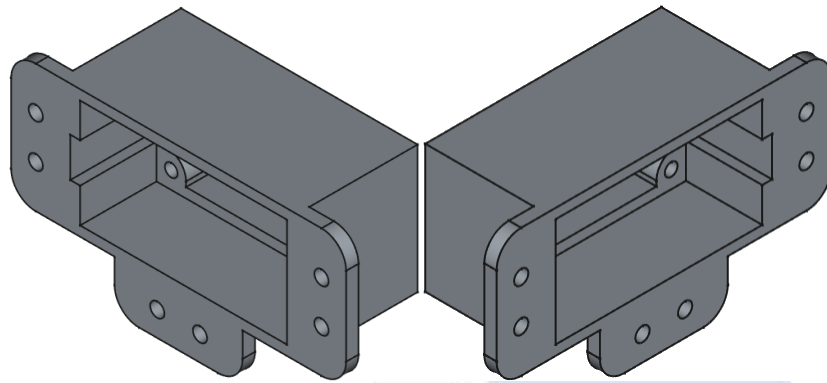
Gambar 3.8 Desain *Main Body* Robot

Gambar di atas merupakan gambar *main body* robot yang diprint menggunakan 3D *printing*. Bagian ini adalah tempat komponen elektrik robot serta dudukan untuk tangan robot.



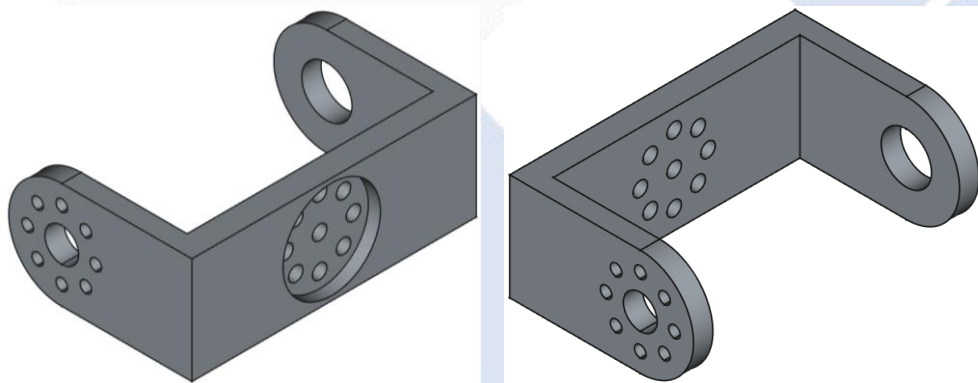
Gambar 3.9 Desain Penutup *Body* Robot

Gambar di atas merupakan gambar penutup *body* robot yang diprint menggunakan 3D *printing*. Bagian ini adalah penutup dari *main body* sekaligus tempat servo *head yaw*.



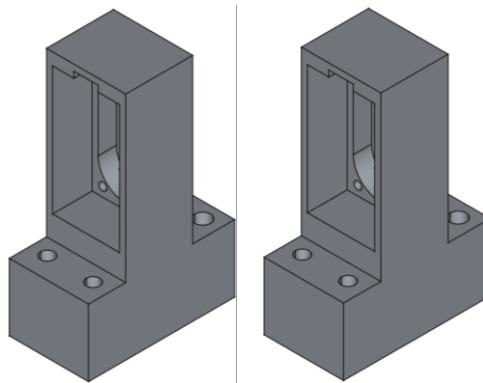
Gambar 3.10 (a) Desain *Right Shoulder Pitch* (b) Desain *Left Shoulder Pitch*

Gambar di atas merupakan gambar *Right Shoulder Pitch* dan *Left Shoulder Pitch* robot yang diprint menggunakan 3D *printing*. Bagian ini membuat lengan robot dapat menganyun ke depan-belakang.



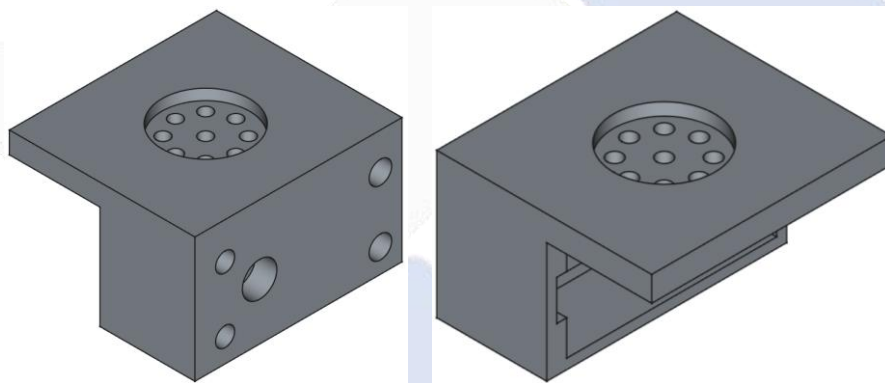
Gambar 3.11 (a) Desain *Right Shoulder Roll* (b) Desain *Left Shoulder Roll*

Gambar di atas merupakan gambar *Right Shoulder Roll* dan *Left Shoulder Roll* robot yang diprint menggunakan 3D *printing*. Bagian ini membuat lengan robot dapat bergerak ke atas-bawah.



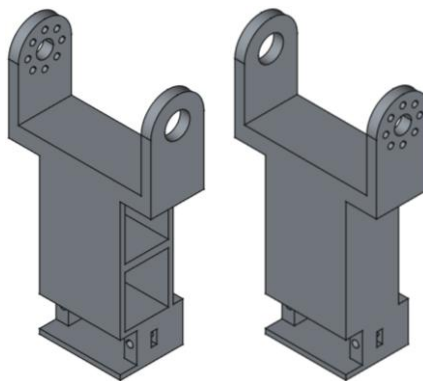
Gambar 3.12 (a) Desain *Right Upper Arm* (b) Desain *Left Upper Arm*

Gambar di atas merupakan gambar *Right Upper Arm* dan *Left Upper Arm* robot yang diprint menggunakan *3D printing*. Bagian ini membuat lengan bagian bawah robot dapat berputar.



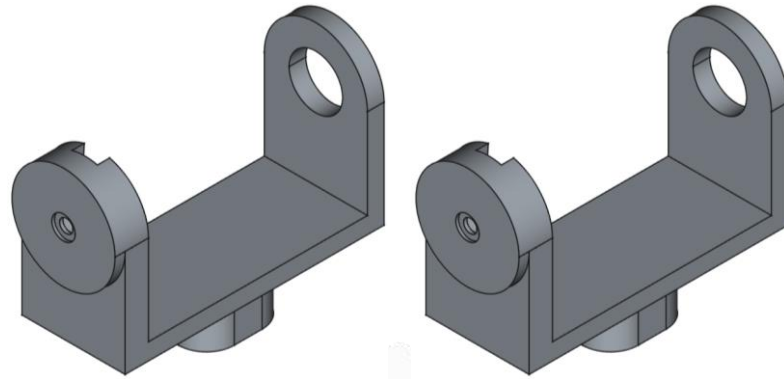
Gambar 3.13 (a) Desain *Right Arm Roll* (b) Desain *Left Arm Roll*

Gambar diatas merupakan gambar *Right Upper Yaw* dan *Left Upper Yaw* robot yang diprint menggunakan *3D printing*. Bagian ini membuat lengan bagian bawah robot dapat bergerak kesamping kanan-kiri.



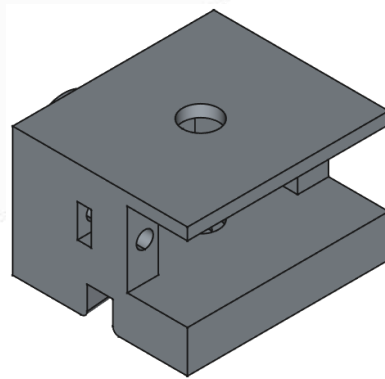
Gambar 3.14 (a) Desain *Right Lower Arm* (b) Desain *Left Lower Arm*

Gambar diatas merupakan gambar *Right Lower Arm* dan *Left Lower Arm* robot yang diprint menggunakan *3D printing*. Bagian ini sekaligus menjadiudukan tangan robot.



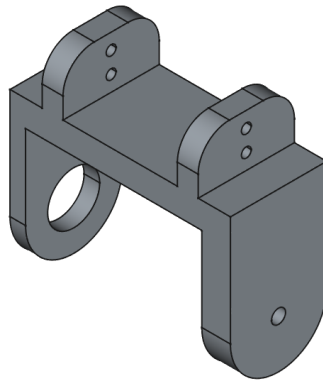
Gambar 3.15 (a) Desain *Right Hand* (b) Desain *Left Hand*

Gambar di atas merupakan gambar *Right Hand* dan *Left Hand* robot yang diprint menggunakan *3D printing*. Bagian ini membuat bagian tangan robot dapat bergerak ke kanan-kiri.



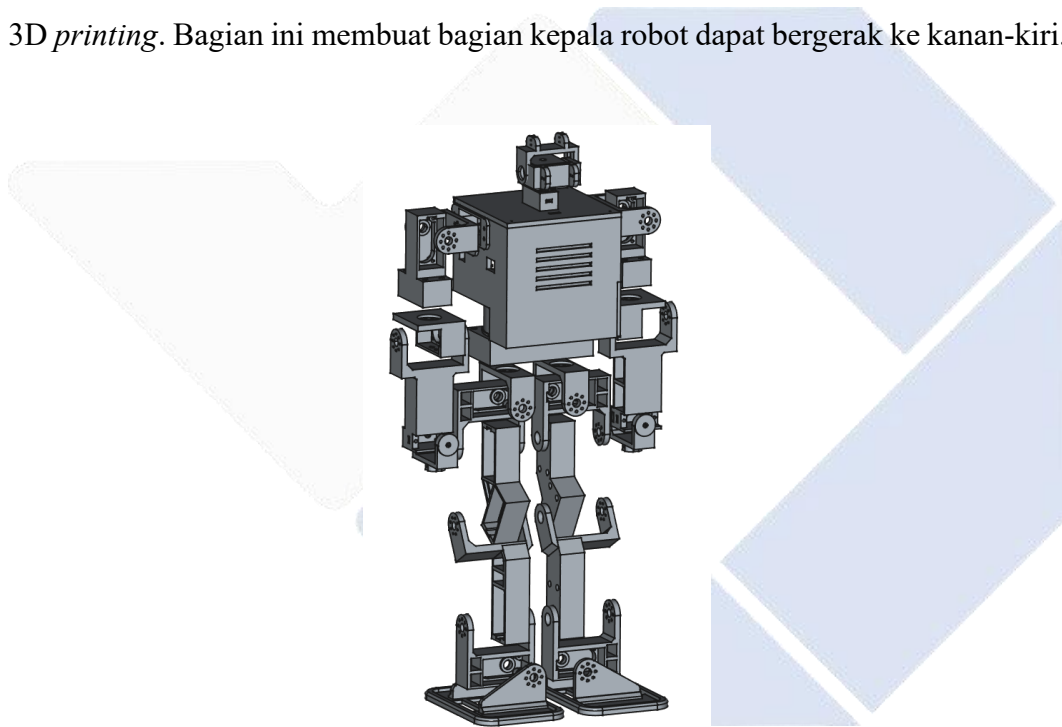
Gambar 3.16 Desain *Head Pitch*

Gambar di atas merupakan gambar *Head Pitch* robot yang diprint menggunakan *3D printing*. Bagian ini membuat bagian kepala robot dapat mengangguk (atas-bawah).



Gambar 3.17 Desain *Head Bracket*

Gambar di atas merupakan gambar *Head Bracket* robot yang diprint menggunakan *3D printing*. Bagian ini membuat bagian kepala robot dapat bergerak ke kanan-kiri.



Gambar 3.18 Desain Keseluruhan Robot Humanoid

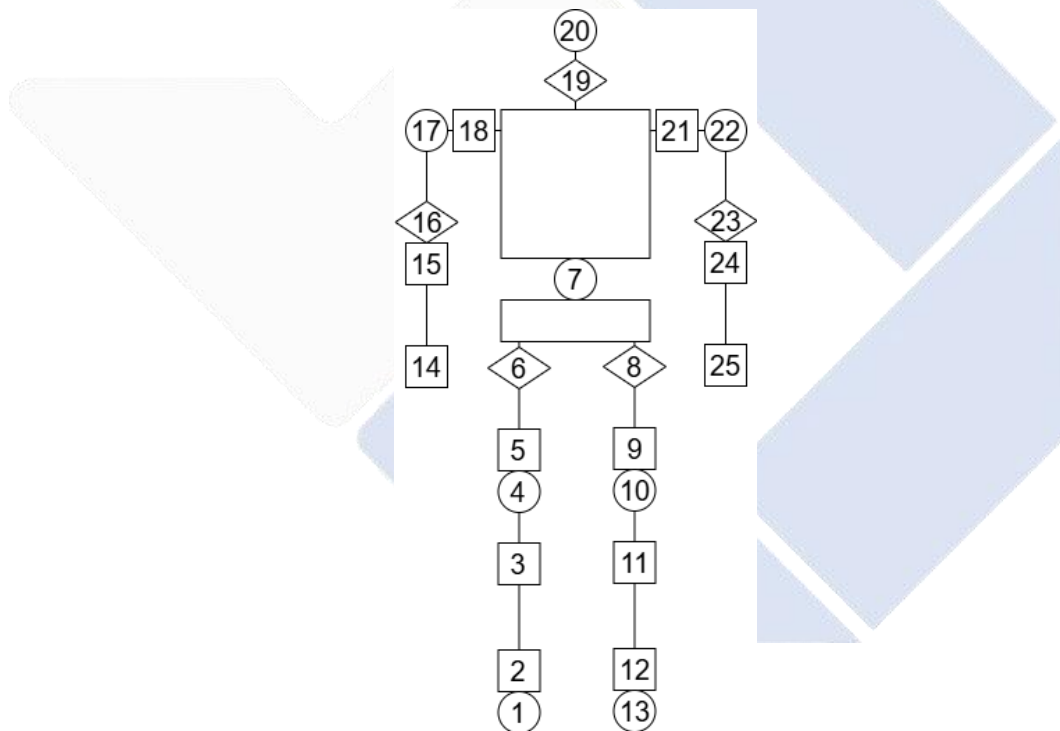
Gambar di atas merupakan gambar desain keseluruhan robot humanoid yang diprint menggunakan *3D printing*.

Desain *hardware* robot humanoid seni tari ini mengikuti aturan pada buku Pedoman Kontes Robot Seni Tari Indonesia yang memiliki beberapa ketentuan sebagai berikut :

Tabel 3.1 Tabel dimensi robot

No	Deskripsi	Status	
		Ukuran Maksimal	Ukuran Robot
1	Tinggi robot	55±5cm	55 cm
2	Rentang tangan dari ujung-ujung jari	60 cm	60 cm
3	Lebar telapak kaki	150 cm ²	98,4 cm ²
4	Berat robot	20 kg	2,65 kg
5	Derajat kebebasan	24 DoF (minimal)	25 DoF

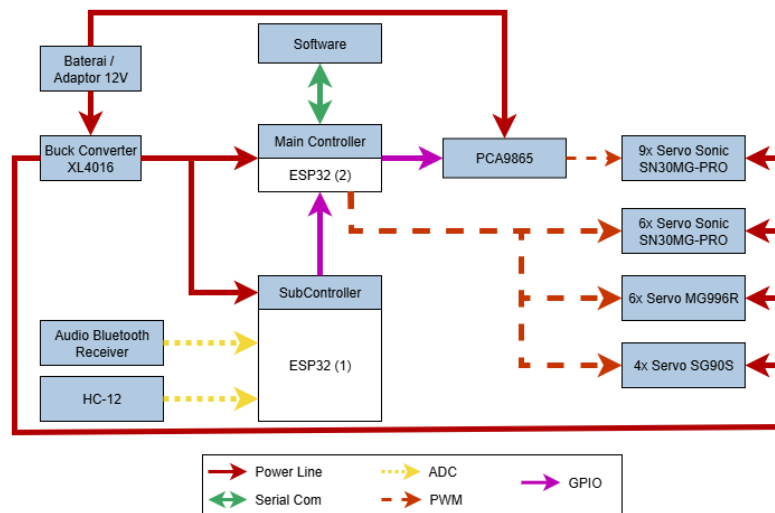
Tabel di atas menunjukkan bahwa robot humanoid ini sesuai dengan peraturan dari buku pedoman KRSTI 2024. Diagram derajat kebebasan dapat dilihat pada gambar dibawah



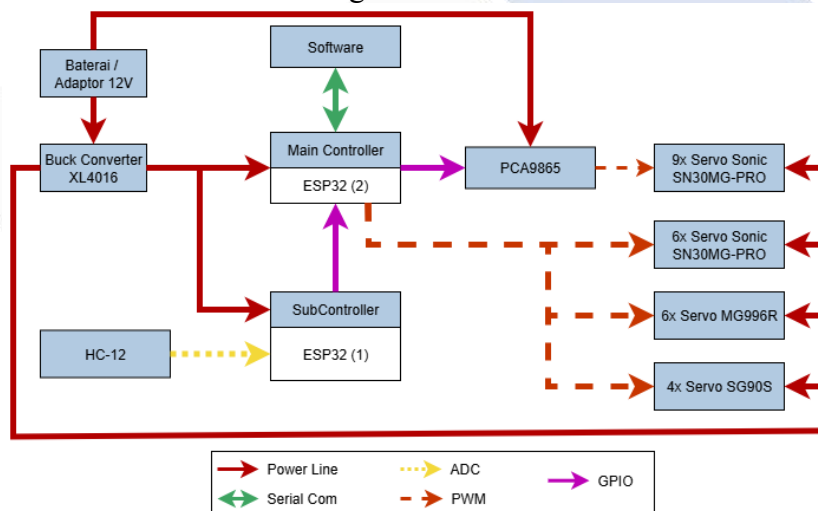
Gambar 3.19 Diagram derajat kebebasan robot humanoid

3.2. Perancangan dan Implementasi Rangkaian Elektrik

Robot humanoid yang sudah dirancang tentu tidak bisa berjalan sesuai yang kita harapkan apabila tidak ada kecerdasan buatan yang ditanam pada robot, sehingga diperlukan sistem elektronik yang dapat menjalankan seluruh program. Berikut desain rangkaian elektrik robot humnoid seni tari sebagai berikut:



Gambar 3.20 Rangkaian Elektrik Robot *Master*



Gambar 3.21 Rangkaian Elektrik Robot *Slave*

Pada desain di atas, merupakan bentuk rangkaian yang digunakan dalam sistem robot *slave* dan *master*, yang masing-masing dilengkapi dengan dua buah ESP32. ESP32 (1) yaitu *subcontroller* sebagai pengatur komunikasi antarrobot dan sebagai penerima sinyal audio, sedangkan ESP32 (2) yaitu *main controller* berperan sebagai pengontrol 25 motor servo dengan bantuan modul PCA9685.

Terdapat modul HC-12 yang terhubung dengan ESP32 *subcontroller*, yang berfungsi sebagai media komunikasi nirkabel antara robot *slave* dan *master*. Sementara itu, modul PCA9685 terhubung ke pin SCL dan SDA pada ESP32 *main controller* melalui komunikasi I2C. Seluruh 25 buah motor servo yang terdapat

pada setiap robot dikendalikan secara penuh oleh ESP32 *main controller* berdasarkan perintah yang dikirim dari ESP32 *subcontroller*.

Sistem ini membutuhkan baterai yang cocok agar semua rangkaian dapat berjalan dengan baik, maka dari itu arus dari setiap servo harus diukur agar dapat menentukan baterai dengan kapasitasnya yang cocok dengan sistem ini. Pada proses pendataan pengukuran arus bahwa tidak semua servo bergerak secara bersamaan, misalnya pada saat berjalan hanya beberapa servo di bagian kaki yang bergerak, begitu juga ketika sedang menari hanya beberapa servo di bagian tangan yang bergerak. Perhitungan di bawah ini menghitung total arus jika semua servo dalam kondisi bergerak.

Tabel 3.2 Tabel hasil pengukuran arus total semua servo

No	Nama Servo	Jumlah	Arus max. (A)	Arus Total (A)
1	Sonic SN30MG-PRO	15	1,93	28,95
2	TowerPro MG996R	6	1,02	6,12
3	TowerPro MG90S	4	0,45	1,8
Total				36,87



Gambar 3.22 Hasil pengukuran arus maksimum servo

Penentuan kapasitas dan ketahanan baterai sangat penting dalam merancang sistem daya robot humanoid, terutama pada sistem yang melibatkan banyak aktuator servo seperti pada proyek ini. Maka dilakukan pengukuran arus maksimum masing-masing jenis servo dalam kondisi tanpa beban. Berdasarkan tabel 3.2 total arus servo keseluruhan adalah 36,87A, angka tersebut merupakan nilai teoritis ketika seluruh servo aktif secara simultan dalam kondisi beban maksimal. Namun dalam praktiknya, tidak semua servo bekerja dalam waktu yang bersamaan, dan gerakan servo yang diberikan cenderung bergantian atau sebagian bersamaan, sesuai dengan pola koreografi robot humanoid tari.

Baterai yang digunakan dalam sistem ini adalah Li-Po 1800mAh 11,1V dengan rating 25C. Nilai 25C menunjukkan bahwa baterai mampu menyediakan arus keluaran maksimal dengan rumus:

$$I_{maks} = C_{rating} \times Kapasitas(Ah) \quad (1)$$

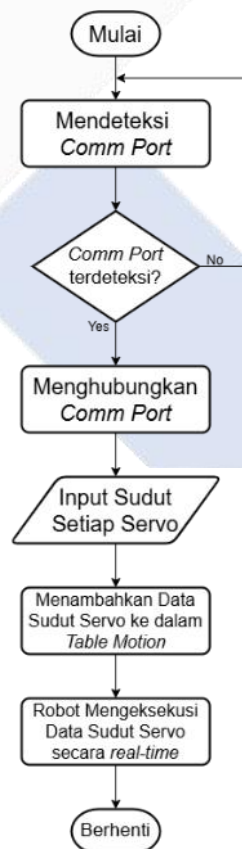
Sehingga,

$$I_{maks} = 25 \times 1,8(Ah) = 45A$$

Baterai memiliki kemampuan untuk memasok arus hingga 45A secara kontinu, yang berarti sudah lebih besar dari arus maksimum sistem sebesar 36,87A, sehingga secara teoritis baterai ini memadai dari sisi *discharge current capacity*.

3.3. Desain Software

Berikut desain *software* robot humanoid seni tari berbasis dekstop sebagai berikut :



Gambar 3.23 *Flowchart* Desain Software

Flowchart ini menggambarkan alur kerja dari sistem robot yang menggunakan data sudut pada setiap servo yang diinput pengguna melalui slider, data yang diinput pengguna dapat disimpan ke *table motion* sebagai step gerakan. Data yang ada pada *table motion* adalah kumpulan data step gerakan robot yang dapat dieksekusi oleh robot secara *real-time* sebagai suatu gerakan yang utuh. Karena robot dan *software* terintegrasi maka setiap pengguna mengatur *slider* secara *real-time* robot akan menyesuaikan sudut sesuai dengan nilai pada *slider* sehingga memungkinkan pengguna dapat melihat pergerakan sudut setiap servo.

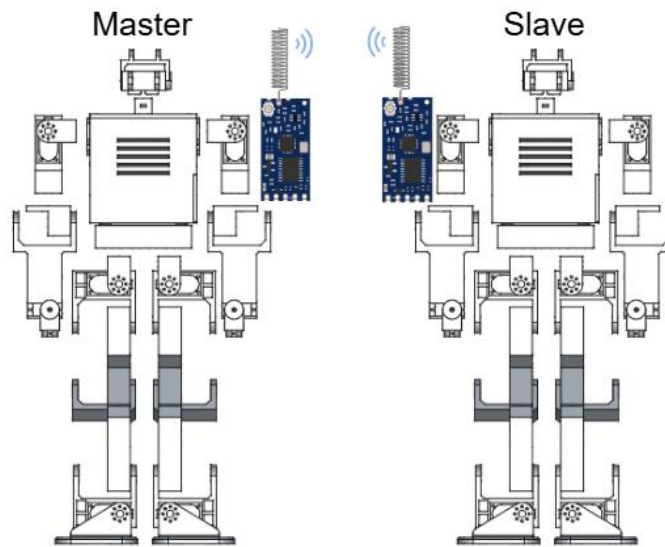
Pada desain *software* ini terdapat beberapa komponen penting seperti *Combobox CommPort*, *Servo Slider Panel*, *Table Motion*, dan berbagai komponen dan fitur lainnya. Di bawah ini merupakan fungsi dari komponen dan fitur tersebut:

1. *Combobox CommPort* : berfungsi menampilkan semua *communication port* yang terhubung ke komputer dan dapat di-select sesuai dengan *communication port* yang dibutuhkan.
2. *Servo Slider Panel* : berfungsi sebagai panel yang berisi *slider* dan *textbox* untuk semua servo yang digunakan untuk mengatur sudut servo.
3. *Table Motion* : berfungsi untuk menyimpan step-step gerakan yang sudah diinput sebelumnya pada *slider*.
4. *Combobox Baudrate* : berfungsi untuk mengatur *baud rate* sesuai dengan kebutuhan.
5. *Open & Close Button* : berfungsi sebagai pembuka dan penutup akses *communication port* ke *software*.
6. *Project Table* : berfungsi untuk menampilkan *project-project* gerakan robot yang sudah dibuat sebelumnya.
7. *Add Data Button* : tombol yang berfungsi menginput data sudut servo yang sudah diatur ke dalam *table motion*.
8. *Insert Data Above* : tombol yang berfungsi untuk menginput data sudut servo ke bagian atas dari data yang di-select pada *table motion*.
9. *Insert Data Below* : tombol yang berfungsi untuk menginput data sudut servo ke bagian bawah dari data yang di-select pada *table motion*.

10. *Edit Data* : tombol yang berfungsi untuk mengedit data sudut servo yang di-*select* pada *table motion*.
11. *Remove Data* : tombol yang berfungsi untuk menghapus data yang di-*select* pada *table motion*.
12. *Move Up Data* : tombol yang berfungsi untuk memindahkan data yang di-*select* berpindah 1 baris ke atas pada *table motion*.
13. *Move Down Data* : tombol yang berfungsi untuk memindahkan data yang di-*select* berpindah 1 baris ke atas pada *table motion*.
14. *Set Data* : tombol yang berfungsi agar data yang di-*select* pada *table motion* akan langsung ke robot untuk menyesuaikan data.
15. *Play & Stop Button* : berfungsi sebagai tombol yang dapat menjalankan dan memberhentikan semua step gerakan yang ada di *table motion*.
16. *Reset Data* : tombol yang berfungsi untuk mereset data semua *slider* ke nilai *default* yaitu 90.

3.4. Desain Komunikasi Robot

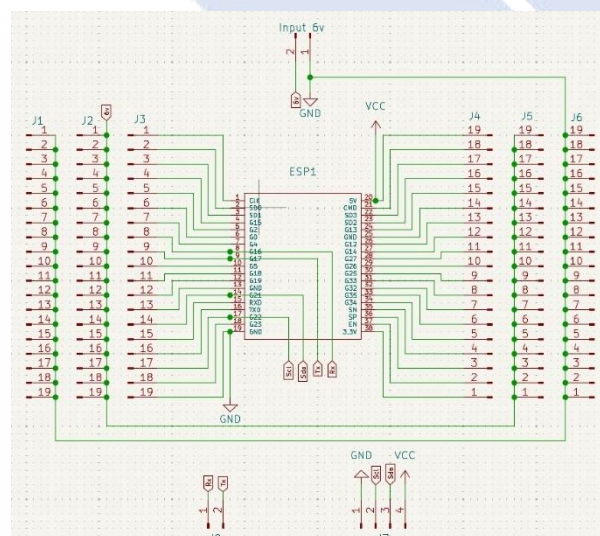
Kontes Robot Seni Tari Indonesia (KRSTI) mengharuskan untuk menggunakan dua unit robot untuk menari bersama-sama yang disebut *master* dan *slave*. Komunikasi antarrobot sangat dibutuhkan agar keduanya dapat melakukan gerakan dengan serasi. Perancangan elektronik robot *slave* tidak menggunakan sensor suara dikarenakan kendali dilakukan oleh robot *master*. Komunikasi antarrobot menggunakan HC-12 sebagai perantara pengiriman data dan proses *pairing* yang cepat dan aman. Kelebihannya ada pada jarak jangkauannya hingga 1000 meter karena HC-12 bekerja pada frekuensi 433,4 hingga 473,0 MHz menurut datasheet. Cara kerjanya cukup sederhana dimana HC-12 pada robot *master* akan mengirim perintah ke HC-12 pada robot *slave* apakah musik terdeteksi atau tidak. Data yang dikirim hanya berupa karakter "1" ketika musik terdeteksi dan "0" ketika musik tidak terdeteksi. Berikut adalah ilustrasi komunikasi antarrobot.



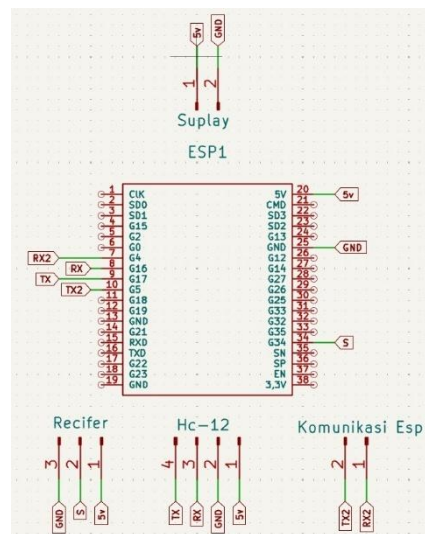
Gambar 3.24 Komunikasi Antar Robot

3.5. Pembuatan Rangkaian Elektrik dan PCB

Tahapan ini merupakan kelanjutan dari proses konstruksi alat. Pada tahap ini, rangkaian dan skematik sistem dirancang terlebih dahulu dalam bentuk sketsa menggunakan *software* KiCAD. Untuk proses pencetakan PCB, penulis memanfaatkan layanan cetak PCB secara online. PCB yang dicetak merupakan papan sirkuit untuk kontroler berbasis ESP32. Setiap robot menggunakan dua buah modul ESP32, dengan skematik rangkaian seperti ditunjukkan berikut.

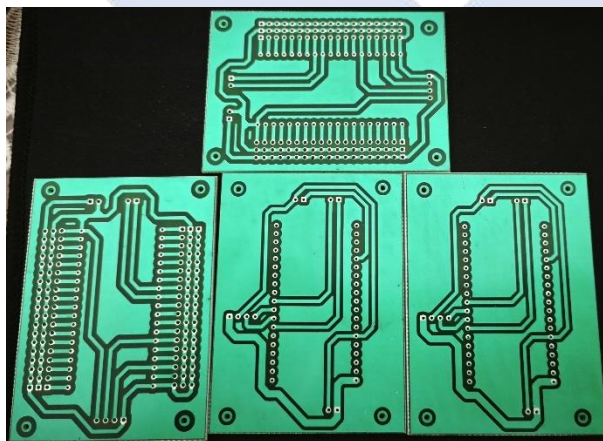


Gambar 3.25 Schematic Kontroler ESP32(1) Robot *Master* dan *Slave*



Gambar 3.26 Schematic Kontroler ESP32(2) Robot *Master* dan *Slave*

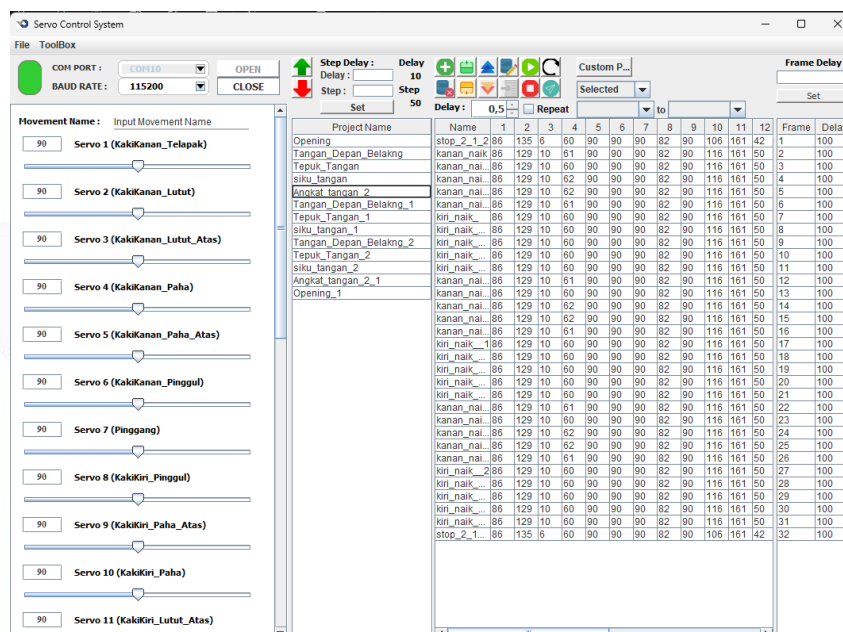
Pada gambar diatas, skematik kontrol robot menggunakan mikrokontroler ES32. Seluruh pin disambungkan ke pin header dan bersampingan dengan dua pin *header* lainnya yang merupakan VCC 6V dan GND. Tiga pin *header* yang disusun sejajar digunakan untuk menghubungkan servo ke PCB kontroler. Komponen eksternal seperti servo, *push button*, *audio bluetooth*, dan HC-12 dihubungkan oleh pin GPIO pada PCB kontroler. GND dari ESP32 juga dihubungkan dengan GND dari *Buck Converter* 200W yang digunakan sebagai sumber tegangan motor servo.



Gambar 3.27 Hasil cetak PCB

3.6. Pembuatan Software

Tahap berikutnya adalah pembuatan program untuk *software*. Program *software* akan diintegrasikan dengan program ESP32 agar dapat beroperasi sesuai dengan keinginan. Program *software* dibuat menggunakan *Eclipse IDE* dengan bahasa pemrograman *Java Swing*. *Software* yang dibuat nantinya akan terhubung ke ESP32 melalui *communication port* sehingga semua servo pada robot dapat dikendalikan secara bersamaan dan *real-time*. Tampilan *software* dapat dilihat pada gambar 3.27.

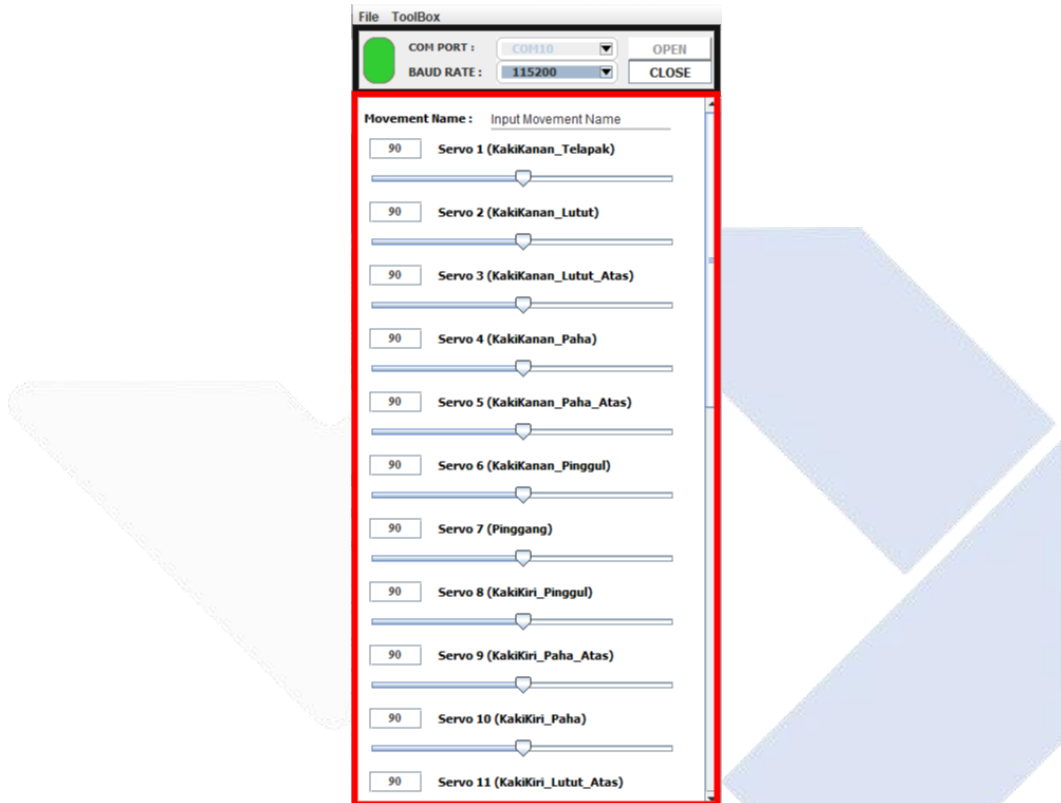


Gambar 3.28 Tampilan Software Kontrol Servo

3.7. Proses Pembuatan Gerakan Tari

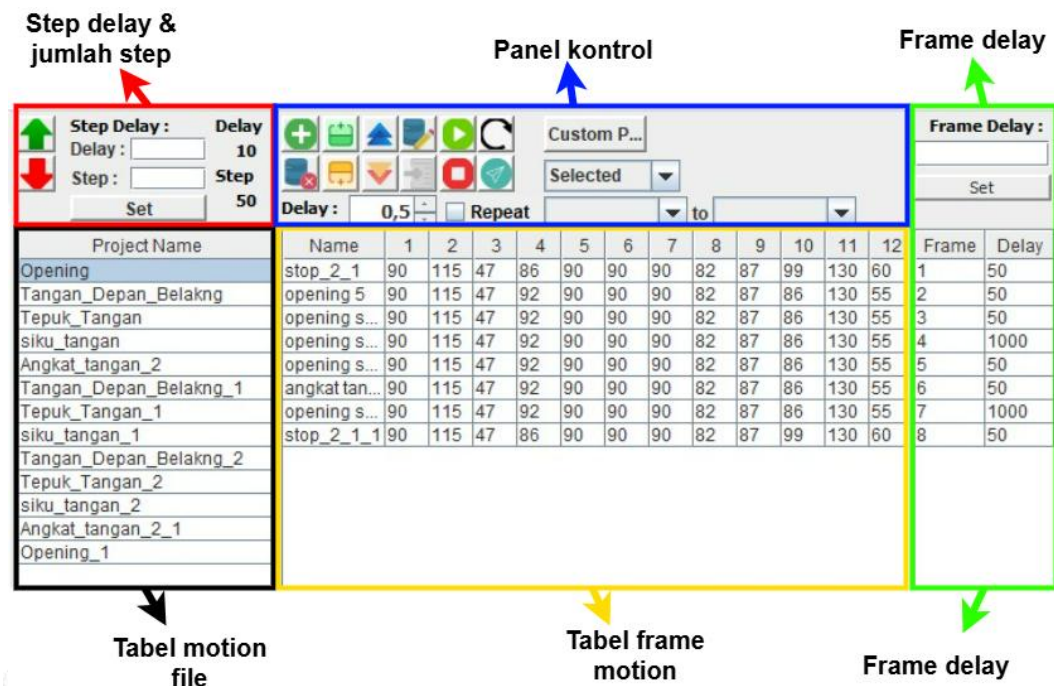
Antarmuka pengguna aplikasi terdiri dari beberapa bagian utama. Bagian sebelah kiri software, terdapat daftar servo yang masing-masing dilengkapi dengan kolom nilai sudut dan kontrol penggeser (slider). Pengguna dapat memasukkan nilai sudut secara manual atau menggeser slider untuk menentukan posisi setiap servo dalam rentang 0 hingga 180 derajat. Sebelumnya pengguna harus menghubungkan *comm port* dan *baud rate* sesuai dengan *main controller* pada robot humanoid, kemudian menekan tombol “open” maka *comm port* akan terhubung. Setiap perubahan pada slider akan langsung dikirim ke mikrokontroler secara otomatis

dalam format data biner, sehingga memungkinkan kontrol gerak yang responsif dan serempak. Bagian yang ditandai dengan kotak berwarna hitam pada gambar 3.28 merupakan bagian untuk menghubungkan *comm port*, sedangkan bagian yang ditandai oleh kotak warna merah adalah bagian *slider* untuk mengontrol sudut servo.



Gambar 3.29 Bagian Kontrol Sudut Servo pada *Software*

Bagian kanan *software*, terdapat dua tampilan tabel untuk menyusun urutan gerakan (*motion*), di mana pada tabel motion file berisi satu paket gerakan yang di dalamnya berisi banyak pose yang sebelumnya sudah ditambahkan pengguna sesuai dengan *input slider*. Sedangkan untuk tabel *frame motion*, setiap baris mewakili satu pose servo (atau langkah gerakan). Pengguna dapat menambahkan, menyisipkan, menghapus, mengatur ulang urutan, serta mengedit data gerakan dengan tombol-tombol yang tersedia pada panel kontrol. Fitur ini sangat membantu dalam menyusun koreografi atau pola gerakan yang kompleks.



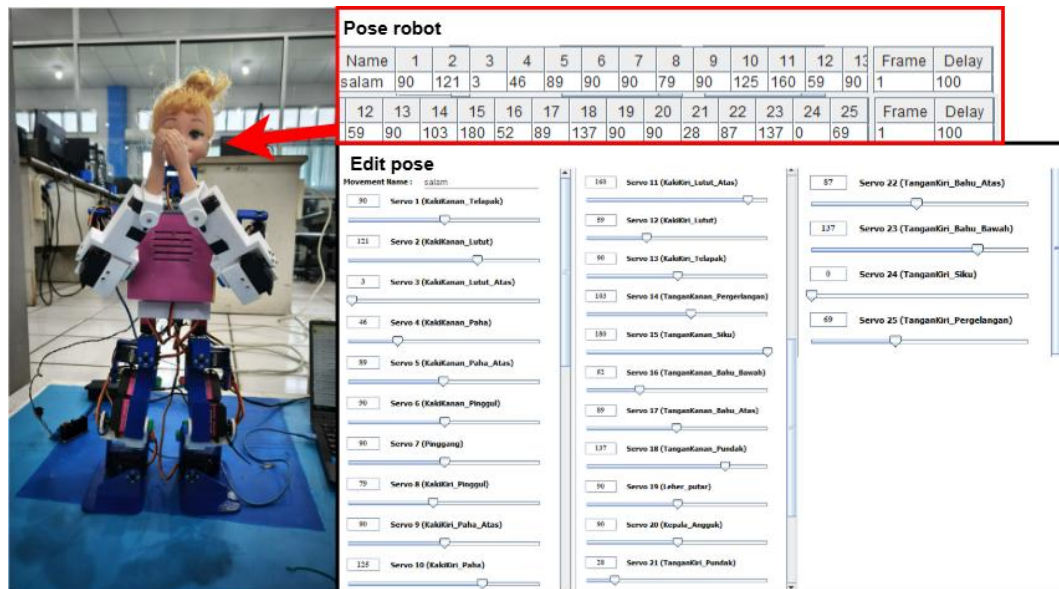
Gambar 3.30 Bagian Data Gerakan pada *Software*

Software ini juga menyediakan pengaturan waktu tunda atau *delay* yang bersifat fleksibel dan disesuaikan dengan kebutuhan pergerakan robot. Pengaturan ini dibagi menjadi tiga komponen utama, yaitu *step delay*, jumlah *step*, dan *frame delay*, yang masing-masing berperan dalam menentukan kehalusan dan kecepatan gerakan secara keseluruhan.

Fitur-fitur ini sangat membantu dalam proses pembuatan gerakan tari, yang mana pada fitur *step delay* akan menentukan seberapa cepat sudut servo akan berpindah dari posisi satu ke posisi berikutnya dalam satu baris data gerakan. *Step delay* juga mempengaruhi jumlah *step* yang merupakan banyaknya langkah interpolasi yang akan dilakukan antara dua baris gerakan yang berurutan. Sementara itu, *frame delay* adalah waktu tunda yang diberikan setelah seluruh posisi servo dalam satu baris data (atau satu *frame* gerakan) telah dicapai. dalam menyusun koreografi dengan ritme yang variatif.

Setelah semua gerakan sudah dibuat dan diuji coba dengan menekan *button play*, maka selanjutnya adalah mengupload program data gerakan ke dalam program *main controller*. *Software* ini dapat meng-*export* data gerakan kedalam file

dengan format *.ino* sehingga pengguna hanya perlu menyalin dan menempelkannya ke dalam program *main controller*.



Gambar 3.31 Pose Robot dengan *software* kontrol servo

BAB IV PEMBAHASAN

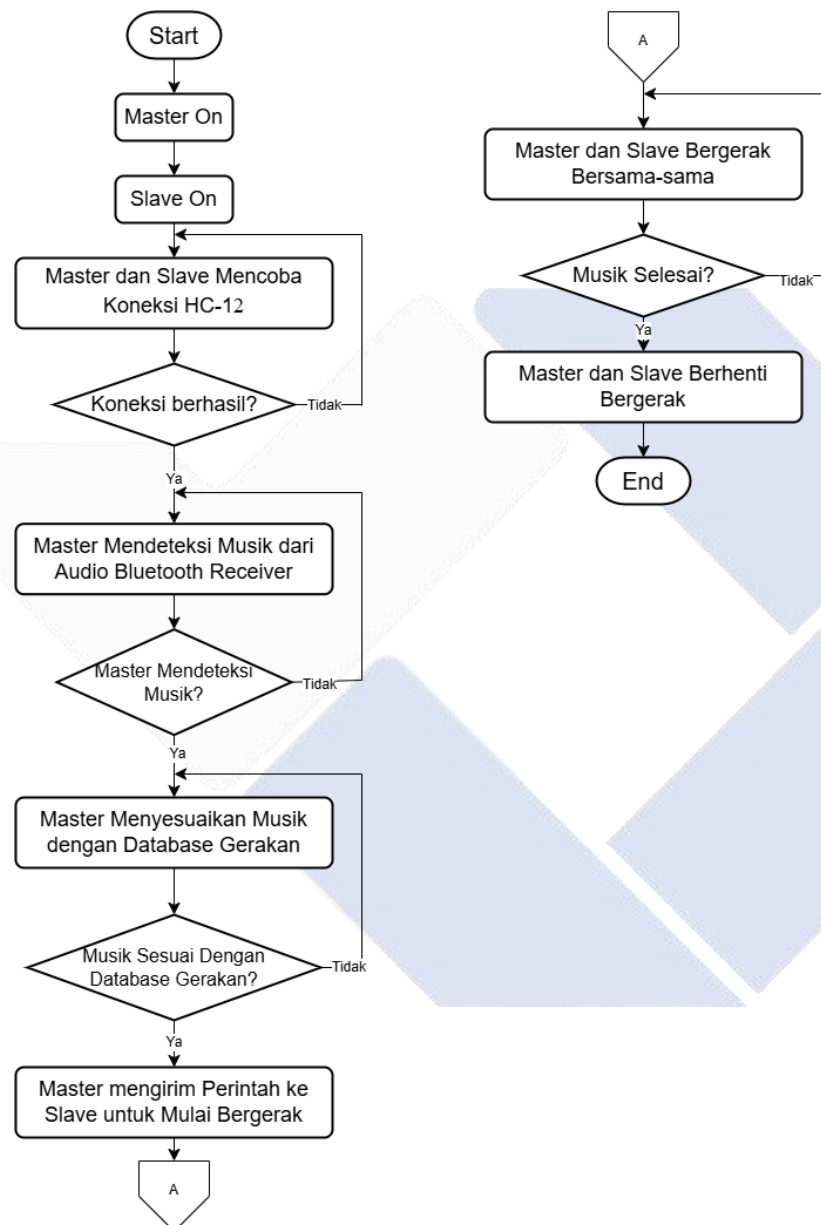
Bab ini membahas secara rinci proses pelaksanaan proyek akhir, yang mencakup tahapan konstruksi alat, perancangan sistem kendali, serta pengujian-pengujian yang berkaitan dengan pengembangan sistem. Selain itu, bab ini juga mengulas berbagai permasalahan yang muncul selama proses pembuatan dan pengujian alat. Setelah seluruh pengujian dilakukan, data yang diperoleh akan dianalisis dan dievaluasi guna menilai kinerja sistem secara keseluruhan.

4.1 Deskripsi Alat

Sistem robot tari humanoid yang meniru tubuh manusia telah dirancang dalam proyek akhir ini, dengan 25 derajat kebebasan (DoF) yang tersebar pada bagian kepala, tangan, badan, dan kaki. Robot ini dapat dikendalikan oleh perangkat lunak yang dirancang khusus untuk kebutuhan pemrograman gerakan, dan dapat menampilkan gerakan tari secara sinkron dan dinamis sesuai dengan irama musik. Menggunakan teknologi 3D printing, struktur mekanis robot ini sebagian besar terbuat dari PLA+ yang ringan dan kuat, yang memungkinkan gerakan servo yang presisi tanpa memberikan beban berlebih pada rangka. Mikrokontroler ESP32 menggunakan modul PWM eksternal PCA9685 untuk mengendalikan motor servo torsi tinggi (DS5160) yang menggerakkan setiap bagian tubuh. Sebagai prosesor utama dalam sistem ini, ESP32 juga mengontrol perintah untuk gerakan servo dan menerima perintah dari komputer melalui komunikasi serial.

Robot humanoid ini dilengkapi dengan modul komunikasi HC-12 yang memungkinkan sinkronisasi gerakan antara dua robot (*master* dan *slave*), serta modul penerima audio Bluetooth yang dapat mengenali sinyal musik sebagai pemicu gerakan. Baterai Li-Po 11,1V berfungsi sebagai sumber daya utama sistem, dan konverter buck memastikan bahwa setiap komponen menerima distribusi tegangan yang stabil. Perangkat lunak pendukung, yang dikembangkan dengan Java Swing di IDE Eclipse, memungkinkan pengguna untuk mengatur urutan pose gerakan ke dalam tabel gerakan, mengontrol sudut setiap servo menggunakan

slider, dan mengeksekusinya satu per satu. Akibatnya, sistem ini memungkinkan pemrograman gerakan tari secara visual dan efektif tanpa harus melalui langkah kompilasi dan unggah lagi, yang diperlukan untuk pengembangan tradisional Arduino IDE.



Gambar 4.1 *Flowchart* sistem keseluruhan

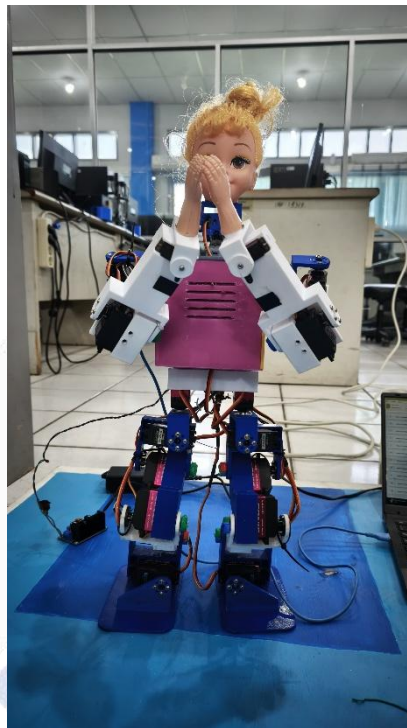
4.2 Perakitan *Hardware* Robot

Tahap perencanaan dan pembuatan robot humanoid seni tari, dilakukan proses konstruksi rangka tubuh robot serta pemasangan berbagai komponen dan

perangkat pendukung. Seluruh proses ini dibagi menjadi beberapa tahapan, antara lain:

4.2.1 Instalasi Servo

Servo dipasang disemua dudukan yang ada pada kerangka robot humanoid, yang nantinya servo akan menggerakkan kerangka sesuai dengan *database* gerakan.



Gambar 4.2 Instalasi Servo

4.2.2 Pemasangan ESP32

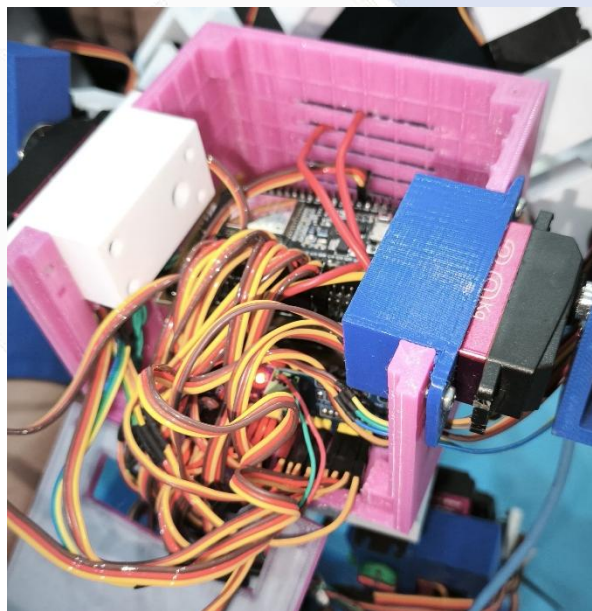
Pemasangan ESP32 pada bagian dalam *body* robot. ESP 32 berfungsi sebagai pemrosesan sinyal, kontrol motor servo, dan pengolah data komunikasi.



Gambar 4.3 Pemasangan ESP32

4.2.3 Pemasangan dan Perakitan Rangkaian Komponen Elektrik

Pemasangan PCA9685, dll. Dipasang dibagian dalam *body* robot.

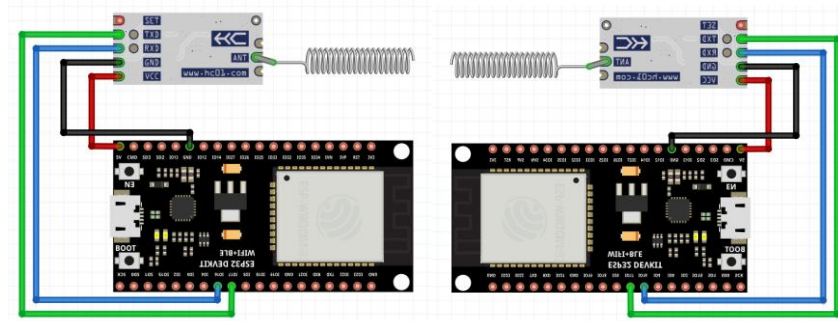


Gambar 4.4 Pemasangan Rangkaian Elektrik

4.3 Pengujian HC-12

Komponen yang digunakan agar robot humanoid dapat berkomunikasi adalah HC-12 yang dimana robot dapat saling bertukar informasi. Pengujian modul HC-12 dilakukan untuk memastikan kemampuan komunikasi antarrobot secara cepat dan akurat. Pengujian ini menggunakan dua perangkat sebagai *transmitter (master)* dan *receiver (slave)* dihubungkan menggunakan modul HC-12 untuk mentransmisikan data berupa karakter

atau *string*. Aspek yang diuji meliputi keberhasilan pengiriman data, jangkauan maksimal komunikasi yang masih efektif, serta waktu tunda (latensi) dari pengiriman hingga penerimaan data. Hasil pengujian dicatat melalui tampilan serial monitor dan dianalisis untuk mengetahui performa komunikasi pada berbagai jarak.



Gambar 4.5 (a). Rangkaian Pengujian *Master Robot* (b). Rangkaian Pengujian *Slave Robot*

Hasil pengujian didapat hasil sebagai berikut.



Gambar 4.6 Grafik selisih waktu kirim dan terima HC-12

Grafik di atas menunjukkan visualisasi perbedaan waktu antara saat perintah dikirim oleh perangkat master dan saat diterima oleh perangkat *slave* menggunakan modul komunikasi nirkabel HC-12. Grafik ini memetakan nilai selisih waktu dari setiap sampel pengujian sebanyak 10 kali, yang sebelumnya telah ditampilkan dalam bentuk tabel. Terlihat bahwa nilai selisih waktu bervariasi dengan fluktuasi yang cukup signifikan, di mana pada beberapa titik seperti pengujian ke-3, ke-4, dan ke-7 nilai selisih waktu mendekati nol, yang menunjukkan proses pengiriman data berlangsung

sangat cepat. Sebaliknya, pada pengujian ke-6 dan ke-8 nilai selisih waktu mencapai angka tertinggi, yaitu sekitar 46 milidetik.

Tabel 4.1 Tabel selisih waktu kirim dan terima HC-12

No	Waktu Kirim	Kirim Perintah	Waktu Terima	Terima Perintah	Selisih Waktu (ms)
1	14:08:29.001	"1"	14:08:29.037	"1"	36
2	14:08:34.000	"1"	14:08:34.033	"1"	33
3	14:08:39.033	"1"	14:08:39.033	"1"	0
4	14:08:44.015	"1"	14:08:44.016	"1"	1
5	14:08:48.998	"1"	14:08:49.035	"1"	37
6	14:08:54.006	"1"	14:08:54.052	"1"	46
7	14:08:59.015	"1"	14:08:59.015	"1"	0
8	14:09:04.007	"1"	14:09:04.053	"1"	46
9	14:09:09.028	"1"	14:09:09.030	"1"	2
10	14:09:13.998	"1"	14:09:14.034	"1"	36
Rata-rata					23,7

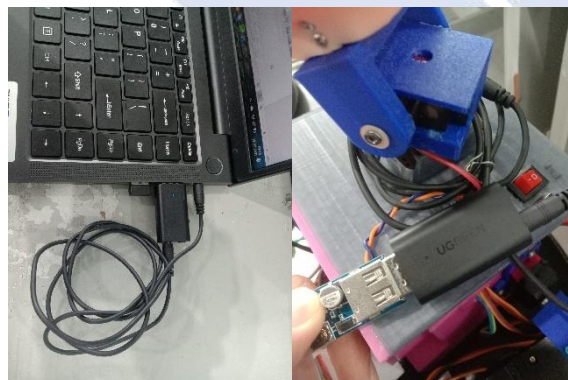
Berdasarkan hasil pengujian modul komunikasi nirkabel HC-12 yang ditampilkan pada tabel, diperoleh data waktu pengiriman dan penerimaan perintah dari perangkat master ke *slave* dalam sepuluh kali percobaan. Terdapat perbedaan waktu antara timestamp master saat perintah dikirim dan timestamp *slave* saat perintah diterima, yang dihitung sebagai selisih waktu transmisi. Dari sepuluh sampel data, selisih waktu bervariasi antara 0 hingga 46 milidetik, dengan nilai rata-rata sebesar 23,7 milidetik. Nilai ini menunjukkan bahwa HC-12 memiliki waktu latensi yang relatif rendah dan cukup konsisten untuk komunikasi dua arah dalam sistem kendali waktu nyata (real-time). Hasil ini menegaskan bahwa modul HC-12 layak digunakan dalam aplikasi robotik atau kendali nirkabel jarak jauh yang membutuhkan respons cepat dan andal.



Gambar 4.7 Dokumentasi Pengujian HC-12

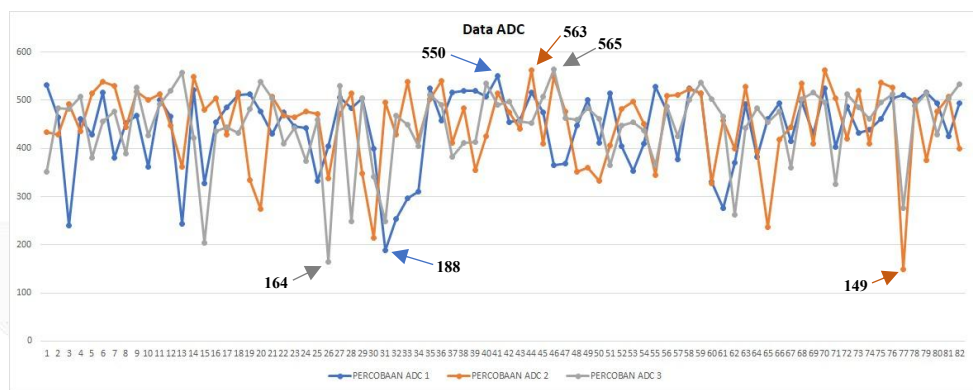
4.4 Pengujian Sinyal ADC Terhadap Kondisi Suara Musik

Pengujian ini bertujuan untuk mengevaluasi kinerja perangkat UGREEN CM523 dalam mengirimkan sinyal audio dari laptop ke sistem kendali robot berbasis ESP32. Implementasi pada sistem robot tari ini, suara lagu yang dikirim melalui UGREEN akan dideteksi oleh ESP32 melalui pembacaan sinyal analog dari pin ADC. Keberadaan lagu akan menjadi pemicu untuk mengaktifkan gerakan tari robot secara otomatis. Pengujian dilakukan dengan konfigurasi di laptop, di mana laptop berfungsi sebagai *transmitter* yang memutar lagu dan mengirimkan sinyal audio melalui jack 3.5 mm ke UGREEN transmitter, sedangkan *receiver* dihubungkan ke input analog ESP32 yang bertugas membaca sinyal tersebut dan di baca melalui serial monitor di laptop.



Gambar 4.8 (a) Audio transmitter terpasang pada laptop (b) Audio receiver terpasang pada robot humanoid

ESP32 menggunakan pin analog 35 (audioPin) untuk membaca sinyal audio secara langsung dari output UGREEN receiver. Pada setiap siklus loop, program melakukan pembacaan ADC sebanyak 82 kali (sampleCount) dan mencatat nilai minimum serta maksimum dari sinyal yang terbaca. Jika nilai ADC lebih besar dari threshold (100), maka sistem menganggap bahwa terdapat sinyal lagu aktif, dan menampilkan pesan “ADA lagu”. Jika tidak, maka sistem menyatakan “TIDAK ada lagu”.



Gambar 4.9 Grafik sinyal ADC yang diterima receiver

Pada gambar 4.8, data maksimum dan minimum sinyal ADC dicantumkan pada tabel berikut.

Tabel 4.2 Tabel data sinyal ADC

Percobaan	Nilai Maksimum	Nilai Minimum	Selisih
Percobaan ADC 1	550	188	362
Percobaan ADC 2	563	149	414
Percobaan ADC 3	565	164	401

Pengujian ini dilakukan untuk mengevaluasi performa sistem deteksi sinyal audio menggunakan modul UGREEN CM523 sebagai pemancar (*transmitter*) dan penerima (*receiver*), serta pembacaan sinyal oleh mikrokontroler ESP32 melalui pin analog. Tujuan utama dari pengujian ini adalah untuk memastikan bahwa sistem mampu mendeteksi keberadaan lagu secara akurat berdasarkan variasi sinyal analog yang diubah ke dalam bentuk digital oleh ADC (*Analog-to-Digital Converter*). Sistem ini selisih ADC

sinyal dihitung sebagai selisih antara nilai maksimum dan minimum dari hasil pembacaan ADC sebanyak 82 sampel setiap siklusnya.

Hasil pengujian menunjukkan bahwa pada percobaan pertama, nilai ADC maksimum sebesar 550 dan minimum sebesar 188, dengan selisih sebesar 362. Pada percobaan kedua, nilai maksimum 563 dan minimum 149 menghasilkan selisih sebesar 414. Sedangkan pada percobaan ketiga, nilai maksimum mencapai 565 dan minimum 164, memberikan selisih sebesar 401. Ketiga percobaan menunjukkan nilai ADC yang signifikan dan secara konsisten berada di atas ambang batas deteksi (*threshold*) sebesar 100. Hal ini menunjukkan bahwa sistem mampu mendeteksi adanya sinyal lagu dengan baik.

4.5 Pengujian Data Serial Sudut Servo

Sistem robot humanoid seni tari agar dapat merespons perintah sudut dengan tepat dan akurat diperlukan pengujian sudut servo. Servo merupakan aktuator utama yang menggerakkan bagian tubuh robot sesuai koreografi, sehingga diperlukan verifikasi fungsionalitas sebelum digunakan dalam keseluruhan sistem. Pengujian difokuskan pada pengendalian satu buah servo yang dihubungkan ke salah satu pin PWM pada board ESP32. Tujuannya adalah untuk memverifikasi apakah servo dapat bergerak ke sudut tertentu yang dikirimkan melalui komunikasi serial dari komputer.

Berikut program pengujian servo :

```
#include <ESP32Servo.h>
const int servoPin = 13;
Servo myServo;
void setup() {
    Serial.begin(115200);
    myServo.attach(servoPin);
    myServo.write(90);
}
```

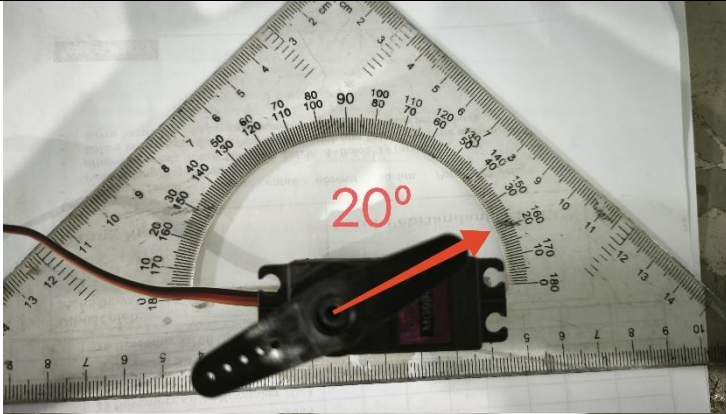
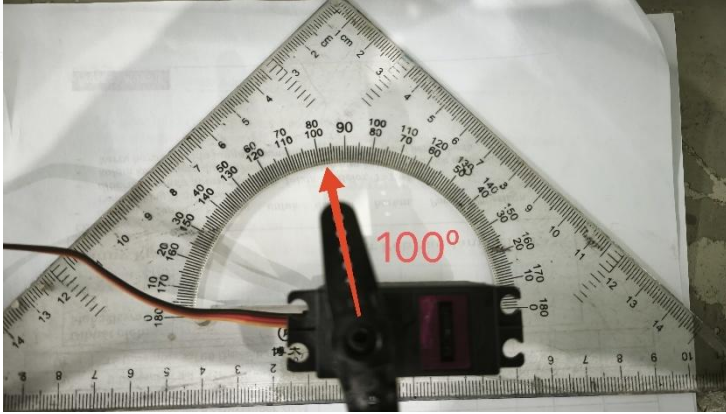
```

void loop() {
    if (Serial.available()) {
        String input = Serial.readStringUntil('\n');
        input.trim();
        if (input.length() > 0) {
            int angle = input.toInt();
            if (angle >= 0 && angle <= 180) {
                myServo.write(angle);
                Serial.print("Servo bergerak ke sudut: ");
                Serial.println(angle);
            } else {
                Serial.println("Sudut harus antara 0 - 180
derajat.");
            }
        }
    }
}

```

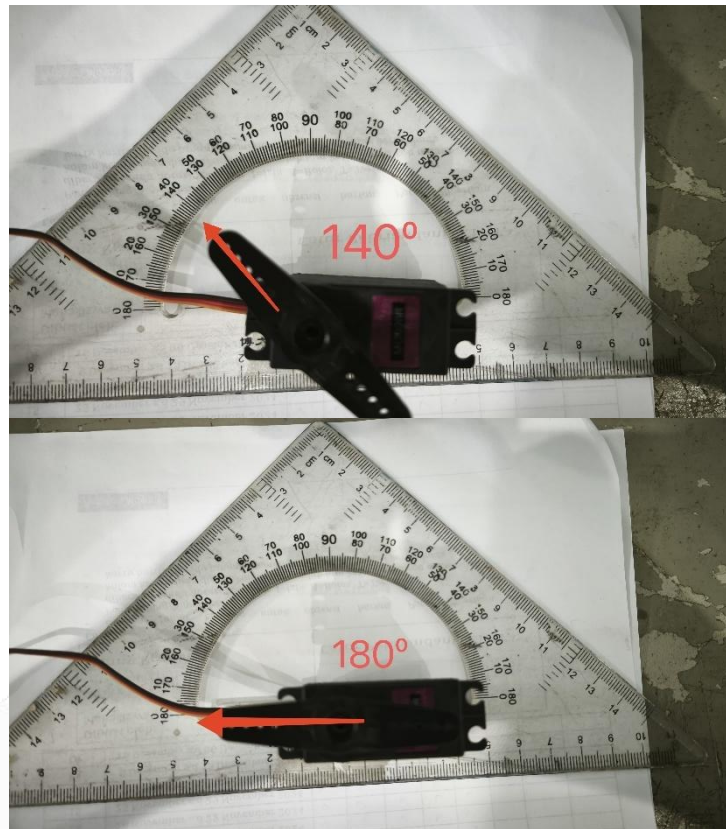
Bagian program *setup()*, servo dihubungkan ke pin GPIO 13 dan diatur ke posisi awal 90°. Selanjutnya, memasukkan nilai sudut antara 0 hingga 180 melalui serial monitor. Pada fungsi *loop()*, program secara kontinu memeriksa apakah terdapat *input* dari serial. Jika *input* terdeteksi dan nilainya berada dalam rentang valid (0–180 derajat), maka servo akan digerakkan ke sudut tersebut menggunakan perintah *myServo.write(angle)*. Sistem juga akan memberikan *feedback* melalui serial monitor terkait sudut yang dicapai atau pesan kesalahan jika *input* tidak valid.

Tabel 4.3 Tabel Pengukuran Servo

No	Sudut Input (°)	Gambar Servo
1	20	
2	60	
3	100	

4 140

5 180



Tabel 4.4 Tabel Pengujian Data Serial Sudut Servo

No.	Sudut <i>Input</i> (°)	Respons Servo	Error (%)
1	20	20°	0
2	60	60°	0
3	100	100°	0
4	140	140°	0
5	180	180°	0

Selama pengujian, pengguna mengirimkan beberapa nilai sudut melalui serial monitor untuk melihat bagaimana servo merespons. Output menunjukkan bahwa servo berhasil bergerak ke sudut 20°, 60°, 100°, 140°, dan akhirnya 180°. Setiap perintah menghasilkan respons balik dari ESP32 yang mencetak informasi sudut ke serial monitor. Tidak terdapat pesan kesalahan, yang menandakan bahwa semua *input* berada dalam rentang yang diperbolehkan dan sistem berhasil memprosesnya. Dapat disimpulkan bahwa komunikasi serial berjalan lancar, parsing *input* tidak bermasalah, dan fungsi kendali servo bekerja sesuai spesifikasi.

Pengujian dilakukan sebanyak 5 kali dengan sudut berbeda, seluruh perintah berhasil dijalankan dan servo merespons dengan tepat. Maka, tingkat keberhasilan dan kesalahan dapat dihitung dengan persamaan sebagai berikut:

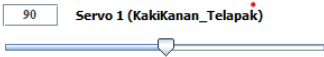
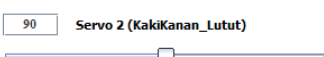
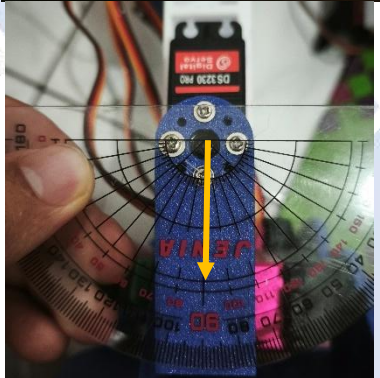
$$\%ERROR = \frac{|Sudut\ Input - Respons\ Servo|}{Respons\ Servo} \times 100\% \quad (2)$$

Hasil pengujian menunjukkan bahwa sistem kontrol servo dengan ESP32 bekerja secara optimal dan sesuai spesifikasi. Servo berhasil merespons lima perintah sudut yang berbeda tanpa adanya kesalahan *input* atau anomali pergerakan. Hal ini membuktikan bahwa baik komunikasi serial, parsing *input*, hingga eksekusi sinyal PWM ke servo dapat diandalkan. Dengan tingkat keberhasilan 100%, kontrol servo ini dapat dilanjutkan ke tahap integrasi dengan sistem robot humanoid secara keseluruhan untuk gerakan koreografi yang lebih kompleks.

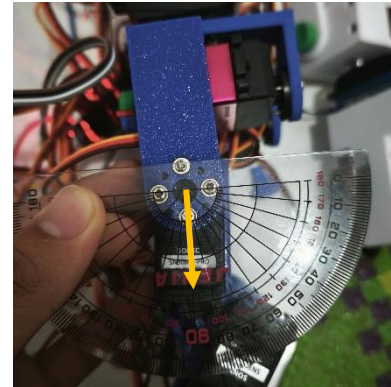
4.6 Pengujian Komunikasi *Software* dengan Servo

Pengujian ini dilakukan untuk mengevaluasi akurasi pergerakan sudut servo pada robot seni tari 25 DoF yang dikendalikan melalui sebuah *software* kontrol berbasis *Java Swing*. Tujuan utama dari pengujian ini adalah untuk memastikan bahwa sudut *input* dari *software* dapat diterjemahkan dengan benar oleh servo fisik, serta mengidentifikasi adanya deviasi atau *error* dalam pergerakan aktual servo terhadap nilai yang diharapkan. Pengujian ini juga penting untuk menilai performa tiap servo dan mendeteksi gangguan potensial, baik dari sisi perangkat lunak, perangkat keras, maupun kalibrasi. Berikut adalah tabel kondisi awal servo:

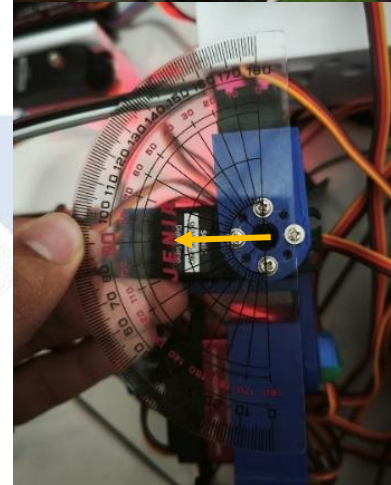
Tabel 4.5 Tabel Kondisi Awal Servo

No	Sudut <i>Input</i> pada <i>Software</i>	Sudut <i>Input</i> (°)	Respons Servo
1		90	
2		90	
3		90	

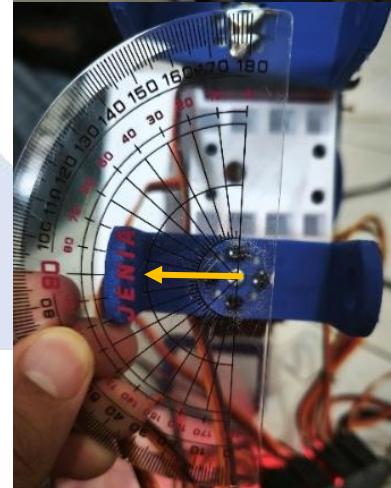
4 Servo 4 (KakiKanan_Paha) 90



5 Servo 5 (KakiKanan_Paha_Atas) 90



6 Servo 6 (KakiKanan_Pinggul) 90



7 Servo 7 (Pinggang) 90



8



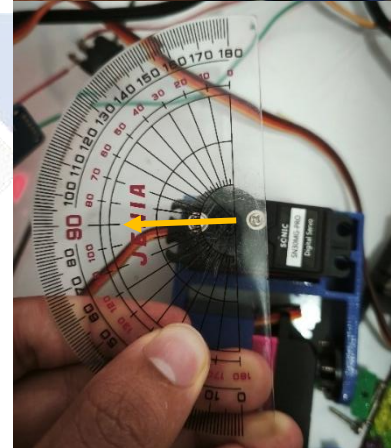
90



9



90



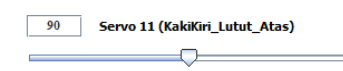
10



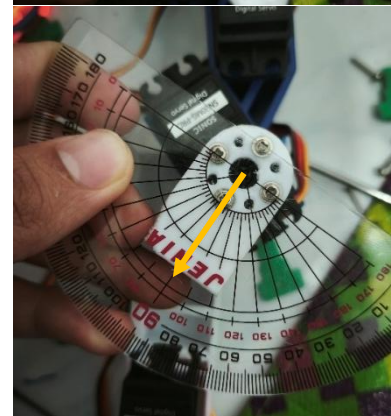
90



11



90



12



90



13



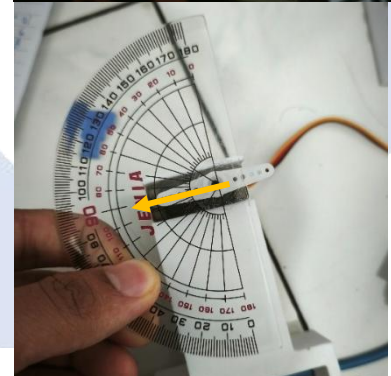
90



14



90



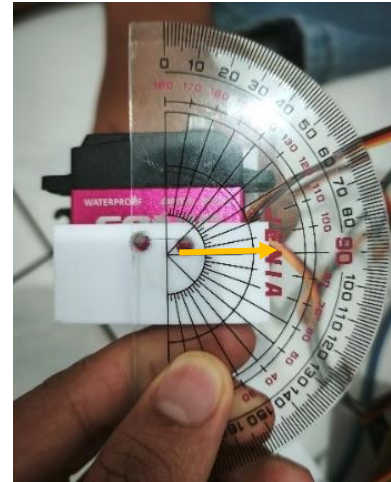
15



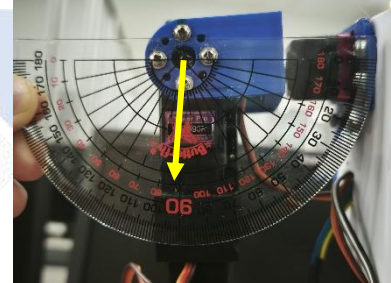
90



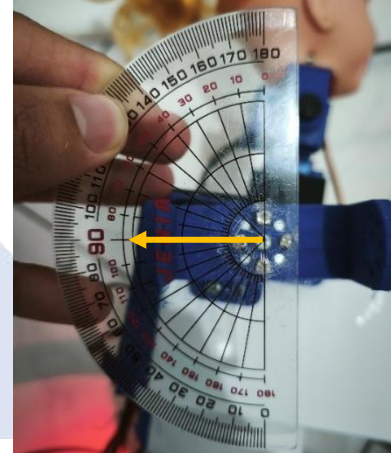
16 Servo 16 (TanganKanan_Bahu_Bawah) 90



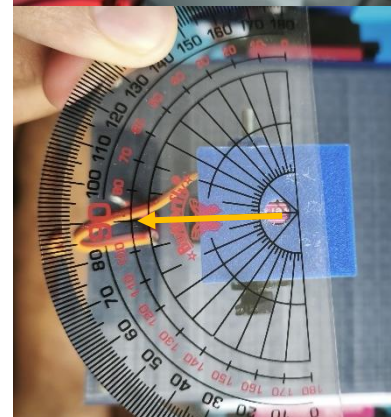
17 Servo 17 (TanganKanan_Bahu_Atas) 90



18 Servo 18 (TanganKanan_Pundak) 90



19 Servo 19 (Leher_putar) 90



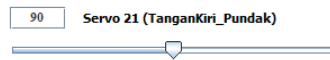
20



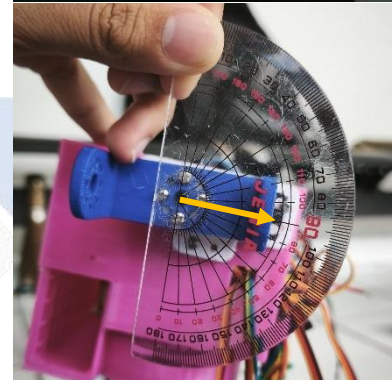
90



21



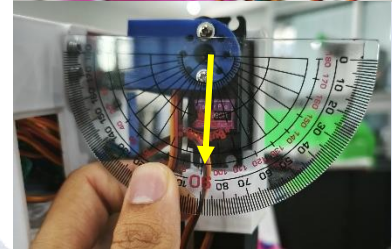
90



22



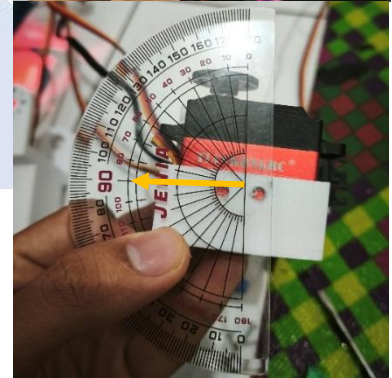
90



23



90

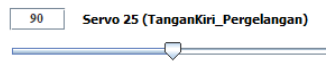


24

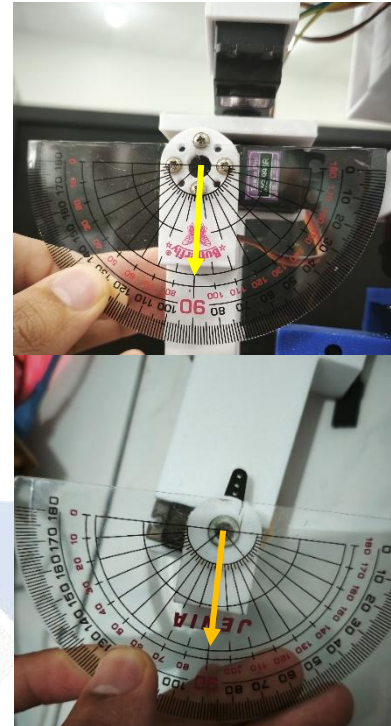


90

25

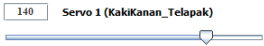
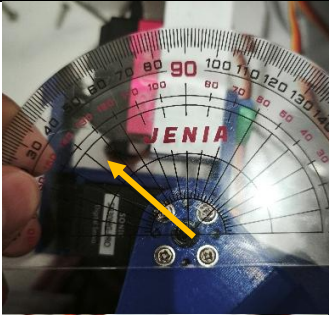
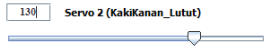
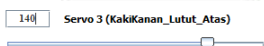
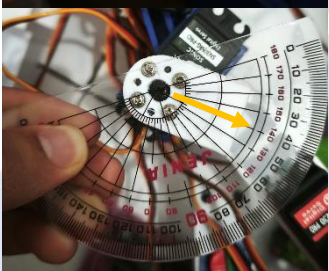
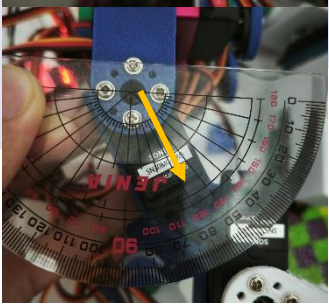


90



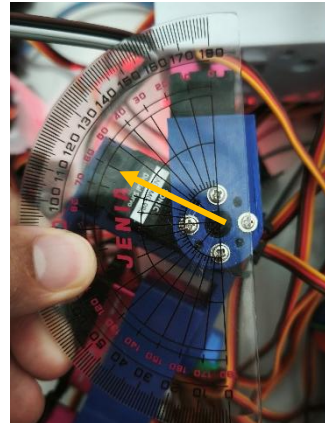
Pada tabel 4.5 mendokumentasikan kondisi awal dari seluruh 25 servo, di mana setiap slider pada antarmuka *software* diset pada 90° . Data yang ditampilkan berupa gambar fisik masing-masing servo beserta tangkapan layar (screenshot) dari tampilan *software* yang menunjukkan sudut slider berada tepat di posisi 90° . Tujuan dari tabel ini adalah untuk memastikan bahwa seluruh servo dimulai dari posisi awal yang seragam dan terkalibrasi, sehingga dapat dijadikan sebagai patokan awal (baseline) dalam proses pengujian berikutnya. Dengan dokumentasi ini, setiap penyimpangan yang terjadi selama pengujian dapat dilacak dengan lebih akurat.

Tabel 4.6 Tabel Pergerakan Servo Terhadap *Software*

No	Sudut <i>Input</i> pada <i>Software</i>	Sudut <i>Input</i> (°)	Gambar Respons Servo	Respon Servo (°)
1		140		137
2		130		132
3		140		140
4		120		118

5 Servo 5 (KakiKanan_Paha_Atas)

120



118

6 Servo 6 (KakiKanan_Pinggul)

130



130

7 Servo 7 (Pinggang)

150



149

8 Servo 8 (KakiKiri_Pinggul)

50

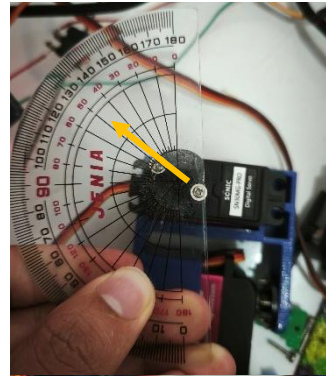


50

9



50

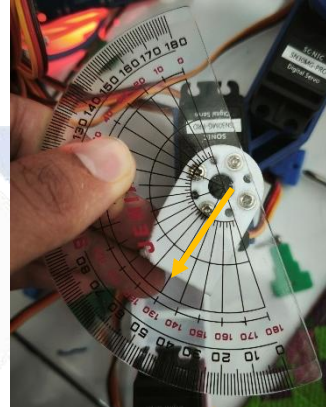


50

10



130



133

11

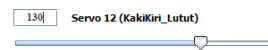


130



135

12



130



133

13



60

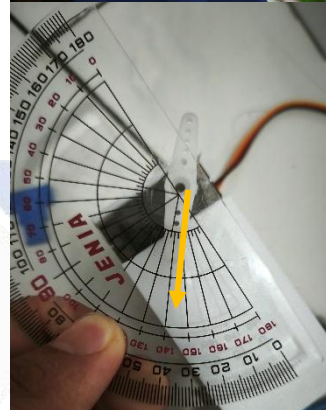


60

14

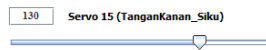


140



140

15



130

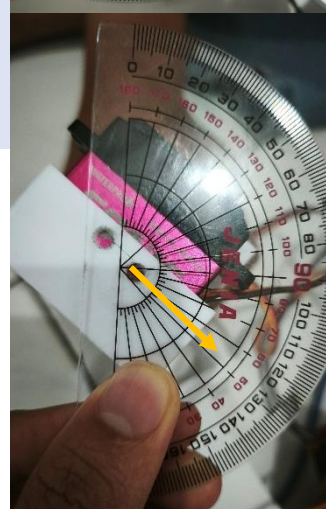


129

16



50

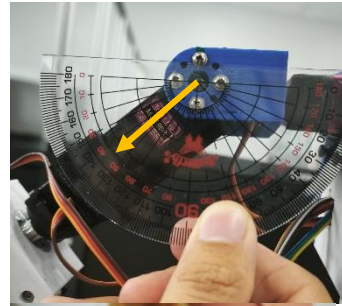


50

17



140

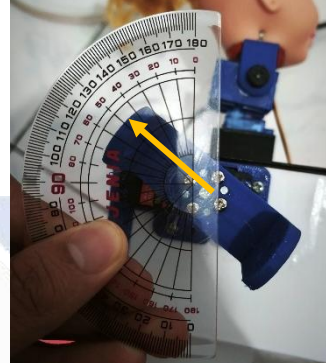


145

18



50

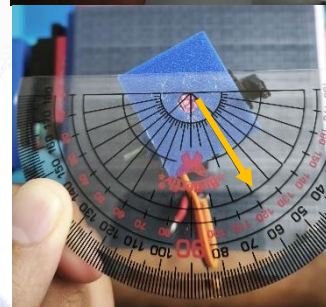


50

19

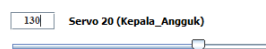


120



119

20



130

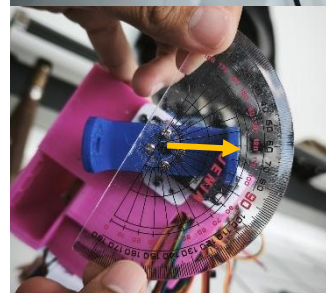


130

21

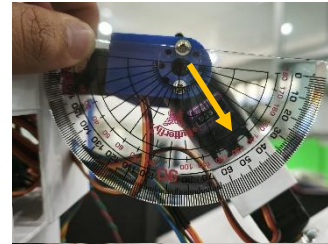


60

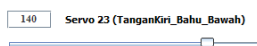


63

22  50



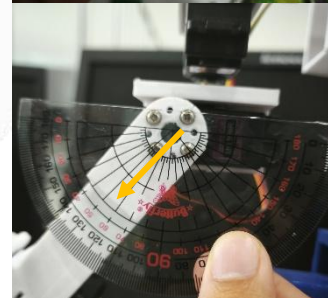
49

23  140

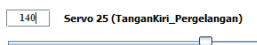


140

24  130



133

25  140



140

Berdasarkan tabel 4.6, ditunjukkan gambar dokumentasi untuk setiap servo yang digerakkan dari posisi awal menuju posisi tertentu. Setiap baris mencakup gambar fisik posisi servo setelah bergerak serta screenshot slider *software* yang menunjukkan sudut input terbaru, misalnya 130°, 140°, atau 50°. Tujuan dari dokumentasi ini adalah untuk memberikan bukti visual bahwa perintah dari *software* telah dijalankan, serta untuk mendukung proses analisis visual terhadap kesesuaian gerakan servo. Gambar-gambar ini membantu mengidentifikasi servo yang mungkin tidak merespons dengan

benar terhadap input software, baik karena masalah mekanik, koneksi, maupun kegagalan sinyal PWM.

Tabel 4.7 Tabel Pengujian Keseluruhan Servo Terhadap *Software*

No.	Sudut Input pada <i>Software</i> (°)	Respons Servo (°)	Error (°)	%Error
1	140	137	3	2,14%
2	130	132	2	1,54%
3	140	140	0	0%
4	120	118	2	1,67%
5	120	118	2	1,67%
6	130	130	0	0%
7	150	149	1	0,67%
8	50	50	0	0%
9	50	50	0	0%
10	130	133	3	2,31%
11	130	135	5	3,85%
12	130	133	3	2,31%
13	60	60	0	0%
14	140	140	0	0%
15	130	129	1	0,77%
16	50	50	0	0%
17	140	145	5	3,57%
18	50	50	0	0%
19	120	119	1	0,83%
20	130	130	0	0%
21	60	63	3	5%
22	50	49	1	2%
23	140	140	0	0%
24	130	133	3	2,31%
25	140	140	0	0%
Rata-rata %Error				1,22%

Tabel ini merupakan hasil akhir dari pengujian yang berisi data kuantitatif yang membandingkan antara sudut input dari *software*, sudut respons servo secara aktual, serta sudut yang diharapkan. Selanjutnya, dihitung juga nilai *Error*(°) menggunakan rumus (3).

$$\%ERROR = \frac{|Sudut Input pada Software - Respons Servo|}{Respons Servo} \times 100\% \quad (3)$$

Dari tabel yang ditampilkan, beberapa servo menunjukkan nilai error kecil, seperti 1 hingga 5 derajat, dengan persentase error bervariasi antara 0.8% hingga 3.8%. Sebagian besar servo menunjukkan akurasi tinggi dengan error 0%. Nilai rata-rata error yang dihitung hanya dari data yang valid (tanpa mengikutsertakan kolom yang kosong akibat kendala servo) adalah 0.90%, menunjukkan bahwa sistem kontrol servo dari *software* ke perangkat fisik bekerja dengan cukup baik dan presisi.

4.7 Pengujian Keseluruhan

Pengujian ini dilakukan untuk mengevaluasi performa sistem komunikasi antar perangkat pada robot humanoid yang menggunakan modul HC-12 sebagai media transmisi data nirkabel antar tiga buah mikrokontroler ESP32, yaitu: *ESP1_Master*, *ESP1_Slave*, dan *ESP2_Slave*. Tujuan utama dari pengujian ini adalah untuk memastikan bahwa perintah yang dikirimkan secara berjenjang, mulai dari deteksi suara oleh *ESP1_Master* hingga eksekusi pergerakan oleh *ESP2_Slave* berjalan dengan akurat, sinkron, andal dan *realtime*. Sistem ini dirancang agar robot *slave* dapat secara otomatis melakukan gerakan tari hanya ketika terdapat input musik, sehingga memastikan efisiensi dan responsivitas dalam pertunjukan robotik berbasis musik.

Pada bagian awal sistem, *ESP1_Master* menerima sinyal audio melalui *receiver UGREEN* dan melakukan analisis sinyal ADC tersebut. Jika terdeteksi adanya musik (ditandai dengan nilai $ADC > 100$), *ESP1_Master* akan mengirimkan perintah '1' ke *ESP1_Slave* melalui HC-12, yang berarti perintah untuk "BERGERAK". Sebaliknya, jika tidak ada musik ($ADC = 0$), perintah '0' akan dikirim sebagai instruksi untuk "DIAM". Selanjutnya, *ESP1_Slave* menerima perintah tersebut dan segera meneruskannya ke *ESP2_Slave*. *ESP2_Slave* kemudian mengeksekusi perintah dengan menjalankan seluruh servo (jika diam) atau menghentikannya (jika bergerak).

Berikut adalah tabel dari ESP1_Master saat mengirim perintah melalui HC-12 (*transmitter*) disaat ESP1_Master mendeteksi musik atau tidak.

Tabel 4.8 Tabel Data Deteksi Musik dari ESP1 Master

Waktu	ADC	Threshold	Keterangan
11:57:00.320	0	100	Tidak Ada Musik
11:57:00.957	0	100	Tidak Ada Musik
11:57:01.536	0	100	Tidak Ada Musik
11:57:02.162	0	100	Tidak Ada Musik
11:57:02.720	348	100	Ada Musik
11:57:03.321	132	100	Ada Musik
11:57:03.963	402	100	Ada Musik
11:57:04.557	594	100	Ada Musik
11:57:05.146	108	100	Ada Musik
11:57:05.720	384	100	Ada Musik
11:57:06.337	558	100	Ada Musik
11:57:06.920	294	100	Ada Musik
11:57:07.535	366	100	Ada Musik
11:57:08.136	450	100	Ada Musik
11:57:08.736	0	100	Tidak Ada Musik
11:57:09.336	0	100	Tidak Ada Musik
11:57:09.920	0	100	Tidak Ada Musik
11:57:10.520	0	100	Tidak Ada Musik
11:57:11.156	0	100	Tidak Ada Musik
11:57:11.735	0	100	Tidak Ada Musik

Berikut adalah tabel dari ESP1_Slave saat menerima perintah melalui HC-12 (*receiver*) dari ESP1_Master.

Tabel 4.9 Tabel Data yang diterima dari ESP1 Slave

Waktu	Perintah	Keterangan
11:57:00.366	Kirim ke ESP2: '0' (DIAM)	Perintah diterima dari Master: DIAM
11:57:00.957	Kirim ke ESP2: '0' (DIAM)	Perintah diterima dari Master: DIAM
11:57:01.537	Kirim ke ESP2: '0' (DIAM)	Perintah diterima dari Master: DIAM
11:57:02.163	Kirim ke ESP2: '0' (DIAM)	Perintah diterima dari Master: DIAM
11:57:02.755	Kirim ke ESP2: '1' (GERAK)	Perintah diterima dari Master: GERAK
11:57:03.363	Kirim ke ESP2: '1' (GERAK)	Perintah diterima dari Master: GERAK
11:57:03.964	Kirim ke ESP2: '1' (GERAK)	Perintah diterima dari Master: GERAK

11:57:04.558	Kirim ke ESP2: '1' (GERAK)	Perintah diterima dari Master: GERAK
11:57:05.146	Kirim ke ESP2: '1' (GERAK)	Perintah diterima dari Master: GERAK
11:57:05.754	Kirim ke ESP2: '1' (GERAK)	Perintah diterima dari Master: GERAK
11:57:06.338	Kirim ke ESP2: '1' (GERAK)	Perintah diterima dari Master: GERAK
11:57:06.956	Kirim ke ESP2: '1' (GERAK)	Perintah diterima dari Master: GERAK
11:57:07.536	Kirim ke ESP2: '1' (GERAK)	Perintah diterima dari Master: GERAK
11:57:08.183	Kirim ke ESP2: '1' (GERAK)	Perintah diterima dari Master: GERAK
11:57:08.737	Kirim ke ESP2: '0' (DIAM)	Perintah diterima dari Master: DIAM
11:57:09.336	Kirim ke ESP2: '0' (DIAM)	Perintah diterima dari Master: DIAM
11:57:09.959	Kirim ke ESP2: '0' (DIAM)	Perintah diterima dari Master: DIAM
11:57:10.566	Kirim ke ESP2: '0' (DIAM)	Perintah diterima dari Master: DIAM
11:57:11.156	Kirim ke ESP2: '0' (DIAM)	Perintah diterima dari Master: DIAM
11:57:11.783	Kirim ke ESP2: '0' (DIAM)	Perintah diterima dari Master: DIAM

Berikut adalah tabel dari ESP2_Slave saat menerima perintah dari ESP1_Slave untuk mengeksekusi semua servo.

Tabel 4.10 Tabel ESP2_Slave saat menerima perintah dari ESP1_Slave

Waktu	Perintah Diterima	Status
11:57:00.412	Terima sinyal: '0' (BERHENTI)	Dijalankan
11:57:01.003	Terima sinyal: '0' (BERHENTI)	Dijalankan
11:57:01.621	Terima sinyal: '0' (BERHENTI)	Dijalankan
11:57:02.209	Terima sinyal: '0' (BERHENTI)	Dijalankan
11:57:02.802	Terima sinyal: '1' (MULAI GERAK)	Dijalankan
11:57:03.410	Terima sinyal: '1' (MULAI GERAK)	Dijalankan
11:57:04.011	Terima sinyal: '1' (MULAI GERAK)	Dijalankan
11:57:04.605	Terima sinyal: '1' (MULAI GERAK)	Dijalankan
11:57:05.184	Terima sinyal: '1' (MULAI GERAK)	Dijalankan
11:57:05.786	Terima sinyal: '1' (MULAI GERAK)	Dijalankan
11:57:06.425	Terima sinyal: '1' (MULAI GERAK)	Dijalankan
11:57:07.002	Terima sinyal: '1' (MULAI GERAK)	Dijalankan
11:57:07.621	Terima sinyal: '1' (MULAI GERAK)	Dijalankan
11:57:08.231	Terima sinyal: '1' (MULAI GERAK)	Dijalankan
11:57:08.822	Terima sinyal: '0' (BERHENTI)	Dijalankan

11:57:09.427	Terima sinyal: '0' (BERHENTI)	Dijalankan
11:57:10.007	Terima sinyal: '0' (BERHENTI)	Dijalankan
11:57:10.612	Terima sinyal: '0' (BERHENTI)	Dijalankan
11:57:11.188	Terima sinyal: '0' (BERHENTI)	Dijalankan
11:57:11.830	Terima sinyal: '0' (BERHENTI)	Dijalankan

Tabel 4.11 menyajikan hasil pengujian sinkronisasi waktu pelaksanaan gerakan antara robot 1 (sebagai *master*) dan robot 2 (sebagai *slave*) dalam 17 segmen gerakan tari, dimulai dari kondisi diam hingga gerakan penutup. Setiap baris dalam tabel menunjukkan waktu tempuh masing-masing robot dalam menyelesaikan satu segmen gerakan, dengan satuan detik. Untuk mengukur sejauh mana perbedaan durasi antara kedua robot, dilakukan perhitungan persentase menggunakan rumus:

$$\%ERROR = \frac{\text{Waktu robot } master - \text{Waktu robot } slave}{\text{Waktu robot } master} \times 100\% \quad (4)$$

Tabel 4.11 Pengujian ke-1 sinkronisasi gerakan kedua robot

No	Gerakan robot	Waktu gerakan robot <i>master</i> (s)	Waktu gerakan robot <i>slave</i> (s)	%Error
1	Diam	0	0	0%
2	Sembah pembuka	11,4	12,2	7,0%
3	Gerakan 1	16,2	16,3	0,6%
4	Gerakan 2	19,2	20,7	7,8%
5	Gerakan 3	19,2	20,1	4,7%
6	Gerakan 4	18,4	19,1	3,8%
7	Gerakan 5	14,4	14,5	0,7%
8	Gerakan 6	18,4	18,7	1,6%
9	Gerakan 7	19,2	19,5	1,6%
10	Gerakan 8	19,2	19,6	2,1%
11	Gerakan 9	16,2	17,2	6,2%
12	Gerakan 10	17,9	18	0,6%
13	Gerakan 11	15,9	16,2	1,9%
14	Gerakan 12	19,2	20,1	4,7%
15	Gerakan 13	19,2	19,6	2,1%
16	Gerakan 14	18,4	19,6	6,5%
17	Tarian penutup	11,4	12,4	8,8%
Rata-rata				3,8%

Hasil pengujian ke-1 menunjukkan bahwa seluruh gerakan mengalami perbedaan waktu antara robot 1 dan robot 2, dengan nilai *error* yang bervariasi mulai dari 1% hingga 9%. Nilai *error* tertinggi terjadi pada tarian penutup, yakni sebesar 9%, sedangkan nilai *error* terendah terdapat pada gerakan 1, gerakan 5 dan gerakan 10 sebesar 1%.

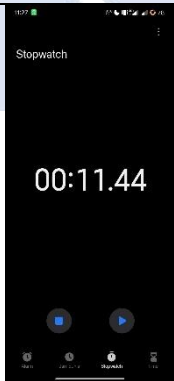
Pengujian keseluruhan dilakukan hingga 5 kali dengan musik dan gerakan yang sama, didapatkan rata-rata *error* adalah 3,2%.

Tabel 4.12 Rata-rata persentase *error* pengujian

No	Pengujian	%Error
1	Pengujian ke-1	3,8%
2	Pengujian ke-2	3,42%
3	Pengujian ke-3	3,29%
4	Pengujian ke-4	2,60%
5	Pengujian ke-5	2,69%
	Rata-rata	3,2%

Pengujian dilakukan dengan membandingkan lama gerakan robot *master* dan *slave* pada setiap gerakan menggunakan *stopwatch* yang tertera pada tabel 4.13.

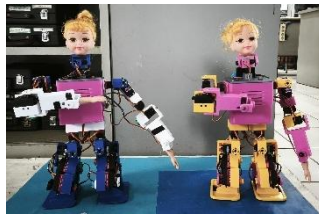
Tabel 4.13 Sinkronisasi gerakan robot

No	Gerakan robot	Dokumentasi gerakan	Waktu gerakan robot <i>master</i>	Waktu gerakan robot <i>slave</i>
1	Sembah pembuka			

2 Gerakan 1



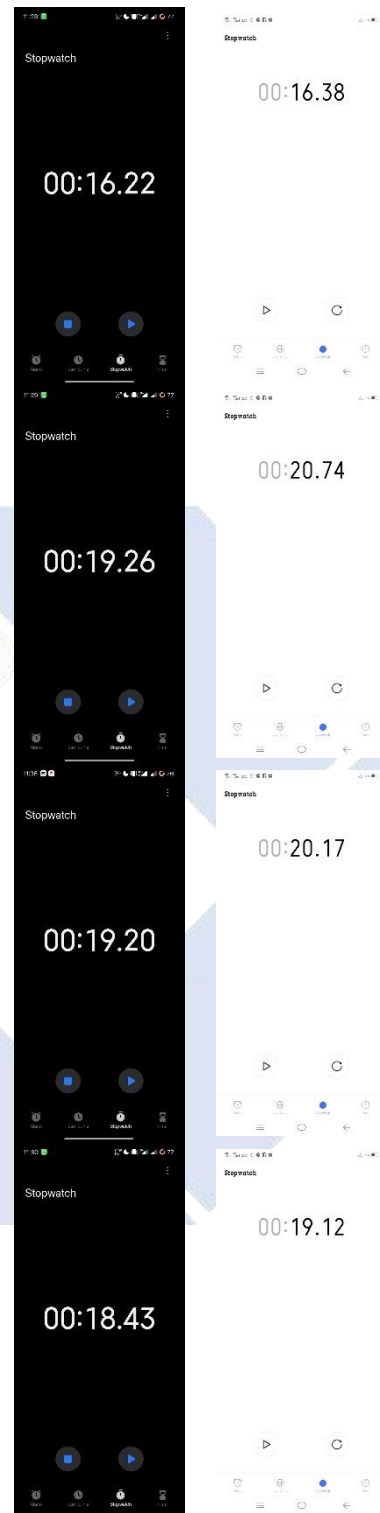
3 Gerakan 2



4 Gerakan 3



5 Gerakan 4



6 Gerakan
5



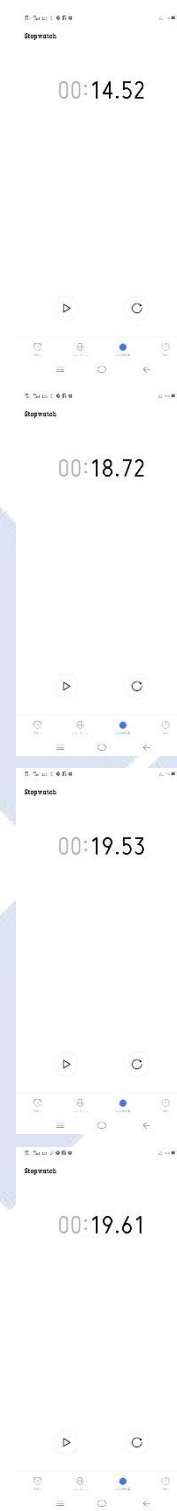
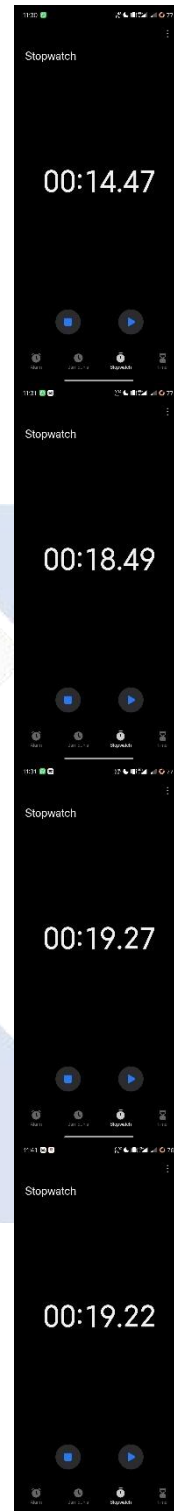
7 Gerakan
6



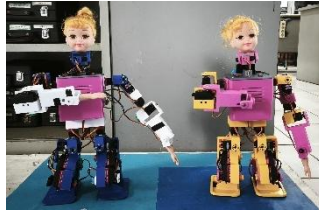
8 Gerakan
7



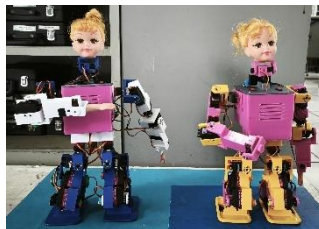
9 Gerakan
8



10 Gerakan
9



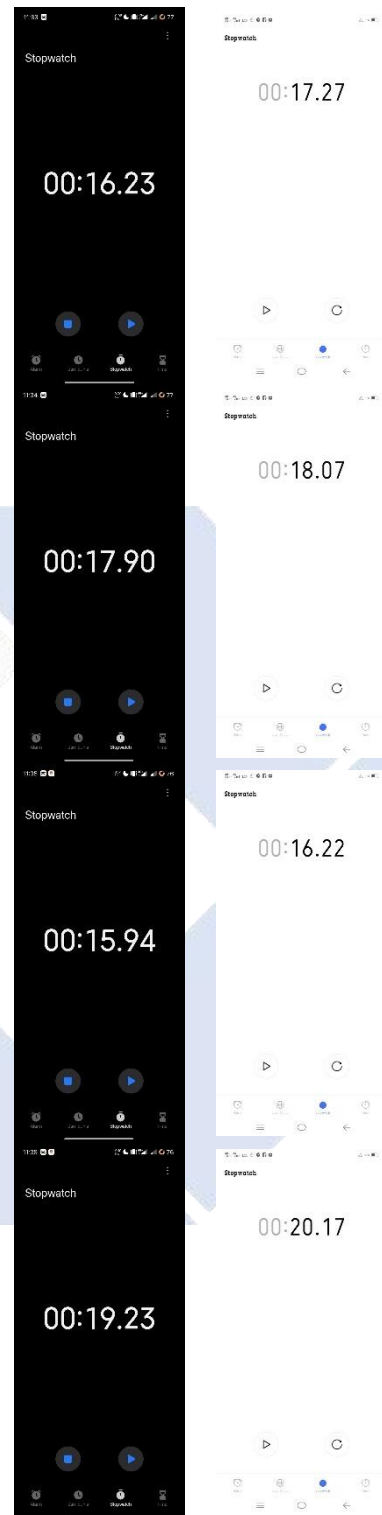
11 Gerakan
10



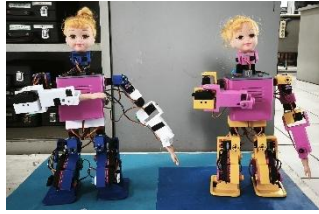
12 Gerakan
11



13 Gerakan
12



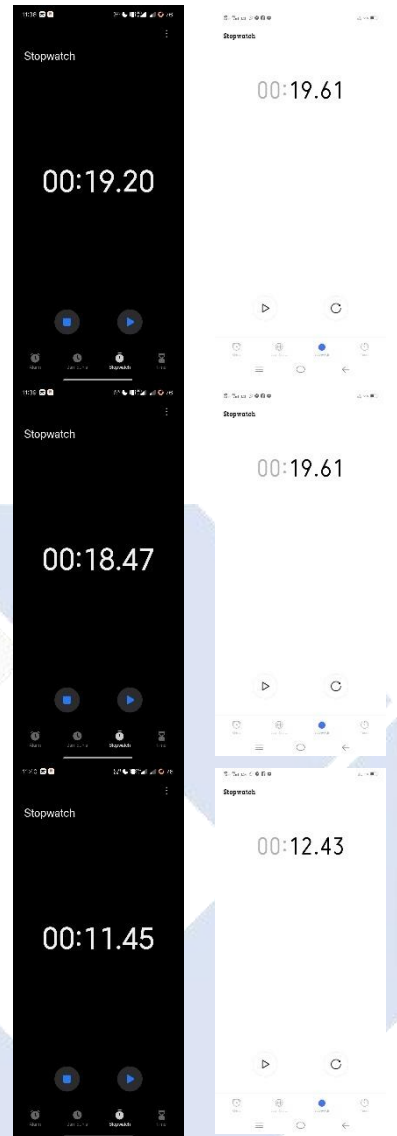
14 Gerakan
13



15 Gerakan
14



16 Tarian
penutup



BAB V PENUTUP

5.1 Kesimpulan

Berdasarkan data dan hasil pengujian yang telah dilakukan, dapat disimpulkan bahwa :

1. Robot yang dikembangkan dalam proyek akhir ini menggunakan 25 buah motor servo, hasil pengujian menunjukkan bahwa setiap servo dapat bergerak dengan baik dan responsif sesuai input yang diberikan dengan persentase *error*nya 1,22%.
2. Permasalahan dalam pemrograman robot secara konvensional melalui Arduino IDE telah diatasi dengan pengembangan *software* GUI yang memanfaatkan *Java Swing*. Fitur visual seperti *slider* untuk setiap servo, tabel gerakan, dan fungsi *play/stop*, proses penyusunan gerakan menjadi jauh lebih interaktif, cepat, dan minim kesalahan. Hal ini terbukti dari pengujian langsung di mana pengguna dapat menyusun urutan gerak dan menyimpan data ke tabel.
3. Sistem komunikasi antar mikrokontroler berhasil menghubungkan ESP1_Master, ESP1_Slave, ESP2_Master dan ESP2_Slave yang dapat adanya musik. Data dari ESP1_Master menunjukkan bahwa musik dapat terdeteksi secara *real-time* berdasarkan sinyal ADC, dan ESP1_Slave berhasil menerjemahkannya menjadi sinyal '1' atau '0' ke ESP2_Slave. Data dari ESP2_Slave membuktikan bahwa perintah dijalankan sesuai waktu dan perintah yang diterima, menunjukkan sistem komunikasi HC-12 dan serial antar ESP32 hampir mendekati keselarasan dengan persentase *error* 3,2%.

5.2 Saran

1. Untuk meningkatkan stabilitas dan efektivitas gerakan, sebaiknya desain mekanik robot dikembangkan lebih lanjut menggunakan *software* CAD yang lebih kompleks.
2. *Software* dapat ditingkatkan dengan fitur visualisasi 3D agar pengguna dapat merancang koreografi lebih intuitif dan presisi.
3. Penambahan sensor keseimbangan seperti IMU (*Inertial Measurement Unit*) agar robot dapat melakukan koreksi postur otomatis selama bergerak.
4. Untuk meningkatkan respons terhadap variasi musik, sistem dapat dikembangkan dengan modul analisis frekuensi atau tempo real-time agar gerakan robot lebih adaptif terhadap perubahan irama lagu.

DAFTAR PUSTAKA

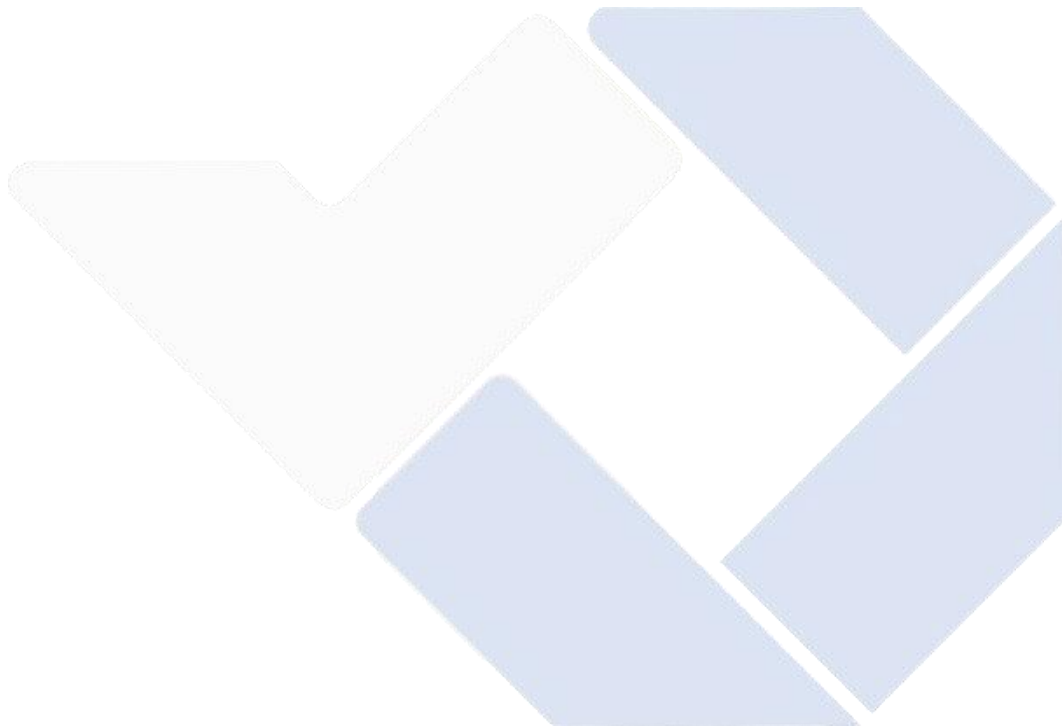
- [1] S. Soim, B. Joni, “PERANCANGAN ROBOT HUMANOID BERBASIS MIKROKONTROLER,” no. 11, November, pp. 1–6, 2015.
- [2] C. Habiba dan A. Dharmawan, “Sistem Kendali Jalan Robot Humanoid Pada Bidang Tidak Rata Menggunakan Lqr,” *IJEIS (Indonesian J. Electron. Instrum. Syst.*, vol. 13, no. 2, p. 135, 2023, doi: 10.22146/ijeis.81570.
- [3] F. Hermawanto, I. Karim, “Rancang Bangun Robot Penari Tidi,” *Energy-Jurnal Ilm. Ilmu-Ilmu Tek.*, vol. 8, no. 2, 2018, [Online]. Available: <https://ejournal.upm.ac.id/index.php/energy/article/view/292%0Ahttps://ejournal.upm.ac.id/index.php/energy/article/download/292/318>
- [4] B. Kusumoputro, M. Hery Purnomo, H. Santoso Budi Rochardjo, G. Prabowo, D. Purwanto, E. Pitowarno, E. Mozef, Indrawanto, K. Mutijarsa, A. Muis, “Pedoman Kontes Robot Indonesia (Kri) Pendidikan Tinggi Tahun 2024,” *Kementerian Pendidikan, Kebudayaan, Riset, dan Teknol.*, pp. 1–164, 2024.
- [5] I. Dwisaputra, O. Ocsirendi, “Teknik Pengenalan Suara Musik Pada Robot Seni Tari,” *Manutech J. Teknol. Manufaktur*, vol. 10, no. 02, pp. 35–39, 2019, doi: 10.33504/manutech.v10i02.66.
- [6] A. Yulita, “OBJEK, ANALISIS PENDETEKSI WARNA ROBOT, MENGGUNAKAN SENSOR KAMERA PADA HUMANOID,” Program Studi Sarjana Terapan Teknik Elektro, Jurusan Teknik Elektro, Politeknik Negeri Sriwijaya, Palembang, 2019.
- [7] I. F. Rozy, V. F. Rizqi, “Pengontrolan Kepala Robot Dengan Gerakan Manusia,” pp. 1–67, 2019.
- [8] “Kontes Robot Indonesia 2023,” Panitia Pusat Kontes Robot Indonesia, Balai Pengembangan Talenta Indonesia, Pusat Prestasi Nasional, Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi Republik Indonesia. Accessed: Apr. 08, 2025. [Online]. Available: <https://kontesrobotindonesia.id/kri-2023.html>

- [9] N. Azizah, "RANCANG BANGUN ROBOT HUMANOID PENARI UNTUK BERJALAN PADA LANTAI BERUNDAK," Program Studi Teknik Elektro, Fakultas Teknologi Industri, Universitas Islam Sultan Agung Semarang, 2023.
- [10] D. Andini, S. Ramadhan, "PERANCANGAN ROBOT PEMINDAH BARANG LINE FOLLOWER DENGAN PATH PLANNING BERBASIS APLIKASI MOBILE," POLITEKNIK MANUFAKTUR NEGERI BANGKA BELITUNG, 2024.
- [11] Y. H. Chang, F. C. Wu, H. W. Lin, "Design and Implementation of ESP32-Based Edge Computing for Object Detection," *Sensors*, vol. 25, no. 6, 2025, doi: 10.3390/s25061656.
- [12] M. Y. MAJID, "RANCANG BANGUN ALAT MONITORING PADA TANAMAN HIAS MENGGUNAKAN ESP32 BERBASIS APLIKASI ANDROID," UNIVERSITAS HASANUDDIN MAKASSAR, 2022.
- [13] D. A. Ayubi, D. A. Prasetya, I. Mujahidin, "Pendeteksi Wajah Secara Real Time pada 2 Degree of Freedom (DOF) Kepala Robot Menggunakan Deep Integral Image Cascade," *Cyclotron*, vol. 3, no. 1, pp. 22–27, 2020, doi: 10.30651/cl.v3i1.4306.
- [14] K. U. Ariawan, G. S. Santyadiputra, I. W. Sutaya, "Design of Hexapod Robot Movement Based on Arduino Mega 2560," *J. Phys. Conf. Ser.*, vol. 1165, no. 1, 2019, doi: 10.1088/1742-6596/1165/1/012011.
- [15] TowerPro, "MG996R Metal Gear Servo Motor Datasheet."
- [16] A. M. S. Nugroho, R. Hidayat, A. Stefanie, "Implementation of Stepper 28Byj-48 and Servo Mg996R As a Roasting Arm Robot in an Arduino Uno-Based Automatic Satay Grill Tool," *JEEMECS (Journal Electr. Eng. Mechatron. Comput. Sci.)*, vol. 5, no. 1, pp. 47–54, 2022, doi: 10.26905/jeemecs.v5i1.5166.
- [17] P. Chotikunnan, Y. Pititheeraphab, T. Angsuwatanakul, J. Prinyakupt, T. Puttasakul, R. Chotikunnan, N. Thongpance, "Enhancing MG996R Servo Motor Performance Using PSO-Tuned PID and Feedforward Control," *Int. J. Robot. Control Syst.*, vol. 5, no. 2, pp. 1120–1138, Apr. 2025, doi:

10.31763/ijrcs.v5i2.1854.

- [18] H. M. Ali, Y. Hashim, G. A. Al-Sakkal, "Design and implementation of Arduino based robotic arm," *Int. J. Electr. Comput. Eng.*, vol. 12, no. 2, pp. 1411–1418, 2022, doi: 10.11591/ijece.v12i2.pp1411-1418.
- [19] T. Zafarudin, E. N. Andriani, M. F. Firdaus, "Pengembangan Prototype Robot Quadruped Dengan Monitoring Control Android," 2019.
- [20] TowerPro, "MG90S servo Datasheet."
- [21] N. FAWAID, "GELANG ANTI BEGAL SEPEDA MOTOR MENGGUNAKAN HC-12 BERBASIS ARDUINO," UNIVERSITAS DIPONEGORO SEMARANG, 2021.
- [22] Wolles Elektronikkiste, "HC-12 Wireless Serial Port Communication Module," 2012.
- [23] UGREEN, "UGREEN Bluetooth 5.1 Transmitter Receiver," UGREEN.
- [24] Super Store, "UGREEN 2-in-1 Wireless Bluetooth 5.1 USB 2.0 Transmitter & Receiver Adapter with 3.5mm Mini Audio Jack Cable, 2400MHz Range, 10M Operating Distance for Desktop, TV, Headphones, Speaker, Amplifier, Speakers, Laptop, Smartphone," Super Store.
- [25] Manual +, "CM523 USB-A ke Jack Adaptor Audio 3.5mm."
- [26] S. G. Ludony, M. Mulyadi, K. Indriati, "Rancang Bangun Purwarupa Lengan Robot Berbantuan Raspberry Pi," *J. Elektro*, vol. 13, no. 2, pp. 115–124, 2021, doi: 10.25170/jurnalelektro.v13i2.1979.
- [27] ALLDATASHEETCOM, "PCA9685 NXP | Alldatasheet," 2009.
- [28] F. R. Utami, M. A. Riyadi, Y. Christyono, "PERANCANGAN CATU DAYA ARUS SEARAH KELUARAN GANDA SEBAGAI PENGGERAK ROBOT LENGAN ARTIKULASI," *Transient*, vol. 9, no. 3, 2020.
- [29] R. F. ASHARI, "PERANCANGAN SISTEM KOMUNIKASI PADA ROBOT HUMANOID PENARI DENGAN AUDIO BLUETOOTH DONGLE DAN BLUETOOTH HC 05 MENGGUNAKAN ARDUINO PRO MICRO," UNIVERSITAS ISLAM SULTAN AGUNG SEMARANG, 2022.
- [30] E. Sulistyono, "Sistem Komunikasi Robot Humanoid Dalam Aplikasi Robot

- Penari,” *Semin. Nas. Sains dan Teknol. 2015 Fak. Tek. Univ. Muhammadiyah Jakarta*, no. November, pp. 1–5, 2015.
- [31] K. Z. Rifal, A. Kusmantoro, I. Harjanto, “Studi Analisa Penggunaan Baterai Li-Ion Dan Li-Po Pada Sistem Iot Hidroponik,” *JETI (Jurnal Elektro dan Teknol. Informasi)*, vol. 1, no. 1, pp. 1–5, 2024, doi: 10.26877/jeti.v1i1.580.
- [32] D. Harjono, “Sistem Monitoring Baterai Lithium Polymer (Lipo) Secara Nirkabel Pada Mobil Listrik PonECar,” *Elit. Electrotech. Inf. Technol.*, vol. 4, no. 2, pp. 1–10, 2023.



LAMPIRAN

Lampiran 1 : Daftar Riwayat Hidup

DAFTAR RIWAYAT HIDUP

1. Data Pribadi

Nama Lengkap : Mardinata Indra Kristianto
Tempat/Tanggal Lahir : Bumi Harjo, 1 Maret 2004
Alamat Rumah : Jl. Sinar Menumbing, Gang
PLTD, Pal 2, RT 003 RW 000,
Kota Muntok, Kab. Bangka
Barat, Prov. Kep. Bangka
Belitung
No. Telp/HP : 085839585774
Email : mardinataindra444@gmail.com
Jenis Kelamin : Laki-laki
Agama : Kristen



2. Riwayat Pendidikan

SD Santa Maria Muntok (2010-2016)
SMP Santa Maria Muntok (2016-2019)
SMK Negeri 1 Muntok (2019-2022)

Sungailiat, 3 Juli 2025

Mardinata Indra Kristianto

DAFTAR RIWAYAT HIDUP

1. Data Pribadi

Nama Lengkap : Danny Pranata
Tempat/Tanggal Lahir : Air Duren, 5 Agustus 2004
Alamat Rumah : Jl. Dr. Setomo, Gang Dumai,
No. 189, Air Duren, RT 002,
RW 000, Kec. Pemali, Kab.
Bangka, Prov. Kep. Bangka
Belitung
No. Telp/HP : 081539394151
Email : dannypranata46@gmail.com
Jenis Kelamin : Laki-laki
Agama : Khonghucu



2. Riwayat Pendidikan

SD Negeri 10 Pemali (2010-2016)

SMP Negeri 2 Pemali (2016-2019)

SMA Negeri 1 Pemali (2019-2022)

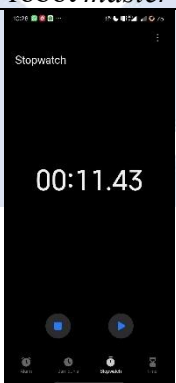
Sungailiat, 3 Juli 2025

Danny Pranata

Lampiran 2 : Tabel Pengujian Sinkronisasi Robot

Pengujian 2

No	Gerakan robot	Waktu gerakan robot <i>master</i> (s)	Waktu gerakan robot <i>slave</i> (s)	%Error
1	Diam	11,4	12,2	7,02%
2	Sembah pembuka	16,2	16,3	0,62%
3	Gerakan 1	19,2	20,7	7,81%
4	Gerakan 2	19,2	20,1	4,69%
5	Gerakan 3	18,4	19,1	3,80%
6	Gerakan 4	14,4	14,6	1,39%
7	Gerakan 5	18,5	18,7	1,08%
8	Gerakan 6	19,3	19,5	1,04%
9	Gerakan 7	19,4	19,6	1,03%
10	Gerakan 8	16,2	17,2	6,17%
11	Gerakan 9	17,9	18	0,56%
12	Gerakan 10	15,9	16,2	1,89%
13	Gerakan 11	19,2	20,2	5,21%
14	Gerakan 12	19,3	19,6	1,55%
15	Gerakan 13	18,4	19,6	6,52%
16	Gerakan 14	11,5	12	4,35%
17	Tarian penutup	11,4	12,2	7,02%
Rata-rata				3,42%

No	Gerakan robot	Dokumentasi gerakan	Waktu gerakan robot <i>master</i>	Waktu gerakan robot <i>slave</i>
1	Sembah pembuka			

2 Gerakan
1



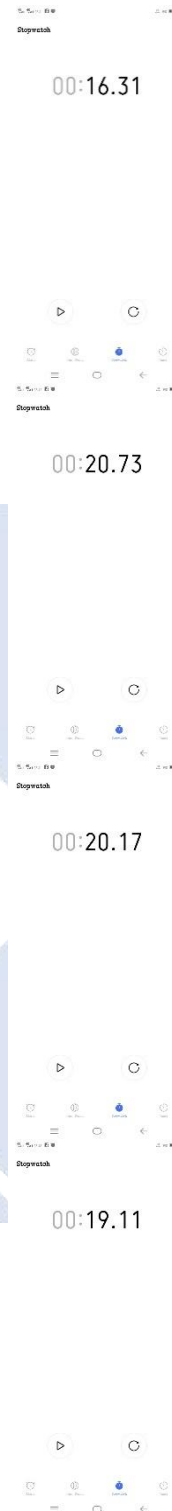
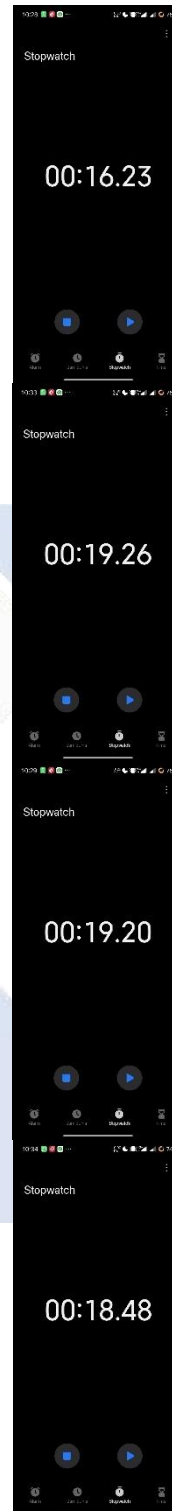
3 Gerakan
2



4 Gerakan
3



5 Gerakan
4



6 Gerakan
5



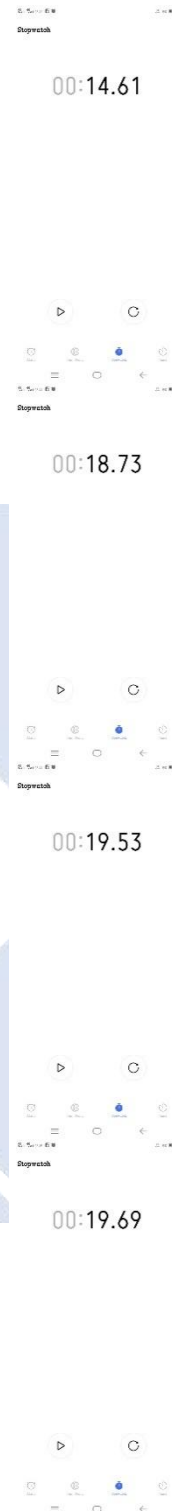
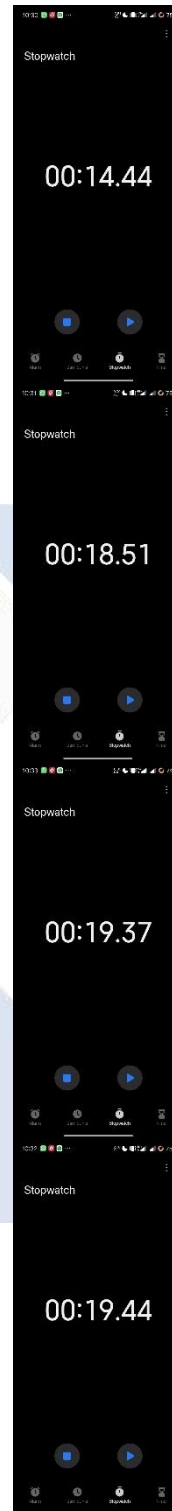
7 Gerakan
6



8 Gerakan
7



9 Gerakan
8



10 Gerakan 9



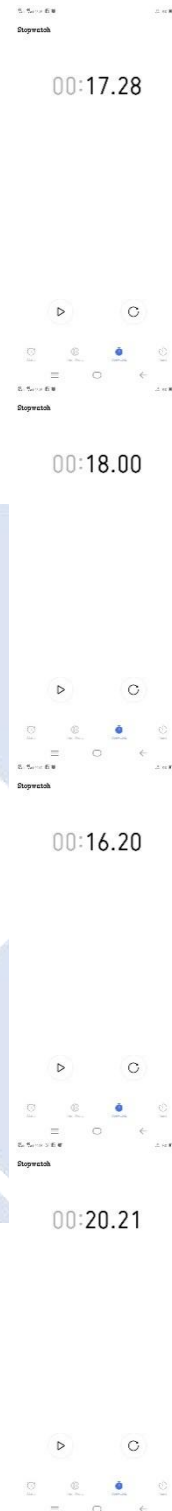
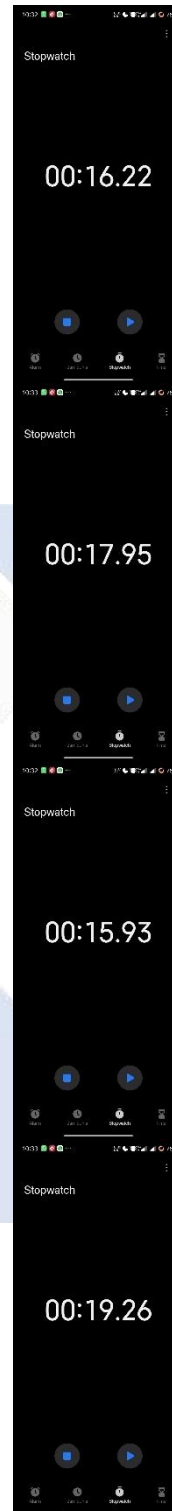
11 Gerakan 10



12 Gerakan 11



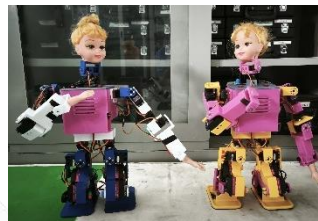
13 Gerakan 12



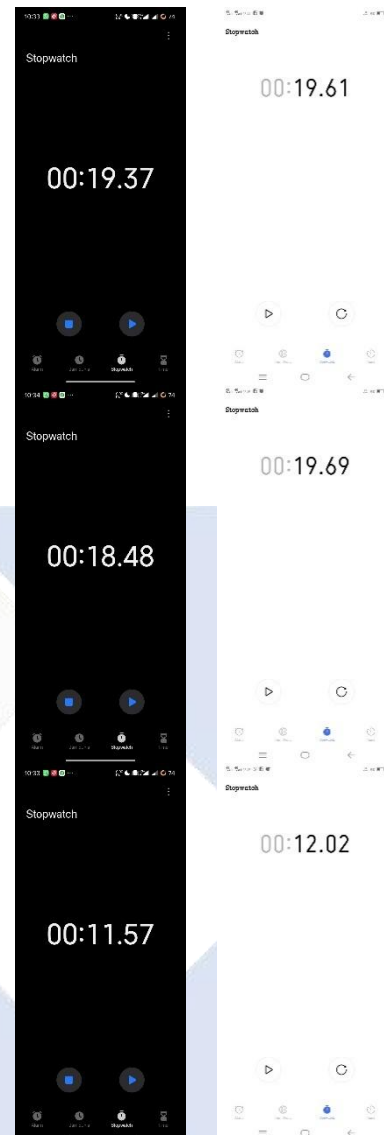
14 Gerakan 13



15 Gerakan 14



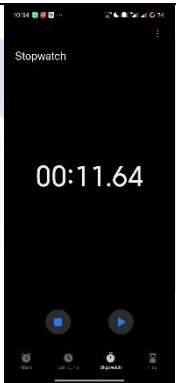
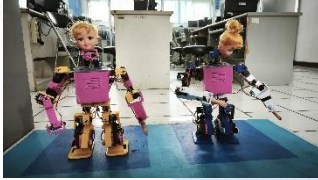
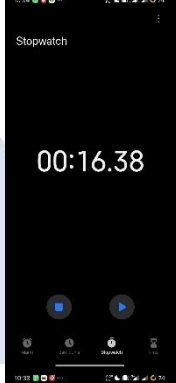
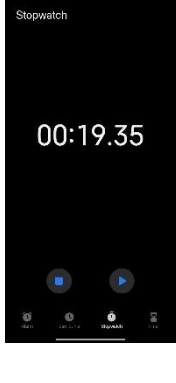
16 Tarian penutup



Pengujian 3

No	Gerakan robot	Waktu gerakan robot <i>master</i> (s)	Waktu gerakan robot <i>slave</i> (s)	%Error
1	Diam	11,6	12,1	4,31%
2	Sembah pembuka	16,3	16,9	3,68%
3	Gerakan 1	19,3	20,4	5,70%
4	Gerakan 2	19,5	20,3	4,10%
5	Gerakan 3	18,7	19,3	3,21%
6	Gerakan 4	14,6	14,8	1,37%
7	Gerakan 5	18,6	18,9	1,61%
8	Gerakan 6	19,3	19,9	3,11%
9	Gerakan 7	19,5	19,7	1,03%
10	Gerakan 8	16,3	17	4,29%

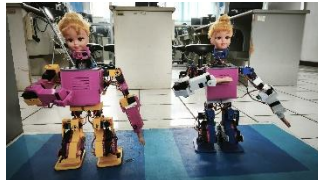
11	Gerakan 9	18	18,7	3,89%
12	Gerakan 10	16	16,5	3,13%
13	Gerakan 11	19,3	20,3	5,18%
14	Gerakan 12	19,4	19,4	0,00%
15	Gerakan 13	18,5	19,2	3,78%
16	Gerakan 14	11,6	12,1	4,31%
17	Tarian penutup	11,6	12,1	4,31%
Rata-rata				3,29%

No	Gerakan robot	Dokumentasi gerakan	Waktu gerakan robot <i>master</i>	Waktu gerakan robot <i>slave</i>
1	Sembah pembuka			
2	Gerakan 1			
3	Gerakan 2			

4 Gerakan
3



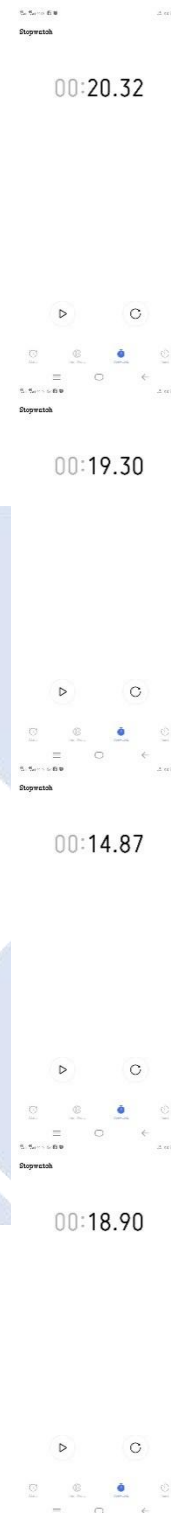
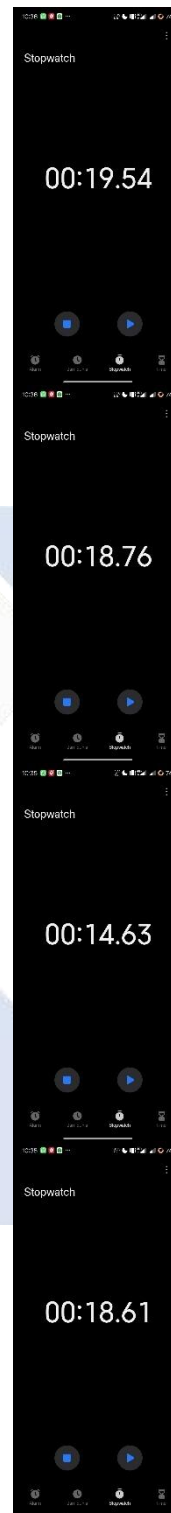
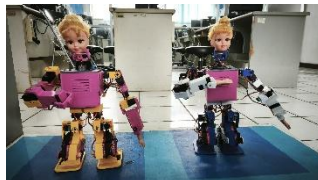
5 Gerakan
4



6 Gerakan
5



7 Gerakan
6



8 Gerakan
7



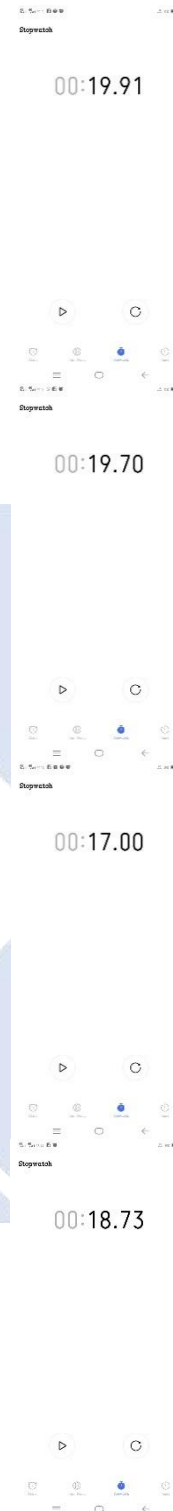
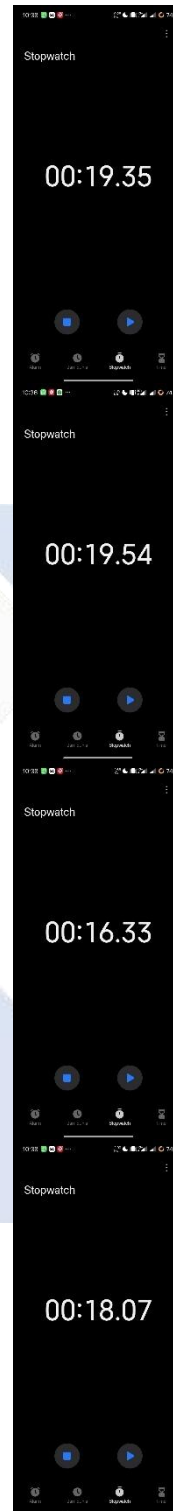
9 Gerakan
8



10 Gerakan
9



11 Gerakan
10



12 Gerakan 11



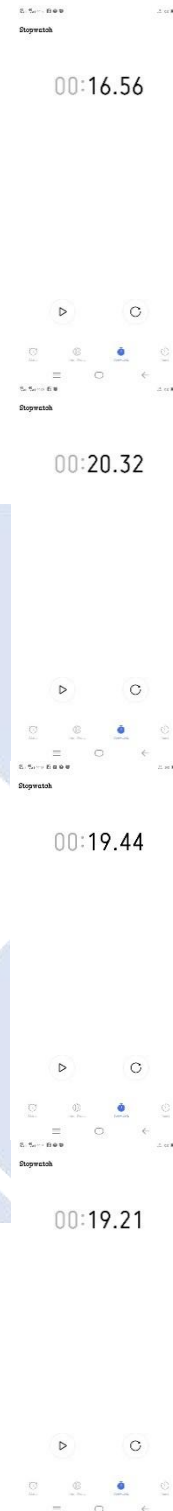
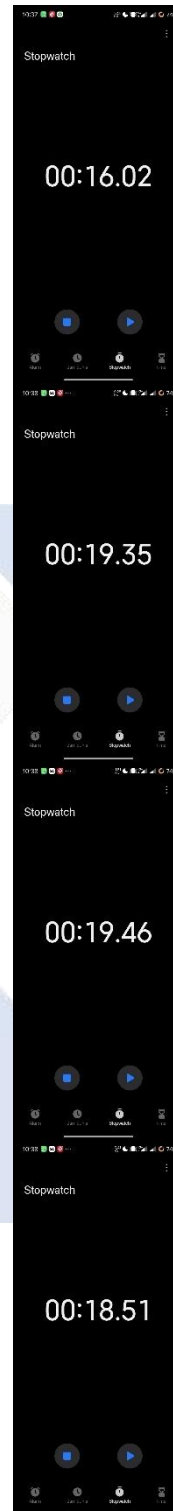
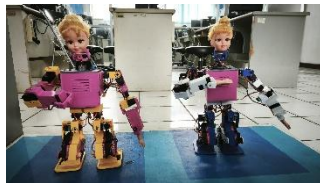
13 Gerakan 12



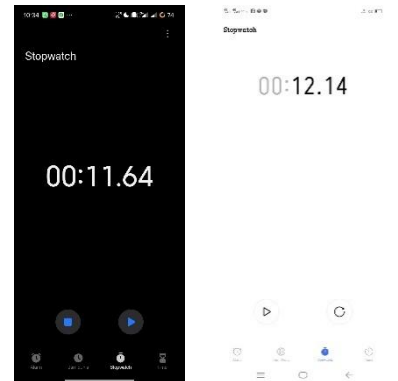
14 Gerakan 13



15 Gerakan 14

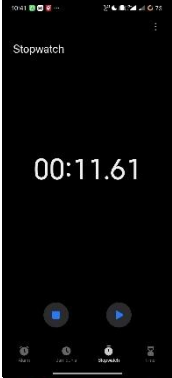
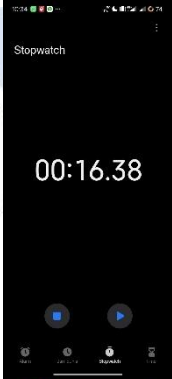
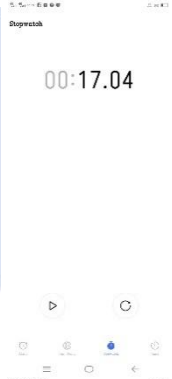
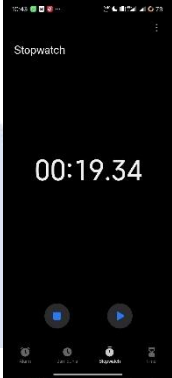
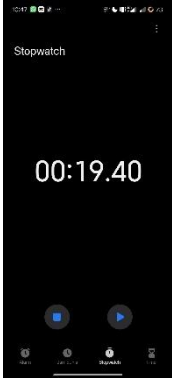


16 Tarian penutup



Pengujian 4

No	Gerakan robot	Waktu gerakan robot <i>master</i> (s)	Waktu gerakan robot <i>slave</i> (s)	%Error
1	Diam	11,6	11,8	1,72%
2	Sembah pembuka	16,3	17	4,29%
3	Gerakan 1	19,3	20,3	5,18%
4	Gerakan 2	19,4	20	3,09%
5	Gerakan 3	18,4	19,2	4,35%
6	Gerakan 4	14,5	14,7	1,38%
7	Gerakan 5	18,5	18,9	2,16%
8	Gerakan 6	19,3	19,6	1,55%
9	Gerakan 7	19,5	19,5	0,00%
10	Gerakan 8	16,4	17	3,66%
11	Gerakan 9	18	18,3	1,67%
12	Gerakan 10	15,9	16,2	1,89%
13	Gerakan 11	19,2	20	4,17%
14	Gerakan 12	19,4	19,4	0,00%
15	Gerakan 13	18,6	19	2,15%
16	Gerakan 14	11,6	12,1	4,31%
17	Tarian penutup	11,6	11,8	1,72%
Rata-rata				2,60%

No	Gerakan robot	Dokumentasi gerakan	Waktu gerakan robot <i>master</i>	Waktu gerakan robot <i>slave</i>
1	Sembah pembuka			
2	Gerakan 1			
3	Gerakan 2			
4	Gerakan 3			

5 Gerakan
4



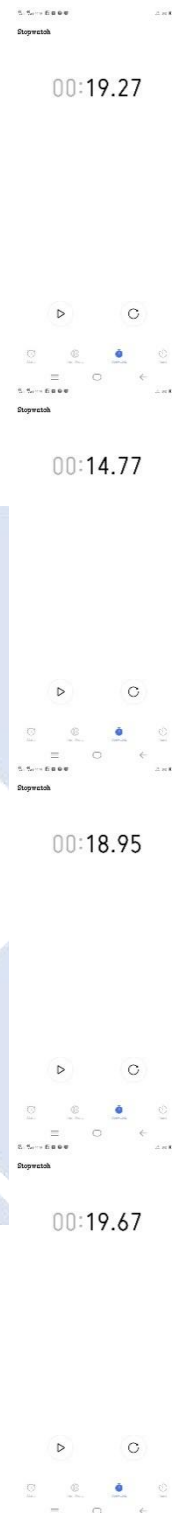
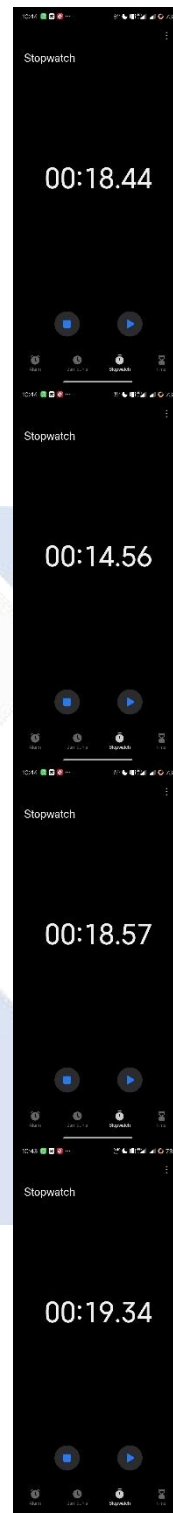
6 Gerakan
5



7 Gerakan
6



8 Gerakan
7



9 Gerakan
8



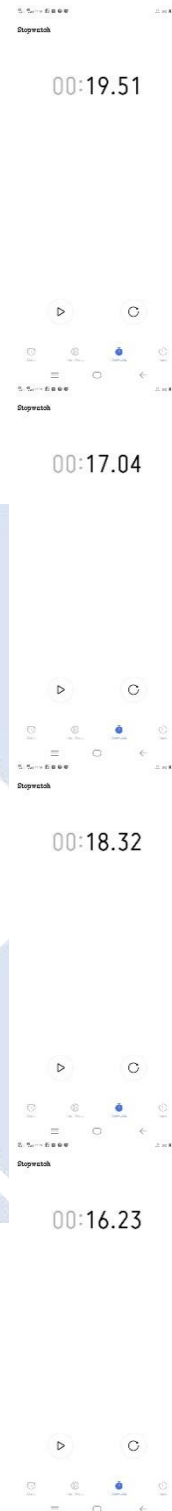
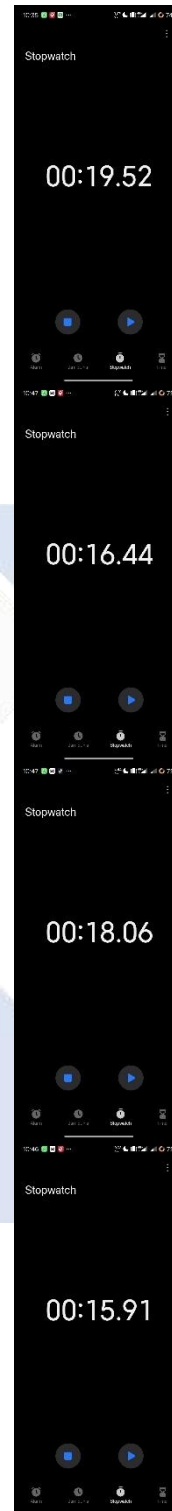
10 Gerakan
9



11 Gerakan
10



12 Gerakan
11



13 Gerakan
12



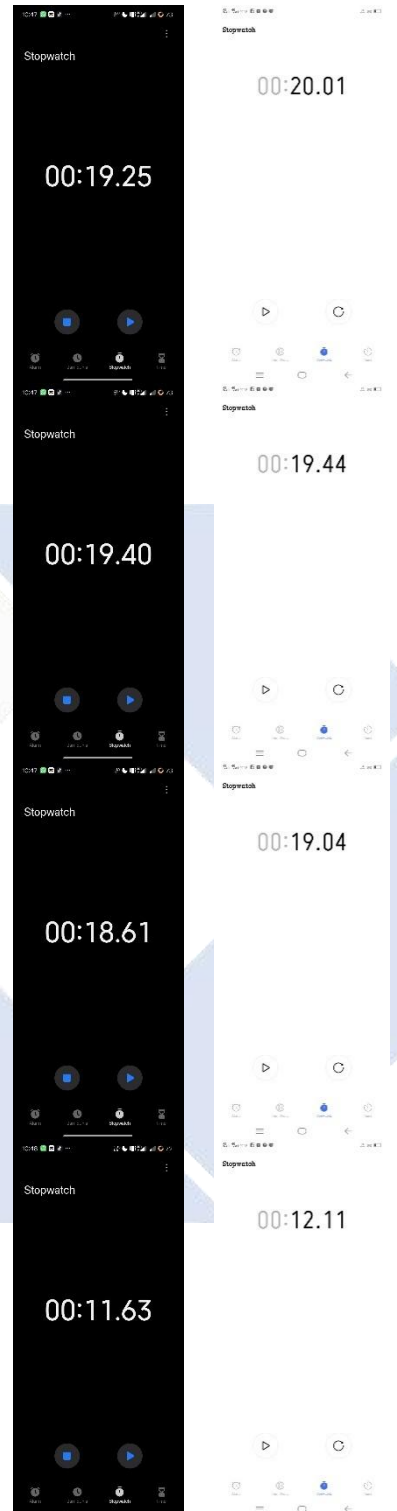
14 Gerakan
13



15 Gerakan
14

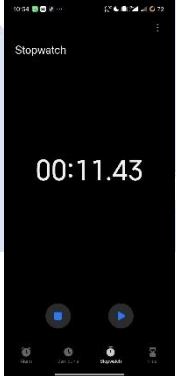


16 Tarian
penutup



Pengujian 5

No	Gerakan robot	Waktu gerakan robot <i>master</i> (s)	Waktu gerakan robot <i>slave</i> (s)	%Error
1	Diam	11,4	12	5,26%
2	Sembah pembuka	16,1	16,3	1,24%
3	Gerakan 1	19,1	20,1	5,24%
4	Gerakan 2	19,2	19,8	3,13%
5	Gerakan 3	18,3	18,9	3,28%
6	Gerakan 4	14,2	14,3	0,70%
7	Gerakan 5	18,3	18,6	1,64%
8	Gerakan 6	19,1	19,3	1,05%
9	Gerakan 7	19,3	19,5	1,04%
10	Gerakan 8	16,1	16,8	4,35%
11	Gerakan 9	17,8	18	1,12%
12	Gerakan 10	15,8	16	1,27%
13	Gerakan 11	19,1	19,6	2,62%
14	Gerakan 12	19,2	19,8	3,13%
15	Gerakan 13	18,4	18,9	2,72%
16	Gerakan 14	11,3	11,9	5,31%
17	Tarian penutup	11,4	12	5,26%
Rata-rata				2,69%

No	Gerakan robot	Dokumentasi gerakan	Waktu gerakan robot <i>master</i>	Waktu gerakan robot <i>slave</i>
1	Sembah pembuka			

2 Gerakan
1



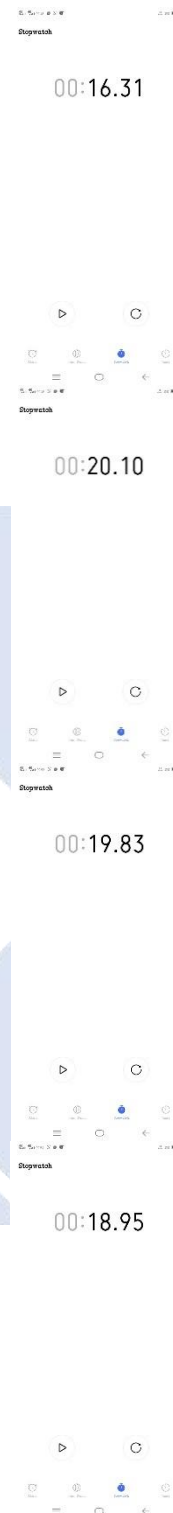
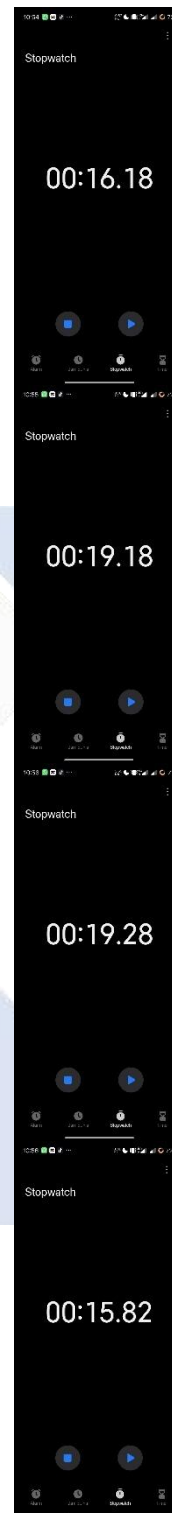
3 Gerakan
2



4 Gerakan
3



5 Gerakan
4



6 Gerakan
5



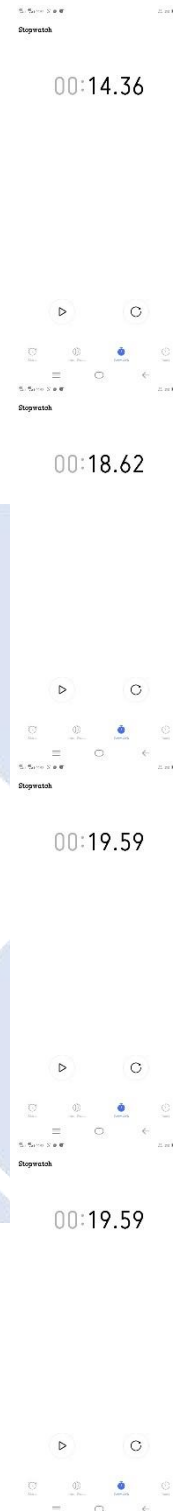
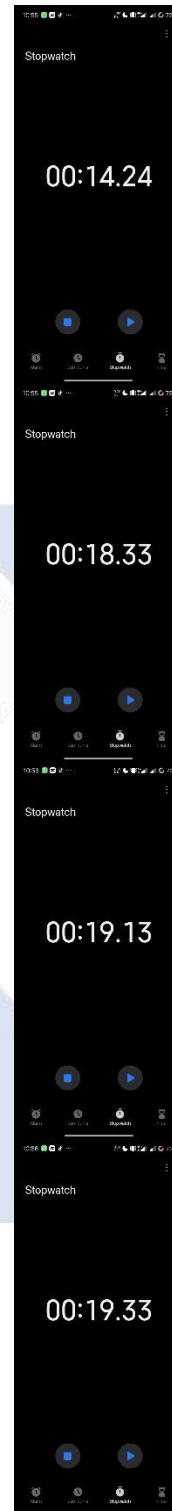
7 Gerakan
6



8 Gerakan
7



9 Gerakan
8



10 Gerakan 9



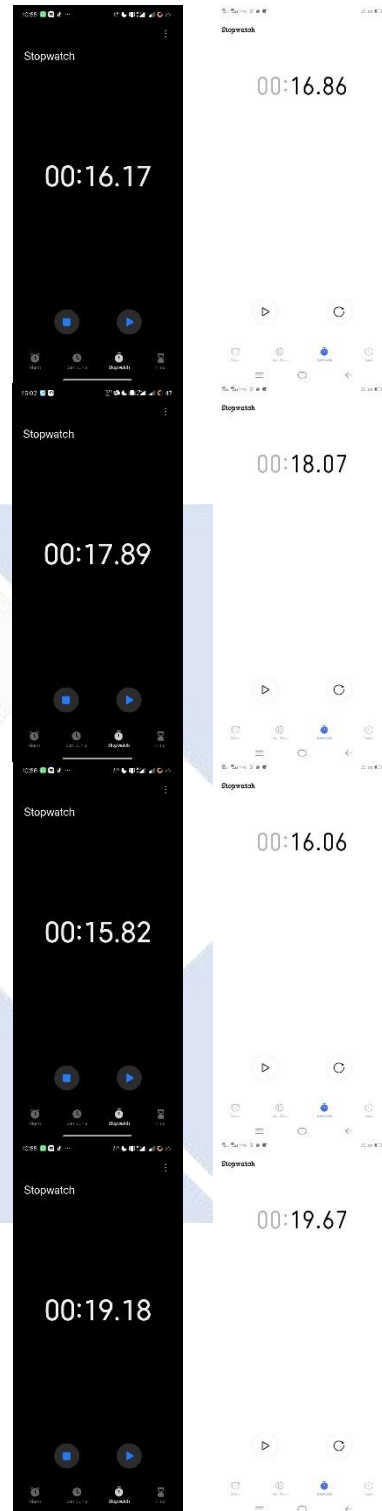
11 Gerakan 10



12 Gerakan 11



13 Gerakan 12



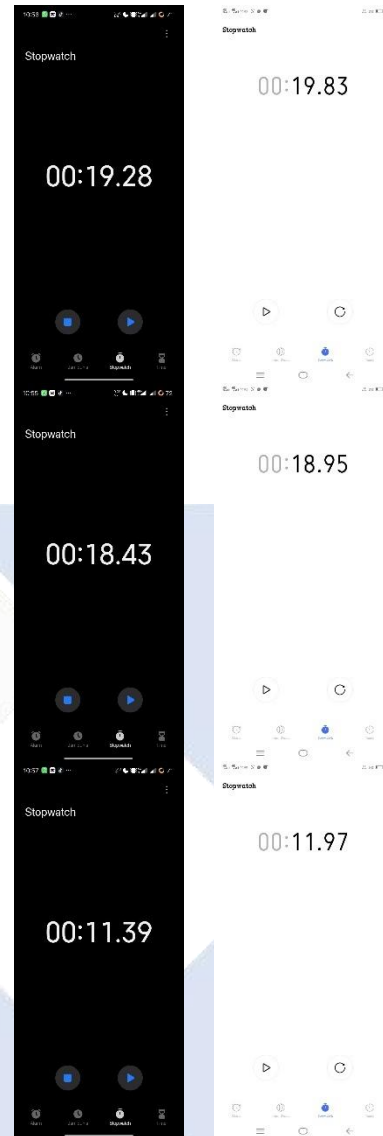
14 Gerakan
13



15 Gerakan
14



16 Tarian
penutup



ORIGINALITY REPORT

14%	14%	4%	4%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	repository.polman-babel.ac.id Internet Source	4%
2	repository.its.ac.id Internet Source	2%
3	repository.unhas.ac.id Internet Source	1%
4	Submitted to Universidad Carlos III de Madrid Student Paper	1%
5	jurnal.umj.ac.id Internet Source	<1%
6	docplayer.info Internet Source	<1%
7	widuri.raharja.info Internet Source	<1%
8	journal.ugm.ac.id Internet Source	<1%
9	taufikoto2.blogspot.com Internet Source	<1%
10	d4trjt.poliupg.ac.id Internet Source	<1%
11	media.neliti.com Internet Source	<1%
12	doaj.org	

<1 %

13 Submitted to University of Lincoln
Student Paper

<1 %

14 journal.um-surabaya.ac.id
Internet Source

<1 %

15 eprints.uny.ac.id
Internet Source

<1 %

16 Submitted to VinUniversity
Student Paper

<1 %

17 Submitted to Unizin, LLC
Student Paper

<1 %

18 intanwblog.wordpress.com
Internet Source

<1 %

19 vdocuments.mx
Internet Source

<1 %

20 repository.umsu.ac.id
Internet Source

<1 %

21 repositori.kemdikbud.go.id
Internet Source

<1 %

22 jurnal.sar.ac.id
Internet Source

<1 %

23 katshuhiko.blogspot.com
Internet Source

<1 %

24 repo.akmet.ac.id
Internet Source

<1 %

25 Dhea Setiawati, Nindia Justita Maryana,
Muhamad Rizal, Muhammad Ilham Syafi', Dwi
Fakhrianto. "ANIMATRONIK HEWAN DENGAN

<1 %

GERAKAN TUBUH", Jurnal Informatika dan Teknik Elektro Terapan, 2025

Publication

26	tonggomelisarobotiktgd.blogspot.com Internet Source	<1 %
27	www.lxrobotics.com Internet Source	<1 %
28	journal.poltekad.ac.id Internet Source	<1 %
29	kontesrobotindonesia.id Internet Source	<1 %
30	seputarrobot.blogspot.com Internet Source	<1 %
31	dit-mawa.upi.edu Internet Source	<1 %
32	ejnteti.jteti.ugm.ac.id Internet Source	<1 %
33	eprints.ums.ac.id Internet Source	<1 %
34	idoc.tips Internet Source	<1 %
35	repository.usu.ac.id Internet Source	<1 %
36	123dok.com Internet Source	<1 %
37	Amaliah Chintami Darti Akhsa, Sajiah Sajiah. "PERANCANGAN APLIKASI CHAT MULTI PLATFORM BERBASIS JAVA GUI (GRAPHIC USER INTERFACE) PADA BADAN ARSIP DAN PERPUSTAKAAN DAERAH (BAPD) PROVINSI	<1 %

SULAWESI SELATAN", ILTEK : Jurnal Teknologi, 2016

Publication

38	doku.pub Internet Source	<1 %
39	ejournal.akprind.ac.id Internet Source	<1 %
40	ejurnal.unima.ac.id Internet Source	<1 %
41	elibrary.unikom.ac.id Internet Source	<1 %
42	elkolind.polinema.ac.id Internet Source	<1 %
43	journal2.uad.ac.id Internet Source	<1 %
44	juniorprogramming.wordpress.com Internet Source	<1 %
45	myusro.id Internet Source	<1 %
46	repository.bsi.ac.id Internet Source	<1 %
47	shirlitalitha.blogspot.com Internet Source	<1 %
48	tr.scribd.com Internet Source	<1 %
49	upcommons.upc.edu Internet Source	<1 %
50	www.notebookcheck.net Internet Source	<1 %

51

www.scribd.com

Internet Source

<1%

52

repository.ub.ac.id

Internet Source

<1%

Exclude quotes Off

Exclude matches Off

Exclude bibliography On

SURAT PERNYATAAN

Saya/Kami yang bertandatangan dibawah ini telah menyelesaikan Proyek Akhir yang berjudul:

Rancang Bangun Robot Humanoid Seni Tari dan Sistem Pengawasan Terintegrasi

Oleh :

1. Mardinata Indra K. /NPM 1052216
2. Danny Pranata /NPM 1052235
3. /NPM

Dengan ini menyatakan bahwa isi laporan akhir proyek akhir sama dengan *hardcopy*.
Demikian surat pernyataan ini dibuat dengan sebenar-benarnya.

Sungailiat, Agustus 2025

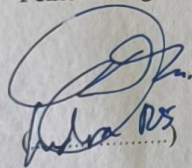
1. Mardinata Indra ()

2. Danny Pranata ()

3. (.....)

Mengetahui,

Pembimbing 1,



Pembimbing 2,

(.....)