

**RANCANG BANGUN ROBOT *HUMANOID* PENARI UNTUK BERJALAN
PADA LANTAI BERUNDAK**

LAPORAN TUGAS AKHIR

Laporan ini disusun untuk memenuhi salah satu syarat
memperoleh gelar S1 pada Program Studi Teknik Elektro
Fakultas Teknologi Insutri
Universitas Islam Sultan Agung Semarang



**DISUSUN OLEH:
NIKMATUL AZIZAH
NIM 30601900038**

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

2023

DESIGN OF A HUMANOID DANCING ROBOT TO WALK ON RELEASES

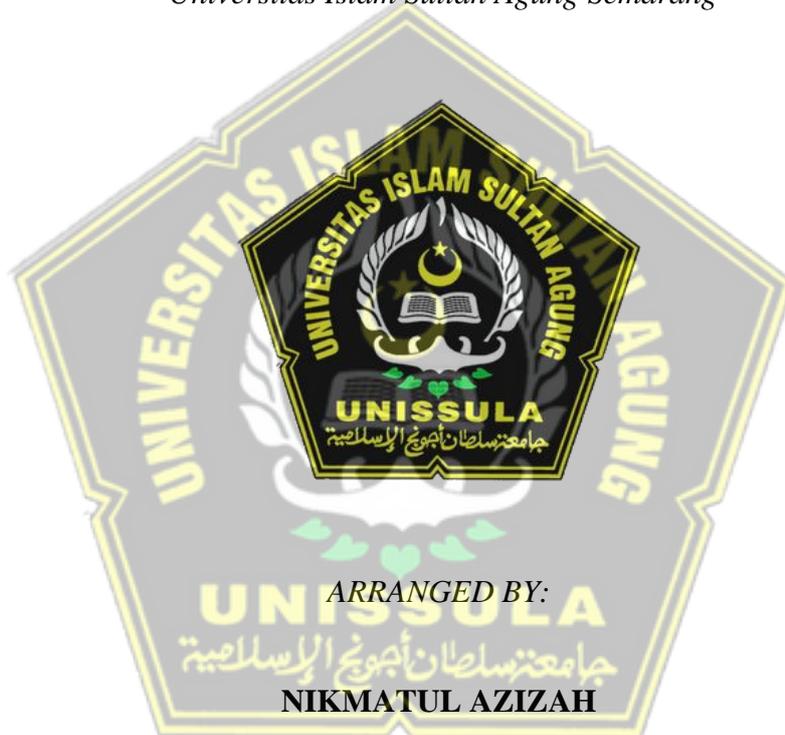
FINAL PROJECT

Proposed to complete the requirement to obtain a bachelor's degree (S1) at

Departement of Industrial Engineering

Faculty of Industrial Technology

Universitas Islam Sultan Agung Semarang



ARRANGED BY:

NIKMATUL AZIZAH

NIM 30601900038

DEPARTMENT OF ELECTRICAL ENGINEERING

FACULTY OF INDUSTRIAL TECHNOLOGY

UNIVERSITAS ISLAM SULTAN AGUNG

SEMARANG

2023

LEMBAR PENGESAHAN PEMBIMBING

Laporan tugas akhir dengan judul **“RANCANG BANGUN ROBOT HUMANOID PENARI UNTUK BERJALAN PADA LANTAI BERUNDAK”** ini disusun oleh

Nama : NIKMATUL AZIZAH

NIM : 30601900038

Program Studi : Teknik Elektro

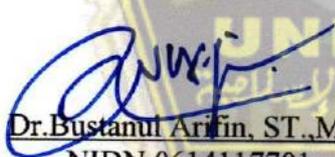
Telah disahkan dan disetujui oleh dosen pembimbing pada :

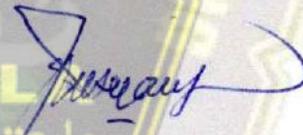
Hari : Rabu

Tanggal : 16 Agustus 2023

Pembimbing I

Pembimbing II


Dr. Bustanul Arifin, ST., MT.
NIDN.0614117701


Dr. Eka Nuryanto Budisusila, ST., MT
NIDN. 0619107301

Mengetahui
Ketua Program Studi Teknik elektro




Jenny Putri Hapsari, S.T., M.T.
NIDN.0607018501

160823

LEMBAR PENGESAHAN PENGUJI

Laporan Tugas Akhir dengan judul “RANCANG BANGUN ROBOT HUMANOID PENARI UNTUK BERJALAN PADA LANTAI BERUNDAK” ini telah dipertahankan di depan dosen penguji sidang Tugas Akhir pada:

Hari : Selasa

Tanggal : 8 Agustus 2023

TIM PENGUJI

Anggota I

Ir. Suryani Alifah, MT., Ph.D.
NIDN.0625036901

Anggota II

Agus Suprajitno, ST., MT.
NIDN.0619076401

Ketua Penguji

Dr. Muhammad Khosyi'in, ST.,MT.
NIDN.0625077901

SURAT PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Nikmatul Azizah
NIM : 30601900038
Fakultas : Fakultas Teknologi Industri
Program Studi : Teknik Elektro

Dengan ini menyatakan bahwa Tugas Akhir yang saya buat dalam rangka menyelesaikan Pendidikan Strata Satu (S1) **Teknik Elektro di Fakultas Teknologi UNISSULA Semarang** dengan judul **“RANCANG BANGUN ROBOT HUMANOID PENARI UNTUK BERJALAN PADA LANTAI BERUNDAK”**, adalah asli (orisinal) dan bukan menjiplak (plagiat) dan belum pernah diterbitkan/dipublikasikan dimanapun dalam bentuk apapun baik sebagian atau keseluruhan, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.

Demikian surat pernyataan ini saya buat dengan sadar dan penuh tanggung jawab. Apabila di kemudian hari ternyata terbukti bahwa Karya Tugas Akhir tersebut adalah hasil karya orang lain atau pihak lain, maka saya bersedia dikenakan sanksi akademis.

Semarang, 8 Agustus 2023

Yang Menyatakan



Nikmatul Azizah

NIM. 30601900038

PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH

Saya yang bertanda tangan di bawah ini:

Nama : Nikmatul Azizah
NIM : 30601900038
Program Studi : Teknik Elektro
Fakultas : Teknologi Industri
Alamat Asal : Jalan Saribowo II RT03/RW01, Ds. Katonsari, Kec. Demak,
Kab. Demak
Email : nikmatulaz@std.unissula.ac.id

Dengan ini menyatakan Karya Ilmiah berupa Tugas Akhir dengan Judul:
**RANCANG BANGUN ROBOT HUMANOID PENARI UNTUK BERJALAN
PADA LANTAI BERUNDAK**

Menyetujui menjadi hak milik Universitas Islam Sultan Agung serta memberikan Hak bebas Royalti Non-Eksklusif untuk disimpan, dialihmediakan, dikelola dan pangkalan data dan dipublikasikan diinternet dan media lain untuk kepentingan akademis selama tetap menyantumkan nama penulis sebagai pemilik hak cipta. Pernyataan ini saya buat dengan sungguh-sungguh. Apabila dikemudian hari terbukti ada pelanggaran Hak Cipta/Plagiatisme dalam karya ilmiah ini, maka segala bentuk tuntutan hukum yang timbul akan saya tanggung secara pribadi tanpa melibatkan Universitas Islam Sultan agung

Semarang, 8 Agustus 2023

Yang Menyatakan



Nikmatul Azizah

NIM. 30601900038

HALAMAN PERSEMBAHAN

Persembahan :

Pertama,

Allah SWT, yang telah memberikan rahmat taufik dan hidayah serta kasih sayangnya, sehingga saya dapat menyelesaikan tugas saya dalam melewati setiap ujian dan cobaan-nya.

Kedua,

Kedua orang tua saya (Bapak Khasirun dan Ibu Nur Siyah) Orang yang hebat yang selalu menjadi penyemangat saya sebagai sandaran terkuat dari kerasnya dunia. Yang tidak henti-hentinya memberikan kasih sayang dengan penuh cinta dan selalu memberikan motivasi. Terimakasih selalu berjuang untuk kehidupan saya. Terimakasih untuk semuanya berkat do'a dan dukungan bapak dan ibu saya bisa berada dititik ini. Sehat selalu dan hiduplah lebih lama lagi. Bapak dan Ibu harus selalu ada disetiap perjalanan dan pencapaian hidup saya,

I love you more

Ketiga,

Untuk seluruh Dosen Fakultas Teknologi Industri Prodi Teknik Elektro Universitas Islam Sultan Agung yang selalu memberikan ilmu yang bermanfaat dan motivasi dalam menyelesaikan studi.

Terakhir,

Untuk diri saya sendiri, terimakasih karena telah mampu berusaha keras dan berjuang sejauh ini dan tak pernah menyerah sesulit apapun proses penyusunan skripsi ini dengan menyelesaikan sebaik dan semaksimal mungkin, ini merupakan pencapaian yang patut dibanggakan untuk diri sendiri.

HALAMAN MOTTO

“Jangan takut atas penilaian orang lain terhadapmu, karena kehidupanmu adalah milikmu. Tapi takutlah atas penilaian Allah terhadapmu, karena kehidupanmu adalah milik-Nya”

(Habib Umar bin Hafidz)

“Karena sesungguhnya sesudah kesulitan itu ada kemudahan, sesungguhnya sesudah kesulitan itu ada kemudahan. Maka apabila kamu telah selesai (dari sesuatu urusan), kerjakanlah dengan sungguh-sungguh (urusan) yang lain, dan hanya kepada Tuhanmulah hendaknya kamu berharap.”

(Q.S Al-Insyirah Ayat 5-8)

“Hadiah terbaik adalah apa yang kamu miliki dan takdir terbaik adalah apa yang sedang kamu jalani”

(Ust. Agam Fachrul)

“Pada akhirnya, ini semua hanyalah permulaan”

(Nadin Amizah)

“Orang lain gak akan bisa paham *struggle* dan masa sulitnya kita yang mereka ingin tahu bagian *succes stories*. Berjuanglah untuk diri sendiri walaupun gak ada yang tepuk tangan. Kelak diri kita di masa depan akan sangat bangga dengan apa yang kita perjuangkan hari ini, tetap berjuang ya”

KATA PENGANTAR

Bismillahirrahmanirrahim

Assalamu'alaikum Warahmatullahi Wabarakatuh

Alhamdulillah puji syukur kehadirat Allah SWT yang telah memberikan Rahmat dan Nikmatnya sehingga masih berkesempatan untuk menuntut ilmu dalam keadaan sehat wal'afiat, Shalawat serta salam semoga selalu tersurahkan kepadabaginda Rasulullah Muhammad SAW, semoga kelak kita mendapatkan syafaatnya. *Aamiin Ya Robbalalamin.*

Penyusunan Tugas Akhir ini adalah salah satu syarat untuk memperoleh gelar sarjana pada Fakultas Teknologi Industri di Universitas Islam Sultan Agung Semarang. Dalam penulisan Tugas Akhir ini tentunya banyak pihak yang memberikan bantuan secara moril maupun materil. Oleh karena itu, penulis menyampaikan upacara terima kasih yang tiada hingganya kepada:

1. Allah SWT yang telah memberikan rahmat dan ridhonya serta memberikan ketabahan, kesabaran dan kelapangan hati serta pikiran dalam menimba ilmu.
2. Kedua orang tua, yakni bapak Khasirun dan ibu Nur Siyah yang telah memberikan dukungan baik materil maupun non materil dan tidak pernah berhenti mendo'akan disetiap sholatnya.
3. Bapak Prof. Dr. Gunarto SH., MHum. selaku Rektor Universitas Islam Sultan Agung Semarang.
4. Ibu Dr. Hj. Novi Marlyana, S.T., M.T. Selaku Dekan Fakultas Teknologi Industri Universitas Islam Sultan Agung Semarang.
5. Ibu Jenny Putri Hapsari, S.T., M.T. Selaku Ketua Jurusan Teknik Elektro Universitas Islam Sultan Agung Semarang.
6. Bapak Dr. Bustanul Arifin, ST., MT. dan Bapak Dr. Eka Nuryanto Budisusila, ST., MT., selaku dosen pembimbing I dan dosen pembimbing II yang memberikan ilmu yang bermanfaat, memberikan banyak arahan, dan dengan sabar membimbing, sehingga dapat menyelesaikan laporan tugas akhir ini.

7. Bapak Dr. Muhammad Khosyi'in, S.T., M.T. Selaku Koordinator Tugas Akhir Program Studi Teknik Elektro Fakultas Teknologi Industri Universitas Islam Sultan Agung.
8. Seluruh dosen dan karyawan Program Studi Teknik Elektro, Fakultas Teknologi Industri, Universitas Islam Sultan Agung Semarang atas ilmu, bimbingan, dan dukungan dalam penyusunan tugas akhir ini.
9. Teman-teman team Robotik yang senantiasa memberikan dukungan, semangat, dan doa juga prestasinya.
10. Teman Teknik Elektro angkatan 2019 senantiasa memberikan dukungan, semangat, dan doa.
11. Semua pihak yang tidak dapat penulis sebutkan satu per satu atas segala dukungan, semangat, ilmu dalam menyelesaikan tugas akhir ini.

Penulis menyadari bahwa didalam penyusunan laporan ini masih jauh dari kesempurnaan, oleh karena itu kritik dan saran sangat diharapkan untuk mencapai hasil yang lebih baik. Semoga laporan ini dapat memberikan manfaat bagi semua pihak pada terutama Mahasiswa Teknik Elektro Universitas Islam Sultan Agung Semarang dan dapat menambah wawasan.

Wassalamu'alaikum Warahmatullahi Wabarakatuh

UNISSULA

بمعتز سلطان أبجوع الإسلامية

Semarang, 8 Agustus 2023



Penulis

DAFTAR ISI

HALAMAN JUDUL.....	iii
<i>FINAL PROJECT</i>	ii
LEMBAR PENGESAHAN PEMBIMBING.....	iii
LEMBAR PENGESAHAN PENGUJI.....	iv
SURAT PERNYATAAN KEASLIAN TUGAS AKHIR.....	v
PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH.....	vi
HALAMAN PERSEMBAHAN.....	vii
HALAMAN MOTTO.....	viii
KATA PENGANTAR.....	ix
DAFTAR ISI.....	xi
DAFTAR GAMBAR.....	xiii
DAFTAR TABEL.....	xv
ABSTRAK.....	xvi
<i>ABSTRACT</i>	xvii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Perumusan Masalah.....	2
1.3 Pembatasan Masalah.....	3
1.4 Tujuan.....	3
1.5 Manfaat.....	3
1.6 Sistematika Penulisan.....	3
BAB II TINJAUAN PUSTAKA DAN LANDASAN TEORI.....	5
2.1 Tinjauan Pustaka.....	5
2.2 Landasan Teori.....	6
2.2.1 Robot Kontes Robot Seni Tari Indonesia (KRSTI).....	6
2.2.2 Robot Humanoid.....	8
2.2.3 Motor Servo Dynamixel MX-28T.....	10
2.2.4 Open CM 9.04.....	11
2.2.5 OpenCM 485.....	14
2.2.6 Aplikasi Program OpenCM IDE.....	16

2.2.7	CM-530	17
2.2.8	RoboPlus	19
BAB III METODE PENELITIAN.....		21
3.1	Waktu dan Penelitian.....	21
3.2	Metode Penelitian.....	21
3.3	Flowchart Sistem.....	23
3.4	Perancangan Alat.....	24
3.4.1	Perancangan Mekanik	24
3.4.2	Perancangan Hardware.....	27
3.4.3	Perancangan Software.....	28
3.5	Skema Pengujian	43
3.5.1	Berjalan pada Permukaan Berundak.....	43
3.5.2	Berjalan pada Permukaan Bidang.....	45
BAB IV DATA DAN ANALISA.....		47
4.1	Perubahan Perancangan Robot.....	47
4.2	Hasil Perancangan Perangkat Lunak.....	49
4.2.1	Mekanisme Jongkok.....	49
4.2.2	Mekanisme <i>Stay</i>	50
4.2.3	Mekanisme Jalan.....	51
4.3	Pengujian Gerakan.....	53
4.3.1	Pengujian Jalan pada Permukaan Bidang	53
4.3.2	Pengujian Jalan pada Permukaan Berundak	60
BAB V PENUTUP.....		70
DAFTAR PUSTAKA		
LAMPIRAN -LAMPIRAN		

DAFTAR GAMBAR

Gambar 2. 1 Pertandingan Kontes Robot Seni Tari Indonesia (KRSTI) UNISSULA.....	7
Gambar 2. 2 Robot Humanoid.....	9
Gambar 2. 3 Goal Position pada Servo Dynamixel MX-28T [12].....	10
Gambar 2. 4 Tipe OpenCM 9.04 [13]	12
Gambar 2. 5 Bagian-bagian OpenCM 9.04 [13]	14
Gambar 2. 6 Bagian-bagian OpenCM 485 [14]	15
Gambar 2. 7 Tampilan Robotis OpenCM IDE.....	17
Gambar 2. 8 Bagian-bagian CM530 [16].....	18
Gambar 2. 9 Tampilan RoboPlus Manager	20
Gambar 3. 1 Diagram Alir Penelitian.....	22
Gambar 3. 2 Flowchart Perancangan Sistem Gerak dan Berjalan Robot Humanoid Penari.....	23
Gambar 3. 3 Desain Perancangan Sistem Mekanik Robot Humanoid Penari.....	25
Gambar 3. 4 Diagram Block Perancangan Sistem Hardware dengan Catu Daya 11,1 volt.....	28
Gambar 3. 5 Flowcart Perancangan Sistem Software	29
Gambar 3. 6 Diagram Alir dari Program Utama	30
Gambar 3. 7 Mekanisme jongkok.....	40
Gambar 3. 8 Bagan Mekanisme Posisi Stay	40
Gambar 3. 9 Diagram Alir Sub Program Jalan.....	41
Gambar 3. 10 Bagan Mekanisme Pergerakan Melangkah Kaki Kanan	42
Gambar 3. 11 Bagan Mekanisme Pergerakan Melangkah Kaki Kiri	43
Gambar 3. 12 Lapangan Tampak Atas	44
Gambar 3. 13 Lapangan Tampak Samping	44
Gambar 3. 14 Skema Pengujian pada Permukaan Bidang (a) tampak depan, (b) tampak samping, dan (c) tampak atas	45
Gambar 4. 1 Perubahan pada pinggang robot.....	48
Gambar 4. 2 Telapak kaki robot (a) sebelumnya dan (b) baru.....	48

Gambar 4. 3 Hasil perancangan mekanisme jongkok	49
Gambar 4. 4 Hasil percobaan mekanisme jongkok dengan kemiringan (a) 2°, (b) 4°, (c) 6°, (d) 8°, dan (e) 10°	50
Gambar 4. 5 Mekanisme Stay.....	50
Gambar 4. 6 Step mekanisme berjalan step (a) pertama, (b) kedua, (c) ketiga, (d) keempat, (e) kelima, (f) keenam, (g) ketujuh, (h) kedelapan, dan (i) kesembilan	51
Gambar 4. 7 Ilustrasi pengukuran menyimpang.....	53
Gambar 4. 8 Hasil percobaan pada permukaan bidang dengan MotionPagePlay 1800.....	55
Gambar 4. 9 Hasil percobaan pada permukaan bidang dengan MotionPagePlay 2200.....	56
Gambar 4. 10 Hasil percobaan pada permukaan bidang dengan MotionPagePlay 2500.....	57
Gambar 4. 11 Hasil percobaan pada permukaan bidang dengan MotionPagePlay 2800.....	59
Gambar 4. 12 Hasil percobaan pada permukaan bidang dengan MotionPagePlay 3200.....	60
Gambar 4. 13 Hasil percobaan pada permukaan berundak dengan MotionPagePlay 1800	62
Gambar 4. 14 Hasil percobaan pada permukaan berundak dengan MotionPagePlay 2200	63
Gambar 4. 15 Hasil percobaan pada permukaan berundak dengan MotionPagePlay 2500	65
Gambar 4. 16 Hasil percobaan pada permukaan berundak dengan MotionPagePlay 2800	67
Gambar 4. 17 Hasil percobaan pada permukaan berundak dengan MotionPagePlay 3200	69

DAFTAR TABEL

Tabel 2. 1 Spesifikasi Motor Servo Dynamixel MX-28T	11
Tabel 2. 2 Spesifikasi OpenCM 9.04.....	12
Tabel 2. 3 Perbedaan OpenCM Tipe A, Tipe B, dan Tipe C [13].....	13
Tabel 2. 4 Bagian-bagian OpenCM 9.04 beserta Fungsinya.....	14
Tabel 2. 5 Spesifikasi OpenCM 485.....	15
Tabel 2. 6 Bagian-bagian Expansion Board OpenCM 485	16
Tabel 2. 7 Spesifikasi Kontroler CM-530	18
Tabel 2. 8 Bagian-bagian CM 530 beserta fungsinya	19
Tabel 3. 1 Keterangan SET ID Robot Humanoid.....	26
Tabel 4. 1 Pengujian berjalan bidang dengan MotionPagePlay 1800	54
Tabel 4. 2 Pengujian berjalan bidang dengan MotionPagePlay 2200	56
Tabel 4. 3 Pengujian berjalan bidang dengan MotionPagePlay 2500	57
Tabel 4. 4 Pengujian berjalan bidang dengan MotionPagePlay 2800	58
Tabel 4. 5 Pengujian berjalan bidang dengan MotionPagePlay 3200.....	59
Tabel 4. 6 Pengujian berjalan pada permukaan berundak dengan MotionPagePlay 1800.....	61
Tabel 4. 7 Pengujian berjalan padapermukaan berundak dengan MotionPagePlay 2200.....	63
Tabel 4. 8 Pengujian berjalan padapermukaan berundak dengan MotionPagePlay 2500.....	64
Tabel 4. 9 Pengujian berjalan pada permukaan berundak dengan MotionPagePlay 2800.....	66
Tabel 4. 10 Pengujian berjalan padapermukaan berundak dengan MotionPagePlay 3200.....	68

ABSTRAK

Robot *humanoid* membutuhkan keseimbangan pada saat berjalan. Pada KRSTI (Kontes Robot Seni Tari Indonesia) robot *humanoid* dapat berjalan dengan baik di atas lapangan dengan ketentuan yang telah ditentukan pada buku pedoman KRI. Permasalahan dari robot *humanoid* yaitu dari gerakan jalan masih belum dapat berjalan dengan lurus, kurangnya keseimbangan pada robot dan belum dapat berjalan pada permukaan berundak yang pada saat itu memiliki diameter 600mm dan ketinggian 4mm. Dari permasalahan tersebut maka, penelitian ini akan membahas tentang sistem berjalannya robot pada permukaan bidang dan berundak yang dirancang dengan menggunakan metode *MotionPagePlay* yang terdapat pada *software* Robotis OpenCM. *MotionPagePlay* ini digunakan untuk mengatur kecepatan gerak atau perpindahan gerak pada robot *humanoid*. Robot *humanoid* ini menggunakan servo *Dynamixel* MX-28T dengan OpenCM9.04 dan OpenCM 485 EXP untuk sistem kontrolnya. Pemrograman yang digunakan pada *software* Robotis OpenCM yaitu menggunakan bahasa C++. Hasil dari penelitian berjalannya robot di permukaan bidang dan berundak dengan menggunakan beberapa variasi *MotionPagePlay* yaitu 1800, 2200, 2500, 2800, dan 3200. Dari hasil pengujian pada permukaan bidang dengan menggunakan variasi *MotionPagePlay* 2200 menghasilkan rata-rata persentase *error* 0,33% dengan jumlah keberhasilan dapat berjalan di dalam batas 26 cm yaitu 70%, sedangkan pada *MotionPagePlay* 2800 rata-rata persentase *error* 0,80% dengan jumlah keberhasilan 50%. Pada hasil pengujian permukaan berundak yang memiliki perbedaan ketinggian 4mm dengan *MotionPagePlay* 2500 dan 2800 menghasilkan jumlah keberhasilan 90%, sedangkan pada *MotionPagePlay* 3200 untuk jumlah keberhasilannya yaitu 10%.

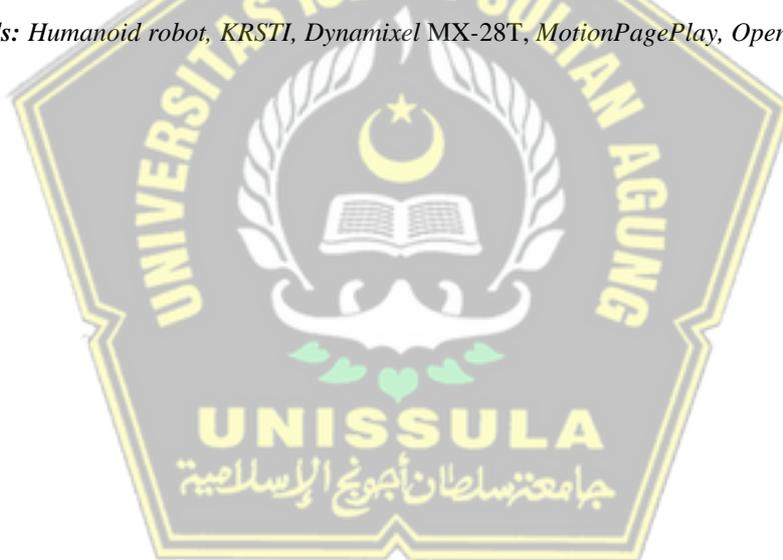
Kata kunci: Robot *Humanoid*, (KRSTI), Servo *Dynamixel* MX-28T, *MotionPagePlay*, OpenCM



ABSTRACT

Humanoid robots need balance when walking. In KRSTI (Indonesian Dance Robot Contest) humanoid robots can run well on the field with the conditions specified in the KRI guidebook. The problem with the humanoid robot is that the movement of the road is still not able to walk straight, the robot lacks balance and cannot walk on a stepped surface which at that time had a diameter of 600mm and a height of 4mm. Based on these problems, this study will discuss the system for walking robots on plane surfaces and steps designed using the MotionPagePlay method found in Robotis OpenCM software. MotionPagePlay is used to adjust the speed of movement or movement of the humanoid robot. This humanoid robot uses a Dynamixel MX-28T servo with OpenCM9.04 and OpenCM 485 EXP for its control system. The programming used in the Robotis OpenCM software is using the C++ language. The results of research on the robot walking on plane surfaces and steps using several variations of MotionPagePlay, namely 1800, 2200, 2500, 2800, and 3200. From the test results on the surface of the plane using the MotionPagePlay 2200 variation, it produces an average error percentage of 0.33% with a total success can run within the 26 cm limit, which is 70%, while on the MotionPagePlay 2800 the average error percentage is 0.80% with a success rate of 50%. The results of testing the stepped surface which has a height difference of 4mm with MotionPagePlay 2500 and 2800 produce a success rate of 90%, while for MotionPagePlay 3200 the success rate is 10%.

Keywords: Humanoid robot, KRSTI, Dynamixel MX-28T, MotionPagePlay, OpenCM



BAB I PENDAHULUAN

1.1 Latar Belakang

Perkembangan dunia robotika di Indonesia sudah maju sangat pesat. Dengan terbuktinya banyak kontes-kontes robot yang telah diselenggarakan dan juga jumlah pesertanyapun semakin meningkat dari tahun ke tahun. Berdasarkan panduan umum Kontes Robot Indonesia 2023 dijelaskan bahwa KRI (Kontes Robot Indonesia) adalah kontes yang diadakan setiap satu tahun sekali yang berskala nasional. Perlombaan ini diikuti oleh perguruan tinggi seluruh Indonesia baik negeri maupun swasta. Kontes Robot Seni Tari Indonesia (KRSTI) adalah kompetisi perancangan, pembuatan dan pemrograman robot *humanoid* yang disertai dengan unsur seni dan budaya bangsa Indonesia khususnya seni tari di tanah air. Untuk pelaksanaan KRSTI tahun 2023 mengambil tema “Robot Penari Denok Semarang atau Robot Penari Gambang Semarang” [1]. Robot *humanoid* merupakan robot yang dirancang memiliki kemampuan yang menyerupai dengan manusia. Secara umum robot humanoid mempunyai bagian-bagian tubuh seperti manusia secara garis besarnya yang diantaranya yaitu kepala, dua buah lengan, dan dua kaki. Robot *humanoid* memiliki persendian yang sama seperti seorang manusia yang disebut dengan sendi gerak. Persendian pada robot ini dirancang dengan cara mengatur besar derajat kebebasan tertentu yang sesuai dengan fungsinya. Gerak pada robot *humanoid* dapat dihasilkan dengan cara mengatur putaran motor servo yang berfungsi sebagai sendi dari robot [2]. Dalam penggunaan motor memerlukan sebuah kontrol agar dapat digunakan sesuai dengan fungsinya [3].

Dalam penelitian ini akan lebih diamati yaitu tujuan utama dari robot *humanoid* penari adalah agar robot dapat berjalan dengan seimbang lurus dan dapat melewati undakan. Terdapat kendala yang dihadapi peserta Kontes Robot Seni Tari Indonesia (KRSTI) adalah kurang seimbang gerak jalan robot *humanoid* sehingga mengakibatkan robot terjatuh. Jika hal tersebut tidak diperbaiki, maka robot dapat saja tidak sampai pada zona *finish* ataupun akan terjadinya pengurangan nilai pada saat tampil.

Permasalahan yang dialami saat Kontes Robot Seni Tari Indonesia di tahun 2022 yang dilaksanakan secara *offline*, pada saat itu tim Robotik UNISSULA bernama Abhipraya dan sekarang menjadi tim SAROSETA. Permasalah dari robot tersebut yaitu dari gerakan jalan masih belum dapat berjalan dengan lurus, kurangnya keseimbangan pada robot dan belum dapat berjalan pada permukaan berundak yang pada saat itu memiliki diameter 600mm dan ketinggian 4mm. Kemudian, kurangnya riset analisa mengenai gerak jalan robot dan pada permukaan berundak.

Solusi dari permasalahan tersebut adalah diperlukannya perbaikan dari segi mekanik dan pemrograman pada robot *humanoid*. Dari segi mekanik terdapat perubahan yaitu luas dari telapak kaki robot *humanoid* menjadi lebih luas dari sebelumnya dan lebih ringan, untuk bahan yang digunakan yaitu *Aluminium Composite Panel* (ACP) dengan luas 124,8 cm. Kemudian, dari segi pemrograman diperlukannya analisa pada pemrograman berjalan robot *humanoid* agar dapat berjalan dengan lebih seimbang dan pada *MotionPagePlay* yang digunakan. *MotionPagePlay* tersebut yaitu untuk mengatur kecepatan gerak atau perpindahan gerak pada robot *humanoid*. Dari solusi yang telah diberikan perubahan pada mekanik dan pemrograman akan menghasilkan robot dapat berjalan dengan baik.

1.2 Perumusan Masalah

Berdasarkan latar belakang yang telah diuraikan di atas, maka dapat diambil rumusan masalah sebagai berikut:

1. Bagaimana merancang robot *humanoid* penari pada bagian kaki?
2. Bagaimana cara mengontrol motor servo Dynamixel MX-28T untuk menggerakkan kaki pada robot *humanoid* KRSTI?
3. Bagaimana hasil gerak jalan robot pada permukaan bidang dapat berjalan dengan lurus?
4. Bagaimana hasil gerak jalan robot pada permukaan berundak?
5. Berapakah hasil akurasi terbaik di *MotionPagePlay* pada gerak jalan robot lurus dan berundak?

1.3 Pembatasan Masalah

Dalam penyusunan penelitian ini, telah ditentukan batasan-batasan masalah sebagai berikut:

1. Perancangan gerak pada robot *humanoid* ini dengan motor servo Dynamixel MX-28T.
2. Robot *humanoid* dapat berjalan lima langkah pada permukaan yang bidang.
3. Pengujian robot dapat berjalan naik dan turun pada lapangan yang berundak. Lapangan memiliki zona dengan ketinggian 4mm dengan terdapat zona larangan dengan ketinggian 10mm dan berdiameter 300mm.

1.4 Tujuan

Tujuan penelitian ini adalah membuat perancangan atau rancang bangun robot *humanoid* dengan melakukan percobaan pergerakan berjalan yaitu dapat melewati jalan dengan ketinggian yang berbeda.

1.5 Manfaat

Adapun manfaat dalam penelitian ini adalah mengetahui prinsip kerja, karakteristik, torsi, beban pada motor servo Dynamixel-28T pada rancang bangun robot *humanoid* penari untuk berjalan pada lantai berundak.

1.6 Sistematika Penulisan

Ada beberapa sistematika penulisan dalam penulisan laporan tugas akhir adalah sebagai berikut:

BAB I : PENDAHULUAN

Bab ini berisikan latar belakang, perumusan masalah, pembatasan masalah, tujuan, manfaat dan sistematika penulisan.

BAB II : TINJAUAN PUSTAKA DAN LANDASAN TEORI

Bab ini akan membahas mengenai rujukan-rujukan dari penelitian sebelumnya yang tentunya berkaitan dengan tugas akhir yang dibuat. Selain itu, bab ini juga berisikan berbagai teori dan juga konsep yang mendukung dalam pembuatan tugas akhir ini.

BAB III : METODE PENELITIAN

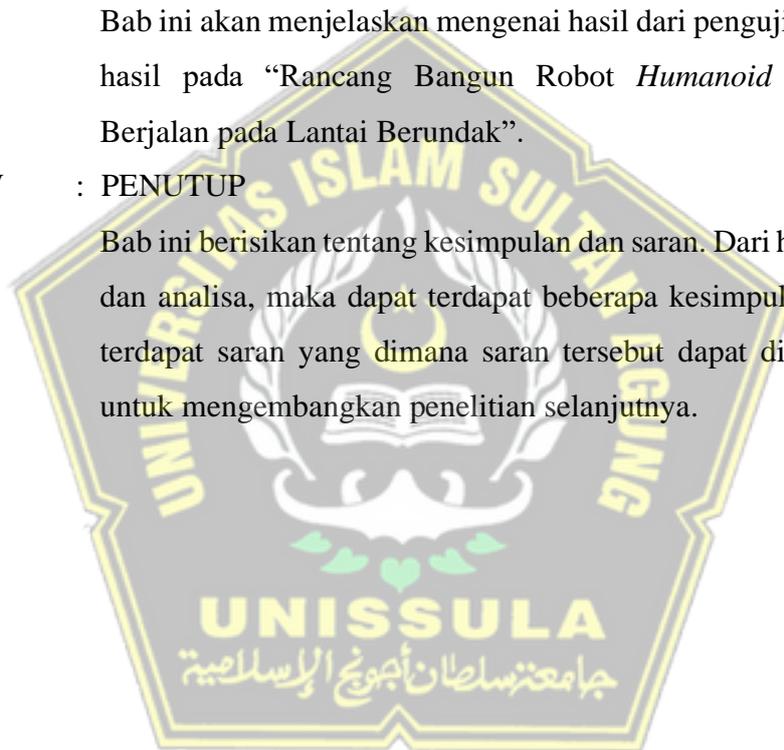
Bab ini akan menguraikan dan memaparkan mengenai metode penelitian yang digunakan yaitu, waktu dan penelitian, metode penelitian yang digunakan, flowcart keseleuruhan sistem, serta perancangan alat yang berupa perancangan mekanik, hardware dan juga software untuk robot *humanoid* dapat berjalan dengan melewati jalan yang berundak.

BAB IV : PENGUJIAN DAN ANALISA

Bab ini akan menjelaskan mengenai hasil dari pengujian dan analisa hasil pada “Rancang Bangun Robot *Humanoid* Penari untuk Berjalan pada Lantai Berundak”.

BAB V : PENUTUP

Bab ini berisikan tentang kesimpulan dan saran. Dari hasil penelitian dan analisa, maka dapat terdapat beberapa kesimpulan. Selain itu, terdapat saran yang dimana saran tersebut dapat digunakan guna untuk mengembangkan penelitian selanjutnya.



BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Tinjauan Pustaka

Pada tugas akhir yang dilakukan oleh Muchammad Ainur Fahd, Djoko Purwanto, dan Muhammad Hilman Fato, meneliti tentang rancang bangun robot penari *humanoid* dengan 25 DoF (Degree of Freedom) yang dapat melakukan gerakan tari remo. Motor servo yang digunakan dalam robot ini yaitu servo MX-28T dan untuk sub controllernya CM-730 [4].

Pada penelitian yang dilakukan oleh Abdul Jalil membuat rancang bangun *humanoid*, yang dapat digunakan sebagai sarana pendidikan. Robot *humanoid* ini dirancang menyerupai manusia untuk ukuran tingginya pun dibentuk menyerupai dengan manusia. Motor Servo yang digunakan yaitu HITEC HS-645MG dan terdapat komponen pendukung lainnya [2].

Zamzami, salahuddin, Anwar dan Yusman telah melakukan penelitian mengenai rancang bangun robot *humanoid*. Penelitian yang dilakukan ini menggunakan mikrokontroler Atmega 2560. Untuk kasus masalah bagaimana perancangan penggabungan semua motor servo sebagai aktuator agar robot dapat berjalan dengan baik [5].

Fajar Hermawanto dan Irwan Karim, melakukan penelitian dengan judul “Rancang Bangun Robot Penari Tidi”. Penelitian tersebut merancang sebuah robot Bioloid Premium Tipe A yang dimana robot ini mampu melakukan gerakan tarian Tidi. Kontrol utama pada robot ini terdapat pada papan kontrol CM-530, pada papan ini terdapat mikrokontroler yang berfungsi sebagai pusat kendali guna penyimpanan program [6].

Tio Haryanti Adi Putra, Dahniyal Syauqy dan Rizal Maulana, melakukan penelitian dengan judul “Sistem Pendeteksi Tempo Lagu Untuk Kontes Robot Seni Tari Indonesia (KRSTI) Berdasarkan Frekuensi Dengan Algoritma Bat This”. Pada penelitian ini membuat suatu sistem agar robot KRSTI dapat mengerti apa arti irama pada ketukan lagu, kecepatan lagu dan tempo [7].

Pada beberapa penelitian yang dilakukan sebelumnya yaitu perancangan sistem kontrol pada robot *humanoid* penari menggunakan mikrokontroler Atmega, CM-730 dan CM-530. Perbedaan dari penelitian yang dilakukan kali ini yaitu menggunakan OpenCM 485 EXP dan OpenCM9.04. Kemudian, sistem kontrol tersebut diprogram dengan menggunakan *software* Robotis.

Pada penelitian ini yang dilakukan adalah perancangan sistem gerak jalan pada robot *humanoid* penari dengan menggunakan servo Dynamixel MX-28T. Perbedaan dengan penelitian yang telah dibuat sebelumnya yaitu penelitian ini merupakan perancangan sistem gerak jalan pada robot *humanoid* yang dapat melewati sebuah jalan yang memiliki ketinggian yang berbeda. Robot *humanoid* dapat berjalan melewati Zona B yang berukuran 1190 x 800 mm dan memiliki permukaan dengan ketinggian 4mm, tanpa menyentuh Zona Larangan yang berada di tengah Zona B yaitu dengan diameter 300mm dan tinggi 10mm [1]. Pemilihan penelitian ini yaitu dikarenakan pada KRI tahun 2023 terdapat rintangan untuk robot dapat melewati undakan. Pada penelitian ini menggunakan *software* Robotis untuk perancangan programnya. Tujuan dari penelitian ini yaitu agar robot dapat melakukan gerakan berjalan dengan melewati alas yang memiliki ketinggian yang berbeda atau dapat disebut dengan undakan.

2.2 Landasan Teori

2.2.1 Robot Kontes Robot Seni Tari Indonesia (KRSTI)

Kontes Robot Indonesia (KRI) merupakan suatu kegiatan kompetisi tahunan bagi mahasiswa dalam bidang desain dan rekayasa robotika yang diselenggarakan oleh Pusat Prestasi Nasional, Kementerian Pendidikan dan Kebudayaan Republik Indonesia. Pada tahun 2003 merupakan pertama kalinya diselenggarakan Kontes Robot Indonesia (KRI) yang berada dibawah Direktorat Jendral Pendidikan Tinggi, Departemen Pendidikan dan Kebudayaan pada masanya. Penyelenggaraan KRI 2023 ini oleh Balai Pengembangan Talenta Indonesia (BPTI) Pusat Prestasi Nasional Puspresnas Kementerian Pendidikan, Kebudayaan, Riset dan Teknologi (Kemendikbudristek) Republik Indonesia [8].

Pada Kontes Robot Indonesia (KRI) 2023 terdapat berbagai divisi, salah satunya yaitu Kontes Robot Seni Tari Indonesia (KRSTI).

Divisi Kontes Robot Seni Tari Indonesia (KRSTI) merupakan salah satu ajang kompetisi dalam perancangan, pembuatan, serta pemrograman robot yang dimana didalamnya disertai dengan unsur tari dan budaya dari bangsa Indonesia. Hal tersebut dibuktikan dengan nama divisinya sendiri yaitu robot seni tari Indonesia yang sudah pastinya robot yang nantinya akan memperlombakan unsur tari daerah di Indonesia. Tahun 2023 ini dengan tema “Robot Denok Semarangan” yang berasal Semarang, Jawa Tengah.

Pertandingan Kontes Robot Seni Tari Indonesia (KRSTI) robot diharuskan dapat melakukan beberapa gerakan tarian apabila terdapat alunan musik yang dinyalakan dan berhenti ketika tidak adanya alunan musik. Robot diharuskan untuk tetap menjaga keseimbangan sampai dengan selesainya zona lapangan. Dalam perlombaan terdapat papan lapangan yang digunakan sebagai alas untuk melakukan perlombaan, yang dimana dalam lapangan tersebut terdapat zona-zona diantara yaitu Zona A, Zona B, dan Zona C. Pada Zona A terdapat Zona Mulai untuk 2 robot. Pada Zona B yang memiliki ketinggian 4mm dibandingkan dengan Zona A dan Zona C, selain itu di Zona terdapat Zona Larangan yang tidak boleh untuk dilewati yaitu dengan diameter 300mm dan tinggi 10mm [1].



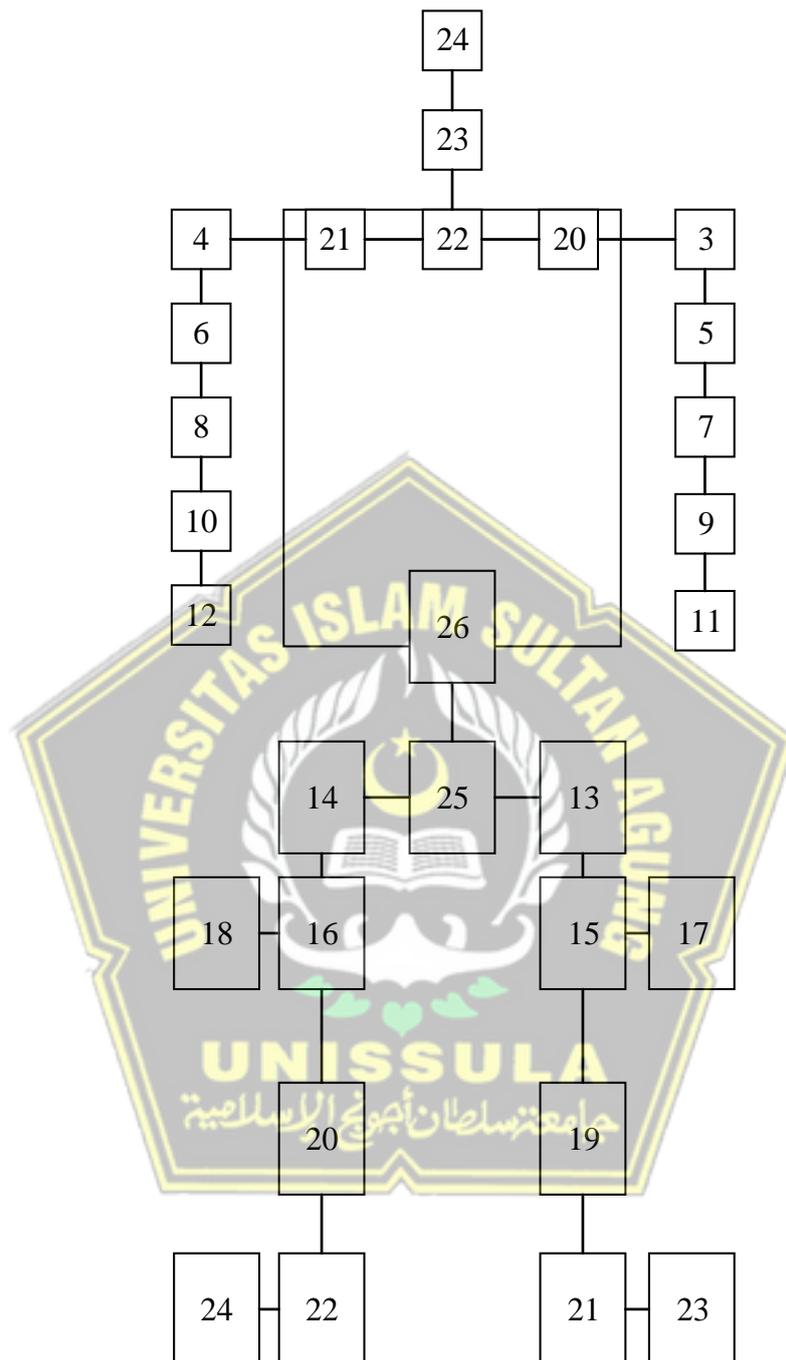
Gambar 2. 1 Pertandingan Kontes Robot Seni Tari Indonesia (KRSTI) UNISSULA

2.2.2 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). Nama “*robot*” merupakan istilah dari robot yang berasal dari bahasa Cheko yang memiliki arti pekerja atau kuli yang tidak mengenal lelah atau bosan. Pada perkembangan zaman ini robot mulai memasuki pasaran konsomen di bidang hiburan [9].

Dari istilah robot *humanoid* sendiri terdapat dua unsur dalam pembentukan kata, yaitu robot dan *humanoid*. Robot merupakan perangkat cerdas mekanik yang dapat melakukan tugas secara otomatis maupun dengan adanya instruksi dari pengguna, sedangkan *humanoid* sendiri yaitu segala sesuatu yang memiliki bentuk maupun struktur yang menyerupai manusia. Dari penjelasan keduanya yaitu dapat disimpulkan bahwa definisi dari robot *humanoid* adalah perangkat cerdas mekanik yang dapat melakukan tugas secara otomatis maupun dengan adanya instruksi dari pengguna atau dapat dikatakan telah diberikan masukan input program oleh pengguna, yang dimana robot ini berbentuk layaknya menyerupai manusia baik dari struktur maupun pergerakannya. Robot *humanoid* dibuat untuk meniru gerakan tubuh manusia, sehingga mereka menggunakan aktuator yang berfungsi sebagai sendi dan otot, meskipun dengan struktur yang berbeda. Robot *humanoid* menggunakan aktuator untuk mencapai efek yang sama seperti gerakan manusia [10].

Secara visual, struktur bentuk fisik robot humanoid dapat dilihat pada Gambar 2.2.



Gambar 2. 2 Robot Humanoid

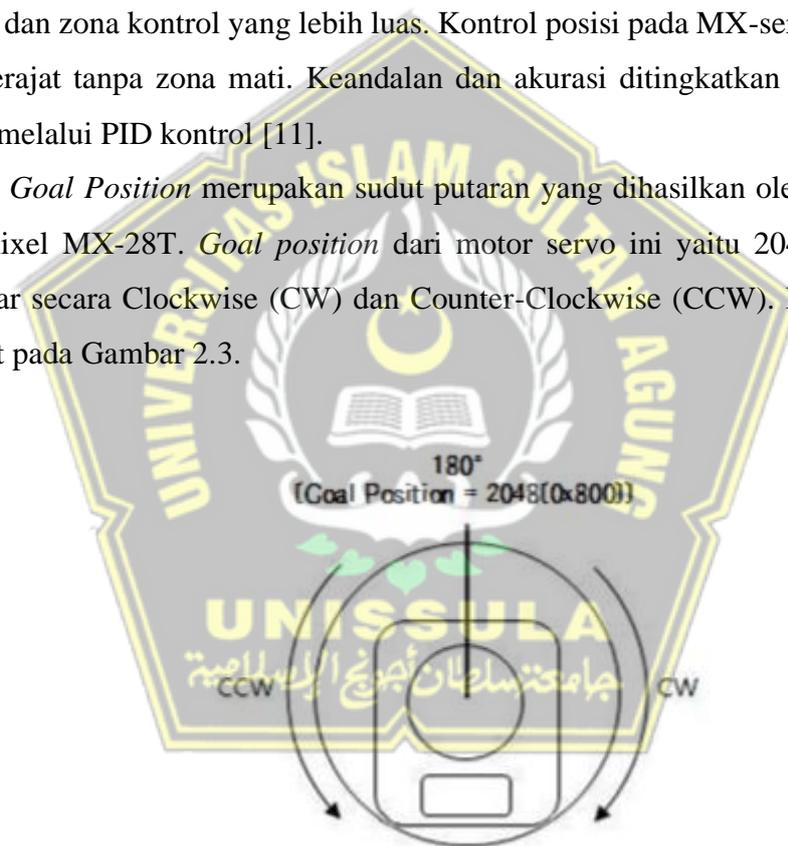
Pada Gambar 2.2, dapat dijelaskan bahwa robot *humanoid* memiliki 27 *joint*. Robot humanoid ini menggunakan 2 jenis motor servo yaitu servo dynamixel XL-320 dan MX-28T. *Joint* atau dapat disebut sendi dari robot humanoid menggunakan jenis *revolute*. Sendi *revolute* merupakan sebuah sendi seperti engsel

dan memungkinkan gerak rotasi antara dua *link*. Robot humanoid ini menggunakan kontroler OpenCM485 dan openCM 9.04.

2.2.3 Motor Servo Dynamixel MX-28T

Servo Dynamixel MX series merupakan konsep baru dari Dynamixel dengan fungsi lanjutan, seperti kontrol presisi, kontrol PID, kontrol posisi 360 derajat dan komunikasi kecepatan tinggi. Berkat adanya *Contactless Absolute Encoder* yang baru diterapkan mempengaruhi daya tahan tingkat lanjut, tingkat presisi dan zona kontrol yang lebih luas. Kontrol posisi pada MX-series ini sebesar 360 derajat tanpa zona mati. Keandalan dan akurasi ditingkatkan dalam kontrol posisi melalui PID kontrol [11].

Goal Position merupakan sudut putaran yang dihasilkan oleh motor servo dynamixel MX-28T. *Goal position* dari motor servo ini yaitu 2048 yang dapat berputar secara Clockwise (CW) dan Counter-Clockwise (CCW). Lebih jelasnya terlihat pada Gambar 2.3.



Gambar 2. 3 Goal Position pada Servo Dynamixel MX-28T [12]

Spesifikasi motor servo dynamixel MX-28T dapat dilihat pada Tabel 2.1 di bawah ini:

Tabel 2. 1 Spesifikasi Motor Servo Dynamixel MX-28T

Keterangan	Spesifikasi
MCU	ARM CORTEX-M3 (72MHz, 32 Bit)
Sensor posisi	Encoder mutlak tanpa kontak (12Bit, 360°)
Kecepatan komunikasi	8000 [bps]~4,5[Mbps]
Algoritma kontrol	PID
Resolusi	4096 [pulse/rev]
Reaksi	20[armin] (0.33°)
Mode Operasi	Mode Roda (Belok Tak Berujung) Mode Bersama (0 ~ 360 [°]) Mode Multi-putaran
Bobot	72 gram
Dimensi (LxTxD)	35.6x50.6x35.5 mm
Perbandingan gigi	193:1
Torsi Kios	2,3 [Nm] pada catu daya 11,1 [V], 1,3 [A]
Kecepatan tanpa beban	55 [putaran/menit] Pada catu daya 12 [V]
Suhu operasi	-5~+80°C
Input tegangan	10.0~14.8 V Direkomendasikan 12.0V
Sinyal perintah	Paket Digital
Koneksi fisik	RS485 / TTL Multidrop Bus TTL Half Duplex Asynchronous Serial Communication dengan 8bit, 1stop, No Parity RS485 Asynchronous Serial Communication dengan 8bit, 1stop, No Parity
ID servo	254 ID (0~253)
Umpan balik	Posisi, Suhu, Beban, Tegangan Input, dll
Material	Plastik Rekayasa
Material gigi	Penglengkapan logam penuh
Arus siaga	100 mA

2.2.4 Open CM 9.04

OpenCM 9.04 merupakan papan mikrokontroler yang dirilis oleh perusahaan ROBOTIS yang dimana OpenCM ini bersifat *Open Source*. Kontroler ini berbasis STM32F103CB dengan ARM Cortex-M3 CPU sebagai *chip* mikrokontroler, memori *flash* 128 kB dan memori SRAM 20kB. Kontroler ini dilengkapi dengan skematik catu daya yang mampu menyuplai tegangan 5 volt dan 3.3 volt DC. Tegangan 5 volt untuk menyuplai pada jalur TTL sedangkan tegangan 3.3 volt digunakan untuk menyuplai mikrokontroler, port sponsor 5 pin dan port

komunikasi 4 pin. OpenCM 9.04 memiliki 3 tipe, yakni tipe A, tipe B dan tipe C seperti terlihat pada Gambar 2.4. Perbedaan antara tiga jenis ini dapat dilihat pada Tabel 2.3 [13].

Mikrokontroler ini dilengkapi dengan konektor 3 pin TTL untuk mengoperasikan *smart servo* dynamixel jenis AX, MX, XL-series. Selain itu terdapat pula pin GPIO sebanyak 26 pin yang nantinya dapat digunakan untuk *input* analog, USB, CAN, SPI, I2C, dan serial port. Penggunaan pin GPIO hanya dapat digunakan apabila openCM 9.04 diprogram menggunakan *software* Robotis OpenCM IDE. Spesifikasi dari *hardware* OpenCM 9.04 sebagai berikut [13]:

Tabel 2. 2 Spesifikasi OpenCM 9.04

Keterangan	Spesifikasi
CPU	STM32F103CB (ARM Cortex-M3)
Tegangan kerja	5V~16V (5V USB; 12V DXL Port; 7,4V XL-Series Port)
Input/Output	26 GPIO
Timer	4 (16 bit)
ADC	10 (12 bit)
Flash	128 Kb
SRAM	20 Kb
Clock	72 Mhz
USB	1 mikro USB Tipe B 2.0
USART	3 buah
Debug	JTAG dan SWD
3 Pin TTL	4 buah
Dimensi	27mmx66.6mm



[OpenCM9.04 A-Type]



[OpenCM9.04 B-Type]



[OpenCM9.04 C-Type]

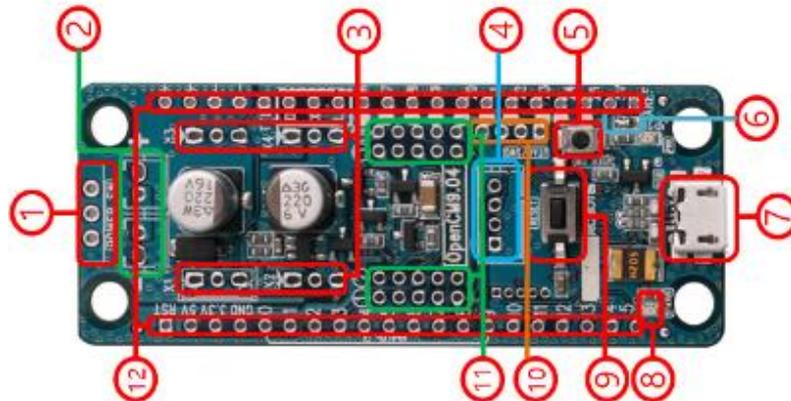
Gambar 2. 4 Tipe OpenCM 9.04 [13]

Pada Gambar 2.4 merupakan macam-macam tipe dari mikrokontroler OpenCM yaitu terdapat tipe A, tipe B, dan tipe C. Dapat dilihat dari gambar tersebut bahwa ketiganya jelas memiliki sebuah perbedaan. Perbedaan dari ketiganya hanya terdapat pada soket yang berada di OpenCM9.04. Pada tipe A hanya memiliki tombol user, sedangkan pada tipe B dan C memiliki saklar lengkap yang membedakan hanya pada tipe B terdapat 2 sambungan Dynamixel seri AX/MX dan terdapat 2 sambungan Dynamixel seri XL, sedangkan yang C hanya memiliki sambungan Dynamixel TTL seri XL yang berjumlah 4. Untuk perbedaan lebih jelasnya dapat dilihat pada Tabel 2.3 berikut ini:

Tabel 2. 3 Perbedaan OpenCM Tipe A, Tipe B, dan Tipe C [13]

Fitur	Tipe A	Tipe B	Tipe C
Saklar Catu Daya	X	1	1
Tombol Pengguna	1	1	1
2-Pin Baterai (LBS-40)	X	2	2
JTAG/SWD	X	1	1
Port Mikro USB	X	1	1
Port 5-Pin	X	4	4
Sambungan Dynamixel TTL seri AX/MX	X	2	X
Sambungan Dynamixel TTL seri XL	X	2	4
4-Pin komunikasi	X	1	1

Selain spesifikasi tersebut, terdapat pula bagian-bagian dari OpenCM 9.04 tipe C yang ditunjukkan oleh Gambar 2.5. Adapun fungsi-fungsi dari bagian OpenCM yang nantinya dijelaskan pada Tabel 2.4. Pada Gambar 2.4 terdapat nomer-nomer yang dimana merupakan inisialisasi yang akan dijelaskan pada Tabel 2.4. Komponen-komponen pada openCM tersebut diantaranya yaitu tombol power, soket baterai atau catu daya, pin Dynamixel TTL, port komunikasi, tombol user, eksternal ADC, mikro USB, status LED, tombol reset, 4-pin JTAG/SWD, 5 pin sensor eksternal, dan pin GPIO2.5 mm.



Gambar 2. 5 Bagian-bagian OpenCM 9.04 [13]

Tabel 2. 4 Bagian-bagian OpenCM 9.04 beserta Fungsinya

No	Keterangan	Fungsi
1	Tombol power	Saklar catu daya
2	Soket baterai	Soket penghubung baterao
3	3-Pin dynamixel TTL	Konektor dynamixel berbasis TTL
4	Port komunikasi	Komunikasi UART BT-110A, BT-210, ZIG-110A, LN-101. 4 Pin port komunikasi pada OpenCM 9.04 menggunakan Serial2 (USART2)
5	Tombel <i>user</i>	Digunakan untuk memasukkan mode pemulihan (<i>recovery</i>) firmware pada OpenCM9.04
6	Ext. ADC Ref <i>Jumper</i>	Merubah tegangan referensi analog
7	Mikro USB-B	Jalur komunikasi, mengunduh program, dan sebagai catu daya 5 volt
8	Status LED	Indikator percobaan program LED OpenCM 9.04
9	Tombol <i>reset</i>	Digunakan untuk <i>reset CPU</i>
10	4-Pin JTAG/SWD	Digunakan untuk menghubungkan ST-LINK dan sejenisnya
11	5-Pin sensor eksternal	Digunakan untuk mengakses produk sensor ROBOTIS
12	Pin GPIO 2.5 mm	Digunakan untuk mengakses perangkat eksternal ke CPU OpenCM 9.04

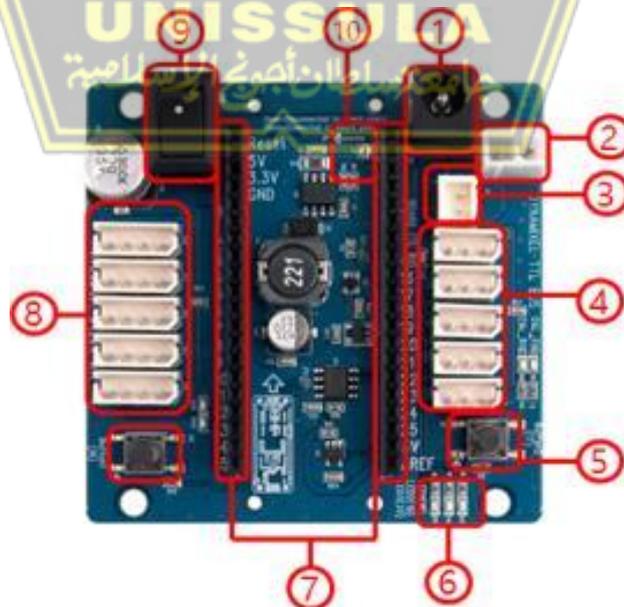
2.2.5 OpenCM 485

OpenCM 485 merupakan *board ekspansi* sebagai pengontrol servo dynamixel. OpenCM ini terhubung dengan pengendali OpenCM 9.04. Papan kontroler OpenCM 9.04 diperlukan untuk digunakan OpenCM 485 *ekspansi board*. Kontroler OpenCM 9.04 sebagai pendukung Dynamixel menggunakan TTL dan RS-485. Spesifikasi dari OpenCM 485 *ekspansi board* sebagai berikut [14]:

Tabel 2. 5 Spesifikasi OpenCM 485

Keterangan	Spesifikasi
Tegangan kerja	5V~30V
Catu daya	SMPS, LiPo, DXL PRO 24V
Saklar daya	1
Port Dynamixel	4Pin x 5, 3Pin x 5
Tombol	2
LED	5
Dimensi	68mm x 66.5 mm
Berat	32g
Serial3 TX	Header Pin#24
Serial3 RX	Header Pin#25
Direction Control	Header Pin#22

Selain spesifikasi tersebut, terdapat pula bagian-bagian dari *ekspansi board* yang ditunjukkan oleh Gambar 2.6. Adapun fungsi-fungsi dari bagian OpenCM ini yang nantinya akan dijelaskan pada Tabel 2.6. Pada Gambar 2.6 terdapat nomer-nomer yang dimana merupakan inisialisasi yang akan dijelaskan pada Tabel 2.6. Komponen-komponen pada OpenCM tersebut diantaranya yaitu Konektor SMPS DC, Konektor Daya DXL Pro, Konektor baterai Li-Po, Dynamixel TTL 3-Pin Bus, Tombol Button, LED, Input/Output, Dynamixel 485 4-Pin Bus, Saklar Daya, dan Jumper JP1.

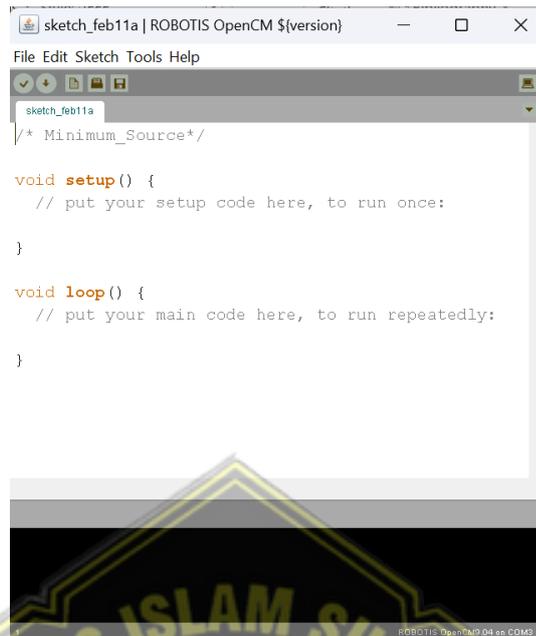
**Gambar 2. 6** Bagian-bagian OpenCM 485 [14]

Tabel 2. 6 Bagian-bagian *Expansion Board* OpenCM 485

No	Nama	Fungsi
1	Konektor SMPS DC (2.5 mm ID/ 5.5mm OD, <i>Center Posistive</i>)	Digunakan untuk menggunakan adaptor DC SMPS untuk memasok daya ke <i>Expansion Board</i> OpenCM 485
2	Konektor Daya DXL Pro	Konektor daya untuk Dynamixel Pro (24V)
3	Konektor baterai Li-Po	Konektor untuk baterai Li-Po 11.1V
4	Dynamixel TTL 3-Pin Bus	Digunakan untuk menghubungkan kabel 3-Pin (Dynamixel TTL Bus) serta untuk daisy chaining Dynamixel
5	Tombol <i>Button</i>	Digunakan untuk memasukkan mode pemulihan (<i>recovery</i>) firmware pada papan ekspansif OpenCM 485
6	LED	Indikator percobaan program LED pada pada papan ekspansif OpenCM 485
7	Input/Output	Pin header yang diguankan untuk me-mount <i>Expansion Board</i> OpenCM485 (pitch 2.54mm)
8	Dynamixel 485 4-Pin Bus	Digunakan untuk menghubungkan Kabel 4-Pin (Dynamixel TTL Bus) serta untuk daisy chaining Dynamixel.
9	Saklar Daya	Saklar yang memberi daya pada board dan juga Dynamixel
10	Jumper JP1	Jumper yang menentukan apakah daya yang dimasukkan ke OpenCM 485 EXP akan disuplai ke board OpenCM 9.04 ataupun tidak

2.2.6 Aplikasi Program OpenCM IDE

Software Robotis OpenCM IDE merupakan perangkat lunak yang dapat digunakan untuk memprogram mikrokontroler OpenCM menggunakan bahasa pemrograman C [15]. Robotis OpenCM IDE dapat digunakan untuk mengoperasikan motor servo dynamixel AX, MX dan XL menggunakan bahasa C. *Software* ini bersifat *open source* dan berbasis *software* Arduino IDE. Tampilan Robotis OpenCM IDE juga hampir mirip dengan Arduino IDE. Tampilan Robotis dapat dilihat pada Gambar 2.7.



Gambar 2. 7 Tampilan Robotis OpenCM IDE

Fitur yang disediakan pada OpenCM IDE ini mirip dengan *software* Arduino IDE, perbedaannya hanya terletak pada penambahan *library* yang khusus untuk mengakses motor servo dengan jenis dynamixel dengan beberapa contoh program didalamnya. Selain itu terdapat pula *library* khusus untuk mengoperasikan produk-produk platform keluaran dari perusahaan ROBOTIS seperti Darwin Mini, OLLO, Bioloid, dan lain-lainnya disertai dengan contoh program. Terdapat beberapa *example* dalam *software* Robotis OpenCM ini, salah satunya adalah *Motion_Page_Play*. *MotionPagePlay* dapat digunakan untuk merekam gerakan robot dan mengatur kecepatan pemutaran atau perpindahan gerakan pada robot *humanoid*.

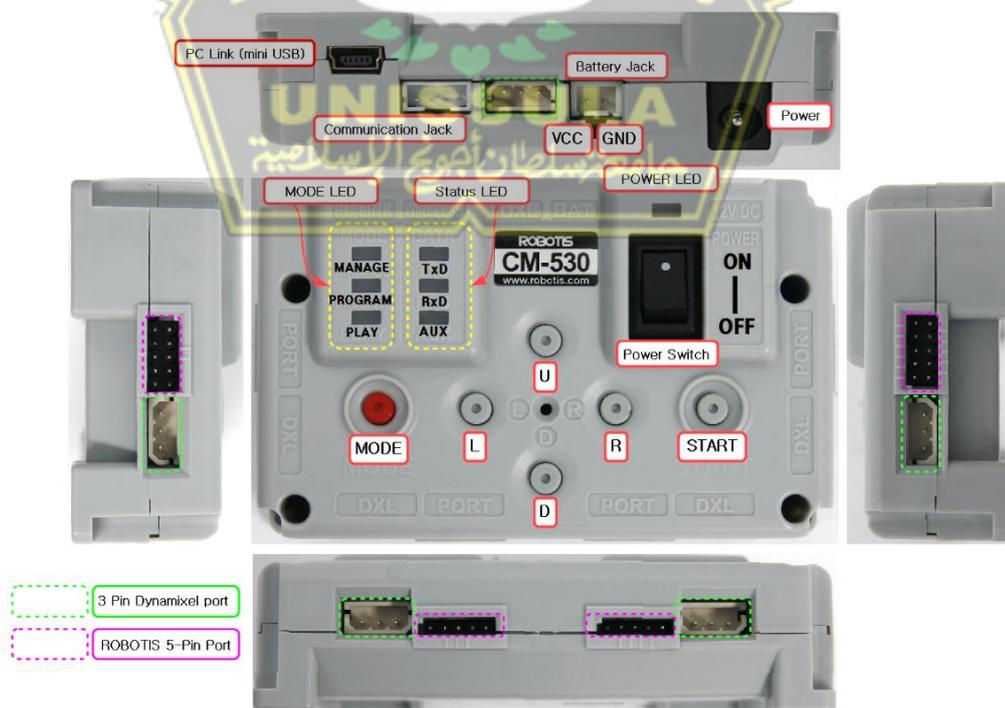
2.2.7 CM-530

CM 530 merupakan suatu kontroler robot yang dikembangkan oleh ROBOTIS yang terkandung dalam setiap pembelian Bioloid premium kit. Kontroler robot ini berfungsi selayaknya kontroler pada umumnya, tetapi kontroler ini tidak bersifat *open source* sehingga memiliki keterbatasan dalam pemrosesan data masukan yang diberikan pengguna. Spesifikasi dari kontroler ini dapat dilihat pada Tabel 2.7 [16].

Tabel 2. 7 Spesifikasi Kontroler CM-530

Keterangan	Spesifikasi
CPU	STM32F103RE
Sumber tegangan	6V~15V Direkomendasikan 11.1V (3S1P Li-Po)
Berat	54 g
Suhu operasi	-5°C~70°C
Perangkat internal I/O	Tombol x 5 (Reset 1, Port 5) MIC untuk deteksi suara x 1 Sensor Tegangan x 1
Perangkat eksternal I/O	Port ROBOTIS 5-Pin x 6 Konektor DYNAMIXEL Seri AX/MX x 5

Selain spesifikasi tersebut, terdapat pula bagian-bagian dari CM 530 yang ditunjukkan oleh Gambar 2.8. Adapun fungsi-fungsi dari bagian CM yang nantinya dijelaskan pada Tabel 2.8. Pada Gambar 2.8 terdapat nama-nama bagian yang nanti akan dijelaskan pada Tabel 2.8. Komponen-komponennya yaitu *PC Link (mini USB)*, *battery jack*, *communication jack*, *VCC*, *GND*, *power*, *mode LED*, *status LED*, *power LED*, *MODE*, *L*, *U*, *D*, *R*, *START*, *3 Pin dynamixel* dan *Robotis 5-Pin Port* [16].



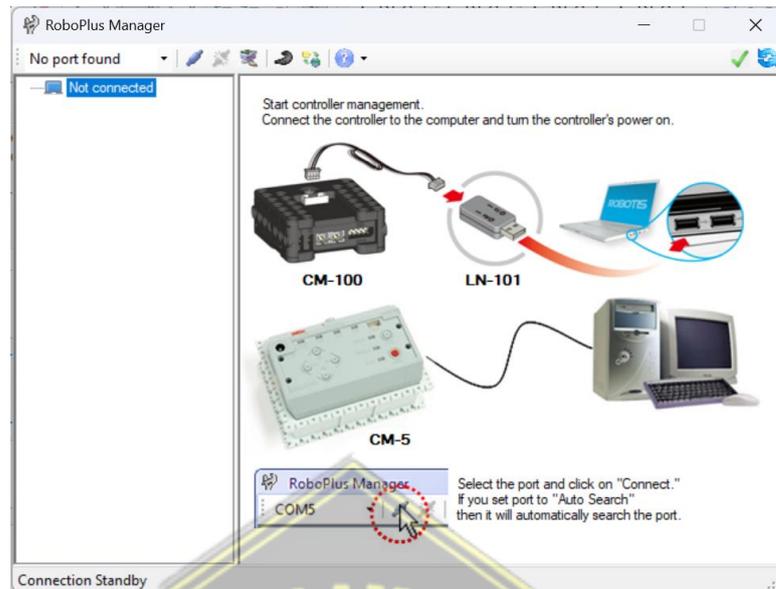
Gambar 2. 8 Bagian-bagian CM530 [16]

Tabel 2. 8 Bagian-bagian CM 530 beserta fungsinya

No	Keterangan	Fungsi
1	<i>PC Link (Serial Cable)</i>	Digunakan untuk menghubungkan kabel serial ke CM-530 dan PC melalui USB. Digunakan untuk komunikasi dengan PC lain atau mengunduh kode tugas.
2	<i>Communication Device Connection Jack</i>	Digunakan untuk komunikasi nirkabel dengan ZIG-110A, BT-110A, modul penerima IR atau papan lainnya.
3	<i>Battery Jack</i>	Digunakan untuk menghubungkan dengan baterai.
4	<i>Power Jack</i>	Digunakan untuk menghubungkan catu daya SMPS.
5	<i>Power LED</i>	Status LED ON dan OFF untuk daya
6	<i>Power Switch</i>	Digunakan untuk menghidupkan / mematikan robot.
7	<i>MODE Button</i>	Digunakan untuk mengubah mode operasi CM-530.
8	<i>START Button</i>	Digunakan untuk MULAI mode yang dipilih.
9	<i>U / L / D / R Button</i>	Digunakan untuk tujuan input saat program diputar. Tombol-tombol ini dapat digunakan untuk mengirim perintah ke robot.
10	<i>AX/MX Serise Bus Port</i>	Digunakan untuk menghubungkan AX/MX DYNAMIXEL dalam metode daisy chain.
11	<i>ROBOTIS 5-Pin Port</i>	Digunakan untuk menghubungkan Sensor Pengukuran Jarak, Sensor Sentuh, Sensor IR, dan perangkat periferal. Nomor port untuk setiap port direpresentasikan dalam bar seperti I, II, III, IIII, IIII, dan IIIII.
12	<i>Mode Display LED</i>	LED untuk menampilkan mode operasi CM-530 saat ini.

2.2.8 RoboPlus

Roboplus merupakan *software* dari robotis yang berfungsi untuk memprogram CM-530. Roboplus merupakan gabungan dari 3 *software* yaitu RoboPlus Task, RoboPlus Motion dan RoboPlus Manager yang masing-masing mempunyai fungsi yang berbeda-beda [17]. *Software* yang sering digunakan dalam pengaturan piranti-piranti robot yaitu menggunakan *software* RoboPlus manager. RoboPlus manager merupakan salah satu *software* dari roboplus yang berfungsi untuk mengatur piranti-piranti yang tersambung dengan CM-530. Tampilan *software* roboplus manager dapat dilihat pada Gambar 2.9.



Gambar 2. 9 Tampilan RoboPlus Manager



BAB III

METODE PENELITIAN

Pada bab ini mempunyai tahapan metode penelitian yaitu waktu dan tempat penelitian, metodologi penelitian, alat dan bahan, rancangan alat, pemrograman alat dan prosedur penelitian. Prosedur penelitian melakukan beberapa pengujian supaya mengetahui kinerja dari Rancang Bangun Robot *Humanoid* Penari untuk Berjalan Pada Lantai Berundak TIM SAROSETA. Berikut penjelasan lebih dalam mengenai bab ini.

3.1 Waktu dan Penelitian

Pada waktu penelitian ini dilaksanakan pada bulan Desember 2022 dengan objek perancangan sistem gerak berjalan pada undakan robot *humanoid* penari. Perancangan sistem pemrograman ini menggunakan pengontrol dari OpenCM9.04 dan OpenCM 485 yang dapat mengontrol atau memprogram sistem gerak berjalan pada robot yang dapat melalui undakan sehingga robot dapat melewati jalan yang memiliki ketinggian yang berbeda.

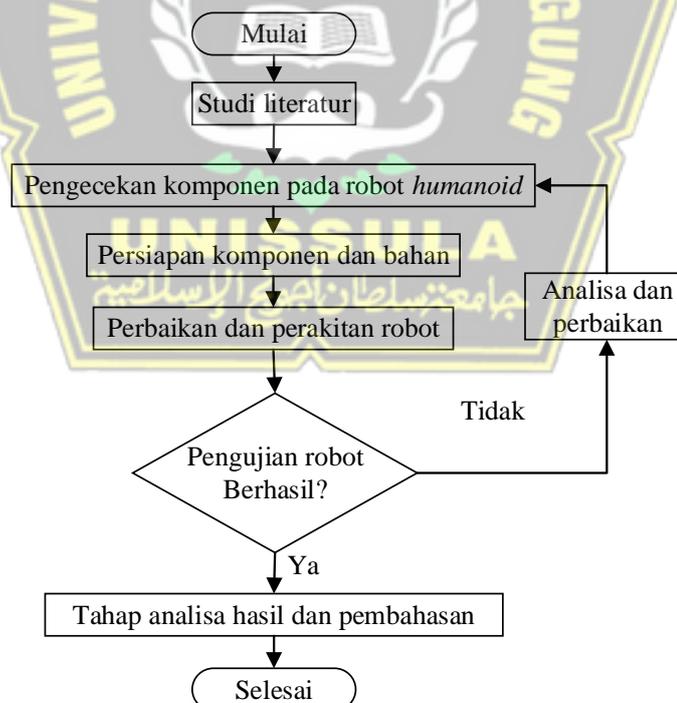
3.2 Metode Penelitian

Pada metode penelitian merupakan salah satu syarat dalam mendapatkan studi-studi yang dilakukan sehingga penelitian tersebut dapat diselesaikan. Terdapat metodologi penelitian pada perancangan penelitian ini adalah sebagai berikut:

1. Mempelajari konsep, ilmu dan teori-teori yang mengangkut tentang penelitian Rancang Bangun Robot *Humanoid* Penari untuk Berjalan pada Lantai Berundak yang dimana untuk motor servonya menggunakan Dynamixel MX-28T yang dapat mendukung penelitian ini. Penelitian ini diprogram dengan menggunakan *software* ROBOTIS. Sumber referensi dari penelitian ini dapat diambil dari jurnal ilmiah, website, paper maupun laporan penelitian.

2. Mempersiapkan untuk pengecekan komponen-komponen pada robot *humanoid* dikarenakan menggunakan robot yang sudah ada guna mendukung penelitian pada tugas akhir ini.
3. Tahap perbaikan dan perakitan yang dimana dimulai dari membuat mekanik untuk kerja, elektronik, dan program.
4. Tahap pengujian robot yaitu tahapan untuk mengetahui supaya nantinya alat ini dapat bekerja sebagaimana mestinya. Apabila “Ya” berarti robot sudah bekerja dengan baik dan apabila “Tidak” berarti robot belum bekerja dengan baik. Apabila robot belum bekerja dengan baik maka akan dilakukan analisa dan perbaikan lagi yang dimulai dari tahap pengecekan komponen pada robot *humanoid*.
5. Pengolahan data dan menganalisa hasil dari pengujian apakah dapat sesuai dengan hipotesa, serta apakah teknologi dapat dikembangkan.

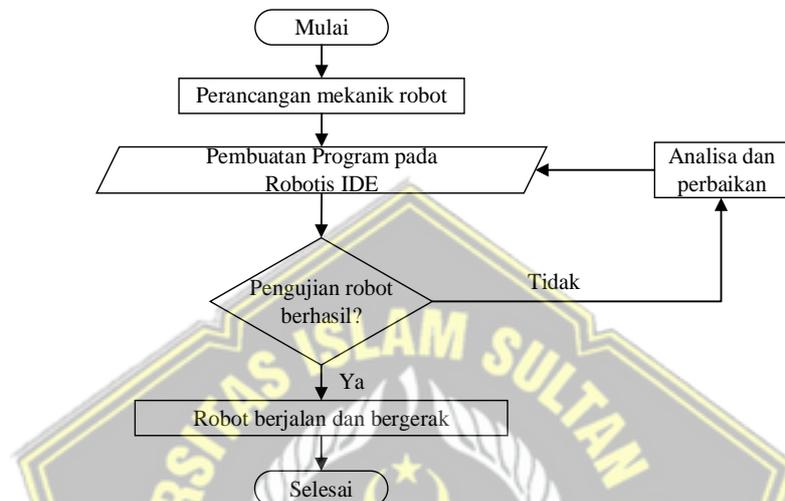
Untuk lebih jelasnya dapat dilihat pada Gambar 3.1 keterangan dari diagram alir penelitian



Gambar 3. 1 Diagram Alir Penelitian

3.3 Flowchart Sistem

Flowchart menjelaskan tentang alur kerja dari keseluruhan sistem yang akan dirancang, berikut ini merupakan penjelasan *flowchart* dari Perancangan Sistem Gerak Berjalan Robot *Humanoid* Penari pada Jalan yang Berundak. *Flowchart* sistem terlihat pada Gambar 3.2.



Gambar 3. 2 *Flowchart* Perancangan Sistem Gerak dan Berjalan Robot *Humanoid* Penari

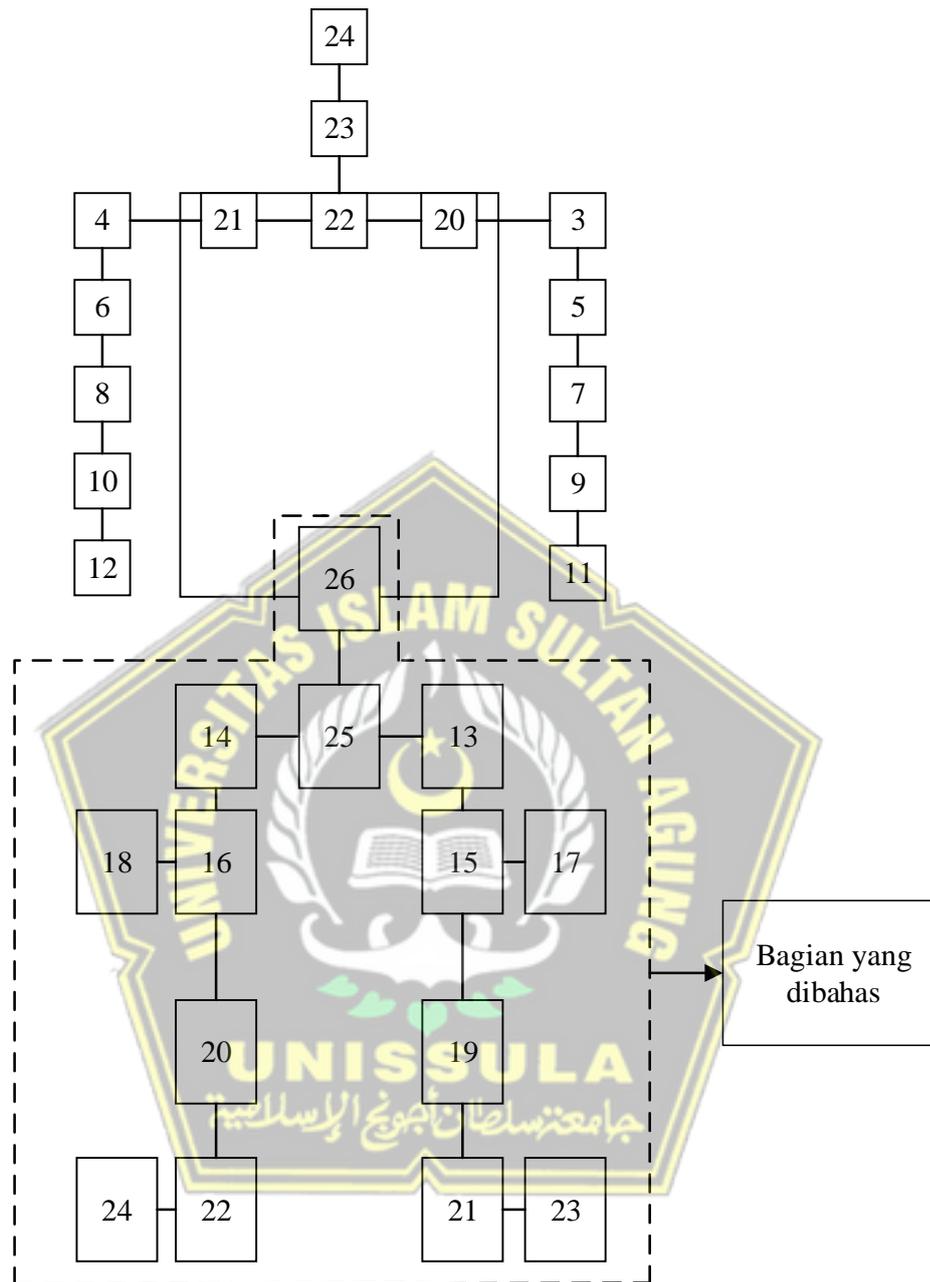
Penjelasan *flowchart* dimulai dari perancangan mekanik robot dimana pada mekanik robot ini terdapat servo Dynamixel MX-28T yang berjumlah 14 buah dan motor servo Dynamixel XL-320 yang berjumlah 13 buah. Pembuatan program pada software Robotis yang akan diuploadkan pada mikrokontroler OpenCM9.04 tipe C. OpenCM9.04 tipe C yang terhubung dengan OpenCM485 digunakan untuk inialisasi program pada servo Dynamixel MX-28T pada bagian program kaki dan perut pada robot *humanoid*. Langkah selanjutnya yaitu dilakukan pengujian robot apakah robot dapat berjalan dan bergerak? Apabila “Ya” robot akan berjalan dan bergerak sesuai dengan keinginan yang tertuang pada inialisasi program yang sebelumnya dibuat, sedangkan apabila “Tidak” sesuai maka akan dilakukan analisa dan perbaikan pemrograman dengan kembali pada tahap inialisasi program pada OpenCM9.04.

3.4 Perancangan Alat

Pada tahap perancangan alat ini akan menjelaskan mengenai perancangan alat mulai dari mekanik robot, perancangan *hardware* dan perancangan *software*. Perancangan pada alat ini dibuat secara bertahap dimulai dari desain mekanik robot, lalu dilanjutkan untuk tahapan perancangan *hardware*, selanjutnya untuk tahapan akhir yaitu perancangan *software*.

3.4.1 Perancangan Mekanik

Perancangan mekanik pada robot *humanoid* penari ini didesain seperti dengan manusia yang pada umumnya yang dapat melakukan gerakan tari. Robot yang digunakan menggunakan 2 jenis motor servo. Pada kerangka tubuh, tangan dan juga kepala menggunakan motor servo Dynamixel XL-320 dengan jumlah 13 buah motor servo. Pada bagian kedua kaki dan bagian perut menggunakan jenis servo Dynamixel MX-28T dengan jumlah 14 buah motor servo sedangkan untuk kerangkanya sendiri menggunakan alumunium. Jadi, total servo yang digunakan pada satu robot *humanoid* penari ini berjumlah 27 motor servo. Pemilihan motor servo *Dynamixel MX-28T* untuk bagian kaki robot *humanoid* yaitu untuk mendapatkan torsi yang tinggi dan akurat pada output gear. Pada motor servo ini telah menggunakan gear yang terbuat dari *full metal*, dan pada servo ini dapat bergerak sampai 360° yang dapat berputar searah jarum jam dan berlawanan arah jarum jam (CW dan CCW). Servo MX-28T dipilih untuk bagian kaki dikarenakan spesifikasi dari torsi dan juga gear rasio yang sesuai untuk kebutuhan gerak kaki. Pada Kontes Robot Seni Tari Indonesia robot *humanoid* (KRSTI) robot akan menampilkan gerakan menaiki undakan dengan ketinggian yang berbeda. Desain untuk perancangan mekanik dapat dilihat pada Gambar 3.3. Kemudian, untuk penjelasannya dapat dilihat pada Tabel 3.1. Kerangka pada bagian robot ini terdapat bagian bagian seperti motor servo Dynamixel XL-320 dan MX-28T, pada kerangka bagian tubuh yaitu terdapat penyimpanan baterai dengan tegangan 8V dan 12V, openCM9.04 dan 485 sebagai kontroler robot, dan juga terdapat tempat sistem komunikasi untuk mendeteksi suara.



Gambar 3.3 Desain Perancangan Sistem Mekanik Robot *Humanoid* Penari

Keterangan Gambar 3.3:

 = Motor Servo Dynamixel MX-28T

 = Motor Servo Dynamixel XL-320

Tabel 3. 1 Keterangan SET ID Robot Humanoid

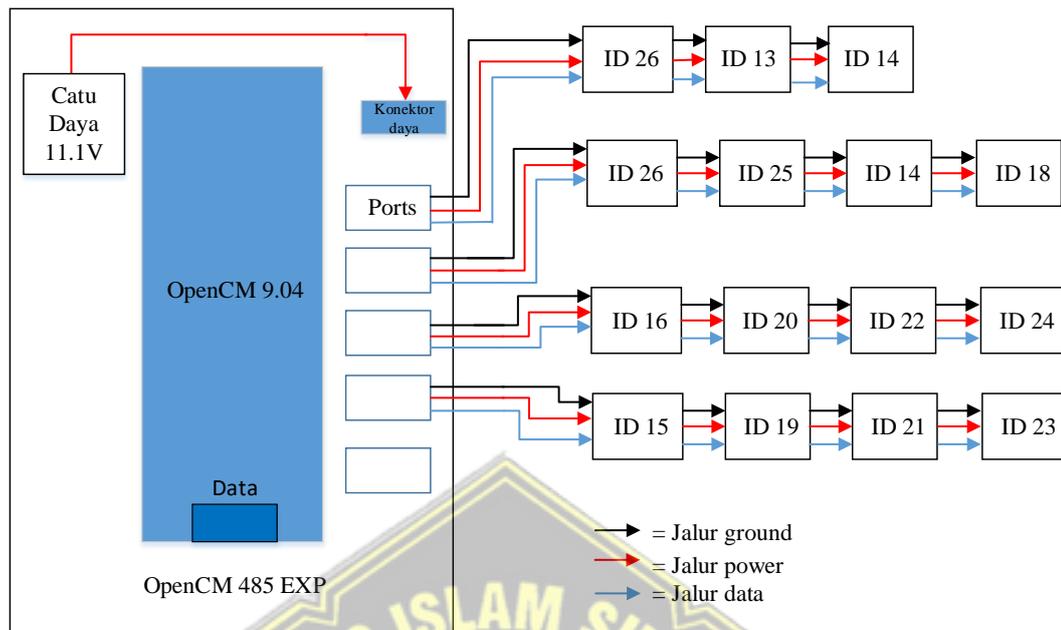
No	ID	Keterangan
Motor Servo Dynamixel XL-320		
1	24	Kepala
2	23	Leher
3	22	Pundak bagian tengah
4	21	Pundak kanan <i>pitch</i>
5	20	Pundak kiri <i>pitch</i>
6	3	Pundak kiri <i>roll</i>
7	4	Pundak kanan <i>roll</i>
8	5	Lengan atas kiri
9	6	Lengan atas kanan
10	7	Lengan bawah kiri
11	8	Lengan bawah kanan
12	9	Pergelangan tangan kiri
13	10	Pergelangan tangan kanan
Motor Servo Dynamixel MX-28T		
14	13	Pinggul kiri
15	14	Pinggul kanan
16	15	Pangkal paha kiri <i>pitch</i>
17	16	Pangkal paha kanan <i>pitch</i>
18	17	Pangkal paha kiri <i>roll</i>
19	18	Pangkal paha kanan <i>roll</i>
20	19	Lutut kiri
21	20	Lutut kanan
22	21	Pergelangan kaki kiri <i>pitch</i>
23	22	Pergelangan kaki kanan <i>pitch</i>
24	23	Pergelangan kaki kiri <i>roll</i>
25	24	Pergelangan kaki kanan <i>roll</i>
26	25	Pinggul bagian tengah
27	26	Perut

Pada Tabel 3.1 terdapat istilah *pitch* dan *roll*. Istilah tersebut telah umum digunakan dalam konteks mekanik, yang khususnya pada kendali, termasuk servo motor. Istilah *pitch* ini merupakan pergerakan yang mengacu pada objek di sekitar sumbu lateral (sumbu x). Istilah tersebut akan mengubah posisi di sekitar sumbu x, untuk mengangkat dan menurunkan motor servo. Istilah yang kedua yaitu *roll* yang

mengacu pada gerakan objek di sekitar sumbu longitudinal (sumbu y), untuk menggerakkan servo ke kanan dan ke kiri.

3.4.2 Perancangan Hardware

Pada perancangan sistem hardware pengecekan servo Dynamixel MX-28T harus dilakukan guna mengetahui apakah kondisi servo masih dapat berfungsi dengan layak ataupun tidak. Apabila kondisinya layak untuk digunakan maka servo tersebut akan digunakan dalam pembuatan robot *humanoid*. Perancangan hardware pada robot penari *humanoid* ini yaitu dengan menggunakan dua buah catu daya dengan baterai lipo 3S 11,1V 2200Mah. Baterai tersebut digunakan untuk mensuplay motor servo Dynamixel MX-28T yang berjumlah 14 buah motor servo. Terdapat penghubung antara motor servo dengan *board* kontroler yaitu dengan kabel yang memiliki 3 saluran yaitu ground, daya, dan data. Untuk lebih jelasnya terlihat pada Gambar 3.4 yaitu diagram block perancangan sistem *hardware*. Catu daya 11,1V akan mensuplay pada servo melalui openCM 485 *exp* dan openCM 9.04, seperti yang dijelaskan pada dasar teori mengenai OpenCM 485 *exp* sendiri tidak dapat beroperasi tanpa adanya dukungan atau bantuan dari openCM 9.04 hal tersebut dikarenakan openCM 485 merupakan *ekspansion board* sebagai pengontrol servo. Sebelumnya openCM 9.04 harus terdapat program di dalamnya untuk dapat menggerakkan motor servo dynamixel MX-28T.

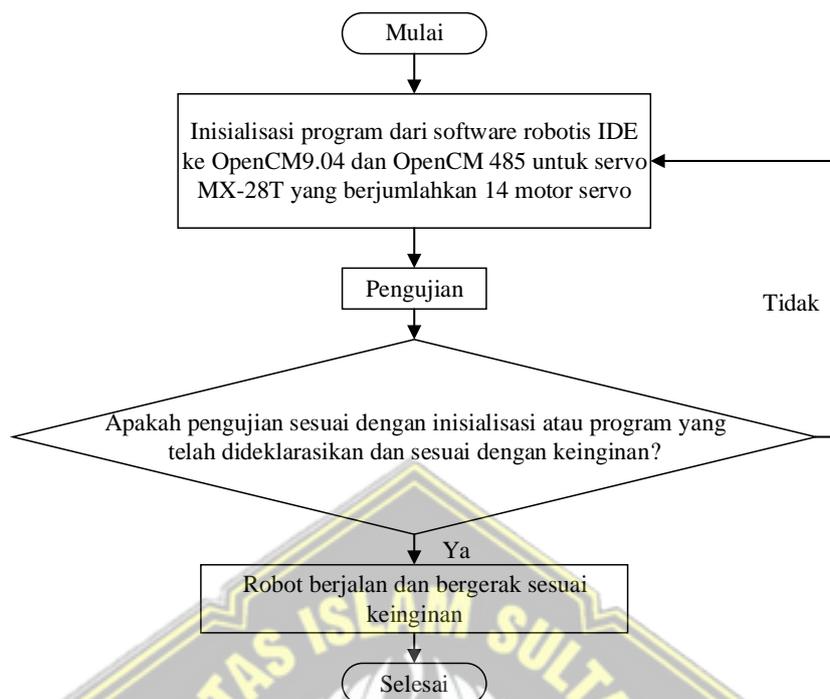


Gambar 3. 4 Diagram Block Perancangan Sistem Hardware dengan Catu Daya 11,1 volt

3.4.3 Perancangan Software

OpenCM 9.04 dan OpenCM 485 merupakan mengatur pergerakan pada robot dengan memberikan perintah berupa posisi servo yang diinginkan dan kecepatan servo. Gerakan yang dihasilkan oleh robot *humanoid* ini semuanya telah terprogram. Robot *humanoid* ini memiliki 3 gerakan yaitu posisi *stay* atau berdiri, jongkok, dan berjalan. Gerakan yang dihasilkan ini diperoleh dengan mengatur pergerakan dari motor servo Dynamixel MX-28T yang berjumlah 14 buah servo yang terdapat pada robot.

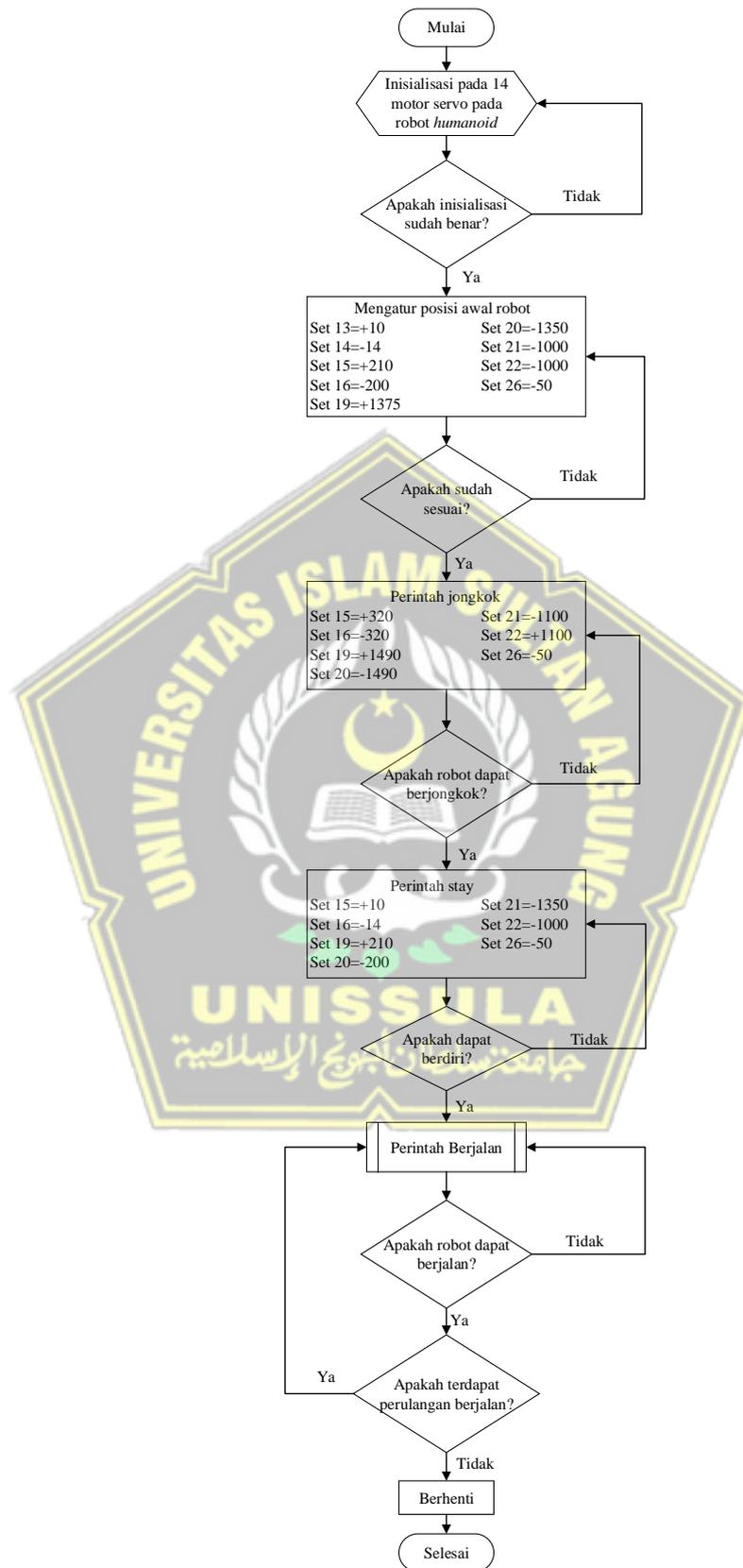
Perancangan perangkat lunak atau biasa disebut dengan perancangan *software* yaitu inialisasi (pemberian nilai awal pada proses pemrograman) program robot *humanoid* yang menggunakan *software* ROBOTIS dengan pemrograman Bahasa C. Untuk lebih jelasnya dapat dilihat pada Gambar 3.5 yaitu *flowcart* perancangan sistem *software*.



Gambar 3.5 Flowcart Perancangan Sistem *Software*

1. Algoritma Gerak Robot

Pada algoritma gerak proses awal yang dilakukan yaitu dengan memberikan inialisasi port pada mikrokontroler lalu dilanjutkan dengan proses pengaturan posisi robot dalam keadaan jongkok terlebih dahulu, kemudian *stay* berdiri, lalu dilanjut dengan berjalan. Pengaturan posisi tersebut diatur dengan menggunakan nilai *goal position* untuk dapat menggerakkan motor servo dengan memasukkan nilai yang berbeda pada setiap servonya untuk menghasilkan gerakan yang diinginkan. Batas *goal position* pada motor servo dynamixel MX-28T yaitu 2048. Setiap gerakan yang dihasilkan oleh robot *humanoid* ini menggunakan nilai pergeseran servo yang berbeda-beda. Gambar 3.6 di bawah ini merupakan gambaran yang menunjukkan program utama proses mikrokontroler secara keseluruhan. Selain gambaran tersebut, juga terdapat keterangan desain gerak motor servo dengan sudut, yang terdapat pada lampiran 3.



Gambar 3. 6 Diagram Alir dari Program Utama

Untuk perancangan *software* algoritma gerak robot *humanoid* ini membutuhkan kontroler yaitu OpenCM 485 yang dimana terhubung dengan pengendali openCM 9.04. Dibutuhkan bagian awal untuk pendeklarasian guna untuk menghasilkan algoritma gerak yang diinginkan untuk gerak robot. Berikut ini adalah algoritma pemrograman pada OpenCM 9.04.

```

unsigned int x ;
word GoalPos[14], PrevGoalPos[14];
byte id[14];
word wGoalPos[14];
word SyncPage1[28]=
{
  13,0, 14,0,
  15,0, 16,0,
  17,0, 18,0,
  19,0, 20,0,
  21,0, 22,0,
  23,0, 24,0,
  25,0, 26,0
};

```

Pada program diatas terdapat deklarasi variable dan juga deklarasi array. Deklarasi variable pada program ini yaitu pada ‘**unsigned int x**’ untuk mendeklarasikan variabel x sebagai bilangan bulat positif tanpa tanda. Sedangkan deklarasi array merupakan perkumpulan elemen dengan tipe data yang sama. Pada program ini terdapat beberapa deklarasi array yaitu pada ‘**word GoalPos[14], PrevGoalPos[14], byte id[14], word wGoalPos[14]**’ merupakan deklarasi yang memiliki ukuran 14 elemen, dimana pada array tersebut dapat menampung 14 elemen yaitu dari 0 hingga 13 . sedangkan pada ‘**word SyncPage1[28]**’ merupakan array yang dapat menampung 28 elemen yang dimana elemen tersebut yaitu 0 sampai 27 seperti yang terlihat pada program di atas.

```

Dynamixel Dxl(3);

void setup() {
  int i = 0;

  Dxl.begin(3);
  Dxl.writeWord(13, 30, 2048+10);
  Dxl.writeWord(14, 30, 2048-18);
  Dxl.writeWord(15, 30, 2048+210);
  Dxl.writeWord(16, 30, 2048-200);

```

```

Dxl.writeWord(17, 30, 2048);
Dxl.writeWord(18, 30, 2048);
Dxl.writeWord(19, 30, 2048+1375);
Dxl.writeWord(20, 30, 2048-1350);
Dxl.writeWord(21, 30, 2048-1000);
Dxl.writeWord(22, 30, 2048+1000);
Dxl.writeWord(23, 30, 2048);
Dxl.writeWord(24, 30, 2048);
Dxl.writeWord(25, 30, 2048);
Dxl.writeWord(26, 30, 2048-50);

```

```

wGoalPos[0] = 2048+10 ;
wGoalPos[1] = 2048-18 ;
wGoalPos[2] = 2048+210 ;
wGoalPos[3] = 2048-200 ;
wGoalPos[4] = 2048 ;
wGoalPos[5] = 2048 ;
wGoalPos[6] = 2048+1375 ;
wGoalPos[7] = 2048-1350 ;
wGoalPos[8] = 2048-1000 ;
wGoalPos[9] = 2048+1000 ;
wGoalPos[10] = 2048 ;
wGoalPos[11] = 2048 ;
wGoalPos[12] = 2048 ;
wGoalPos[13] = 2048-50 ;

```

```

PrevGoalPos[0] = 2048+10 ;
PrevGoalPos[1] = 2048-18 ;
PrevGoalPos[2] = 2048+210 ;
PrevGoalPos[3] = 2048-200 ;
PrevGoalPos[4] = 2048 ;
PrevGoalPos[5] = 2048 ;
PrevGoalPos[6] = 2048+1375 ;
PrevGoalPos[7] = 2048-1350 ;
PrevGoalPos[8] = 2048-1000 ;
PrevGoalPos[9] = 2048+1000 ;
PrevGoalPos[10] = 2048 ;
PrevGoalPos[11] = 2048 ;
PrevGoalPos[12] = 2048 ;
PrevGoalPos[13] = 2048-50 ;
delay(1000);

```

```

for(i = 3;i<=6;i++){
    Dxl.writeByte(i, 28, 128);
    Dxl.writeByte(i, 29, 128);
}
}

```

Pada program di atas terdapat deklarasi objek dan juga void setup. Deklarasi objek pada program ini adalah **'Dynamixel Dxl(3)'** yang akan mendeklarasikan sebuah objek yang bernama dxl atau dynamixel dengan

menggunakan konstruktor yang menerima parameter 3 yang menggunakan opencm 485 EXP. Void setup merupakan program yang akan dieksekusi hanya sekali pada awal program. Dalam void setup terdapat beberapa langkah inisialisasi diantaranya yaitu, ‘`int i = 0;`’ yaitu untuk mendeklarasikan dan juga menginisialisasi variabel ‘`i`’ yaitu bernilai 0, ‘`Dxl.begin(3);`’ yaitu untuk menginisialisasi komunikasi dengan protokol 1Mbps pada servo dynamixel, ‘`Dxl.writeWord(13,30, 2048+10)`’ yaitu untuk mengirimkan perintah ke servo dengan ID 13 menjadi 2048+10, array ‘`wGoalPos`’ diisi dengan posisi tujuan yang sesuai, array ‘`PrevGoalPos`’ untuk memperbarui letak tujuan yang sebelumnya dengan letak tujuan yang baru, dan ‘`for`’ untuk mengirimkan perintah ke beberapa servo untuk mengatur parameter dengan nilai yang telah ditentukan.

```
void MotionPagePlay(word * wGoalPos, word wTimeMill, word wPauseTime)
{
  delay(wPauseTime);
  word wNumOfStep = wTimeMill / 28;
  word wCount;
  int i = 0;
  int ii = 1;
  word wTempGoalPos[14];
  for (wCount = 0; wCount < wNumOfStep; wCount++)
  {
    for (i = 0; i<14; i++)
    {
      if(wGoalPos[i] > PrevGoalPos[i])
        GoalPos[i] = PrevGoalPos[i] + (wGoalPos[i] - PrevGoalPos[i]) * wCount /
wNumOfStep;
      else
        GoalPos[i] = PrevGoalPos[i] - (PrevGoalPos[i] - wGoalPos[i]) * wCount /
wNumOfStep;
    }
    i=0;
    for(ii = 1;ii<=29;ii+=2){
      SyncPage1[ii] = GoalPos[i];
      i++;
    }
  }
}
```

Pada program di atas merupakan fungsi ‘`void MotionPagePlay`’ yang dimana untuk mengatur pergerakan servo sesuai letak tujuan yang ditentukan. ‘`void MotionPagePlay(word * wGoalPos, word wTimeMill, word wPauseTime)`’ fungsi ini memiliki 3 parameter yaitu yang pertama berisi *goal position* untuk

servo yang dikendalikan, yang kedua yaitu untuk menentukan waktu total dalam milidetik yang digunakan untuk dapat sampai pada posisi tujuan, dan yang ketiga yaitu untuk menentukan jeda dalam milidetik sebelum pergerakan dimulai. Langkah-langkah yang terjadi pada void MotionPagePlay ini antara lain yaitu, `'delay(wPauseTime);'` merupakan waktu persiapan sebelum servo bergerak, `'word wNumOfStep = wTimeMill / 28;'` merupakan variabel untuk menyimpan jumlah pergerakan yang dilakukan berdasarkan waktu total yang dimana setiap pergerakan akan memiliki durasi 28 milidetik, `'word wCount;'` merupakan deklarasi untuk menyimpan bilangan bulat ta negatif dan digunakan untuk menghitung jumlah langkah dalam berulang, `'for (wCount = 0; wCount < wNumOfStep; wCount++)'` merupakan fungsi perulangan yang akan berjalan sesuai dengan perhitungan sebelumnya pada `'wNumOfStep'`, `'for (i = 0; i < 14; i++)'` perulangan yang akan dijalankan untuk menghitung nilai posisi dari setiap motor servo berdasarkan langkah dan letak tujuan, `'if(wGoalPos[i] > PrevGoalPos[i])'` merupakan sebuah pertanyaan apakah posisi tujuan `'wGoalPos[i]'` lebih besar dari posisi awal `'PrevGoalPos[i]'` apabila iya maka servo akan bergerak maju begitupun kebalikannya, `'for(ii = 1; ii <= 29; ii += 2)'` perulangan yang digunakan untuk mengisi array `'SyncPage1'` dengan nilai `'GoalPos'` yang sebelumnya telah dihitung.

```

Dxl.syncWrite(30,1, SyncPage1, 29);
delay(8);
}
for (i = 0; i < 14; i++)
{
  PrevGoalPos[i] = wGoalPos[i];
}
delay(100);
}

```

Penjelasan pada bagian program diatas adalah `'Dxl.syncWrite(30,1, SyncPage1, 29);'` digunakan untuk mengirimkan perintah secara sinkronis ke beberapa motor servo. Terdapat 4 parameter pada fungsi ini yaitu nilai `'30'` merupakan alamat register untuk mengatur goal position dari servo, nilai `'1'` ukuran data yang dikirimkan ke servo dengan jumlah yang dikirimnya 2-byte

data(word), 'SyncPage1' yang berisikan nilai goal position dari setiap servo yang sebelumnya telah dihitung, dan parameter '29' merupakan jumlah servo yang akan menerima perintah, dikarenakan terdapat 14 servo yang dikendalikan maka parameternya adalah 29 hasil dari $(14*2+1)$. Pada 'delay(8);' program akan menghentikan eksekusi selama 8 milidetik setelah mengirimkan perintah kepada semua servo. Pada program 'for (i = 0; i<14; i++)' merupakan perulangan yang digunakan untuk memperbarui array pada 'PrevGoalPos' dengan nilai tujuan terbaru yang disimpan pada array 'wGoalPos' sebelumnya. Terdapat 'delay(100)' program akan menghentikan eksekusi selama 100 milidetik.

```
//=====jongkok=====
void jongkok(){
  wGoalPos[0]= 2048 ;
  wGoalPos[1]= 2048 ;
  wGoalPos[2]= 2048+320 ;
  wGoalPos[3]= 2048-320 ;
  wGoalPos[4]= 2048 ;
  wGoalPos[5]= 2048 ;
  wGoalPos[6]= 2048+1490 ;
  wGoalPos[7]= 2048-1490 ;
  wGoalPos[8]= 2048-1100 ;
  wGoalPos[9]= 2048+1100 ;
  wGoalPos[10]= 2048 ;
  wGoalPos[11]= 2048 ;
  wGoalPos[12]= 2048 ;
  wGoalPos[13]= 2048-50 ;
  MotionPagePlay(wGoalPos, 3200,0);
}
//=====stay=====
void stay(){
  wGoalPos[0]= 2048 ;//13
  wGoalPos[1]= 2048 ; //14
  wGoalPos[2]= 2048+210 ;//15
  wGoalPos[3]= 2048-210 ;//16
  wGoalPos[4]= 2048 ;//17
  wGoalPos[5]= 2048 ;//18
  wGoalPos[6]= 2048+320 ;//19
  wGoalPos[7]= 2048-320 ;//20
  wGoalPos[8]= 2048-150 ;//21
  wGoalPos[9]= 2048+150 ;//22
  wGoalPos[10]= 2048 ;//23
  wGoalPos[11]= 2048 ;//24`
  wGoalPos[12]= 2048 ;
  wGoalPos[13]= 2048-50 ;
  MotionPagePlay(wGoalPos, 2000,0);
```

```

}
//=====jalan=====
void jalan_1(){
    wGoalPos[0]= 2048 ;//13
    wGoalPos[1]= 2048 ; //14
    wGoalPos[2]= 2048+90 ;//15
    wGoalPos[3]= 2048-90 ;//16
    wGoalPos[4]= 2048+250 ;//17
    wGoalPos[5]= 2048+250 ;//18
    wGoalPos[6]= 2048+450 ;//19
    wGoalPos[7]= 2048-450 ;//20
    wGoalPos[8]= 2048-300 ;//21
    wGoalPos[9]= 2048+300 ;//22
    wGoalPos[10]= 2048+270 ;//23
    wGoalPos[11]= 2048+280 ;//24
    wGoalPos[12]= 2048 ;
    wGoalPos[13]= 2048 ;
    MotionPagePlay(wGoalPos, 2500,0);
}
void jalan_2(){
    wGoalPos[0]= 2048 ;//13
    wGoalPos[1]= 2048-10 ; //14 //+20
    wGoalPos[2]= 2048+90 ;//15 90 120
    wGoalPos[3]= 2048-745 ;//16
    wGoalPos[4]= 2048+280 ;//17
    wGoalPos[5]= 2048+400 ;//18
    wGoalPos[6]= 2048+450 ;//19
    wGoalPos[7]= 2048-980 ;//20
    wGoalPos[8]= 2048-280 ;//21
    wGoalPos[9]= 2048+400 ;//22
    wGoalPos[10]= 2048+410 ;//23
    wGoalPos[11]= 2048+250 ;//24
    wGoalPos[12]= 2048 ;
    wGoalPos[13]= 2048 ;
    MotionPagePlay(wGoalPos, 2500,0);
}
void jalan_3(){
    wGoalPos[0]= 2048 ;//13
    wGoalPos[1]= 2048-10 ; //14
    wGoalPos[2]= 2048+90 ;//15
    wGoalPos[3]= 2048-480 ;//16
    wGoalPos[4]= 2048+80 ;//17
    wGoalPos[5]= 2048+430 ;//18
    wGoalPos[6]= 2048+430 ;//19
    wGoalPos[7]= 2048-540 ;//20
    wGoalPos[8]= 2048-350 ;//21
    wGoalPos[9]= 2048+10 ;//22
    wGoalPos[10]= 2048+320 ;//23
    wGoalPos[11]= 2048+350 ;//24
    wGoalPos[12]= 2048 ;
    wGoalPos[13]= 2048 ;
    MotionPagePlay(wGoalPos, 2500,0);
}
void jalan_4(){

```

```

wGoalPos[0]= 2048 ;//13
wGoalPos[1]= 2048 ; //14
wGoalPos[2]= 2048+220 ;//15
wGoalPos[3]= 2048-290 ;//16
wGoalPos[4]= 2048-50 ;//17
wGoalPos[5]= 2048-70 ;//18
wGoalPos[6]= 2048+460 ;//19
wGoalPos[7]= 2048-420 ;//20
wGoalPos[8]= 2048-400 ;//21
wGoalPos[9]= 2048+50 ;//22
wGoalPos[10]= 2048-50 ;//23
wGoalPos[11]= 2048-50 ;//24
wGoalPos[12]= 2048 ;
wGoalPos[13]= 2048 ;
MotionPagePlay(wGoalPos, 2500,0);
}
void jalan_5(){
wGoalPos[0]= 2048 ;//13
wGoalPos[1]= 2048; //14
wGoalPos[2]= 2048+150 ;//15
wGoalPos[3]= 2048-300 ;//16
wGoalPos[4]= 2048-320 ;//17
wGoalPos[5]= 2048-270 ;//18
wGoalPos[6]= 2048+480 ;//19
wGoalPos[7]= 2048-300 ;//20
wGoalPos[8]= 2048-420 ;//21
wGoalPos[9]= 2048+120 ;//22
wGoalPos[10]= 2048-290 ;//23
wGoalPos[11]= 2048-290 ;//24
wGoalPos[12]= 2048 ;
wGoalPos[13]= 2048 ;
MotionPagePlay(wGoalPos, 2500,0);
}
void jalan_6(){
wGoalPos[0]= 2048+50;//13
wGoalPos[1]= 2048 ; //14
wGoalPos[2]= 2048+780 ;//15
wGoalPos[3]= 2048-200 ;//16
wGoalPos[4]= 2048-450 ;//17
wGoalPos[5]= 2048-280 ;//18
wGoalPos[6]= 2048+900 ;//19
wGoalPos[7]= 2048-350 ;//20
wGoalPos[8]= 2048-330 ;//21
wGoalPos[9]= 2048+190 ;//22
wGoalPos[10]= 2048-200 ;//23
wGoalPos[11]= 2048-342 ;//24
wGoalPos[12]= 2048 ;
wGoalPos[13]= 2048 ;
MotionPagePlay(wGoalPos, 2500,0);
}
void jalan_7(){
wGoalPos[0]= 2048+50;//13
wGoalPos[1]= 2048 ; //14
wGoalPos[2]= 2048+550 ;//15

```

```

wGoalPos[3]= 2048-250 ;//16
wGoalPos[4]= 2048-420 ;//17
wGoalPos[5]= 2048-290 ;//18
wGoalPos[6]= 2048+470 ;//19
wGoalPos[7]= 2048-350 ;//20
wGoalPos[8]= 2048+100 ;//21
wGoalPos[9]= 2048+200 ;//22
wGoalPos[10]= 2048-320 ;//23
wGoalPos[11]= 2048-240 ;//24
wGoalPos[12]= 2048 ;
wGoalPos[13]= 2048 ;
MotionPagePlay(wGoalPos, 2500,0);
}
void jalan_8(){
wGoalPos[0]= 2048 ;//13
wGoalPos[1]= 2048 ; //14
wGoalPos[2]= 2048+370 ;//15
wGoalPos[3]= 2048-270 ;//16
wGoalPos[4]= 2048+60 ;//17
wGoalPos[5]= 2048+80 ;//18
wGoalPos[6]= 2048+470 ;//19
wGoalPos[7]= 2048-380 ;//20
wGoalPos[8]= 2048-90 ;//21
wGoalPos[9]= 2048+260 ;//22
wGoalPos[10]= 2048+150 ;//23
wGoalPos[11]= 2048+90 ;//24
wGoalPos[12]= 2048 ;
wGoalPos[13]= 2048 ;
MotionPagePlay(wGoalPos, 2500,0);
}
void jalan_9(){
wGoalPos[0]= 2048;//13
wGoalPos[1]= 2048 ; //14
wGoalPos[2]= 2048+100 ;//15
wGoalPos[3]= 2048-100 ;//16
wGoalPos[4]= 2048+180 ;//17
wGoalPos[5]= 2048+350 ;//18
wGoalPos[6]= 2048+380 ;//19
wGoalPos[7]= 2048-380 ;//20
wGoalPos[8]= 2048-180 ;//21
wGoalPos[9]= 2048+280 ;//22
wGoalPos[10]= 2048+320 ;//23
wGoalPos[11]= 2048+250 ;//24
wGoalPos[12]= 2048 ;
wGoalPos[13]= 2048 ;
MotionPagePlay(wGoalPos, 2500,0);
}

```

Penjelasan dari program ini terdapat beberapa blok kode yang diberikan yang dimana blok kode tersebut mengandung beberapa fungsi gerakan robot. 'wGoalPos[0]' diberikan nilai 2048, nilai tersebut merupakan *center point* pada

servo atau nilai tengah, nilai tersebut juga dapat dijumlahkan atau dikurangkan sesuai dengan posisi tujuan yang diinginkan. Terdapat fungsi `'void jongkok()'` digunakan untuk mengatur servo untuk melakukan gerakan "jongkok" pada robot, fungsi `'void stay()'` digunakan untuk mengatur servo untuk melakukan gerakan stay (diam). Void jalan yaitu untuk melakukan gerakan jalan yang terdapat beberapa 9 *step* untuk menghasilkan 2 langkah robot. `'void jalan_1()'` akan menghasilkan gerakan miring ke kiri, `'void jalan_2()'` akan menghasilkan gerakan mengangkat robot kaki kanan, `'void jalan_3()'` akan menghasilkan kaki kanan turun, `'void jalan_4()'` posisi kedua kaki stabil ditengah dengan kaki kanan berapa di depan, `'void jalan_5()'` akan menghasilkan gerakan miring ke kanan, `'void jalan_6()'` akan menghasilkan gerakan mengangkat robot kaki kiri, `'void jalan_7()'` akan menghasilkan kaki kiri turun, `'void jalan_8()'` posisi kedua kaki stabil ditengah dengan kaki kanan berapa di depan, `'void jalan_9()'` akan menghasilkan gerakan miring ke kiri. Pada `'void jalan_9()'` akan menggantikan `'void jalan_1()'` apabila dilakukan pemanggilan program yang berulang.

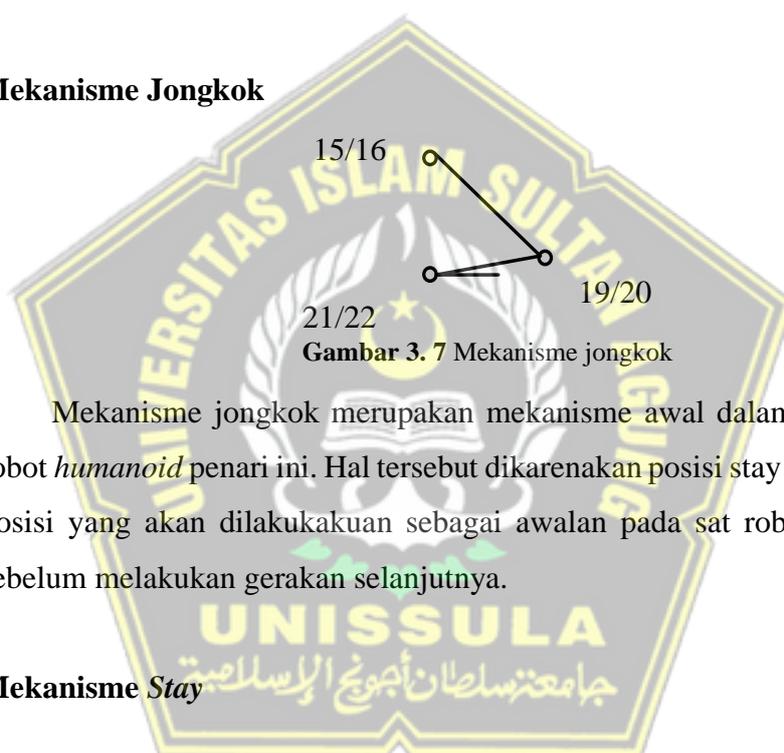
```
void loop () {
  delay(3000);
  jongkok();
  stay();
  jalan_1();
  jalan_2();
  jalan_3();
  jalan_4();
  jalan_5();
  jalan_6();
  jalan_7();
  jalan_8();
  jalan_9();

  jalan_2();
  jalan_3();
  jalan_4();
  jalan_5();
  jalan_6();
  jalan_7();
  jalan_8();

  for(;;);
}
```

Pada program ini terdapat ‘`void loop()`’ yang berisi rangkaian perintah gerak yang akan dieksekusi secara berurutan saat program dijalankan. ‘`delay(3000);`’ akan membuat penundaan 3 detik sebelum memulai gerakan. Terdapat eksekusi void berjalan yang dilakukan secara berulang sebanyak 5 kali pengulangan. ‘`for(;;);`’ merupakan program untuk menghentikan loop. Terdapat akhir dari program yaitu ‘`}`’ yang menutup blok program, ini menandakan akhir dari program dan menunjukkan bahwa tidak ada lagi kode yang dieksekusi setelah loop dihentikan.

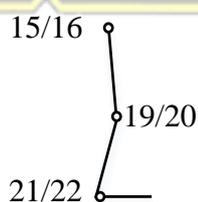
2. Mekanisme Jongkok



Gambar 3. 7 Mekanisme jongkok

Mekanisme jongkok merupakan mekanisme awal dalam penggerakan robot *humanoid* penari ini. Hal tersebut dikarenakan posisi stay ini merupakan posisi yang akan dilakukakan sebagai awalan pada saat robot dihidupkan sebelum melakukan gerakan selanjutnya.

3. Mekanisme Stay



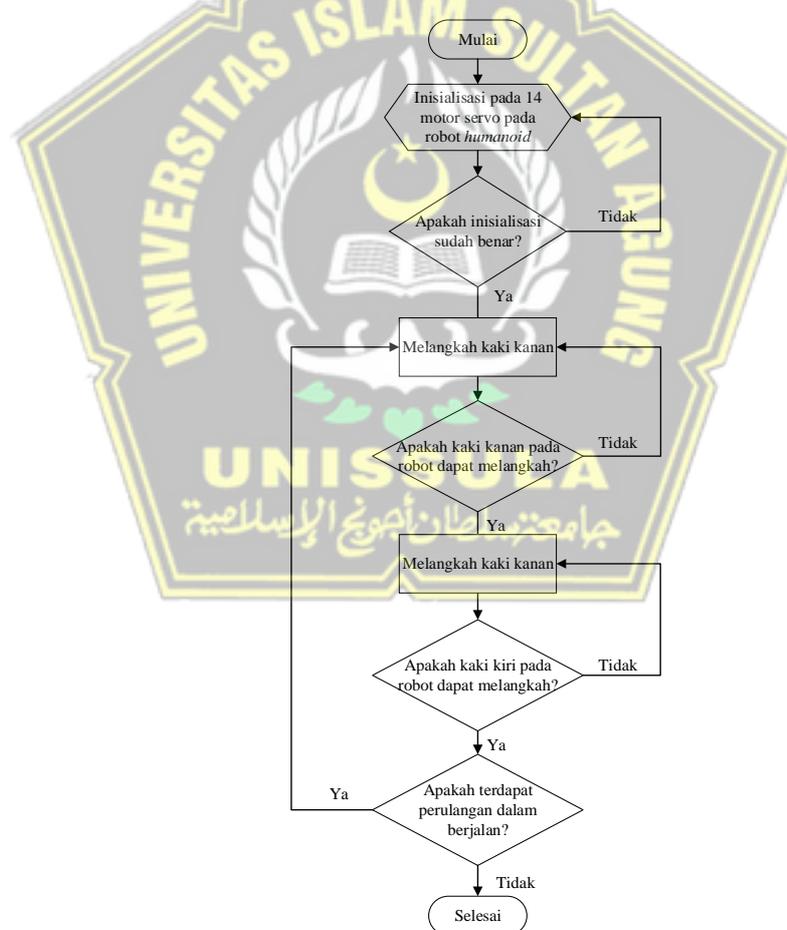
Gambar 3. 8 Bagan Mekanisme Posisi Stay

Mekanisme *stay* atau dapat dikatakan dengan berdiri merupakan salah satu mekanisme yang sangat penting dalam penggerakan robot *humanoid* penari ini. Hal tersebut dikarenakan posisi stay ini merupakan posisi yang paling ideal dalam memberikan perintah guna untuk melakukan gerakan

berjalan yang nantinya robot akan melakukan gerakan berjalan, dapat dilihat pada Gambar 3.8.

4. Mekanisme Berjalan

Pada saat posisi *stay* sudah dapat dilakukan maka setelah itu akan dilakukan mekanisme berjalan. Sebenarnya sama dengan mekanisme pada *stay* yang dimana nantinya akan dilakukan inialisasi sesuai dengan yang diinginkan. Pada mekanisme berjalan robot melakukan gerakan berjalan mengangkat kaki kanan terlebih dahulu setelah itu dilanjutkan dengan mengangkat kaki kiri. Untuk lebih jelasnya terlihat pada Gambar 3.9 yaitu diagram alir dari mekanisme program berjalan.



Gambar 3. 9 Diagram Alir Sub Program Jalan

Algoritma program pada mekanisme gerak berjalan ini dengan intrupsi void yang terdiri dari 9 macam void untuk menghasilkan satu langkah robot berjalan, yaitu sebagai berikut:

a. Mekanisme Melangkah Kaki Kanan

Pada mekanisme melangkah kaki kanan ini terdapat 4 *step* yang harus dilakukan. *Step* pada robot *humanoid* ini tentunya berbeda, yang diantaranya yaitu:

```

jalan_1();
jalan_9();
jalan_2();
jalan_3();
jalan_4();

```

Pada void “*jalan_1()*,” dan “*jalan_9()*,” akan menghasilkan gerakan robot miring ke kiri, akan menghasilkan gerakan robot miring ke kiri, “*jalan_2()*,” akan menghasilkan gerakan robot mengangkat kaki kanan, “*jalan_3()*,” akan menghasilkan gerakan kaki kanan memijak lapangan, dan “*jalan_4()*,” akan menghasilkan posisi robot stabil. Perubahan tersebut terjadi diantaranya pada paha kanan, lutut kanan, bagian pinggang, lalu pada betis kanan. Dapat dilihat pada Gambar 3.10.



Gambar 3. 10 Bagan Mekanisme Pergerakan Melangkah Kaki Kanan

b. Mekanisme Melangkah Kaki Kiri

Pada mekanisme melangkah kaki kiri ini terdapat 4 *step* yang harus dilakukan. *Step* pada robot *humanoid* ini tentunya berbeda, yang diantaranya yaitu:

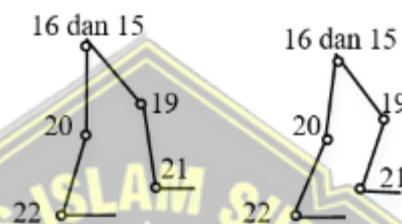
```

jalan_5();
jalan_6();
jalan_7();

```

`jalan_8()`;

Pada void `"jalan_5()"` akan menghasilkan gerakan robot miring ke kanan, `"jalan_6()"` akan menghasilkan gerakan robot kaki kiri mengangkat, `"jalan_7()"` akan menghasilkan gerakan kaki kiri memijak lapangan, dan `"jalan_8()"` akan menghasilkan posisi robot stabil. Perubahan tersebut terjadi diantaranya pada paha kiri, lutut kiri, bagian pinggang, lalu pada betis kiri. Dapat dilihat pada Gambar 3.11.



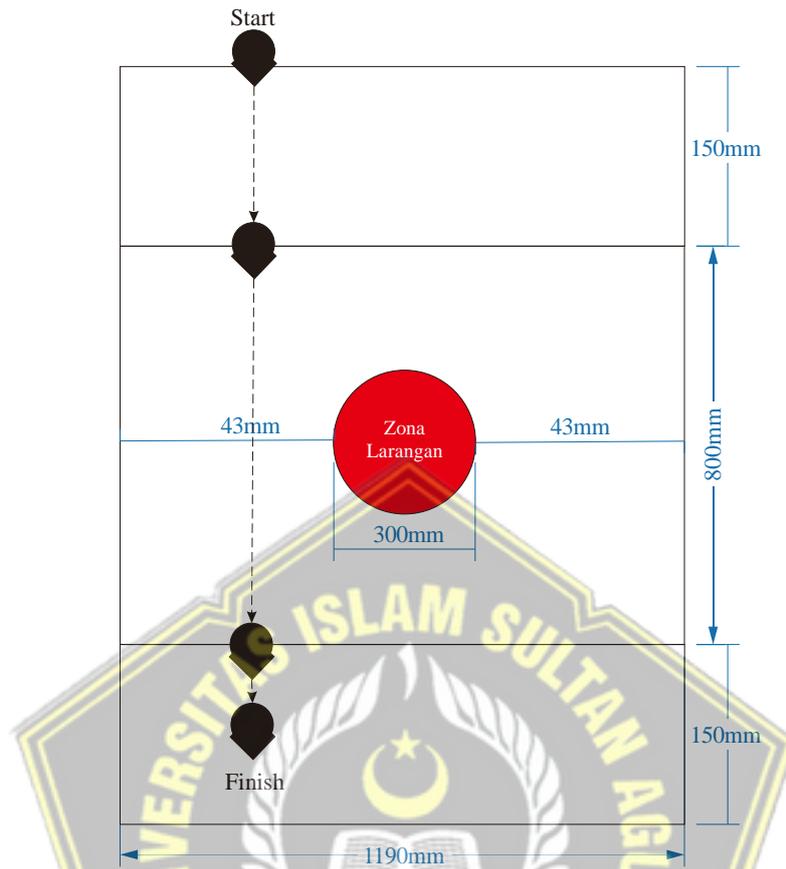
Gambar 3. 11 Bagan Mekanisme Pergerakan Melangkah Kaki Kiri

3.5 Skema Pengujian

Pada saat pengujian robot *humanoid* yang berjalan di lantai yang memiliki undakan atau dapat dikatakan memiliki tinggi yang bertingkat. Dalam pengujian ini membutuhkan skema pengujian pendukung untuk mengetahui hasil dari pengujian robot berjalan pada lantai yang berundak. Pada skema pengujian ini, data lapangan direkam berdasarkan pedoman Kontes Robot Seni Tari Indonesia (KRSTI) pada tahun 2023. Terdapat beberapa skema pengujian robot berjalan pada lantai yang berundak.

3.5.1 Berjalan pada Permukaan Berundak

Berdasarkan judul penelitian yang akan dilakukan yaitu robot akan berjalan dengan melewati permukaan yang berundak. Robot akan melewati permukaan yang berundak dengan ketinggian 4mm yaitu Zona B memiliki permukaan yang lebih tinggi dari Zona A dan Zona C. Selain itu, Zona B terdapat sebuah lingkaran di tengah dengan diameter 300mm dan ketinggian 10mm zona ini disebut Zona Larangan. Berikut Gambar 3.12 untuk gambar lapang tampak atas dan Gambar 3.13 untuk gambar lapangan tampak samping.



Gambar 3. 12 Lapangan Tampak Atas



Gambar 3. 13 Lapangan Tampak Samping

Seperti yang terlihat pada Gambar 3.12 di atas terdapat pemisalan sebuah robot *humanoid* yang nantinya akan berjalan dari permukaan bidang lalu setelah itu akan naik melewati Zona B dengan ketinggian lapangan 4 mm, tanpa melewati Zona Larangan di Zona B. Pengujian nantinya akan dilakukan 10 kali uji coba untuk mengetahui persentase keberhasilan robot. Untuk mengetahui persentase keberhasilan menggunakan persamaan (3.2) sebagai berikut:

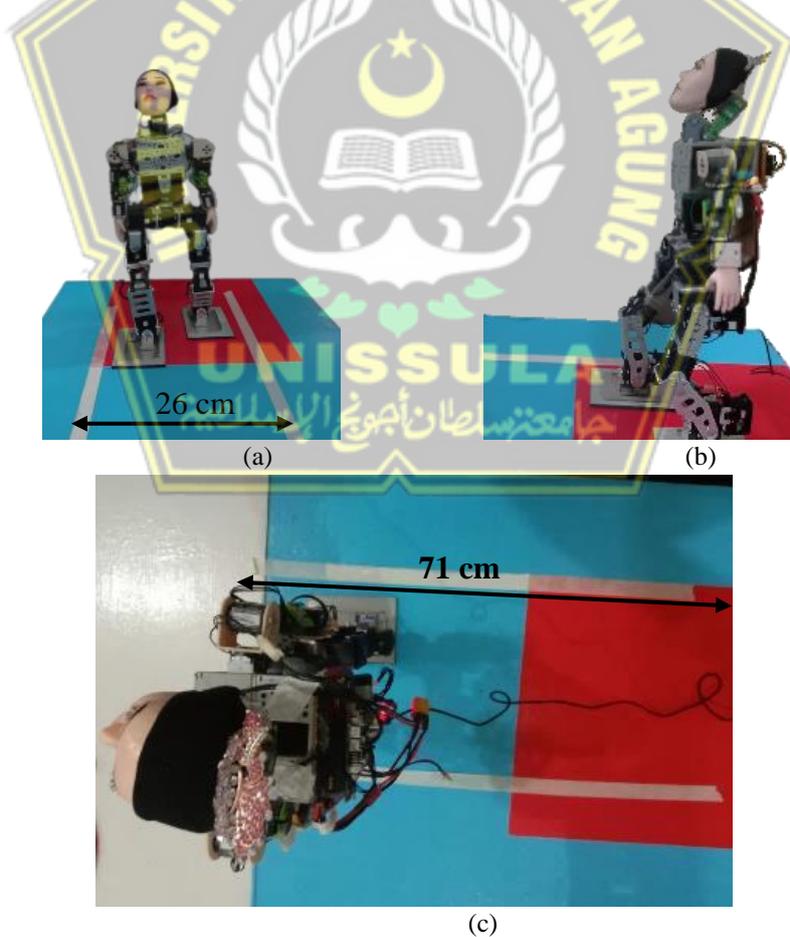
$$\text{Persentase Keberhasilan} = \frac{\text{Jumlah keberhasilan percobaan}}{\text{Jumlah percobaan}} \times 100\% \dots\dots\dots(3.4)$$

Robot berjalan pada permukaan berundak akan dilakukan pengujian diantaranya:

1. Pengujian robot dapat berjalan melewati permukaan berundak yang akan dilakukan sebanyak 10 kali percobaan.
2. Mengubah variasi delay setiap gerakan atau dapat disebut dengan mengubah *MotionPagePlay* dari yang lambat ke cepat yaitu 1800, 2200, 2500, 2800, dan 3200.

3.5.2 Berjalan pada Permukaan Bidang

Skema pengujian ini dengan menjalankan robot *humanoid* pada permukaan yang bidang. Pengujian dilakukan dengan menggunakan satu robot *humanoid*. Robot *humanoid* akan berjalan pada permukaan yang bidang sejauh 71 cm atau 5 langkah robot *humanoid* penari berjalan kaki kiri dan kanan. Pengujian akan dilakukan sebanyak 10 kali uji coba. Dapat dilihat pada Gambar 3.14 sebagai berikut:



Gambar 3. 14 Skema Pengujian pada Permukaan Bidang (a) tampak depan, (b) tampak samping, dan (c) tampak atas

Pada gambar 3.14 terlihat bahwa terdapat jarak lebar permukaan yaitu 26 cm, lebar dari robot yaitu 25 cm. Dilakukan pengukuran dari posisi tengah yang bermula pada garis *start* sampai dengan *finish*, untuk mengetahui jarak menyimpang yang dilalui oleh robot. Untuk mengetahui nilai *error* dilakukan dengan menggunakan perhitungan trigonometri [18] terdapat pada persamaan (3.1) dan (3.2) sebagai berikut:

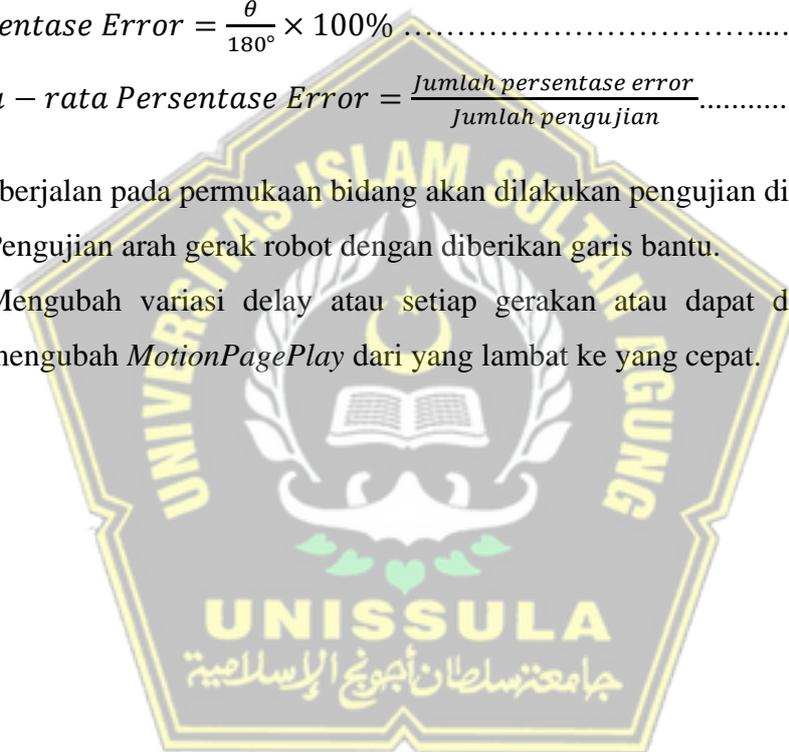
$$\theta = \tan^{-1} \left| \frac{\text{Jarak menyimpang}}{\text{Jarak tempuh}} \right| \dots\dots\dots(3.1)$$

$$\text{Persentase Error} = \frac{\theta}{180^\circ} \times 100\% \dots\dots\dots(3.2)$$

$$\text{Rata - rata Persentase Error} = \frac{\text{Jumlah persentase error}}{\text{Jumlah pengujian}} \dots\dots\dots(3.3)$$

Robot berjalan pada permukaan bidang akan dilakukan pengujian diantaranya:

1. Pengujian arah gerak robot dengan diberikan garis bantu.
2. Mengubah variasi delay atau setiap gerakan atau dapat disebut dengan mengubah *MotionPagePlay* dari yang lambat ke yang cepat.



BAB IV

DATA DAN ANALISA

Bab empat menjelaskan mengenai bentuk fisik robot dan juga membahas tentang hasil dari implementasi perancangan pergerakan kaki robot. Pada penelitian ini nantinya lebih meneliti bagaimana robot dapat berjalan pada lantai yang berundak atau memiliki ketinggian yang berbeda. Dalam bab ini nantinya akan dibagi menjadi tiga buah bagian yaitu, hasil perancangan perangkat keras, hasil perancangan perangkat lunak dan juga pengujian alat.

4.1 Perubahan Perancangan Robot

Pada penelitian ini mengalami perubahan pada perancangan robot dengan yang sebelumnya. Perubahan perancangan ini terjadi pada bagian robot tubuh robot dan juga pada bagian telapak kaki robot. Dengan adanya perubahan pada mekanik robot tersebut mempengaruhi gerakan dari robot. Pada penelitian ini yaitu menggunakan robot yang sudah jadi sebelumnya.

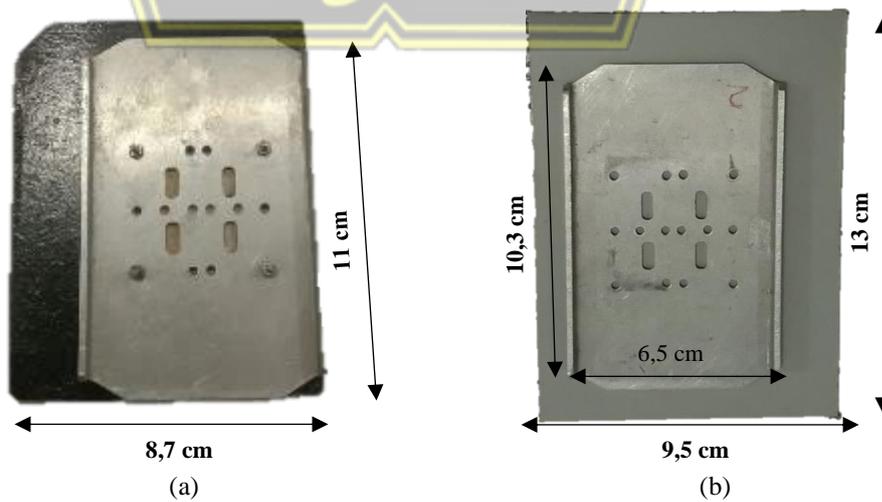
Perubahan yang terjadi pada bagian pinggung kanan dan kiri seperti yang terlihat pada Gambar 3.3 yang dimana posisi servo Dynamixel MX-28T yang awal mulanya dipasangkan secara vertikal kini telah berubah menjadi horizontal. Perubahan ini dapat dilihat pada Gambar 4.1. Dengan adanya perubahan tersebut membuat jarak antaran kaki kanan dan kiri menjadi lebih lebar apabila dibandingkan dengan pemasangan servo sebelumnya secara vertikal.

Perubahan yang kedua yaitu terjadi pada telapak kaki yang sebelumnya memiliki luas $\pm 93.5 \text{ cm}^2$ dengan menggunakan bahan telapak kaki plat alumunium. Perubahan terjadi pada luas dan juga bahan yang digunakan untuk telapak kaki yaitu *Alumunium Composite Panel (ACP)* untuk bagian bawahnya dan atasnya terdapat plat alumunium. Jika dibandingkan dengan bahan material telapak kaki yang sebelumnya dengan material yang baru ini lebih ringan dengan berat 68 gram sedangkan material plat alumunium sebelumnya memiliki berat 82 gram. Luas dari telapak kaki yang baru dibuat lebih luas yaitu 124.8 cm^2 . Dengan luas

telapak kaki yang lebih luas maka, tumpuan pada kaki robot humanoid akan lebih baik dalam menopang robot humanoid. Perubahan dapat dilihat pada Gambar 4.2.



Gambar 4. 1 Perubahan pada pinggang robot



Gambar 4. 2 Telapak kaki robot (a) sebelumnya dan (b) baru

4.2 Hasil Perancangan Perangkat Lunak

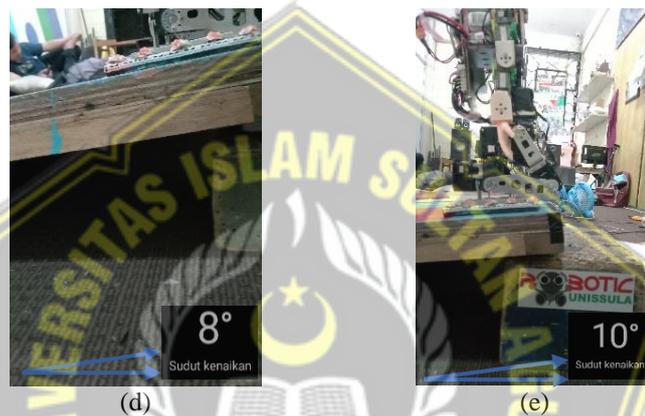
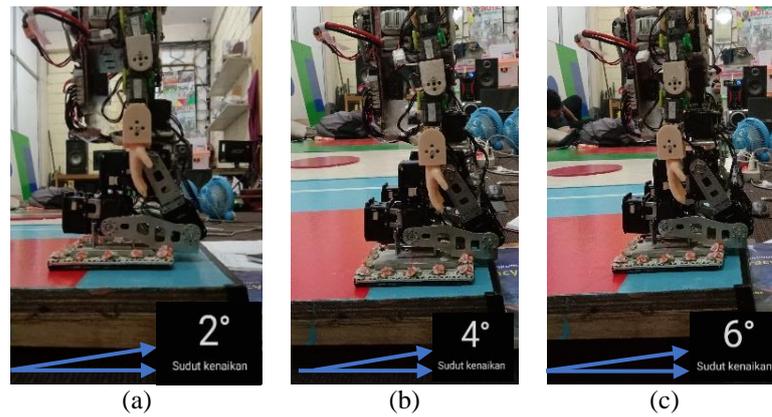
Pada hasil perancangan perangkat lunak akan melakukan *step* variasi gerakan seperti jongkok, *stay*, dan berjalan.

4.2.1 Mekanisme Jongkok



Gambar 4.3 Hasil perancangan mekanisme jongkok

Pada Gambar 4.3 merupakan hasil dari perancangan perangkat lunak dengan mekanisme jongkok. Terlihat pada gambar bahwa robot *humanoid* penari seperti sedang dalam posisi jongkok. Pada saat robot *humanoid* diberi hambatan dengan angin maupun sentuhan pada robot *humanoid*, robot tetap dapat berjongkok dengan baik. Begitupun pada saat percobaan pada permukaan dengan kemiringan 2° , 4° , 6° , 8° dan 10° robot *humanoid* masih dapat berjongkok dengan baik. Pada Gambar 4.4 merupakan hasil dari percobaan pada permukaan dengan kemiringan 2° , 4° , 6° , 8° dan 10° .



Gambar 4. 4 Hasil percobaan mekanisme jongkok dengan kemiringan (a) 2°, (b) 4°, (c) 6°, (d) 8°, dan (e) 10°

4. 2. 2 Mekanisme *Stay*

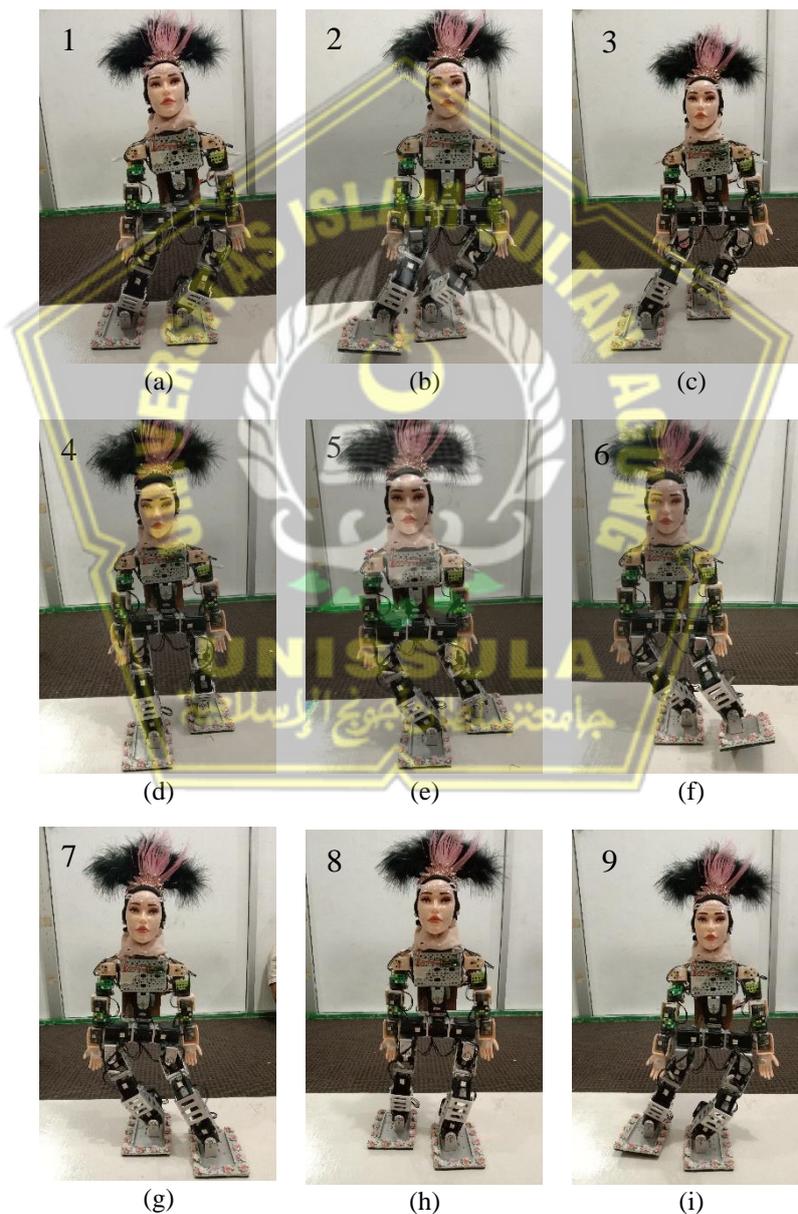


Gambar 4. 5 Mekanisme *Stay*

Pada Gambar 4.5 merupakan hasil dari perancangan perangkat lunak dengan mekanisme *stay* pada robot menghasilkan robot *humnaoid* penari berdiri

secara tegap. Pada saat pengujian mekanisme *stay* dengan rentan waktu robot berdiri selama 1 menit 54 detik robot masih dapat berdiri dengan baik. Namun, pada tumpuan lutut robot *humanoid* penari sedikit mengalami perubahan suhu. Pada saat diberi hambatan angin dan sentuhan sedikit robot *humanoid* penari masih dapat berdiri dengan baik.

4. 2. 3 Mekanisme Jalan



Gambar 4. 6 *Step* mekanisme berjalan *step* (a) pertama, (b) kedua, (c) ketiga, (d) keempat, (e) kelima, (f) keenam, (g) ketujuh, (h) kedelapan, dan (i) kesembilan

Pada Gambar 4.6 merupakan hasil dari perancangan perangkat lunak pada mekanisme berjalan dua langkah terdapat 9 *step* program. Gerakan berjalan pada robot *humanoid* penari ini langkahnya hampir menyerupai berjalannya manusia, yaitu gerakan pergantian langkah kaki kanan dan juga kaki kiri yang dilakukan secara bergiliran. Berikut ini penjelasan tiap *step* dari pergerakan berjalan pada Gambar 4.6 :

1. *Step* pertama:

Robot miring ke kiri, pada saat percobaan dengan bantuan *stopwatch* pada menit ke 1 robot tidak dapat mempertahankan gerak miring tersebut.

2. *Step* kedua:

Robot *humanoid* penari dalam posisi miring ke kiri dengan posisi kaki kanan melangkah, namun pada detik ke 23 servo tidak kuat untuk menahan beban robot jika berdiam lama.

3. *Step* ketiga:

Alas kaki kanan robot menyentuh permukaan tanah dengan jarak perpindahan 8,8 cm. Namun, pada detik ke 17 servo tidak dapat menopang berat daripada robot *humanoid* penari.

4. *Step* keempat:

Robot stabil ke tengah. Pada saat *step* stabil ini robot hanya dapat berdiam diri selama 17 detik.

5. *Step* kelima:

Step kelima sama dengan *step* pertama hanya saja pada pergerakan kaki mengalami kebalikan. Pada saat percobaan dengan bantuan dari *stopwatch* robot dapat bertahan selama 50 detik.

6. *Step* keenam:

Kaki kiri diangkat ke atas dengan badan miring ke kanan. Pada saat percobaan dengan *stopwatch* robot hanya dapat bertahan selama 35 detik.

7. *Step* ketujuh:

Gerakan robot alas kaki kiri robot menyentuh permukaan lapangan dengan posisi badan miring ke kanan. Robot dapat bertahan dalam posisi tersebut selama 1 menit dengan perpindahan jarak sejauh 15,4 cm dari posisi semula.

8. *Step* kedelapan:

Posisi stabil, robot dapat bertahan selama 1 menit masih aman.

9. *Step* kesembilan:

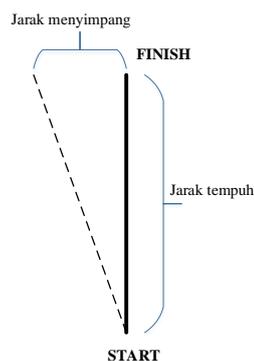
Step pengganti dari *step* pertama untuk melanjutkan langkah berikutnya pada saat robot *humanoid* berjalan. *Step* kesembilan robot dapat bertahan dalam waktu 31 detik.

4.3 Pengujian Gerakan

Pengujian dilakukan guna mengetahui keberhasilan dari pembuatan perancangan perangkat lunak. Pengujian alas robot menggunakan papan triplek dengan diberi garis penanda untuk mengukur kelurusan dari robot berjalan pada permukaan bidang.

4.3.1 Pengujian Jalan pada Permukaan Bidang

Pengujian robot berjalan pada permukaan yang bidang dilakukan dengan meletakkan robot di atas lapangan yang telah disediakan. Pada penelitian ini dalam pengambilan data menggunakan variasi delay pada setiap gerak robot. Rentan delay pada program yang dibuat akan menentukan apakah gerakan robot berhasil atau belum berhasil dalam berjalan lurus ke depan dengan menggunakan program yang telah dibuat. Pengujian delay atau *MotionPagePlay* yang akan dilakukan pengujian sebanyak 10 kali mulai dari *MotionPagePlay* 1800, 2200, 2500, 2800 dan 3200. Pengujian dilakukan dengan mengukur pergeseran posisi tengah pada robot *humanoid*. Berikut hasil dari pengujian dengan variasi delay pada robot *humanoid* penari yang berjalan pada permukaan bidang.



Gambar 4.7 Ilustrasi pengukuran menyimpang

Pada Tabel 4.1 merupakan hasil dari pengujian pergerakan robot yang berjalan pada permukaan yang datar dengan pengaturan *MotionPagePlay* yang digunakan adalah 1800. Hal tersebut berarti pergantian antar perpindahan gerak robot dengan selang waktu 1,8 detik. Pada saat pengujian sebanyak 10 kali dengan *MotionPagePlay* 1800 menghasilkan keberhasilan berjalan di dalam jalur yaitu 40%.

Tabel 4. 1 Pengujian berjalan bidang dengan *MotionPagePlay* 1800

Percobaan	Jarak menyimpang (cm)	Sudut	Persentase error
1	2,9	2,34°	1,30%
2	1,5	1,21°	0,67%
3	0,5	0,40°	0,22%
4	0,5	0,40°	0,22%
5	0,5	0,40°	0,22%
6	0,9	0,73°	0,40%
7	0,7	0,56°	0,31%
8	1,7	1,37°	0,76%
9	3,7	2,98°	1,66%
10	0,5	0,40°	0,22%
Rata-rata			0,60%

Gambar 4.7 merupakan ilustrasi dilakukannya pengukuran. Sudut didapatkan dari perhitungan persamaan 3.1, maka diperoleh sebagai berikut:

$$\theta = \tan^{-1} \left(\frac{2,9}{71} \right) \\ = 2,34^\circ$$

Maka, sudut yang diperoleh dari persamaan 3.1 yaitu 2,34°. Nilai *error* didapatkan dari perhitungan persamaan 3.1, maka diperoleh sebagai berikut:

$$\text{Persentase Error} = \frac{2,34^\circ}{180^\circ} \times 100\% \\ = 1,30\%$$

Kemudian, dilakukan perhitungan persentase error dengan menggunakan persamaan (3.2) menghasilkan 1,30%. Rata-rata persentase error dari jumlah

percobaan dapat diperhitungkan dengan persamaan (3.3), maka diperoleh sebagai berikut:

$$\begin{aligned} \text{Rata - rata Persentase Error} &= \frac{6,90\%}{10} \\ &= 0,60\% \end{aligned}$$

Sehingga, rata-rata persentase error pada percobaan dengan *MotionPagePlay* yaitu 0,69%. Dalam tabel terdapat jarak menyimpang sebesar 0,5 cm, pada jarak menyimpang tersebut posisi robot berada masih berada dalam jalur yang telah ditentukan yaitu 26 cm.

Pada Gambar 4.8 merupakan hasil percobaan pada permukaan bidang dengan *MotionPagePlay* 1800. Pada gambar tersebut terlihat bahwa kaki kiri robot telah melewati garis yang telah ditetapkan. Jadi, posisi robot agak menyimpang ke kiri atau berjalan dengan kurang lurus. Terlihat pada gambar bahwa robot *humanoid* penari melampaui garis batas yang telah dibuat, jarak pada gambar tersebut adalah 28,4 cm. Dengan keluarnya robot dari jalur, menandakan bahwa robot telah bergerak menyimpang. Terjadinya hal tersebut dikarenakan pada setiap langkah robot terjadi pergeseran dengan kecepatan 1,8 detik untuk perpindahan gerak servo dari satu gerakan ke gerakan satunya. Dengan kecepatan tersebut juga mengakibatkan servo berjalan dengan kurang seimbang dikarenakan terlalu cepat.



Gambar 4. 8 Hasil percobaan pada permukaan bidang dengan *MotionPagePlay* 1800

Pada Tabel 4.2 merupakan hasil dari pengujian pergerakan robot yang berjalan pada permukaan yang datar dengan pengaturan *MotionPagePlay* yang digunakan adalah 2200. Hal tersebut berarti pergantian antar perpindahan gerak robot dengan selang waktu 2,2 detik. Presentase rata rata *error* dari percobaan ini yaitu 0,33%. Presentase keberhasilan robot dapat melewati jalur yaitu 70%, yang

dimana pergeserannya kisaran 1 cm yang dikarenakan oleh pergeseran pada saat robot menapakkan kaki ke lapangan. Untuk dapat dikatakan berhasil yaitu apabila posisi robot masih berada dalam garis yang telah ditentukan.

Tabel 4. 2 Pengujian berjalan bidang dengan *MotionPagePlay* 2200

Percobaan	Jarak menyimpang (cm)	Sudut	Persentase error
1	0,5	0,40°	0,22%
2	0,5	0,40°	0,22%
3	0,5	0,40°	0,22%
4	1,5	1,21°	0,67%
5	0,5	0,40°	0,22%
6	0,5	0,40°	0,22%
7	0,7	0,56°	0,31%
8	0,5	0,40°	0,22%
9	0,5	0,40°	0,22%
10	1,6	1,29°	0,72%
Rata-rata			0,33%

Pada Gambar 4.9 merupakan hasil percobaan pada permukaan bidang dengan *MotionPagePlay* 2200. Pada gambar tersebut terlihat bahwa robot dapat berjalan dengan lurus berada dalam garis yang telah ditetapkan yaitu 26 cm. Posisi tubuh robot menghadap ke kedepan dengan dengan kaki kiri hampir mengenai garis. Walaupun, dalam Gambar 4.9 menunjukkan robot dapat berjalan dalam garis, tetapi terdapat penyimpangan sebesar 0,5 cm.



Gambar 4. 9 Hasil percobaan pada permukaan bidang dengan *MotionPagePlay* 2200

Pada Tabel 4.3 merupakan hasil dari pengujian pergerakan robot yang berjalan pada permukaan yang datar dengan pengaturan *MotionPagePlay* yang digunakan adalah 2500. Hal tersebut berarti pergantian antar perpindahan gerak

robot dengan selang waktu 2,5 detik. Presentase rata rata error dari percobaan ini yaitu 0,48%. Pada saat percobaan dengan delay 2,5 detik pada setiap langkah geraknya menghasilkan gerakan yang lebih pas pada saat telapak kaki menyentuh permukaan, walaupun terdapat sedikit pergeseran pada kaki robot *humanoid* penari. Pengujian sebanyak 10 kali dengan *MotionPagePlay* 2500 menghasilkan keberhasilan berjalan didalam jalur yaitu 60%.

Tabel 4. 3 Pengujian berjalan bidang dengan *MotionPagePlay* 2500

Percobaan	Jarak menyimpang (cm)	Sudut	Persentase error
1	0,5	0,40°	0,22%
2	1,6	1,29°	0,72%
3	0,5	0,40°	0,22%
4	0,5	0,40°	0,22%
5	2	1,61°	0,90%
6	0,5	0,40°	0,22%
7	0,5	0,40°	0,22%
8	1,7	1,37°	0,76%
9	0,5	0,40°	0,22%
10	2,4	1,94°	1,08%
Rata-rata			0,48%

Pada Gambar 4.10 merupakan hasil percobaan pada permukaan bidang dengan *MotionPagePlay* 2500. Pada gambar tersebut terlihat bahwa robot dapat berjalan pada garis yang telah ditetapkan. Namun, kaki kiri pada robot hampir menyentuh garis yang telah ditetapkan. Jarak keseluruhan pada gambar tersebut adalah 26 cm, dengan jarak penyimpangan sejauh 0,5 cm yang diambil dari posisi tengah.



Gambar 4. 10 Hasil percobaan pada permukaan bidang dengan *MotionPagePlay* 2500

Pada Tabel 4.4 merupakan hasil dari pengujian pergerakan robot yang berjalan pada permukaan yang datar dengan pengaturan *MotionPagePlay* yang digunakan adalah 2800. Hal tersebut berarti pergantian antar perpindahan gerak robot dengan selang waktu 2,8 detik. Presentase rata rata error dari percobaan ini yaitu 0,80%. Pada saat pengujian sebanyak 10 kali dengan *MotionPagePlay* 2800 menghasilkan keberhasilan berjalan didalam jalur sebanyak 50%. Pada saat pengujian terjadi pergeseran sedikit pada saat kaki kanan menyentuh permukaan lapangan sehingga mengakibatkan beberapa percobaan robot keluar pada garis ketentuan.

Tabel 4. 4 Pengujian berjalan bidang dengan *MotionPagePlay* 2800

Percobaan	Jarak menyimpang (cm)	Sudut	Persentase error
1	0,5	0,40°	0,22%
2	5,3	4,27	2,37%
3	4,5	3,63°	2,01%
4	0,5	0,40°	0,22%
5	2,3	1,86°	1,03%
6	0,5	0,40°	0,22%
7	0,5	0,40°	0,22%
8	2,5	2,02°	1,12%
9	0,8	0,65°	0,36%
10	0,5	0,40°	0,22%
Rata-rata			0,80%

Pada Gambar 4.11 merupakan hasil percobaan pertama pada permukaan bidang dengan *MotionPagePlay* 2800. Pada gambar tersebut terlihat bahwa robot dapat berjalan dalam garis yang telah ditetapkan. Walaupun pada kaki kanan terlihat hampir mengenai garis yang telah ditentukan. Jarak pada robot *humanoid* penari yaitu 26 cm, dengan jarak penyimpangan sejauh 0,5 cm yang diambil dari posisi tengah.



Gambar 4. 11 Hasil percobaan pada permukaan bidang dengan *MotionPagePlay* 2800

Pada Tabel 4.5 merupakan hasil dari pengujian pergerakan robot yang bejalan pada permukaan yang datar dengan pengaturan *MotionPagePlay* yang digunakan adalah 3200. Hal tersebut berarti pergantian antar perpindahan gerak robot dengan selang waktu 3,2 detik, mengakibatkan perpindahan gerak satu ke gerak satunya lambat. Presentase rata rata error dari percobaan ini yaitu 0,43%. Pengujian dilakukan sebanyak 10 kali dengan *MotionPagePlay* 3200 menghasilkan keberhasilan berjalan didalam jalur sebanyak 40%.

Tabel 4. 5 Pengujian berjalan bidang dengan *MotionPagePlay* 3200

Percobaan	Jarak menyimpang (Cm)	Sudut	Persentase error
1	1,4	1,13°	0,63%
2	1,4	1,13°	0,63%
3	0,5	0,40°	0,22%
4	0,5	0,40°	0,22%
5	1	0,81°	0,45%
6	0,5	0,40°	0,22%
7	1,4	1,13°	0,63%
8	1,9	1,53°	0,85%
9	0,5	0,40°	0,22%
10	0,5	0,40°	0,22%
Rata-rata			0,43%



Gambar 4. 12 Hasil percobaan pada permukaan bidang dengan MotionPagePlay 3200

Gambar 4.12 merupakan hasil percobaan pertama pada permukaan bidang dengan *MotionPagePlay* 3200. Pada gambar tersebut terlihat bahwa kaki robot melewati garis yang telah ditetapkan. Jadi, posisi robot agak menyimpang ke kiri dengan jarak pada gambar tersebut adalah 26,9 cm, dengan jarak penyimpangan sejauh 1,4 cm yang diambil dari posisi tengah.

4.3.2 Pengujian Jalan pada Permukaan Berundak

Pengujian robot berjalan pada permukaan yang terdapat undakan dengan ketinggian 4mm dilakukan dengan meletakkan robot di atas lapangan yang telah disediakan. Pada penelitian ini dalam pengambilan data menggunakan variasi delay pada setiap gerak robot. Rentan delay pada program yang dibuat akan menentukan apakah gerakan robot dapat melewati undakan dan menuruni undakan serta tidak menyentuh zona larangan dengan diameter 300mm. Pengujian delay atau *MotionPagePlay* yang akan dilakukan pengujian sebanyak 10 kali pengujian mulai dari *MotionPagePlay* 1800, 2200, 2500, 2800, dan 3200. Pengujian dilakukan dengan ketentuan apabila robot *humanoid* penari dapat melewati undakan dan turunan serta tidak menyentuh zona larangan maka robot *humanoid* dikatakan berhasil.

Pada Tabel 4.6 merupakan hasil dari pengujian pergerakan robot yang berjalan pada permukaan berundak dengan pengaturan *MotionPagePlay* yang digunakan adalah 1800. Hal tersebut berarti pergantian antar perpindahan gerak robot dengan selang waktu 1,8 detik. Pada percobaan tersebut robot dapat berjalan dengan benar hanya sebanyak 3 kali pengujian yang menghasilkan robot berhasil.

Dari jumlah hasil pengujian pada Tabel 4.6 maka dapat dihitung untuk persentase keberhasilan dengan menggunakan persamaan (3.4) berikut:

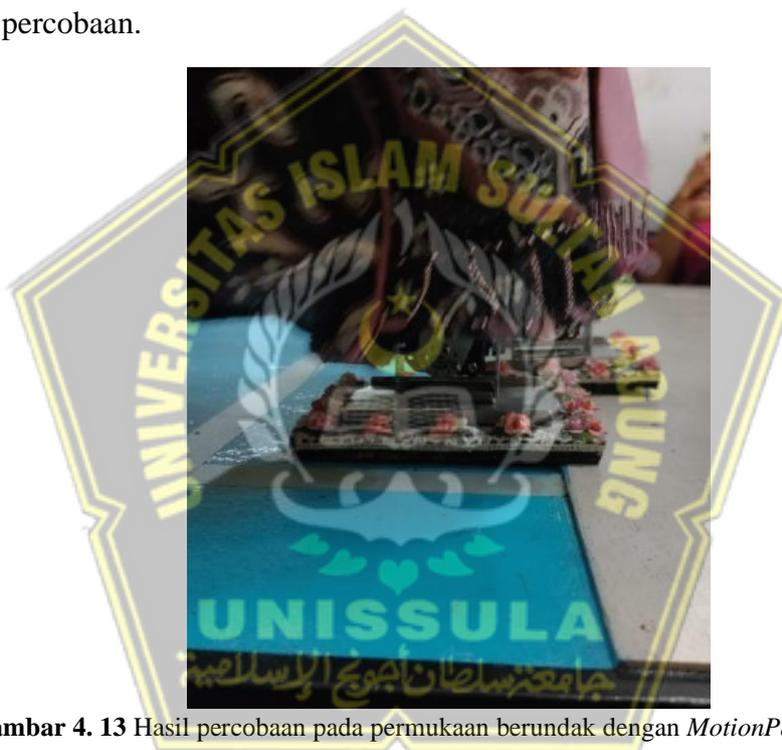
$$\begin{aligned} \text{Persentase Keberhasilan} &= \frac{3}{10} \times 100\% \\ &= 30\% \end{aligned}$$

Dari hasil perhitungan tersebut maka didapatkan jumlah presentase keberhasilan adalah 30%.

Tabel 4. 6 Pengujian berjalan pada permukaan berundak dengan *MotionPagePlay* 1800

Percobaan	Hasil Pengujian	Keterangan Hasil
1	Tidak berhasil	Pada saat menaiki undakan robot kurang seimbang tetapi tidak jatuh, pergerakan kurang seimbang dan pada saat menuruni undakan robot hampir jatuh
2	Berhasil	Robot berjalan kurang seimbang, hampir jatuh dikarenakan kecepatan servo sehingga robot juga menyimpang ke kiri
3	Tidak berhasil	Robot berjalan kurang seimbang, pada zona tengah hampir jatuh mengakibatkan robot menyimpang ke kiri
4	Tidak berhasil	Robot menyentuh zona larangan dan hampir terjatuh, lalu karena keseimbangan yang kurang ditambah dengan kecepatan robot yang begitu cepat maka pada saat turun robot hilang keseimbangan
5	Tidak berhasil	Robot tidak menyentuh zona merah dan pada saat turun robot tidak terlalu seimbang dan hampir jatuh pada saat turun
6	Berhasil	Robot dapat melewati rintangan tetapi pada zona putih robot seperti hampir jatuh karena pergantian gerak yang terlalu cepat
7	Tidak berhasil	Pada saat menaiki undakan robot kurang seimbang tetapi tidak jatuh, pergerakan kurang seimbang
8	Tidak berhasil	Robot berjalan kurang seimbang, hampir jatuh pada saat naik dan turun undakan dikarenakan kecepatan servo sehingga robot juga menyimpang ke kiri
9	Tidak berhasil	Robot berjalan kurang seimbang sehingga pada saat melewati jalan yang berundak hampir terjatuh dan di zona tengah robot sedikit tidak dapat menjaga keseimbangannya
10	Berhasil	Robot dapat naik dan turun jalan berundak tetapi kurang dapat seimbang

Pada Gambar 4.13 merupakan hasil percobaan pada permukaan berundak dengan *MotionPagePlay* 1800. Pada gambar dapat menaiki undakan tetapi selang beberapa detik robot sedikit kurang seimbang dikarenakan perpindahan gerak berikutnya yang terlalu cepat. Pada saat berjalan robot mengalami pergeseran karena perpindahan gerak robot yang terlalu cepat, sehingga mengakibatkan posisi robot yang kurang seimbang dan bergeser. Pada percobaan robot *humanoid* penari saat menaiki undakan kurang seimbang. Pada saat menuruni undakan robot hampir terjatuh. Robot benar benar berhasil pada saat melewati undakan sebanyak 3 dari 10 kali percobaan.

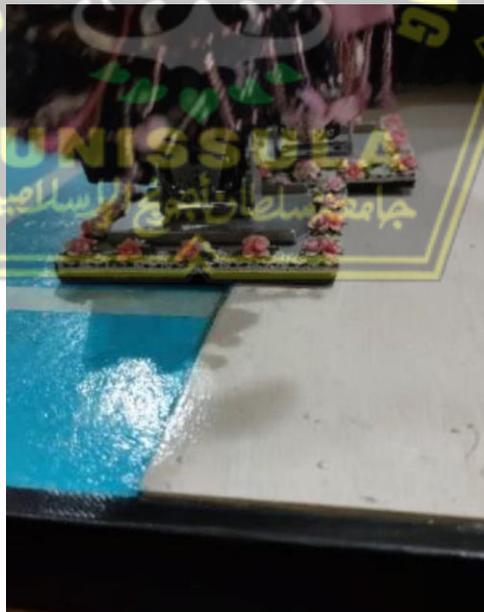


Gambar 4. 13 Hasil percobaan pada permukaan berundak dengan *MotionPagePlay* 1800

Pada Tabel 4.7 merupakan hasil dari pengujian pergerakan robot yang berjalan pada permukaan berundak dengan pengaturan *MotionPagePlay* yang digunakan adalah 2200. Pada percobaan tersebut robot dapat berjalan dengan benar sebanyak 6 kali percobaan, tetapi tidak secara berurutan. Pada saat pengujian delay pada robot 2,2 detik robot dapat berjalan pada permukaan yang berundak tetapi juga terdapat pengujian yang menampilkan bahwa robot kekurangan keseimbangan pada saat naik undakan, ditengah jalan ataupun pada saat menuruni undakan dikarenakan karena delay dari perpindahan gerak robot. Dari perhitungan persamaan (3.3) pada Tabel 4.7 maka persentase keberhasilannya yaitu 60%.

Tabel 4. 7 Pengujian berjalan pada permukaan berundak dengan *MotionPagePlay* 2200

Percobaan	Hasil Pengujian	Keterangan Hasil
1	Berhasil	Robot berjalan dengan baik pada saat naik undakan dan pada saat turun undakan
2	Berhasil	Robot berjalan dengan baik pada saat naik undakan dan pada saat turun undakan
3	Berhasil	Robot berjalan dengan baik pada saat naik undakan dan pada saat turun undakan
4	Tidak berhasil	Saat naik dan turun undakan robot hampir terjatuh
5	Tidak berhasil	Pada saat naik robot sedikit kurang seimbang, pada saat turun robot hampir jatuh
6	Berhasil	Robot berjalan dengan baik pada saat naik undakan dan pada saat turun undakan
7	Tidak berhasil	Pada saat naik undakan robot hampir terjatuh
8	Berhasil	Robot berjalan dengan baik pada saat naik undakan dan pada saat turun undakan
9	Berhasil	Robot berjalan dengan baik pada saat naik undakan dan pada saat turun undakan
10	Tidak berhasil	Robot terjatuh pada saat menuruni undakan

**Gambar 4. 14** Hasil percobaan pada permukaan berundak dengan *MotionPagePlay* 2200

Pada Gambar 4.14 merupakan hasil percobaan pada permukaan berundak dengan *MotionPageplay* 2200. Pada gambar tersebut terlihat bahwa robot masih

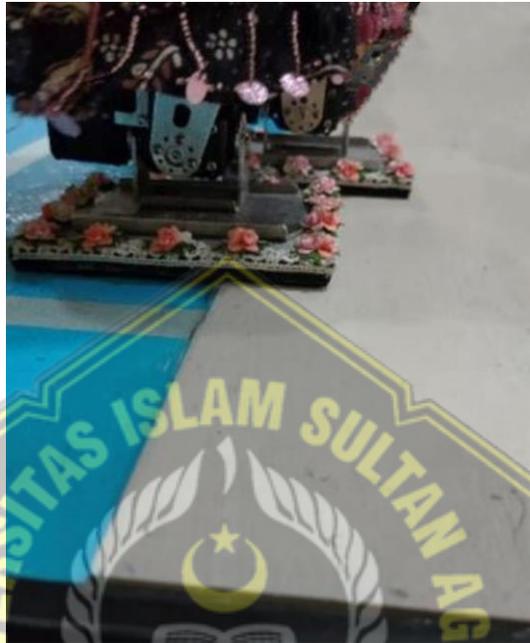
dalam posisi yang menghadap kedepan, adapun pergeseran pada saat robot berjalanpun hanya sedikit dan juga tidak mempengaruhi robot dalam melewati undakan. Pada saat pengujian robot dapat berjalan dengan baik pada saat naik undakan ataupun pada saat menuruni undakan. Robot benar-benar berhasil pada saat dilakukan pengujian sebanyak 6 dari 10 kali percobaan.

Tabel 4. 8 Pengujian berjalan pada permukaan berundak dengan *MotionPagePlay* 2500

Percobaan	Hasil Pengujian	Keterangan Hasil
1	Berhasil	Robot berjalan dengan baik pada saat naik undakan dan pada saat turun undakan, tetapi jalan robot sedikit menyimpang ke kiri
2	Tidak berhasil	Berjalan dengan lurus pada saat naik berhasil tetapi pada saat turun hampir terjatuh
3	Berhasil	Robot berjalan dengan baik pada saat naik undakan dan pada saat turun undakan
4	Berhasil	Robot berjalan dengan baik pada saat naik undakan dan pada saat turun undakan
5	Berhasil	Robot berjalan dengan baik pada saat naik undakan dan pada saat turun undakan
6	Berhasil	Robot berjalan dengan baik pada saat naik undakan dan pada saat turun undakan, tetapi jalan agak menyimpang ke kiri
7	Berhasil	Robot berjalan dengan baik pada saat naik undakan dan pada saat turun undakan, tetapi jalan agak menyimpang ke kiri dan kaki kiri hampir menyentuh dengan zona larangan
8	Berhasil	Robot berjalan dengan baik pada saat naik undakan dan pada saat turun undakan, tetapi jalan agak menyimpang ke kiri
9	Berhasil	Robot berjalan dengan baik pada saat naik undakan dan pada saat turun undakan, tetapi jalan agak menyimpang ke kiri
10	Berhasil	Robot berjalan dengan baik pada saat naik undakan dan pada saat turun undakan

Pada Tabel 4.8 merupakan hasil dari pengujian pergerakan robot yang berjalan pada permukaan yang berundak dengan pengaturan *MotionPagePlay* yang digunakan adalah 2500. Pada percobaan dengan delay 2,5 detik robot dapat berjalan dan melewati undakan dan juga turunan dengan baik, walaupun pada saat perpindahan gerak kaki robot sedikit bergeser ke kiri. Sehingga, ada salah satu

percobaan yang dimana kaki robot hampir menyentuh zona larangan dengan diameter 300 mm. Dari perhitungan persamaan (3.3) pada Tabel 4.8 maka persentase keberhasilannya yaitu 90%.



Gambar 4. 15 Hasil percobaan pada permukaan berundak dengan *MotionPagePlay* 2500

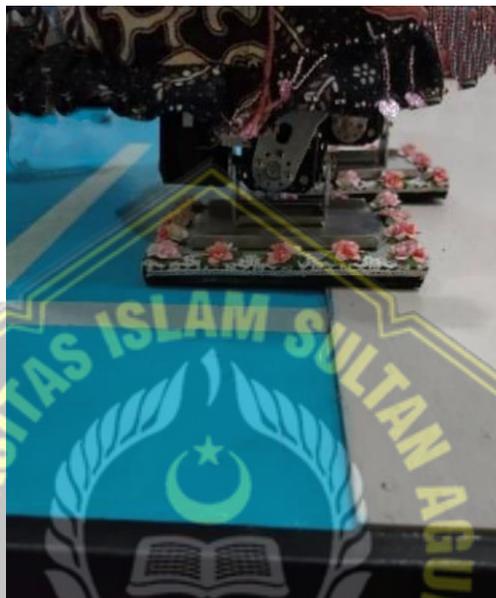
Pada Gambar 4.15 merupakan hasil percobaan pada permukaan berundak dengan *MotionPagePlay* 2500. Pada gambar tersebut terlihat bahwa posisi robot lebih menyimpang jika dibandingkan dengan pengujian *MotionPagePlay* seelumnya. Robot masih dapat berjalan dengan baik pada saat melawati undakan. Robot dapat berjalan dan melewati undakan dan juga turunan dengan baik, walaupun pada saat perpindahan gerak kaki robot sedikit bergeser ke kiri. Sehingga, ada salah satu percobaan yang dimana kaki robot hampir menyentuh zona larangan dengan diameter 300 mm. Robot benar-benar berhasil pada saat dilakukan pengujian sebanyak 9 dari 10 kali percobaan.

Tabel 4. 9 Pengujian berjalan pada permukaan berundak dengan *MotionPagePlay* 2800

Percobaan	Hasil Pengujian	Keterangan Hasil
1	Berhasil	Robot berjalan dengan baik pada saat naik undakan dan pada saat turun undakan, tetapi jalan agak menyimpang ke kiri dan hampir mengenai zona larangan
2	Berhasil	Robot berjalan dengan baik pada saat naik undakan dan pada saat turun undakan, tetapi jalan agak menyimpang ke kiri dan hampir mengenai zona larangan
3	Berhasil	Robot berjalan dengan baik pada saat naik undakan dan pada saat turun undakan, tetapi jalan agak menyimpang ke kiri dan hampir mengenai zona larangan
4	Tidak berhasil	Robot berjalan dengan baik pada saat naik undakan dan pada saat turun undakan, tetapi menyentuh zona larangan
5	Berhasil	Robot berjalan dengan baik pada saat naik undakan dan pada saat turun undakan, tetapi jalan agak menyimpang ke kiri
6	Berhasil	Robot berjalan dengan baik pada saat naik undakan dan pada saat turun undakan, tetapi jalan agak menyimpang ke kiri
7	Berhasil	Robot berjalan dengan baik pada saat naik undakan dan pada saat turun undakan, tetapi jalan agak menyimpang ke kiri dan hampir menyentuh zona larangan
8	Berhasil	Robot berjalan dengan baik pada saat naik undakan dan pada saat turun undakan, tetapi jalan agak menyimpang ke kiri dan pada saat jalan sedikit tidak stabil
9	Berhasil	Robot berjalan dengan baik pada saat naik undakan dan pada saat turun undakan, tetapi jalan agak menyimpang ke kiri
10	Berhasil	Robot berjalan dengan baik pada saat naik undakan dan pada saat turun undakan, tetapi jalan agak menyimpang ke kiri

Pada Tabel 4.9 merupakan hasil dari pengujian pergerakan robot yang berjalan pada permukaan yang berundak dengan pengaturan *MotionPagePlay* yang digunakan adalah 2800. Pada pengujian dengan *MotionPagePlay* tersebut robot dapat melewati undakan dengan baik sebanyak 9 kali, tetapi apabila robot tidak

menyentuh zona larangan maka, percobaan pada *MotionPagePlay* yang berhasil sebanyak 10 kali. Pengujian robot dengan delay setiap pergerakan 2,8 detik mengakibatkan pergeseran kaki pada saat robot berjalan sehingga mengakibatkan robot berjalan sedikit menyimpang ke kiri. Dari perhitungan persamaan (3.3) pada Tabel 4.9 maka persentase keberhasilannya yaitu 90%.



Gambar 4. 16 Hasil percobaan pada permukaan berundak dengan *MotionPagePlay* 2800

Pada Gambar 4.16 merupakan hasil percobaan pada permukaan berundak dengan *MotionPagePlay* 2800. Pada gambar tersebut terlihat bahwa posisi robot sedikit menyimpang ke kiri. Tetapi, robot masih dapat berjalan dengan baik saat melewati undakan maupun turunan, walaupun terdapat pergeseran pada saat kaki mengindak tanah. Pergeseran pada robot mengakibatkan robot menyimpang ke kiri dan hampir mengenai zona larangan. Robot benar-benar berhasil pada saat dilakukan pengujian sebanyak 9 kali dengan persentase keberhasilan 90%.

Tabel 4. 10 Pengujian berjalan padapermukaan berundak dengan *MotionPagePlay* 3200

Percobaan	Hasil Pengujian	Keterangan Hasil
1	Tidak berhasil	Robot sedikit kurang seimbang saat melewati undakan, robot dapat melewati turunan dengan baik tetapi jalan agak menyimpang ke kiri dan mengenai zona larangan
2	Tidak berhasil	Robot tidak dapat melewati undakan, tetapi robot dapat melewati turunan dengan baik dan jalan agak menyimpang ke kiri sehingga mengenai zona larangan
3	Tidak berhasil	Robot tidak dapat melewati undakan, tetapi robot dapat melewati turunan dengan baik dan jalan agak menyimpang ke kiri sehingga mengenai zona larangan
4	Tidak berhasil	Robot robot tidak dapat melewati undakan, tetapi robot dapat melewati turunan dengan baik dan jalan agak menyimpang ke kiri sehingga mengenai zona larangan
5	Tidak berhasil	Robot dapat melewati undakan tetapi langkah berikutnya robot kehilangan keseimbangan. Robot dapat melewati turunan dengan baik dan jalan agak menyimpang ke kiri sehingga mengenai zona larangan
6	Berhasil	Robot robot dapat melewati undakan tetapi sedikit kurang seimbang. Robot dapat melewati turunan dengan baik dan jalan agak menyimpang ke kiri tetapi mengenai zona larangan
7	Tidak berhasil	Robot dapat melewati undakan tetapi setelah itu robot kurang seimbang dan hampir terjatuh. Robot dapat melewati turunan dengan baik dan jalan agak menyimpang ke kiri tetapi mengenai zona larangan
8	Tidak berhasil	Robot tidak dapat melewati undakan, pada pertengahan jalan robot hampir jatuh, tetapi robot dapat melewati turunan dengan baik dan jalan agak menyimpang ke kiri sehingga mengenai zona larangan
9	Tidak berhasil	Robot tidak dapat melewati undakan, pada pertengahan jalan robot hampir jatuh, tetapi robot dapat melewati turunan dengan baik dan jalan agak menyimpang ke kiri
10	Tidak berhasil	Robot tidak dapat melewati undakan, pada pertengahan jalan robot hampir jatuh, robot tidak dapat melewati turunan dan jalan agak menyimpang ke kiri sehingga mengenai zona larangan

Pada Tabel 4.10 merupakan hasil dari pengujian pergerakan robot yang berjalan pada permukaan yang berundak dengan pengaturan *MotionPagePlay* yang digunakan adalah 3200. Pada pengujian dengan *MotionPagePlay* ini hampir 10 kali percobaan robot, robot tidak dapat melewati undakan dan pergeseran jalan robot yang lumayan sehingga, mengakibatkan robot berjalan miring dan menyentuh zona larangan. Pengujian robot dengan delay setiap pergerakan yaitu 3,2 detik yang mengakibatkan pergeseran robot menyimpang ke arah kiri. Dari perhitungan persamaan (3.3) pada Tabel 4.10 maka persentase keberhasilannya yaitu 10%.



Gambar 4. 17 Hasil percobaan pada permukaan berundak dengan *MotionPagePlay* 3200

Pada Gambar 4.17 merupakan hasil percobaan pada permukaan berundak dengan *MotionPagePlay* 3200. Pada gambar tersebut terlihat bahwa posisi robot menyimpang ke arah kiri. Pergerakan robot sangat lambat jika dibandingkan dengan percobaan sebelumnya yang dimana itu mempengaruhi kerja robot pada saat perpindahan pergerakan yang bergeser pada saat kaki robot menginjak tanah. Pada saat pengujian robot kurang seimbang pada saat melewati undakan maupun turunan sehingga mengakibatkan pergeseran yang menyentuh zona larangan. Robot benar-benar berhasil pada saat dilakukan pengujian sebanyak 1 dari 10 kali percobaan.

BAB V

PENUTUP

5.1 Kesimpulan

Dari hasil penelitian yang telah dilakukan maka dapat disimpulkan sebagai berikut:

1. Bagian kaki robot *humanoid* terdiri dari 14 joint yang masing-masing digerakkan dengan motor servo Dynamixel 28T. motor servo tersebut memiliki ukuran yang fleksibel dan torsi 2.3 Nm yang digunakan untuk dapat menggerakkan sendi, serta menghasilkan gerakan yang halus.
2. Motor servo Dynamixel MX-28T digunakan untuk mengontrol gerakan kaki robot *humanoid* yang terdiri dari gerak jongkok, stay, dan berjalan yang terdiri dari 9 tahapan. Pengontrolan gerak tersebut yaitu dengan mengatur sudut serta joint yang terdapat pada lampiran 2. Pengontrol sudut dilakukan dengan mengubah nilai encoder. Hasil pengujian gerakan kaki robot dengan posisi jongkok dan *stay*, robot dapat bertahan setelah diberikan hambatan berupa dorongan kecil. Pada pengujian jongkok dapat mempertahankan posisi dalam bidang miring dengan tingkat kemiringan 2°, 4°, 6°, 8°, dan 10°. Pada saat pengujian *step* berjalan robot hanya dapat bertahan selama 30 detik.
3. Pada pengujian gerak jalan robot humanoid pada permukaan bidang dengan panjang arena 71 cm yang dilakukan pengujian dengan menggunakan variasi *MotionPagePlay*. Akurasi terbaik terdapat pada *MotionPagePlay* 2200 dengan rata-rata persentase error 0,33%.
4. Pada pengujian gerak jalan robot humanoid pada permukaan berundak dengan ketinggian 4mm dengan menggunakan variasi *MotionPagePlay*. Akurasi terbaik terdapat pada *MotionPagePlay* 2500 dan 2800 dengan persentase keberhasilan 90%. Pada *MotionPagePlay* tersebut robot lebih dapat melewati permukaan yang berundak saat naik dan turun yang dimana robot lebih dapat menjaga keseimbangan.

5. Hasil akurasi terbaik pada gerak jalan robot lurus dan berundak yaitu di *MotionPagePlay* 2200 sampai dengan 2500. Jika, *MotionPagePlay* diatur dengan lebih lambat maupun cepat dari nilai 2200 sampai dengan 2500 maka dapat mempengaruhi gerakan robot.

5.2 Saran

Saran yang dapat dilakukan untuk mengembangkan penelitian robot *humanoid* penari ini adalah

1. Sebelum dilakukannya perancangan perangkat lunak, kondisi dari mekanik robot *humanoid* harus dalam keadaan yang sudah benar benar siap untuk digunakan. Pemasangan baut yang kurang kencang juga dapat mempengaruhi pergerakan dari robot itu sendiri.
2. Robot *humanoid* memerlukan pemeriksaan secara berkala, terutama pada bagian dengan beban tumpuan yang berat. Pemeriksaan visual meliputi kerusakan dan keausan pada servo, uji gerak robot untuk memastikan bahwa servo bergerak dengan lancar, pemeriksaan suhu untuk memastikan bahwa servo telah bekerja lebih baik, pemeriksaan koneksi antar servo untuk memastikan bahwa antar servo telah terhubung dengan baik dan pemeriksaan catu daya yang cukup.
3. Diperlukannya management energi serta kapasitas baterai guna untuk menjaga gerakan robot agar tetap optimal, selain itu diperlukan pendinginan pada servo untuk menjaga kualitas dari gerakan robot *humnaoid* dan untuk mencegah terjadinya kerusakan pada servo dikarenakan panas servo tersebut.
4. Pada robot *humanoid* ini dapat dikembangkan dengan menambahkan sensor *Gyroskop* dan *accelerometer* yang digunakan untuk menunjang dalam pengenalan gerakan robot, menjaga keseimbangan serta mengontrol postur tubuh.

DAFTAR PUSTAKA

- [1] B. Kusumoputro *et al.*, *Buku Pedoman Kontes Robot Indonesia (Kri) Tahun 2023*. Jakarta: Balai Pengembangan Talenta Indonesia Pusat Prestasi Nasional Kementerian Pendidikan, Kebudayaan, Riset dan Teknologi, 2023.
- [2] A. Jalil, "Rancang Bangun Robot Humanoid," *Konf. Nas. Ilmu Komput.*, pp. 2338–2899, 2016.
- [3] B. Arifin, A. A. Nugroho, B. Suprpto, S. A. D. Prasetyowati, and Z. Nawawi, "Review of Method for System Identification on Motors," in *2021 8th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*, 2021, pp. 257–262.
- [4] M. A. Fahd, D. Purwanto, and M. H. Fatoni, "Rancang Bangun Robot Penari Humanoid dengan 25 DoF untuk Melakukan Gerakan Tari Remo," *J. Tek. ITS*, vol. 7, no. 2, pp. 2337–3539, 2018.
- [5] Zamzami, Salahuddin, Anwar, and Yusman, "Rancang Bangun Robot Humanoid Berbasis Mikrokontroler ATmega 2560," *J. Infomedia Tek. Inform. Multimed. Jar.*, vol. 6, no. 1, pp. 22–26, 2021.
- [6] F. Hermawanto and I. Karim, "Rancang Bangun Robot Penari Tidi," *Energy-Jurnal Ilm. Ilmu-Ilmu Tek.*, vol. 8, no. 2, pp. 35–41, 2018.
- [7] T. H. A. Putra, D. Syauqy, and R. Maulana, "Sistem Pendeteksi Tempo Lagu Untuk Kontes Robot Seni Tari Indonesia (KRSTI) Berdasarkan Frekuensi Dengan Algoritma Beat This," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 3, no. 4, pp. 3986–3992, 2019.
- [8] Kementrian Pendidikan Kebudayaan Riset dan Teknologi, "KRI 2023 - Kontes Robot Indonesia," *Panitia Pusat Kontes Robot Indonesia*, 2023. <https://kontesrobotindonesia.id/kri-2023.html> (accessed Apr. 20, 2023).
- [9] Ariani, "Robot," in *Politeknik Negeri Sriwijaya*, 2019. Accessed: Feb. 11, 2023. [Online]. Available: <https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ad=rja&uact=8&ved=2ahUKEwiHi8K-vIz9AhXBVmwGHeyRDHQQFnoECA8QAQ&url=http%3A%2F%2Fepri>

nts.polsri.ac.id%2F8009%2F3%2FFile%2520III.pdf&usg=AOvVaw0mPR
NtkAJ8NzcYO_GFWwX4

- [10] R. Supriyanto, Hustinawati, R. W. Nugraini, A. B. Kurniawan, Y. Permadi, and A. Sa'ad, *Robotika*, vol. 1. 2010.
- [11] Robotis, “Dynamixel MX-28T - Robotis,” 2022.
<https://www.robotis.us/dynamixel-mx-28t/> (accessed Sep. 26, 2022).
- [12] Robotis, “Dynamixel XL-320 - Robotis,” 2022.
<https://www.robotis.us/dynamixel-xl-320/> (accessed Sep. 26, 2022).
- [13] Robotis, “OpenCM9.04-C (with onboard XL-type connectors) - Robotis.”
<https://www.robotis.us/opencm9-04-c-with-onboard-xl-type-connectors/>
(accessed Sep. 26, 2022).
- [14] Robotis, “OpenCM 485 EXP.”
<https://emanual.robotis.com/docs/en/parts/controller/opencm485exp/>
(accessed Sep. 26, 2022).
- [15] Robotis, “OpenCM IDE.”
https://emanual.robotis.com/docs/en/software/opencm_ide/getting_started/#
(accessed Jul. 19, 2023).
- [16] Robotis, “CM-530 - Robotis.” <https://www.robotis.us/cm-530/> (accessed Nov. 13, 2022).
- [17] Robotis, “RoboPlus 1.0 - Software.” <https://robotis.co.uk/software/roboplus-1-0.html> (accessed Nov. 13, 2022).
- [18] A. A. Syakur, R. Maulana, and E. Setiawan, “Perancangan Dan Implementasi Sistem Pola Berjalan Pada Robot Hexapod Menggunakan Metode Inverse Kinematic,” *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 4, no. 6, pp. 1875–1881, 2020.